University of the Western Cape

Faculty of Natural Sciences

Department of Statistics and Population Studies

Handling Heteroskedasticity in the Linear Regression Model

A thesis submitted in fulfilment of the requirements for the degree of
Doctor of Philosophy in Statistics

By

Thomas Farrar

Supervisor: Prof. Rénette Blignaut
Co-Supervisors: Prof. Sarel Steel, Dr. Retha Luus

Date: November 2022

Contents

Abstract

This research project delves into the problem of heteroskedasticity in the linear regression model. Having defined the problem and its consequences for estimation and inference, a comprehensive literature review of existing methods for diagnosing and correcting for heteroskedasticity is undertaken, with special emphasis on heteroskedasticity tests.

New theory on the statistical properties of the Ordinary Least Squares residuals is developed, leading to new models for estimating linear regression error variances. The most important of these models is the Auxiliary Linear Variance Model, which is further classified into sub-types (e.g., clustering, linear, penalised polynomial, spline). Model fitting techniques are discussed, which reduce to quadratic programming problems. An Auxiliary Nonlinear Variance Model is also developed, which can be fitted using a maximum quasi-likelihood method. Techniques for tuning of model hyperparameters and feature selection are discussed. Bootstrap methods of obtaining interval estimates for error variances are also proposed. A new heteroskedasticity test is constructed based on the auxiliary linear variance model.

To make existing and new methods of handling heteroskedasticity more accessible to the practitioner, a new package called **skedastic** has been developed for R statistical software. Its functionality is described in detail.

Various empirical results are obtained using Monte Carlo simulation experiments. A comparison between heteroskedasticity tests is made using an average excess power over size metric. The new error variance estimation methods are assessed under a variety of conditions in terms of four distinct mean squared error metrics, and are found to outperform existing methods under some conditions. Coverage probabilities of bootstrap confidence intervals are estimated. Finally, illustrative case studies are undertaken with three real-world data sets.

The new variance models are found to be competitive methods for handling heteroskedasticity in linear regression. Possible avenues for further refining the new methods are proposed for future research.

Keywords— model assumptions, model adequacy, Monte Carlo simulation, bootstrap, power, robustness, variance estimation

Declaration

I declare that this thesis is my own work, that it has not been submitted for any degree or examination in any other university, and that all the sources I have used or quoted have been indicated and acknowledged by complete references.

Thomas Joel Farrar

6 November 2022
Date

Acknowledgments

Albert Einstein wrote that ‘the eternal mystery of the world is its comprehensibility’ (Einstein [1936] 2003). In that spirit, I must first acknowledge Almighty God, who has willed all things into existence, and gifted us humans with the capacity to do science—that is, to study the world in all of its mysterious intelligibility.

I must express my sincere gratitude to my supervisor, Prof. Rénette Blignaut, and my co-supervisors, Dr. Retha Luus and Prof. Sarel Steel, for their constant encouragement and expert guidance throughout this project. To maintain bi-weekly meetings over four years, during a pandemic and sometimes meeting across continents and time zones, showed a matchless depth of commitment and was instrumental in keeping my progress inching forward.

My family, both near and far, have been an abiding source of love and support throughout my life and PhD journey. My dear parents first instilled in me a strong work ethic and an enthusiasm for reading and learning. My late grandparents seemed to believe that I was capable of anything, and it was my grandfather who started me down the path to a career in statistics by arranging a lunch meeting for me with a professor that he knew. My siblings, Jonathan, Sarah, Zachary, and Caitlin, showed interest in my progress and provided encouraging words.

My beloved wife, Ayanda, has been a rock of support throughout these years of study. Together with my adopted son, Sphesihle, and nephew, Smiso, she has made many sacrifices to give me the time and space needed to complete this work.

A special word of gratitude is due to my Head of Department, Mr. John Farmer, who relieved me of teaching duties in the first half of 2022 to allow me to focus on my research, and has always taken a strong interest in my professional development. The camaraderie of my departmental colleagues and the enthusiasm of my students have also made balancing my workload with my studies more manageable.

Finally, I wish to acknowledge a few friends who have been a particular source of strength and prayer during this academic journey: Eric, Gary, Abie, Deacon Saville and Lynn, Benjy, Alphonce, and Ahmed.

In loving memory of M.J.

List of Acronyms

ABC Approximate Bootstrap Confidence

AEPS Average Excess Power over Size

ALRT Approximate Likelihood Ratio Test

ALS Adaptive Least Squares

ALVM Auxiliary Linear Variance Model

ANLVM Auxiliary Nonlinear Variance Model

BAMSET Bartlett's M Specification Error Test

BCa Bias-Corrected and accelerated

BLUE Best Linear Unbiased Estimator

BLUP Best Linear Unbiased Predictor

BLUS Best Linear Unbiased Scalar-Covariance-Matrix

BSS Best Subset Selection

CDF cumulative distribution function

CI Confidence Interval

CRAN Comprehensive R Archive Network

CV Cross-Validation

CVT Coefficient of Variation Test

DGP Data Generating Process

FICGLS Feasible Inequality-Constrained Generalised Least Squares

FWLS Feasible Weighted Least Squares

GCV Generalised Cross-Validation

GLS Generalised Least Squares

GSS Golden Section Search

HCCME Heteroskedasticity-Consistent Covariance Matrix Estimator

ICGLS Inequality-Constrained Generalised Least Squares

ICLASSO Inequality-Constrained LASSO

ICLS Inequality-Constrained Least Squares

ICRR Inequality-Constrained Ridge Regression

iid independent and identically distributed

IR information ratio

IVH independent variable hull

LASSO Least Absolute Shrinkage and Selection Operator

LM Lagrange Multiplier

LR likelihood ratio

MAD median absolute deviation

MC Monte Carlo

MGF moment-generating function

ML Maximum Likelihood

MPLR Modified Profile Likelihood Ratio

MQL Maximum Quasi-Likelihood

MSE Mean Squared Error

MWD Maximum Within-Cluster Distance

OLS Ordinary Least Squares

PDF probability density function

PMF probability mass function

QP Quadratic Programming

QGCV Quasi-Generalised Cross-Validation

RCEV Regression on Centered External Variable

ReML Residual Maximum Likelihood

ROC Receiver Operating Characteristic

RQF Ratio of Quadratic Forms

RR Ridge Regression

SE Standard Error

SVR Support Vector Regression

SWD Sum of Within-Cluster Distances

UIK Unit Invariant Knee

WLS Weighted Least Squares

List of Symbols

The following is by no means a complete list of mathematical symbols used in this thesis but explains some of the more important and frequently used symbols.

Boldface herein is used for a vector or matrix variable. Matrices are denoted by upper case letters and vectors by lower case letters. Scalars are denoted with normal font type and either upper or lower case.

Operators and Miscellaneous Symbols

$\mathbf{0}^{+}$	a vector of very small positive numbers
$\succ, \succeq$	inequalities applied elementwise to a vector
$1_{\bullet}$	an indicator function taking a value of 1 if the condition in the subscript is satisfied and 0 otherwise
$\Vert\boldsymbol{a}\Vert_p$	the L_p-norm of a vector $\boldsymbol{a}$
$\arg\min$	the argument of the minimum value of a function
$\boldsymbol{a} \circ \boldsymbol{b}$	denotes the Hadamard (elementwise) product of two vectors (or matrices)
$\hat{\theta}$	an estimator or predicted value of θ
$\breve{\theta}$	a parameter estimate computed from a subset or fold of the data
$[a]$	the integer part of a real number a
$\mathrm{diag}\,\{\boldsymbol{a}\}$	denotes a diagonal matrix with the vector $\boldsymbol{a}$ as its diagonal
$\mathrm{diag}\,(\boldsymbol{A})$	denotes the diagonal elements of the matrix $\boldsymbol{A}$
$\boldsymbol{A}'$	the transpose of a matrix $\boldsymbol{A}$
$\boldsymbol{A}^{-1}$	the inverse of a matrix $\boldsymbol{A}$
$\mathrm{tr}(\boldsymbol{A})$	the trace of a matrix $\boldsymbol{A}$
$\mathrm{E}(\cdot)$	the expectation of a random variable or random vector
$\mathrm{Var}(\cdot)$	the variance of a random variable
$\mathrm{Cov}(\cdot)$	the covariance of two scalar random variables, or the variance-covariance matrix of a random vector
$\mathrm{Corr}(\cdot)$	the correlation of two scalar random variables, or the correlation matrix of a random vector
$\mathrm{Bias}(\cdot)$	the bias of an estimator
$\mathrm{MSE}(\cdot)$	the mean squared error of an estimator
$\mathrm{SE}(\cdot)$	the standard error of an estimator
$\xrightarrow{D}$	converges in distribution to
$a^{(b)}$	the bth bootstrap replication of a
$a^{(r)}$	the rth Monte Carlo replication of a

Variables and Parameters

This list follows Greek alphabetical order first, then Latin.

$\boldsymbol{\beta}$	a p-vector of coefficients in the mean function of a linear regression model, with jth element β_j
$\hat{\boldsymbol{\beta}}_{\mathrm{OLS}}$	the Ordinary Least Squares estimator of $\boldsymbol{\beta}$
$\boldsymbol{\gamma}$	a q-vector of parameters used in one parametrisation of the $g(\cdot)$ heteroskedastic function
δ_i	a power used to construct heteroskedasticity-consistent estimators of ω_i of the form $e_i^2(1-h_{ii})^{-\delta_i}$
$\boldsymbol{\epsilon}$	an n-vector of random errors in a linear regression model, with ith element ϵ_i
$\boldsymbol{\zeta}$	a p'-vector of parameters used in one parametrisation of the $g(\cdot)$ heteroskedastic function
$\hat{\boldsymbol{\eta}}$	the Lagrangian vector of the solution to a quadratic programming problem
λ	the penalty parameter in a penalised regression model
$\boldsymbol{\Xi}$	a design matrix composed of the union of columns of $\boldsymbol{X}$ and $\boldsymbol{Z}$
ω	the (scalar) variance of the errors in a homoskedastic linear regression model
$\bar{\omega}$	the (biased) maximum likelihood estimator of ω under homoskedasticity
$\hat{\omega}_{\mathrm{ub}}$	an unbiased estimator of ω under homoskedasticity

$\boldsymbol{\omega}$	an n-vector of variances of the errors in a linear regression model, with ith element ω_i
$\boldsymbol{\Omega}$	an $n \times n$ variance-covariance matrix of the errors in a linear regression model
D_i	the Cook's Distance of the ith observation in a linear regression model
$\boldsymbol{D}$	the design matrix $(\boldsymbol{M} \circ \boldsymbol{M})\boldsymbol{L}$ in an auxiliary linear variance model, which maps the parameters $\boldsymbol{\gamma}$ onto the expected Ordinary Least Squares squared residuals
$\boldsymbol{e}$	an n-vector of Ordinary Least Squares residuals in a linear regression model, with ith element e_i
$\boldsymbol{e}_{\text{BLUS}}$	an $n-p$-vector of Best Linear Unbiased Scalar-Covariance-Matrix residuals
$\boldsymbol{f}(\boldsymbol{\gamma})$	the conditional mean function of an auxiliary variance model, taken as a function of parameters $\boldsymbol{\gamma}$
$\boldsymbol{F}_{\bullet}(\boldsymbol{\gamma})$	the Jacobian of $\boldsymbol{f}(\boldsymbol{\gamma})$, with respect to parameters $\boldsymbol{\gamma}$
$g(\cdot)$	a continuous, twice-differentiable, positive real-valued function by which the error variances ω_i are related to an observed covariate matrix $\boldsymbol{Z}$
$\mathfrak{g}(\cdot)$	a rescaled version of $g(\cdot)$
$\boldsymbol{H}$	the $n \times n$ 'hat' or 'projection' matrix in a linear regression model, with i,jth element h_{ij}, and with diagonal elements h_{ii} referred to as 'leverage scores'
$\boldsymbol{H}_{\boldsymbol{\Omega}}$	a generalisation of $\boldsymbol{H}$ used in Weighted Least Squares
$I_{\nu}(\cdot)$	the modified Bessel function of the first kind and order ν
$\boldsymbol{I}_a$	an $a \times a$ identity matrix
$\boldsymbol{L}$	an $n \times q$ linear predictor matrix in an auxiliary linear variance model, which maps parameters $\boldsymbol{\gamma}$ onto the error variances $\boldsymbol{\omega}$
$M_{\boldsymbol{a}}(\boldsymbol{t})$	the moment-generating function of a random vector $\boldsymbol{a}$
$\boldsymbol{M}$	the $n \times n$ 'annihilator' matrix in a linear regression model, with i,jth element m_{ij}
$\boldsymbol{M}_{\boldsymbol{\Omega}}$	a generalisation of $\boldsymbol{M}$ used in Weighted Least Squares
n_c	the number of clusters used in a clustering algorithm
$\boldsymbol{P}$	a $q \times q$ penalty matrix used in some auxiliary linear variance models
$t_{p,\nu}$	the pth upper quantile of Student's t distribution with ν degrees of freedom
T	a test statistic used in a test of hypotheses
$\boldsymbol{u}$	an n-vector of random errors in an auxiliary linear variance model
$\boldsymbol{V}(\boldsymbol{\gamma})$	the $n \times n$ variance-covariance matrix of the errors in an auxiliary variance model, written as a function of the parameters $\boldsymbol{\gamma}$
$\boldsymbol{W}$	an $n \times n$ diagonal weights matrix used in Weighted Least Squares, with ith diagonal element w_{ii}
$\boldsymbol{X}$	$n \times p$ design or covariate matrix in a linear regression model, with i,jth element X_{ij}, ith row $\boldsymbol{X}_{i\cdot}$, $i = 1, 2, \ldots, n$, and jth column $\boldsymbol{X}_{\cdot j}$, $j \in \{1, 2, \ldots, p\}$
$\boldsymbol{\mathcal{X}}$	an $n \times (p-1)$ matrix formed by removing an intercept column from $\boldsymbol{X}$ (if present) and then centering the remaining columns
$\boldsymbol{y}$	response vector in a linear regression model, with ith element y_i
z_p	the pth upper quantile of the standard normal distribution
$\boldsymbol{Z}$	$n \times p'$ design or covariate matrix in an auxiliary regression model, with i,jth element Z_{ij}

List of Figures

List of Tables

A Note to the Reader

Here are a few preliminary remarks to ensure the reader has a pleasant experience navigating within this document. Numerous entities within the document are cross-referenced using clickable hyperlinks (sections and subsections down to the fourth level; equations; figures and tables; some numbered list items; acronyms). These hyperlinks will appear in blue font. Clicking on a hyperlink will take the reader to the location in the document where that entity is defined.

Suppose the reader encounters an acronym on page 100 of the thesis and does not remember what it stands for. The reader can click on the acronym's blue hyperlink and will be taken to the List of Acronyms. Now, the reader knows what the acronym stands for, but suppose s/he does not remember what page s/he was on! No problem: the reader can go back to the previous location by pressing `Alt` ← on the keyboard. This is equivalent to the Previous View option on the Page Navigation submenu accessible from the View tab in Adobe Acrobat.[1]

[1] Note that `Alt Gr` ← does not have the same function as `Alt` ←.

1 Introduction

When applying statistical models to data, model adequacy—verifying that the assumptions of a model are satisfied—is an important consideration. Equally important is to have access to robust methods that retain good statistical properties in the event that model assumptions are violated. This research project focuses on the violation of a particular assumption of a particular statistical model. The specific case that will be studied is heteroskedasticity—the violation of the assumption of homoskedasticity or homogeneity of variances—in the linear regression model.[2] Having broadly sketched out the context of this study, it is necessary to provide some background on the linear regression model, its assumptions, and some of its statistical characteristics, so that the research problem and objectives can be properly framed.

The classical linear regression model is a very widely used statistical method for analysing relationships between a continuous response variable and one or more predictor variables. This research project is devoted to the study of addressing heteroskedasticity in the linear regression model: detecting it, estimating it, and correcting for it in the estimation of and inference on the model parameters.

In this introductory chapter, the linear regression model will be defined, along with some important vector and matrix quantities. The classical assumptions of the linear model will be stated and heteroskedasticity defined. This will be followed by an overview of estimation theory pertaining to the linear regression model, both under the full classical assumptions and under heteroskedasticity.

Next, special attention will be given to the model residuals, since these are of great importance for detecting and modelling heteroskedasticity. Statistical properties of the ordinary least squares residuals will be stated and, in some cases, proven. Other types of residuals will be introduced.

Turning to the problem of inference, the distributions of quantities of interest under the full classical assumptions will be stated and the t-test for inference on model parameters derived. The distributions of quantities of interest will likewise be derived under heteroskedasticity, and the effect of heteroskedasticity on the validity of the t-test discussed. A brief introduction to the notions of leverage and influence in the linear model will be provided, along with the related concepts of studentised residuals and Cook's Distance.

With all of this background in hand, the introduction will end with a statement of the research problem and an enumeration of the research objectives.

1.1 The Linear Regression Model

1.1.1 Definition and Basic Notation

The linear regression model is specified in matrix form as follows:

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \tag{1.1}$$

where $\boldsymbol{y}$ is an n-vector of observed responses (sometimes referred to as the regressand or dependent variable), $\boldsymbol{\beta} = [\beta_1, \beta_2, \ldots, \beta_p]'$ is a p-vector of unknown constant parameters, $\boldsymbol{X}$ is an $n \times p$ observed predictor matrix (sometimes referred to as the design matrix), and $\boldsymbol{\epsilon}$ is an n-vector of unobserved random errors or disturbances. Conventionally, $\boldsymbol{X}$ has a column of ones as its first column (corresponding to the model intercept), and the remaining columns, $\boldsymbol{X}_{\cdot j}$, $j = 2, 3, \ldots, p$, are n-vectors of predictor variables (sometimes referred to as regressors, covariates, design variables, explanatory variables, or independent variables).[3] In the case of simple linear regression (where there is only one predictor variable), the notation $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]'$ will also be used for the predictor variable. It is normally required that $n > p$ (ideally, $n \gg p$). (1.1) can also be referred to as the Data Generating Process (DGP) of the response $\boldsymbol{y}$.

Let $\hat{\boldsymbol{\beta}}$ denote any statistical estimator of $\boldsymbol{\beta}$. The predicted or fitted response vector (subject to the estimator $\hat{\boldsymbol{\beta}}$) is defined as $\hat{\boldsymbol{y}} := \boldsymbol{X}\hat{\boldsymbol{\beta}}$, and the residual vector (an observable predictor of the random errors) is defined as $\boldsymbol{e} := \boldsymbol{y} - \hat{\boldsymbol{y}}$. Throughout this document, $\circ$ denotes the Hadamard product (elementwise product) of two vectors or matrices. Thus the vector of squared residuals will be denoted $\boldsymbol{e} \circ \boldsymbol{e}$.

1.1.2 The Hat Matrix and the Annihilator Matrix

Certain matrices are of special importance in the linear regression model. The 'hat' or 'projection' matrix $\boldsymbol{H} = \boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}' = (h_{ij})$, denoted in some literature by $\boldsymbol{P}$, is an $n \times n$ hat matrix satisfying $\hat{\boldsymbol{y}} = \boldsymbol{H}\boldsymbol{y}$; that

[2] On 'heteroskedasticity' (as opposed to 'heteroscedasticity') as the preferred spelling, see Paloyo (2011).

[3] Herein, $\boldsymbol{X}'_{i\cdot}$ denotes the ith row of $\boldsymbol{X}$, $i = 1, 2, \ldots, n$, while $\boldsymbol{X}_{\cdot j}$ denotes the jth column of $\boldsymbol{X}$, $j = 1, 2, \ldots, p$.

is, it projects the observed response vector onto the predicted or fitted response vector. The matrix $\boldsymbol{H}$ is symmetric ($h_{ij} = h_{ji}$) and idempotent ($\boldsymbol{HH} = \boldsymbol{H}$). From the commutative property of the trace operator, it follows that $\operatorname{tr}(\boldsymbol{H}) = p$. Moreover, like all idempotent matrices other than the identity matrix, $\boldsymbol{H}$ is singular. This can easily be proven by contradiction. Assume that $\boldsymbol{H}$ has inverse $\boldsymbol{H}^{-1}$ such that $\boldsymbol{H}^{-1}\boldsymbol{H} = \boldsymbol{I}_n$. Then,

$$\begin{aligned}\boldsymbol{H}^{-1}\boldsymbol{H} &= \boldsymbol{H}^{-1}\boldsymbol{HH} \text{ (by idempotence)}\\ &= \boldsymbol{I}_n\boldsymbol{H} \text{ (by inverse assumption)}\\ &= \boldsymbol{H} \neq \boldsymbol{I}_n \text{ (contradicts inverse assumption).}\end{aligned}$$

The annihilator matrix, denoted here by $\boldsymbol{M} = (m_{ij})$ (in some literature by $\boldsymbol{Q}$), is an $n \times n$ matrix defined as $\boldsymbol{M} := \boldsymbol{I}_n - \boldsymbol{H}$, where $\boldsymbol{I}_n$ is the identity matrix. The name 'annihilator' comes from the property that $\boldsymbol{MX} = \boldsymbol{0}$; that is, $\boldsymbol{M}$ 'annihilates' the design matrix. Like $\boldsymbol{H}$, the matrix $\boldsymbol{M}$ is symmetric and idempotent, and therefore singular; its trace is $\operatorname{tr}(\boldsymbol{M}) = n - p$.

1.1.3 Classical Linear Model Assumptions

The exact list of assumptions for the multiple linear regression model varies from one text to another.[4] Certain assumptions are already inherent in (1.1), such as linearity in the parameters $\boldsymbol{\beta}$, and correct specification of the model. Another assumption that is often made is that the predictors are fixed or nonstochastic (they do not vary in repeated samples).[5] This assumption is reasonable in an experimental context, but not in the case of observational data. To avoid making this assumption, all statistical results can be conditioned on the predictor matrix. To enable concise notation, all statistical results herein should be taken as conditioned on $\boldsymbol{X}$ and/or any other relevant predictor matrix that is introduced.[6]

With this conditioning approach and the assumptions built into (1.1) (i.e., that it gives the true data generating process and thus that the linear predictor $\boldsymbol{X\beta}$ is correctly specified), the following are the remaining model assumptions.

A1. The error terms all have conditional mean zero ($\mathrm{E}(\boldsymbol{\epsilon}) = \boldsymbol{0}$). This directly implies that the conditional mean response equals the linear predictor ($\mathrm{E}(\boldsymbol{y}) = \boldsymbol{X\beta}$).

A2. The error terms all have the same conditional variance, $\omega > 0$ ($\mathrm{Var}(\epsilon_i) = \omega, i \in \{1, 2, \ldots, n\}$). This is known as the assumption of *homoskedasticity*.

A3. The error terms are all conditionally uncorrelated ($\mathrm{Cov}(\epsilon_i, \epsilon_j) = 0$, $i \neq j$). This is known as the assumption of *no autocorrelation*.

A4. No predictor can be expressed as a linear combination of other predictor(s). This is known as the assumption of *no perfect multicollinearity*. Another way of stating this assumption is that the predictor matrix $\boldsymbol{X}$ must have rank p.[7]

A5. The joint distribution of the error terms, conditioned on the corresponding predictor values, is multivariate Normal (Gaussian).

Let $\boldsymbol{\Omega} = (\omega_{ij})$ be the variance-covariance matrix of the random errors, $\mathrm{Cov}(\boldsymbol{\epsilon})$,[8] and let $\boldsymbol{\omega} = \operatorname{diag}(\boldsymbol{\Omega})$.[9] Then A2-A3 can be stated jointly by the expression $\boldsymbol{\Omega} = \omega\boldsymbol{I}_n$, where $\omega > 0$ is a scalar,[10] while A1-A3 and A5 can be stated jointly by the expression $\boldsymbol{\epsilon} \sim N(\boldsymbol{0}, \omega\boldsymbol{I}_n)$. Note also that A3 and A5 together imply that the

[4] See, for example, Berry (1993) and Gujarati (2018).

[5] In some treatments of multiple linear regression, a weaker version of this assumption is used, namely that for each $i \in \{1, 2, \ldots, n\}$ and $j \in \{1, 2, \ldots, p\}$, $\mathrm{Cov}(\epsilon_i, X_{ij}) = 0$ (the predictors are uncorrelated with the errors).

[6] Thus, in assumption A1, for instance, $\mathrm{E}(\boldsymbol{\epsilon})$ is really $\mathrm{E}(\boldsymbol{\epsilon}|\boldsymbol{X})$.

[7] Sometimes a separate assumption is made stating that all predictors have some variation in value. However, provided that the model has an intercept ($\boldsymbol{X}$ contains a column of ones), this follows from the assumption of no perfect multicollinearity, since another column with no variation in value would be expressible as a scalar multiple of the intercept.

[8] For conciseness, the notation $\mathrm{Cov}(\boldsymbol{v})$ is used for the variance-covariance matrix of a random vector $\boldsymbol{v}$, i.e., $\mathrm{Cov}(\boldsymbol{v}) = \mathrm{Cov}(\boldsymbol{v}, \boldsymbol{v}) = \mathrm{E}\left[(\boldsymbol{v} - \mathrm{E}(\boldsymbol{v}))\,(\boldsymbol{v} - \mathrm{E}(\boldsymbol{v}))'\right]$.

[9] Herein, $\operatorname{diag}(\boldsymbol{A})$, where $\boldsymbol{A} = (a_{ij})$ is a square matrix, denotes a vector whose elements are the diagonal elements of $\boldsymbol{A}$. $\operatorname{diag}\{\boldsymbol{v}\}$, where $\boldsymbol{v}$ is a vector, denotes a diagonal matrix with $\boldsymbol{v}$ as its diagonal.

[10] Note that, by A1, $\mathrm{Cov}(\boldsymbol{\epsilon})$ can also be written as $\mathrm{E}(\boldsymbol{\epsilon\epsilon}')$. Since, by definition, $\mathrm{Cov}(\boldsymbol{y}) = \mathrm{Cov}(\boldsymbol{\epsilon})$, it also follows from A2-A3 that $\mathrm{Cov}(\boldsymbol{y}) = \omega\boldsymbol{I}_n$.

errors $\boldsymbol{\epsilon}$ are mutually independent, due to the property that the components of a normally distributed random vector are independent if and only if they are uncorrelated (Gut 2005, Theorem 5.3).

Although not formally part of the classical linear model assumptions, a modified and more general version of A5 is introduced here, along with a further assumption, as follows.

A5′. The error terms, conditioned on the corresponding predictor values, all have the same cumulative distribution function (CDF) (though not necessarily the same parameter values).

A6′. The third and fourth moments of the conditional distribution of the error terms exist and are finite ($\mathrm{E}\left(\epsilon_i^s\right) < \infty$ for $s = 3, 4$).

1.1.4 Definition of Heteroskedasticity

Throughout this research, it is assumed that A1 and A3-A4 hold. The primary focus is on the violation of A2, known as *heteroskedasticity*. (A5 may, on occasion, be relaxed as well). The problems of heteroskedasticity and autocorrelation are sometimes treated together, since they are both violations of assumptions concerning $\boldsymbol{\Omega}$. A separate treatment of heteroskedasticity can be justified by noting that, while autocorrelation is often encountered in time series data, it is in practice rare in cross-sectional data (Berry 1993, pp. 71-73). Thus, unless the data contains spatial variables where spatial autocorrelation might occur (Anselin and Bera 1998, p. 237), an *a priori* assumption of no autocorrelation may be reasonable. Heteroskedasticity, on the other hand, is a frequently observed problem in linear models fitted to cross-sectional data (Gujarati 2018, p. 106). When A2 is relaxed but A3 retained, $\boldsymbol{\Omega}$ is assumed to be an $n \times n$ diagonal matrix with ith diagonal element $\omega_{ii} = \omega_i$.

Of special interest for this research is the case, commonly seen in practice with observational data, where the ith error variance, $i = 1, 2, \ldots, n$, is a function of some observed covariates Z_{ij}, $j = 1, 2, \ldots, p'$; that is, $\omega_i = g(\boldsymbol{Z}'_{i\cdot})$, where g is a continuous, twice-differentiable, positive real-valued function. The auxiliary design matrix $\boldsymbol{Z}$ could be identical to $\boldsymbol{X}$ (in which case $p' = p$), but may also consist of a subset of columns of $\boldsymbol{X}$, and/or other covariates not in $\boldsymbol{X}$. By convention, $\boldsymbol{Z}$ will be assumed to include a column of ones as its first column. An auxiliary design matrix that does not include a column of ones will be denoted as $\boldsymbol{Z}_{-1}$.

1.1.5 Estimation of Model Parameters under Assumptions A1-A4

1.1.5.1 Ordinary Least Squares

The Ordinary Least Squares (OLS) estimator of $\boldsymbol{\beta}$ is denoted by

$$\hat{\boldsymbol{\beta}}_{\text{OLS}} = \left(\boldsymbol{X}'\boldsymbol{X}\right)^{-1}\boldsymbol{X}'\boldsymbol{y}, \tag{1.2}$$

where A4 ensures that $\boldsymbol{X}'\boldsymbol{X}$ is invertible. The name ‘least squares’ refers to the fact that this estimator minimises the sum of squared residuals, $SS_{\text{residual}} = \left(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}\right)'\left(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}\right)$, with respect to $\boldsymbol{\beta}$. A derivation of $\hat{\boldsymbol{\beta}}_{\text{OLS}}$ is given in Appendix A.1.

1.1.5.2 The Gauss-Markov Theorem

The Gauss-Markov Theorem states that, under Assumptions A1-A4 (A5 not required), $\hat{\boldsymbol{\beta}}_{\text{OLS}}$ is the Best Linear Unbiased Estimator (BLUE) of $\boldsymbol{\beta}$ (Wooldridge 2013). That is, among the class of linear unbiased estimators of $\boldsymbol{\beta}$, $\hat{\boldsymbol{\beta}}_{\text{OLS}}$ has the smallest variance. A proof of the theorem is given in Appendix A.2. Note that (1.3) also follows from the proof:

$$\text{Cov}(\hat{\boldsymbol{\beta}}_{\text{OLS}}) = \omega(\boldsymbol{X}'\boldsymbol{X})^{-1}. \tag{1.3}$$

1.1.5.3 Maximum Likelihood Estimator

The Maximum Likelihood (ML) estimator of the parameter vector $\boldsymbol{\theta} = [\boldsymbol{\beta}', \omega]'$ under homoskedasticity (A1-A5) is, as derived in Appendix A.3,

$$\hat{\boldsymbol{\theta}}_{\text{MLE}} = \begin{bmatrix} \left(\boldsymbol{X}'\boldsymbol{X}\right)^{-1}\boldsymbol{X}'\boldsymbol{y} \\ \dfrac{1}{n}\boldsymbol{e}'\boldsymbol{e} \end{bmatrix}. \tag{1.4}$$

That is, $\hat{\boldsymbol{\beta}}_{\text{OLS}}$ is the ML estimator of $\boldsymbol{\beta}$, while the mean of the squared OLS residuals, $\bar{\omega} = n^{-1}\boldsymbol{e}'\boldsymbol{e}$, is the ML estimator of the common error variance ω. However, $\bar{\omega}$ is not the most widely used estimator of ω under homoskedasticity because it is biased. That distinction belongs to the unbiased estimator,

$$\hat{\omega}_{\text{ub}} = (n-p)^{-1}\boldsymbol{e}'\boldsymbol{e}, \tag{1.5}$$

whose unbiasedness property is discussed in §3.1.1. The $_{\text{ub}}$ subscript denotes that this is an unbiased estimator of ω.

1.1.6 Estimation of Model Parameters under Heteroskedasticity

1.1.6.1 Ordinary Least Squares

As the conditional expectation of the response, $\text{E}(\boldsymbol{y})$, is unaffected by heteroskedasticity, $\hat{\boldsymbol{\beta}}_{\text{OLS}}$ remains an unbiased estimator. Moreover, the argument of §1.1.5.1 still holds, meaning that $\hat{\boldsymbol{\beta}}_{\text{OLS}}$ still minimises the sum of squared residuals.

However, the variance-covariance matrix of the OLS estimator is now given by,

$$\text{Cov}(\hat{\boldsymbol{\beta}}_{\text{OLS}}) = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{\Omega}\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}. \tag{1.6}$$

$\hat{\boldsymbol{\beta}}_{\text{OLS}}$ is still a consistent estimator of $\boldsymbol{\beta}$ (Greene 2012). However, the argument of §1.1.5.2 that $\hat{\boldsymbol{\beta}}_{\text{OLS}}$ is the BLUE no longer applies. This leads us to consider alternative estimators of $\boldsymbol{\beta}$ under heteroskedasticity.

1.1.6.2 Weighted Least Squares

Under heteroskedasticity (with A1, A3-A4 still satisfied), SS_{residual}, which was minimised in §1.1.5.1 to obtain $\hat{\boldsymbol{\beta}}$, no longer has the same utility as a model precision metric, because observations with larger variances will exert a disproportionate influence on it. A modified metric could thus be considered, $SS^{W}_{\text{residual}} = \left(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}\right)'\boldsymbol{W}\left(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}\right) = \sum_{i=1}^{n} w_{ii}e_i^2$, where $\boldsymbol{W}$ is some diagonal matrix with nonnegative diagonal elements w_{ii}, $i = 1, 2, \ldots, n$.

Following an argument analogous to that by which the OLS estimator is derived (see Appendix A.1), it can be shown that minimising SS^{W}_{residual} with respect to $\boldsymbol{\beta}$ yields the Weighted Least Squares (WLS)[11] estimator,

$$\hat{\boldsymbol{\beta}}_{\text{WLS}} = \left(\boldsymbol{X}'\boldsymbol{W}\boldsymbol{X}\right)^{-1}\boldsymbol{X}'\boldsymbol{W}\boldsymbol{y}. \tag{1.7}$$

A generalisation of the Gauss-Markov Theorem holds under A1, A3-A4, by which $\hat{\boldsymbol{\beta}}_{\text{WLS}}$ is the BLUE of $\boldsymbol{\beta}$ under heteroskedasticity, provided that $\boldsymbol{W} = \boldsymbol{\Omega}^{-1}$ (i.e., $w_{ii} = \omega_i^{-1}$, $i = 1, 2, \ldots, n$). This result is proven in Appendix B.

It can henceforth be assumed that the weight matrix in $\hat{\boldsymbol{\beta}}_{\text{WLS}}$ is $\boldsymbol{W} = \boldsymbol{\Omega}^{-1}$. It also follows from the proof of the generalised Gauss-Markov theorem that,

$$\text{Cov}(\hat{\boldsymbol{\beta}}_{\text{WLS}}) = \left(\boldsymbol{X}'\boldsymbol{\Omega}^{-1}\boldsymbol{X}\right)^{-1}. \tag{1.8}$$

1.1.6.3 Maximum Likelihood Estimator

Under A1 and A3-A5, it is not possible to derive an explicit expression for the ML estimator of the $(n+p)$-vector $\boldsymbol{\theta} = [\boldsymbol{\beta}', \boldsymbol{\omega}']'$, since only n observations are available. However, the ML method does lead to a mutual relation between the parameters:

$$\hat{\boldsymbol{\theta}}_{1,\text{MLE}} = \begin{bmatrix} \hat{\boldsymbol{\beta}}_{\text{MLE}} \\ \hat{\boldsymbol{\omega}}_{\text{MLE}} \end{bmatrix} = \begin{bmatrix} \left(\boldsymbol{X}'\boldsymbol{\Omega}^{-1}\boldsymbol{X}\right)^{-1}\boldsymbol{X}'\boldsymbol{\Omega}^{-1}\boldsymbol{y} \\ (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) \circ (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) \end{bmatrix}, \tag{1.9}$$

where $\boldsymbol{\Omega}$ is a diagonal matrix with ith diagonal element ω_i. If the ω_i are known, the ML estimator of $\boldsymbol{\beta}$ will be $\hat{\boldsymbol{\beta}}_{\text{WLS}}$, defined previously in (1.7). For a derivation of (1.9), see Appendix B.1.

[11] The term Generalised Least Squares (GLS) is used in the literature for the problem of minimising SS^{W}_{residual} where $\boldsymbol{W}$ is not a diagonal matrix. WLS is thus a specialised version of GLS. Interest herein is primarily in WLS and not GLS, since A3 is assumed throughout.

1.1.6.4 Infeasibility of Weighted Least Squares

Under the classical model assumptions, the BLUE, $\hat{\beta}_{\text{OLS}}$, is feasible, being a function of observed variables. When A2 does not hold, however, the BLUE, $\hat{\beta}_{\text{WLS}}$, is a function of the observed variables and the error variances ω_i, $i = 1, 2, \ldots, n$. Hence, apart from the highly exceptional occasion where the error variances are known, the WLS estimator is infeasible: it cannot actually be computed. WLS is thus more useful as a theoretical construct than to the practitioner. Modifications to the WLS estimator to make it 'feasible' will be introduced in §2.2.1.

1.1.7 Model Residuals

1.1.7.1 OLS Residuals

The OLS residuals, $\boldsymbol{e}_{\text{OLS}} = \boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}_{\text{OLS}}$, can also be written as $\boldsymbol{e}_{\text{OLS}} = \boldsymbol{M}\boldsymbol{y}$ or as $\boldsymbol{e}_{\text{OLS}} = \boldsymbol{M}\boldsymbol{\epsilon}$.[12] For the sake of brevity, $\boldsymbol{e}_{\text{OLS}}$ will usually be denoted herein by $\boldsymbol{e}$.

1.1.7.2 OLS Residuals under Classical Linear Model Assumptions

Consider the OLS residuals $\boldsymbol{e}$ under A1-A4. Firstly, it is easily shown that $\boldsymbol{e}$ is an unbiased predictor of $\boldsymbol{\epsilon}$; that is, $\text{E}(\boldsymbol{e}) = \text{E}(\boldsymbol{\epsilon}) = \boldsymbol{0}$:

$$\text{E}(\boldsymbol{e}) = \text{E}(\boldsymbol{M}\boldsymbol{\epsilon}) = \boldsymbol{M}\,\text{E}(\boldsymbol{\epsilon}) = \boldsymbol{0}.$$

Secondly, the variance-covariance matrix of $\boldsymbol{e}$ can be derived as follows:

$$\begin{aligned}\text{Cov}(\boldsymbol{e}) &= \text{E}(\boldsymbol{e}\boldsymbol{e}') = \text{E}(\boldsymbol{M}\boldsymbol{\epsilon}\boldsymbol{\epsilon}'\boldsymbol{M}') \\ &= \boldsymbol{M}\,\text{E}(\boldsymbol{\epsilon}\boldsymbol{\epsilon}')\boldsymbol{M} \text{ (by symmetry of } \boldsymbol{M}) \\ &= \omega\boldsymbol{M} \text{ (by A1-A4 and idempotence of } \boldsymbol{M}). \qquad (1.10)\end{aligned}$$

In scalar form,

$$\text{Var}(e_i) = \text{E}(e_i^2) = \omega m_{ii} = \omega(1 - h_{ii}), \qquad (1.11)$$

where m_{ii} is the ith diagonal element of $\boldsymbol{M}$ and h_{ii} is the ith diagonal element of $\boldsymbol{H}$ (as defined in §1.1.2). Similarly,

$$\text{Cov}(e_i, e_j) = \omega m_{ij},\ i \neq j, \qquad (1.12)$$

and thus

$$\rho_{ij} = \text{Corr}(e_i, e_j) = \frac{m_{ij}}{\sqrt{m_{ii}m_{jj}}} = -\frac{h_{ij}}{\sqrt{(1 - h_{ii})(1 - h_{jj})}}. \qquad (1.13)$$

Thirdly, it can be proven that $\boldsymbol{e}$ is the Best Linear Unbiased Predictor (BLUP) of $\boldsymbol{\epsilon}$ under assumptions A1-A4. The proof is similar to the proof of the Gauss-Markov theorem (see Appendix A.2) and is outlined in Appendix B.2.

1.1.7.3 OLS Residuals under Heteroskedasticity

How are the OLS residuals affected when A1 and A3-A4 hold but A2 does not? The result $\text{E}(\boldsymbol{e}) = 0$ remains. The covariance matrix, however, becomes

$$\begin{aligned}\text{Cov}(\boldsymbol{e}) &= \text{E}(\boldsymbol{e}\boldsymbol{e}') = \text{E}(\boldsymbol{M}\boldsymbol{\epsilon}\boldsymbol{\epsilon}'\boldsymbol{M}') \\ &= \boldsymbol{M}\,\text{E}(\boldsymbol{\epsilon}\boldsymbol{\epsilon}')\boldsymbol{M} \text{ (by symmetry of } \boldsymbol{M}) \\ &= \boldsymbol{M}\boldsymbol{\Omega}\boldsymbol{M}. \qquad (1.14)\end{aligned}$$

[12] Note, however, that this does *not* mean that the unobserved random errors can be recovered from the OLS residuals using the transformation $\boldsymbol{\epsilon} = \boldsymbol{M}^{-1}\boldsymbol{e}$, because $\boldsymbol{M}$ is singular.

By working through the matrix multiplication, the result in (1.14) can be expressed in scalar form as,

$$\mathrm{Var}(e_i) = \mathrm{E}(e_i^2) = (\boldsymbol{M\Omega M})_{ii} = \sum_{k=1}^{n} \omega_k m_{ik}^2,\ i \in \{1, 2, \ldots, n\}\,, \tag{1.15}$$

and

$$\mathrm{Cov}(e_i, e_j) = \sum_{k=1}^{n} \omega_k m_{ik} m_{kj},\ i \in \{1, 2, \ldots, n\}\,, j \in \{1, 2, \ldots, n\}\,, i \neq j. \tag{1.16}$$

Moreover,

$$\mathrm{Corr}(e_i, e_j) = \frac{\displaystyle\sum_{k=1}^{n} \omega_k m_{ik} m_{kj}}{\sqrt{\displaystyle\sum_{k=1}^{n} \omega_k m_{ik}^2 \sum_{\ell=1}^{n} \omega_\ell m_{j\ell}^2}} = \rho_{ij}. \tag{1.17}$$

Note that (1.15) and (1.16) can also be expressed in terms of the elements of $\boldsymbol{H}$:[13]

$$\mathrm{Var}(e_i) = \mathrm{E}(e_i^2) = m_{ii}^2 \omega_i + \sum_{k \neq i} \omega_k m_{ik}^2 = (1 - h_{ii})^2 \omega_i + \sum_{k \neq i} \omega_k h_{ik}^2, \tag{1.18}$$

and

$$\begin{aligned} \mathrm{Cov}(e_i, e_j) &= \omega_i m_{ii} m_{ij} + \omega_j m_{jj} m_{ij} + \sum_{k \notin \{i,j\}} \omega_k m_{ik} m_{kj} \\ &= -h_{ij} \left[\omega_i (1 - h_{ii}) + \omega_j (1 - h_{jj})\right] + \sum_{k \notin \{i,j\}} \omega_k h_{ik} h_{kj}. \end{aligned} \tag{1.19}$$

Note that, following an argument analogous to that of Horn et al. (1975)—who are, however, working with WLS rather than OLS—$\mathrm{E}(e_i^2)$ can be simplified as follows:

$$\begin{aligned} \mathrm{E}(e_i^2) &= (1 - h_{ii})^2 \omega_i + \sum_{k \neq i} \omega_k h_{ik}^2 \\ &= (1 - 2h_{ii}) \omega_i + \sum_{k=1}^{n} \omega_k h_{ik}^2 \\ &= (1 - 2h_{ii}) \omega_i + c_i, \text{ where} \\ c_i &= \sum_{k=1}^{n} \boldsymbol{X}'_{i\cdot} (\boldsymbol{X}'\boldsymbol{X})^{-1} \boldsymbol{X}_{k\cdot} \omega_k \boldsymbol{X}'_{k\cdot} (\boldsymbol{X}'\boldsymbol{X})^{-1} \boldsymbol{X}_{i\cdot} \\ &= \boldsymbol{X}'_{i\cdot} (\boldsymbol{X}'\boldsymbol{X})^{-1} \boldsymbol{X}'\boldsymbol{\Omega}\boldsymbol{X} (\boldsymbol{X}'\boldsymbol{X})^{-1} \boldsymbol{X}_{i\cdot} = (\boldsymbol{H\Omega H})_{ii}. \end{aligned} \tag{1.20}$$

Working from (1.15), the sum of squared residuals $\boldsymbol{e}'\boldsymbol{e}$ has conditional expectation given by (1.21) under heteroskedasticity:

[13] Since $m_{ii} = \sum_{k=1}^{n} m_{ik}^2$, it can be shown that $\sum_{k \neq i} h_{ik}^2 = (1 - h_{ii}) h_{ii}$. From this it follows that $(1 - h_{ii})^2$, the coefficient of ω_i in (1.18), exceeds the sum of all the other coefficients as long as $h_{ii} < \frac{1}{2}$. This will in general be true in most cases, particularly when the sample size n is large relative to p, since the average h_{ii} value is $\frac{p}{n}$. Thus, with the exception of very high-leverage observations, $\mathrm{Var}(e_i)$ will be dominated by the first term in (1.15) and (1.18).

$$
\begin{aligned}
\mathrm{E}(\boldsymbol{e}'\boldsymbol{e}) &= \sum_{k=1}^{n}\sum_{i=1}^{n}\omega_i m_{ki}^2 \\
&= \sum_{i=1}^{n}\omega_i \sum_{k=1}^{n} m_{ki}^2 \\
&= \sum_{i=1}^{n}\omega_i m_{ii} = \sum_{i=1}^{n}\omega_i(1-h_{ii}).
\end{aligned} \tag{1.21}
$$

1.1.7.4 WLS Residuals

Define $\boldsymbol{H}_{\boldsymbol{\Omega}} = \boldsymbol{X}\left(\boldsymbol{X}'\boldsymbol{W}\boldsymbol{X}\right)^{-1}\boldsymbol{X}'\boldsymbol{W} = \boldsymbol{X}\left(\boldsymbol{X}'\boldsymbol{\Omega}^{-1}\boldsymbol{X}\right)^{-1}\boldsymbol{X}'\boldsymbol{\Omega}^{-1}$ as a WLS generalisation of the hat matrix defined previously in §1.1.2. The generalised annihilator matrix is similarly defined as $\boldsymbol{M}_{\boldsymbol{\Omega}} = \boldsymbol{I}_n - \boldsymbol{H}_{\boldsymbol{\Omega}}$. The key properties of these matrices are preserved under this generalisation. Both matrices are symmetric and idempotent, and their traces remain p and $n-p$ respectively. Furthermore, the relations $\hat{\boldsymbol{y}}_{\mathrm{WLS}} = \boldsymbol{H}_{\boldsymbol{\Omega}}\boldsymbol{y}$ (where $\hat{\boldsymbol{y}}_{\mathrm{WLS}} = \boldsymbol{X}\hat{\boldsymbol{\beta}}_{\mathrm{WLS}}$) and $\boldsymbol{e}_{\mathrm{WLS}} = \boldsymbol{M}_{\boldsymbol{\Omega}}\boldsymbol{y} = \boldsymbol{M}_{\boldsymbol{\Omega}}\boldsymbol{\epsilon}$ hold (where $\boldsymbol{e}_{\mathrm{WLS}} = \boldsymbol{y} - \hat{\boldsymbol{y}}_{\mathrm{WLS}}$).

In consequence of this, it can be shown, using the same approaches as those taken in §1.1.7.3, that the following results hold (under A1, A3-A4) concerning the WLS residuals $\boldsymbol{e}_{\mathrm{WLS}}$:

$$\mathrm{E}(\boldsymbol{e}_{\mathrm{WLS}}) = \boldsymbol{0} \tag{1.22}$$

and

$$\mathrm{Cov}(\boldsymbol{e}_{\mathrm{WLS}}) = \boldsymbol{M}_{\boldsymbol{\Omega}}\boldsymbol{\Omega}\boldsymbol{M}_{\boldsymbol{\Omega}}. \tag{1.23}$$

It follows that, in scalar form,

$$\mathrm{Var}(e_{i,\mathrm{WLS}}) = \mathrm{E}(e_{i,\mathrm{WLS}}^2) = (\boldsymbol{M}_{\boldsymbol{\Omega}}\boldsymbol{\Omega}\boldsymbol{M}_{\boldsymbol{\Omega}})_{ii} = \sum_{k=1}^{n}\omega_k\mathfrak{m}_{ik}^2, i\in\{1,2,\ldots,n\}, \tag{1.24}$$

and

$$\mathrm{Cov}(e_{i,\mathrm{WLS}}, e_{j,\mathrm{WLS}}) = \sum_{k=1}^{n}\omega_k\mathfrak{m}_{ik}\mathfrak{m}_{kj}, i\in\{1,2,\ldots,n\}, j\in\{1,2,\ldots,n\}, i\neq j, \tag{1.25}$$

where $\mathfrak{m}_{ik}$ is the (i,k)th element of $\boldsymbol{M}_{\boldsymbol{\Omega}}$.

1.1.7.5 Best Linear Unbiased Scalar-Covariance-Matrix Residuals

Under homoskedasticity, the OLS residuals have covariance matrix $\omega\boldsymbol{M}$ and not $\omega\boldsymbol{I}_n$. Thus the OLS residuals are not themselves homoskedastic or uncorrelated even when the errors are. This creates a problem when using the OLS residuals for heteroskedasticity diagnostics. Graphical diagnostics may suffer, because when heterogeneity in the squared OLS residuals appears on a plot (for instance), one cannot be certain whether one is seeing evidence of heterogeneity in the error variances or merely heterogeneity in the diagonal elements of $\boldsymbol{M}$. The heterogeneity of residual variances can be corrected using a simple transformation $e_i m_{ii}^{-1/2}$, which will have constant variance ω under A1-A4. However, this does not solve the problem of autocorrelation in the OLS residuals, which may hamper inferential diagnostics since it is easier to construct null distributions from independent or at least uncorrelated random variables.

Theil (1965, 1968) sought to address this problem. Specifically, he considered a class of linear unbiased predictors of $\boldsymbol{\epsilon}$ of the form $\boldsymbol{e} = \boldsymbol{A}\boldsymbol{y}$ (see §1.1.7.2) such that $\boldsymbol{A}\boldsymbol{y}$ has a scalar (conditional) covariance matrix. Thus, some increase in the mean squared prediction error is permitted in order to achieve residuals that are uncorrelated. This class of residuals may be termed ‘LUS’ (linear unbiased scalar-covariance-matrix).

Theil (1965) noted from the outset that, besides compromising on the mean squared prediction error, this approach requires that the scalar covariance matrix will (like $\boldsymbol{M}$) be of rank $n-p$ and not n, because p degrees of freedom are inevitably lost due to estimation of $\boldsymbol{\beta}$. The scalar covariance matrix (with the constant factored out) will thus consist of $n-p$ ones and p zeroes on the diagonal, and zeroes elsewhere. One can, however, choose which p errors are 'sacrificed' (not predicted), by specifying an $(n-p) \times n$ selection matrix $\boldsymbol{J}'$ obtained by deleting any p rows from $\boldsymbol{I}_n$. However, a restriction applies to the choice of $\boldsymbol{J}'$: if $\boldsymbol{X}_0$ denotes the $p \times p$ submatrix of $\boldsymbol{X}$ consisting of those rows of $\boldsymbol{X}$ corresponding to the rows deleted from $\boldsymbol{I}_n$ to obtain $\boldsymbol{J}'$, then $\boldsymbol{X}_0$ must be nonsingular.

Among the class of 'LUS' residual vectors $\boldsymbol{u}$, Theil (1965, 1968) derived the 'best' in the sense of minimising the mean squared error for predicting $\boldsymbol{J}'\boldsymbol{\epsilon}$, that is for predicting those $n-p$ errors that have not been jettisoned:

$$\mathrm{E}\left[(\boldsymbol{u}-\boldsymbol{J}'\boldsymbol{\epsilon})'(\boldsymbol{u}-\boldsymbol{J}\boldsymbol{\epsilon})\right].$$

Because p errors are being ignored, the 'LUS' constraints can also be restated as:

(i) $\boldsymbol{u} = \boldsymbol{A}'\boldsymbol{y}$ for some $n \times (n-p)$ matrix $\boldsymbol{A}$;

(ii) $\mathrm{E}(\boldsymbol{u}-\boldsymbol{J}'\boldsymbol{\epsilon}) = \boldsymbol{0}$ (which implies that $\boldsymbol{A}'\boldsymbol{X} = 0$ and $\boldsymbol{u} = \boldsymbol{A}'\boldsymbol{\epsilon}$); and

(iii) $\mathrm{Cov}(\boldsymbol{u}) = \omega \boldsymbol{I}_{n-p}$ (which implies that $\boldsymbol{A}'\boldsymbol{A} = \boldsymbol{I}_{n-p}$).

Theil (1965) shows that the solution to this optimisation problem, which he dubs the Best Linear Unbiased Scalar-Covariance-Matrix (BLUS) residuals (here denoted $\boldsymbol{e}_{\mathrm{BLUS}}$), is achieved by $\boldsymbol{A} = \boldsymbol{M}\boldsymbol{J}(\boldsymbol{J}'\boldsymbol{M}\boldsymbol{J})^{-1/2}$, where $(\boldsymbol{J}'\boldsymbol{M}\boldsymbol{J})^{-1/2}$ is the positive definite square root of $(\boldsymbol{J}'\boldsymbol{M}\boldsymbol{J})^{-1}$. An accessible version of the proof, which uses Lagrange multipliers but follows similar logic to the proof in Appendix B.2 that the OLS residual vector is the BLUP of the random error vector, can be found in Magnus and Sinha (2005).

Huang and Bolch (1974) note that the optimal $\boldsymbol{A}$ matrix that gives rise to the BLUS residuals satisfies (1.26) and (1.27), where a_{ij} is the (i,j)th element of $\boldsymbol{A}$:

$$\sum_{i=1}^{n} a_{ij}a_{ik} = 0, \tag{1.26}$$

and

$$\sum_{i=1}^{n} (a_{ij}a_{ik})^2 \neq 0, \tag{1.27}$$

for all $j \neq k$. They are then able to show that, while the BLUS residuals are by definition *uncorrelated*, they are *independent* if and only if A5 holds. The implication is that, if the errors are not normally distributed, the BLUS residuals are not mutually independent, and may not perform better than the OLS residuals in heteroskedasticity diagnostics in that case.

Using the moment-generating function (MGF) technique employed in §1.1.8, it can be demonstrated that the BLUS residuals are normally distributed under A1-A5. Since $\boldsymbol{e}_{\mathrm{BLUS}} = \boldsymbol{A}'\boldsymbol{y}$,

$$\begin{aligned} M_{\boldsymbol{e}_{\mathrm{BLUS}}}(\boldsymbol{t}) &= M_{\boldsymbol{y}}(\boldsymbol{A}\boldsymbol{t}) \\ &= \exp\left\{\boldsymbol{\beta}'\boldsymbol{X}'\boldsymbol{A}\boldsymbol{t} + \frac{\omega}{2}(\boldsymbol{A}\boldsymbol{t})'\boldsymbol{A}\boldsymbol{t}\right\} \\ &= \exp\left\{(\boldsymbol{A}'\boldsymbol{X}\boldsymbol{\beta})'\boldsymbol{t} + \frac{\omega}{2}\boldsymbol{t}'\boldsymbol{A}'\boldsymbol{A}\boldsymbol{t}\right\} \\ &= \exp\left\{\boldsymbol{0}'\boldsymbol{t} + \frac{\omega}{2}\boldsymbol{t}'\boldsymbol{t}\right\}. \end{aligned}$$

It follows that $\boldsymbol{e}_{\mathrm{BLUS}} \sim N(\boldsymbol{0}, \omega \boldsymbol{I}_{n-p})$.

There are other types of residuals that have been used in connection with the linear regression model, such as recursive residuals (Magnus and Sinha 2005), but these will not be discussed here as they will play no further role in the study.

1.1.8 Inference on Model Parameters under Assumptions A1-A5

Assumptions A1-A5 enable construction of exact confidence intervals and hypothesis tests on $\boldsymbol{\beta}$. It was noted in §1.1.3 that A1-A5 together entail that $\boldsymbol{\epsilon} \sim N(\boldsymbol{0}, \omega \boldsymbol{I}_n)$. Since $\boldsymbol{y}$ differs from $\boldsymbol{\epsilon}$ only by a location shift of $\boldsymbol{X}\boldsymbol{\beta}$, it follows immediately that $\boldsymbol{y} \sim N(\boldsymbol{X}\boldsymbol{\beta}, \omega \boldsymbol{I}_n)$. Now, the MGF of a random vector $\boldsymbol{x} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is given by (1.28) (Kotz et al. 2000, p. 108):

$$M_{\boldsymbol{X}}(\boldsymbol{t}) = \mathrm{E}\left(\exp\left\{\boldsymbol{X}'\boldsymbol{t}\right\}\right) = \exp\left\{\boldsymbol{\mu}'\boldsymbol{t} + \frac{1}{2}\boldsymbol{t}'\boldsymbol{\Sigma}\boldsymbol{t}\right\}. \tag{1.28}$$

It follows immediately from A1-A5 that $\boldsymbol{\epsilon}$ has MGF $M_{\boldsymbol{\epsilon}}(\boldsymbol{t}) = \exp\left\{\boldsymbol{0}'\boldsymbol{t} + \frac{\omega}{2}\boldsymbol{t}'\boldsymbol{t}\right\}$ and that $\boldsymbol{y}$ has MGF $M_{\boldsymbol{y}}(\boldsymbol{t}) = \exp\left\{\boldsymbol{\beta}'\boldsymbol{X}'\boldsymbol{t} + \frac{\omega}{2}\boldsymbol{t}'\boldsymbol{t}\right\}$.

Any linear function of a normally distributed random vector is also normally distributed (Gut 2009, Theorem 3.1). Thus, it follows from the normality of $\boldsymbol{y}$ that the random vectors $\hat{\boldsymbol{\beta}}_{\text{OLS}} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{y}$, $\hat{\boldsymbol{y}} = \boldsymbol{H}\boldsymbol{y}$, and $\boldsymbol{e} = \boldsymbol{M}\boldsymbol{y}$ are all normally distributed under A1-A5. Writing the MGFs of these random vectors in the form of (1.28) is a convenient way to derive their mean vectors and variance-covariance matrices. One can therefore make use of a property of joint MGFs that, if $\boldsymbol{Y}$ is a random vector and $\boldsymbol{A}$ is a nonstochastic matrix, $M_{\boldsymbol{AY}}(\boldsymbol{t}) = M_{\boldsymbol{Y}}(\boldsymbol{A}'\boldsymbol{t})$.[14] Accordingly, the following exact distributional results can be derived under A1-A5:

$$\hat{\boldsymbol{\beta}} \sim N(\boldsymbol{\beta}, \omega(\boldsymbol{X}'\boldsymbol{X})^{-1}), \tag{1.29}$$

$$\hat{\boldsymbol{y}} \sim N(\boldsymbol{X}\boldsymbol{\beta}, \omega\boldsymbol{H}), \text{ and} \tag{1.30}$$

$$\boldsymbol{e} \sim N(\boldsymbol{0}, \omega\boldsymbol{M}). \tag{1.31}$$

More details on these three results are given in Appendix B.3. These distributional findings are consistent with the earlier finding (§1.1.7.2) that, under A1-A4, $\mathrm{E}(\boldsymbol{e}) = \boldsymbol{0}$ and $\mathrm{Cov}(\boldsymbol{e}) = \omega\boldsymbol{M}$. Note, however, that because $\boldsymbol{M}$ is singular, the joint probability density function (PDF) of $\boldsymbol{e}$ does not exist: $\boldsymbol{e}$ has a 'degenerate' multivariate normal distribution.

Two further results, together with (1.29), allow construction of an exact t-test for significance of individual parameters (e.g., of the null hypothesis $\beta_j = 0$ for $j \in \{1, 2, \ldots, p\}$). The first of these results is,

$$\omega^{-1}\boldsymbol{e}'\boldsymbol{e} \sim \chi^2(n-p). \tag{1.32}$$

The second result required for construction of the exact t-test is the independence of $\hat{\boldsymbol{\beta}}$ and $\hat{\omega}_{\text{ub}}$. Proofs of both of these results are given in Appendix B.4. Thus, under the null hypothesis $\beta_j = 0$, from the definition of Student's t distribution,

$$T_{\hat{\beta}_j} = \frac{\hat{\beta}_j - 0}{\sqrt{\omega(\boldsymbol{X}'\boldsymbol{X})^{-1}_{jj}}\sqrt{\frac{\boldsymbol{e}'\boldsymbol{e}}{\omega(n-p)}}} = \frac{\hat{\beta}_j}{\sqrt{\hat{\omega}_{\text{ub}}(\boldsymbol{X}'\boldsymbol{X})^{-1}_{jj}}} \sim t(n-p),\ j \in \{1, 2, \ldots, p\}, \tag{1.33}$$

where $\hat{\beta}_j$ is the jth element of $\hat{\boldsymbol{\beta}}_{\text{OLS}}$. Thus, $T_{\hat{\beta}_j}$ can be used for an exact test of the null hypothesis $\beta_j = 0$ against the alternative $\beta_j \neq 0$, $j \in \{1, 2, \ldots, p\}$. Similarly, an exact $(1-\alpha)100\%$ confidence interval for β_j is given by,

$$\hat{\beta}_j \pm t_{\alpha/2, n-p}\sqrt{\hat{\omega}_{\text{ub}}(\boldsymbol{X}'\boldsymbol{X})^{-1}_{jj}}, \tag{1.34}$$

where $(\boldsymbol{X}'\boldsymbol{X})^{-1}_{jj}$ is the jth diagonal element of $(\boldsymbol{X}'\boldsymbol{X})^{-1}$.

1.1.9 Invalidity of Classical Hypothesis Tests under Heteroskedasticity

Consider the situation of A1 and A3-A5 (homoskedasticity is violated but the other classical linear model assumptions hold). The normality of the random vectors $\hat{\boldsymbol{\beta}}_{\text{OLS}}$, $\hat{\boldsymbol{y}}$, and $\boldsymbol{e}$ still holds, but their moments are different than under homoskedasticity. The MGF technique introduced in §1.1.8 can be used to derive these.

It follows immediately from the assumptions that $\boldsymbol{\epsilon}$ has MGF $M_{\boldsymbol{\epsilon}}(\boldsymbol{t}) = \exp\left\{\boldsymbol{0}'\boldsymbol{t} + \frac{1}{2}\boldsymbol{t}'\boldsymbol{\Omega}\boldsymbol{t}\right\}$ and that $\boldsymbol{y}$ has MGF $M_{\boldsymbol{y}}(\boldsymbol{t}) = \exp\left\{\boldsymbol{\beta}'\boldsymbol{X}'\boldsymbol{t} + \frac{1}{2}\boldsymbol{t}'\boldsymbol{\Omega}\boldsymbol{t}\right\}$. Hence, using properties of MGFs, it can be shown that,

[14] This property follows straightforwardly from the definition of the joint MGF:

$$\mathrm{E}(\exp\left\{(\boldsymbol{AY})'\boldsymbol{t}\right\}) = \mathrm{E}(\exp\left\{\boldsymbol{Y}'\boldsymbol{A}'\boldsymbol{t}\right\}) = M_{\boldsymbol{Y}}(\boldsymbol{A}'\boldsymbol{t}).$$

$$\begin{aligned} \hat{\beta}_{\text{OLS}} &\sim N(\boldsymbol{\beta}, (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{\Omega}\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}), & (1.35)\\ \hat{\boldsymbol{y}} &\sim N(\boldsymbol{X}\boldsymbol{\beta}, \boldsymbol{H}\boldsymbol{\Omega}\boldsymbol{H}), \text{ and} & (1.36)\\ \boldsymbol{e} &\sim N(\boldsymbol{0}, \boldsymbol{M}\boldsymbol{\Omega}\boldsymbol{M}). & (1.37) \end{aligned}$$

These three results are derived explicitly in Appendix B.5. They align with the results on the moments of $\boldsymbol{e}$ obtained in §1.1.7.3.

Under heteroskedasticity, $\hat{\boldsymbol{\beta}}$ is still an unbiased and consistent estimator of $\boldsymbol{\beta}$, but the conditional variances of the elements of $\hat{\boldsymbol{\beta}}$ are different than under homoskedasticity. Thus, the standard error estimates used in the test statistic and confidence interval formulas (1.33) and (1.34) are no longer valid (Greene 2012, Wooldridge 2013).

Approaches in the literature to inference on linear regression model parameters under heteroskedasticity will be discussed in §2.4.

1.1.10 Leverage and Influence in the Linear Model

1.1.10.1 Leverage Scores

The 'hat matrix,' $\boldsymbol{H} = \boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'$, was introduced previously in §1.1.2. Since $\hat{\boldsymbol{y}} = \boldsymbol{H}\boldsymbol{y}$, it follows that $\dfrac{\partial \hat{\boldsymbol{y}}}{\partial \boldsymbol{y}} = \boldsymbol{H}$. Accordingly, one can interpret h_{ij}, the i,jth element of $\boldsymbol{H}$, as measuring the degree of influence exerted by the jth observed response on the ith predicted response. The diagonal elements h_{ii} are measures of self-influence and are referred to as 'leverage scores'; observations with high leverage are particularly influential. Since each h_{ii} can be written as a quadratic form, $\boldsymbol{X}'_{i\cdot}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}_{i\cdot}$ (where $\boldsymbol{X}'_{i\cdot}$ denotes the ith row of $\boldsymbol{X}$), it follows that $h_{ii} \geq 0$ for all $i \in \{1, 2, \ldots, n\}$. Moreover, the symmetry and idempotence properties of $\boldsymbol{H}$ imply that $h_{ii} = \displaystyle\sum_{j=1}^{n} h_{ij}^2 \geq h_{ii}^2$, from which it follows that $h_{ii} \leq 1$. Cook and Weisberg (1982) further show that, for models with an intercept, $h_{ii} \geq \dfrac{1}{n}$. The extreme case $h_{ii} = 1$ implies that $h_{ij} = 0, j \neq i$ and that $y_i = \hat{y}_i$ $(e_i = 0)$.

Rencher and Schaalje (2008) show from properties of the trace that $\displaystyle\sum_{i=1}^{n} h_{ii} = p$, which implies that the average leverage score is $\dfrac{p}{n}$. Cook and Weisberg (1982) express the h_{ii} in terms of the eigenvalues and eigenvectors of $\boldsymbol{\mathcal{X}}'\boldsymbol{\mathcal{X}}$, where $\boldsymbol{\mathcal{X}}$ is a $n \times (p-1)$ centered design matrix without the intercept column, having ith row $\boldsymbol{\mathcal{X}}'_{i\cdot}$. Specifically, if $\mu_2 \geq \mu_2 \geq \ldots \geq \mu_p$ denote the eigenvalues of $\boldsymbol{\mathcal{X}}'\boldsymbol{\mathcal{X}}$ and $\boldsymbol{p}_2, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_p$ denote the corresponding eigenvectors, one can write,

$$\begin{aligned} h_{ii} &= \frac{1}{n} + \sum_{k=2}^{p} \frac{(\boldsymbol{p}'_k \boldsymbol{\mathcal{X}}_{i\cdot})^2}{\mu_k} & (1.38)\\ &= \frac{1}{n} + \boldsymbol{\mathcal{X}}'_{i\cdot}\boldsymbol{\mathcal{X}}_{i\cdot} \sum_{k=2}^{p} \frac{\cos^2(\theta_{ki})}{\mu_k}, & (1.39) \end{aligned}$$

where θ_{ki} is the angle between $\boldsymbol{p}_j$ and $\boldsymbol{\mathcal{X}}'_{i\cdot}$, and $\cos(\theta_{ki}) = \dfrac{\boldsymbol{p}'_k \boldsymbol{\mathcal{X}}'_{i\cdot}}{\sqrt{\boldsymbol{\mathcal{X}}'_{i\cdot}\boldsymbol{\mathcal{X}}_{i\cdot}}}$. On this basis, they argue that h_{ii} is large if (1) $\boldsymbol{\mathcal{X}}'_{i\cdot}\boldsymbol{\mathcal{X}}_{i\cdot}$ is large ($\boldsymbol{X}'_{i\cdot}$ is far from the centre of the design distribution), and (2) $\boldsymbol{\mathcal{X}}'_{i\cdot}$ is substantially in the direction of an eigenvector corresponding to an eigenvalue of $\boldsymbol{\mathcal{X}}'\boldsymbol{\mathcal{X}}$.

Fidell and Tabachnick (2003) give an alternative expression for h_{ii} in terms of Mahalanobis distance. Define the squared Mahalanobis distance of the point $\boldsymbol{\mathcal{X}}_{i\cdot}$ with reference to the empirical mean vector $\bar{\boldsymbol{\mathcal{X}}}$ and the empirical covariance matrix $\boldsymbol{S}$ as,

$$\text{MH}^2(\boldsymbol{\mathcal{X}}_{i\cdot}; \bar{\boldsymbol{\mathcal{X}}}, \boldsymbol{S}) = \left(\boldsymbol{\mathcal{X}}_{i\cdot} - \bar{\boldsymbol{\mathcal{X}}}\right)' \boldsymbol{S}^{-1} \left(\boldsymbol{\mathcal{X}}_{i\cdot} - \bar{\boldsymbol{\mathcal{X}}}\right). \tag{1.40}$$

Then, the ith leverage score, $i = 1, 2, \ldots, n$, can be written as,

$$h_{ii} = \frac{1}{n} + \frac{\text{MH}^2(\boldsymbol{\mathcal{X}}_{i\cdot}; \bar{\boldsymbol{\mathcal{X}}}, \boldsymbol{S})}{n-1}. \tag{1.41}$$

Note that the mean vector $\bar{\boldsymbol{\mathcal{X}}} = \mathbf{0}$ in this case since the $\boldsymbol{\mathcal{X}}$ matrix has been centered. Moreover, in this case $\boldsymbol{S} = (n-1)^{-1}\boldsymbol{\mathcal{X}}'\boldsymbol{\mathcal{X}}$.

The off-diagonal elements of $\boldsymbol{H}$ can similarly be expressed in terms of the spectral decomposition of $\boldsymbol{\mathcal{X}}$, as in (1.42) and (1.43):

$$h_{ij} = \frac{1}{n} + \sum_{k=2}^{p} \frac{\boldsymbol{p}_k'\boldsymbol{\mathcal{X}}_{i\cdot}\boldsymbol{p}_k'\boldsymbol{\mathcal{X}}_{j\cdot}}{\mu_k} \tag{1.42}$$

$$= \frac{1}{n} + \sqrt{\boldsymbol{\mathcal{X}}_{i\cdot}'\boldsymbol{\mathcal{X}}_{i\cdot}\boldsymbol{\mathcal{X}}_{j\cdot}'\boldsymbol{\mathcal{X}}_{j\cdot}} \sum_{k=2}^{p} \frac{\cos(\theta_{ki})\cos(\theta_{kj})}{\mu_k}. \tag{1.43}$$

The off-diagonal elements can also be expressed in a way analogous to (1.41), as per (1.44):

$$h_{ij} = \frac{1}{n} + \frac{(\boldsymbol{\mathcal{X}}_{i\cdot} - \mathbf{0})'\,\boldsymbol{S}^{-1}\,(\boldsymbol{\mathcal{X}}_{j\cdot} - \mathbf{0})}{n-1}. \tag{1.44}$$

Leverage scores h_{ii} (or annihilator matrix elements $m_{ii} = 1 - h_{ii}$) will play a major role in the heteroskedasticity-consistent covariance matrix estimators to be reviewed in §2.3, as well as in the new error variance estimation models to be introduced in Chapter 3.

1.1.10.2 Studentised Residuals

If one defines ‘studentisation’ as the division of an estimator by an estimator of its standard error, the *internally* studentised OLS residuals r_i (assuming A2) can be defined, following Cook and Weisberg (1982), as

$$r_i = \frac{e_i}{\sqrt{\hat{\omega}_{\text{ub}}(1-h_{ii})}},\ i = 1, 2, \ldots, n. \tag{1.45}$$

Cook and Weisberg (1982) show that, under A1-A5, $(n-p)^{-1}r_i^2$ follows a Beta distribution with parameters $1/2$ and $(n-p-1)/2$.

Alternatively, the *externally* studentised OLS residual t_i (again, assuming A2) provides a standardised estimate of the outlyingness of the ith predicted value that does not depend on the ith observation itself. Define $\hat{\boldsymbol{\beta}}_{(-i)}, i \in \{1, 2, \ldots, n\}$, as the OLS coefficient estimate obtained when the ith observation is omitted (sometimes referred to as a jackknife estimate). Similarly, define

$$\hat{y}_{i,(-i)} = \boldsymbol{X}_{i\cdot}'\hat{\boldsymbol{\beta}}_{(-i)}, \tag{1.46}$$

$$\boldsymbol{e}_{i,(-i)} = y_i - \hat{y}_{i,(-i)}, \tag{1.47}$$

and

$$\hat{\omega}_{(-i)} = \frac{(n-p)\hat{\omega}_{\text{ub}} - e_i^2/(1-h_{ii})}{n-p-1} = \hat{\omega}_{\text{ub}}\left(\frac{n-p-r_i^2}{n-p-1}\right). \tag{1.48}$$

The ith externally studentised residual, $i = 1, 2, \ldots, n$, can then be written as,

$$t_i = \frac{e_i}{\sqrt{\hat{\omega}_{(-i)}(1-h_{ii})}}$$

$$= r_i\sqrt{\frac{n-p-1}{n-p-r_i^2}}. \tag{1.49}$$

The studentised residuals are introduced only for the sake of defining Cook’s Distance below, and will not be directly used in this research.

1.1.10.3 Cook's Distance

Cook (1977) proposes "an easily interpretable measure [of outlyingness] that combines information from both [the internally studentised residual] and [the variance of the OLS residual], and that will naturally isolate 'critical' values" (p. 15). This measure, which has become known as Cook's Distance, is based on the formula for the confidence ellipsoid for $\boldsymbol{\beta}$ under A1-A5. It can be expressed in terms of the jackknife OLS parameter estimate $\hat{\boldsymbol{\beta}}_{(-i)}$ as,

$$D_i = \frac{(\hat{\boldsymbol{\beta}}_{(-i)} - \hat{\boldsymbol{\beta}})' \boldsymbol{X}' \boldsymbol{X} (\hat{\boldsymbol{\beta}}_{(-i)} - \hat{\boldsymbol{\beta}})}{p \hat{\omega}_{\text{ub}}}. \tag{1.50}$$

Cook (1977) shows that D_i can also be expressed in terms of the OLS and internally studentised residuals, respectively, as per (1.51) and (1.52):

$$\begin{aligned} D_i &= \frac{e_i^2}{\hat{\omega}_{\text{ub}}(1 - h_{ii})} \frac{h_{ii}}{p(1 - h_{ii})} && (1.51) \\ &= \frac{r_i^2}{p} \frac{h_{ii}}{1 - h_{ii}}, && (1.52) \end{aligned}$$

where $h_{ii}/(1 - h_{ii})$ is, under the classical assumptions, the ratio $\dfrac{\text{Var}(\hat{y}_i)}{\text{Var}(e_i)}$, which measures 'the relative sensitivity of the estimate, $\hat{\boldsymbol{\beta}}$, to potential outlying values at each data point' (Cook 1977, p. 16).

Cook (1977) provides a modified distance metric that can be used if one is interested only in certain elements of $\boldsymbol{\beta}$. For instance, in the special case where one is interested in only one element β_j,

$$D_i(\beta_j) = r_i^2 \left(\frac{h_{ii} - h_{ii}^{(-j)}}{1 - h_{ii}} \right), \tag{1.53}$$

where $h_{ii}^{(-j)}$ is the ith diagonal element of $\boldsymbol{H}^{(-j)}$, the hat matrix constructed from the design matrix with the jth column omitted.

Cook (1979), building on the earlier work, relates influential observations to the independent variable hull (IVH), being the smallest convex set containing all of the design points. He notes that 'a large value of D_i indicates that the associated ith point has a strong influence on the estimate of $\boldsymbol{\beta}$' (Cook 1979, p. 169). He proceeds to show that the point with largest leverage scores h_{ii} must lie on the boundary of the IVH, and that the h_{ii} can thus be thought of as measures of outlyingness (though it does not follow that the point with largest h_{ii} necessarily has the largest Euclidean distance from the centroid of $\boldsymbol{X}$).

Cook's Distance will be encountered again in §2.3.8, as one method for estimating the covariance matrix $\boldsymbol{\Omega}$ under heteroskedasticity makes use of it.

1.2 Research Problem

Heteroskedasticity in the linear model—the violation of the assumption of constant error variance—has been widely researched, both in terms of diagnostic methods and remedial measures. Diagnostic tests continue to proliferate, but many of these remain in obscurity, inaccessible to practitioners in statistical software. Meanwhile, the remedial measures vary depending on the end goal (e.g., estimation of the coefficient vector $\boldsymbol{\beta}$ vs. inference on its elements). There is a need for a unified approach to handling heteroskedasticity in the linear model that can be used regardless of the end goal, and that is accessible to practitioners via an R software package.

1.3 Research Objectives

The objectives of this research project are as follows:

1. To review and catalogue the many heteroskedasticity testing methods that have appeared in the literature over the past few decades;
2. To program these heteroskedasticity tests and make them accessible to practitioners via a package in R statistical software;

3. To evaluate the role (if any) of heteroskedasticity tests in handling the problem of heteroskedasticity in the linear model;
4. To develop a new method of handling heteroskedasticity in the linear model by direct estimation of the error variances using a suitable auxiliary regression model;
5. To show empirically, using Monte Carlo (MC) simulations, that the new method performs well relative to existing methods in terms of meaningful performance metrics;
6. To show empirically, using MC simulations, that the new method is robust in certain meaningful respects; and
7. To make the new method accessible to practitioners via a package in R statistical software.

1.4 Chapter Summary and Way Forward

Thus far, a thorough background has been provided on the linear regression model, its basic notation and classical assumptions, the classical approach (under assumptions A1-A5) to estimation of and inference on parameters, and how estimation and inference are affected by heteroskedasticity. Besides this, two quantities relevant to the study of heteroskedasticity in linear regression—namely, model residuals and leverage scores—have been introduced and discussed in some detail. With this background in hand, the research problem was stated and research objectives were set out.

An overview of the remaining chapters runs as follows. In Chapter 2 (Literature Review), a thorough review of the academic literature will be conducted to take stock of existing methods for detecting and handling heteroskedasticity. This will facilitate the identification of gaps that provide an opportunity for the development of new approaches and methods for handling heteroskedasticity.

Chapter 3 (Methodology) opens with some new theoretical results on the statistical properties of squared OLS residuals. However, the main contribution of this chapter is to propose two new classes of auxiliary regression models that can be used to estimate the error variances $\boldsymbol{\omega}$ in a heteroskedastic linear regression model.

Chapter 4 (Software Implementation: The **skedastic** R Package) describes a new package in R statistical software called **skedastic** that has been developed specially for this research project. The package has two main purposes. The first is fill address gaps in software implementation of *existing* methods for diagnosing and handling heteroskedasticity (those discussed in the Literature Review chapter). The second is to make the new methods proposed in the Methodology chapter accessible to practitioners.

Chapter 5 (Results and Discussion) evaluates the empirical performance of the new methods proposed in the Methodology, using of course the functions included in the **skedastic** package described in Chapter 4. The performance is evaluated by means of an extensive set of Monte Carlo experiments. These simulated results are supplemented by three examples where the new methods are applied to real data sets.

Chapter 6 (Conclusion) summarises the content of the thesis and the contributions of the research, discusses the extent to which the research objectives have been achieved, and proposes several avenues of possible further research.

2 Literature Review

The aim of this chapter is to describe existing diagnostic and remedial methods for handling heteroskedasticity in the linear regression model. §2.1, the most extensive part of the review, looks at hypothesis testing methods designed to detect violations of A2. It was mentioned in §1.1.6.2 that WLS is generally infeasible. Accordingly, various feasible methods of estimating $\boldsymbol{\beta}$ under heteroskedasticity have been proposed, and these are reviewed in §2.2.1.

Section 2.3 describes a class of methods for estimating $\boldsymbol{\Omega}$, the variance-covariance matrix of the errors $\boldsymbol{\epsilon}$, under heteroskedasticity.

Section 2.4 reviews methods of statistical inference on the individual model parameters (elements of $\boldsymbol{\beta}$) under heteroskedasticity. Finally, existing software implementations of the methods discussed in the first four subsections are reviewed in §2.5.

2.1 Testing for Heteroskedasticity

A heteroskedasticity test is a test of the null hypothesis $\mathrm{H}_0 : \omega_i = \omega_j$ for all $i \neq j$, against the alternative hypothesis $\mathrm{H}_1 : \omega_i \neq \omega_j$ for at least one $i \neq j$ (recall that ω_i is the variance of the ith random error, $i = 1, 2, \ldots, n$). If A3 is assumed, these hypotheses could also be stated as $\mathrm{H}_0 : \boldsymbol{\Omega} \propto \boldsymbol{I}_n$ vs. $\mathrm{H}_1 : \boldsymbol{\Omega} \not\propto \boldsymbol{I}_n$. In some cases the alternative hypothesis may be more specific, and the hypotheses may be expressed in terms of various other parameters depending on the heteroskedastic alternative considered.

Table 2.1 provides an overview of heteroskedasticity tests discussed in this section. The table categorises the tests according to three important characteristics: prior information required (if any), whether hyperparameters are involved, and the (asymptotic) null distribution of the test statistic. Hence, Table 2.1 is designed to enable the user to narrow down the options and choose a method or methods suitable for the task at hand.

The 'Function' column in Table 2.1 refers to the name of the function in the **skedastic** R package that implements the test. This R package was developed by the author for this research project and is discussed further in Chapter 4.

Some heteroskedasticity tests require prior information about the form of heteroskedasticity that is suspected under the alternative hypothesis. These requirements are summarised in the 'Prior Info' column of Table 2.1. A number of tests require the practitioner to specify an $n \times p'$ auxiliary design matrix $\boldsymbol{Z}$, as introduced in §1.1.4. These tests assume that, under the alternative hypothesis, the error variances are related—usually monotonically—to a matrix of nonstochastic or exogenous variables (Glejser 1969, Harvey 1976, Breusch and Pagan 1979, Cook and Weisberg 1983, Verbyla 1993, Simonoff and Tsai 1994, Zhou et al. 2015). If no such prior information is available, the auxiliary design matrix can be set to $\boldsymbol{X}$. The Cook-Weisberg and Simonoff-Tsai tests are most demanding in this regard, as they also require specification of the functional form of the relationship between the error variances and $\boldsymbol{Z}$. Numerous other tests require the user to specify an n-vector of observed variables called the deflator—typically a column of $\boldsymbol{X}$—that is believed to be monotonically related to the error variances (Goldfeld and Quandt 1965, Ramsey 1969, Szroeter 1978, Harrison and McCabe 1979, Horn 1981, Evans and King 1988, Honda 1989, Carapeto and Holt 2003). Some of these tests can be two-tailed or one-tailed, depending on whether the direction of relationship between the deflator and the ω_i is posited in the alternative hypothesis. Tests involving a deflator are not very usable in the absence of prior information, unless there is only one regressor or the user is prepared to run the test repeatedly on each regressor and adjust p-values to control the family-wise Type I error rate. There are other tests that do not require prior information about the form of heteroskedasticity, though their power may still vary depending on the nature of heteroskedasticity present (Bickel 1978, White 1980, Diblasi and Bowman 1997, Wilcox and Keselman 2006, Račkauskas and Zuokas 2007, Yüce 2008, Li and Yao 2019).

Many of the heteroskedasticity tests also require the user to specify a hyperparameter (or hyperparameters); that is, a tuning parameter that may strongly influence the test's performance. The 'Hyp.' column of Table 2.1 indicates, for each heteroskedasticity test, whether or not there are any hyperparameters to be set or tuned. The need for hyperparameters is a strength in the sense that the user has added control over the design of the test, but a weakness in the sense that additional thought and effort is required. Such a test lacks an

Table 2.1: Overview of Heteroskedasticity Tests in the Literature

Test	Function	Prior Info	Hyp.	(Asymp.) Null Dist.
Anscombe's (1961) Test	`anscombe`	None	No	Gaussian
Ramsey's (1969) BAMSET Test	`bamset`	Deflator	Yes	Chi-Squared
Bickel's (1978) Test	`bickel`	None	Yes	Gaussian
Breusch and Pagan's (1979) Test	`breusch_pagan`	Aux. Design	No	Chi-Squared
Carapeto and Holt's (2003) Test	`carapeto_holt`	Deflator	Yes	Ratio of Quadratic Forms (RQF) (Imhof)
Cook and Weisberg's (1983) Test	`cook_weisberg`	Aux. Design, Het. Model	No	Chi-Squared
Diblasi and Bowman's (1997) Test	`diblasi_bowman`	None	Yes	Chi-Squared/ Simulated
Dufour et al.'s (2004) Test	`dufour_etal`	Varies	Yes	Simulated
Evans and King's (1988) LM Test	`evans_king`	Deflator	No	RQF (Imhof)
Evans and King's (1988) GLS Test	`evans_king`	Deflator	Yes	RQF (Imhof)
Glejser's (1969) Test	`glejser`	Aux. Design	No	Chi-Squared
Godfrey and Orme's (1999) Test	`godfrey_orme`	Varies	Yes	Simulated
Goldfeld and Quandt's (1965) F Test	`goldfeld_quandt`	Deflator	Yes	F
Goldfeld and Quandt's (1965) Peaks Test	`goldfeld_quandt`	Deflator	No	Exact Nonpar.
Harrison and McCabe's (1979) Test	`harrison_mccabe`	Deflator	Yes	RQF (Imhof)
Harvey's (1976) Test	`harvey`	Aux. Design	No	Chi-Squared
Honda's (1989) Test	`honda`	Deflator	No	RQF (Imhof)
Horn's (1981) Test	`horn`	Deflator	No	Exact Nonpar./ Gaussian
Li and Yao's (2019) ALR Test	`li_yao`	None	No	Gaussian
Li and Yao's (2019) CV Test	`li_yao`	None	No	Gaussian
Račkauskas and Zuokas's (2007) Test	`rackauskas_zuokas`	None	Yes	Simulated
Simonoff and Tsai's (1994) MPLR Test	`simonoff_tsai`	Aux. Design	Yes	Chi-Squared
Simonoff and Tsai's (1994) Score Test	`simonoff_tsai`	Aux. Design, Het. Model	No	Chi-Squared
Szroeter's (1978) Test	`szroeter`	Deflator	Yes	RQF (Imhof)
Verbyla's (1993) Test	`verbyla`	Aux. Design	No	Chi-Squared
White's (1980) Test	`white`	None	No	Chi-Squared
Wilcox and Keselman's (2006) Test	`wilcox_keselman`	None	Yes	Gaussian
Yüce's (2008) Test	`yuce`	None	No	Chi-Squared/t
Zhou et al.'s (2015) Test	`zhou_etal`	Aux. Design	Yes	Simulated

'off-the-shelf' quality, and this may cause practitioners to avoid it in favour of less complicated methods.[15]

Finally, the last column of Table 2.1 reflects how tests can be grouped by the null distribution of the test statistic (whether exact or asymptotic). The most common distribution used in heteroskedasticity tests is the chi-square distribution (Glejser 1969, Ramsey 1969, Harvey 1976, Breusch and Pagan 1979, White 1980, Cook and Weisberg 1983, Verbyla 1993, Simonoff and Tsai 1994, Diblasi and Bowman 1997, Yüce 2008), which is asymptotically valid in every case. Likewise, those tests that use a Gaussian null distribution are asymptotically valid (Anscombe 1961, Bickel 1978, Horn 1981, Wilcox and Keselman 2006, Li and Yao 2019). Several tests involve a Ratio of Quadratic Forms (RQF) in the random errors, and exact p-values can be computed using the Imhof algorithm (Imhof 1961)—provided that the normality assumption A5 holds (Szroeter 1978, Harrison and McCabe 1979, Evans and King 1988, Honda 1989, Carapeto and Holt 2003).[16] Exact distributions are available for the nonparametric tests of Goldfeld and Quandt (1965) and Horn (1981).[17] Goldfeld and Quandt's (1965) parametric test is an exact F test, provided that A5 holds. Finally, several tests rely on empirical distributions obtained through simulation (Godfrey and Orme 1999, Dufour et al. 2004, Račkauskas and Zuokas 2007, Zhou et al. 2015).[18]

The various heteroskedasticity tests will now be described individually in more detail. Every test statistic will be denoted generically by T for simplicity.

2.1.1 Anscombe's Test

Anscombe (1961) suggests an *ad hoc* method for testing for heteroskedasticity; the test is more compactly described by Bickel (1978, pp. 267-68). The test statistic is

$$T = \tilde{\omega}^{-1/2}\sum_{i=1}^{n} e_i^2(\hat{y}_i - \bar{t}), \tag{2.1}$$

where $\bar{t} = (n-p)^{-1}\sum_{i=1}^{n} m_{ii}\hat{y}_i$ and $\tilde{\omega} = \dfrac{2(n-p)}{n-p+2}\hat{\omega}_{\text{ub}}^2 \sum_{i=1}^{n}\sum_{j=1}^{n} m_{ij}^2(\hat{y}_i - \bar{t})(\hat{y}_j - \bar{t})$, m_{ij} are elements of the annihilator matrix $\boldsymbol{M}$, and $\hat{\omega}_{\text{ub}}^2$ is the square of the unbiased estimator (1.5) of the homoskedastic error variance.

The statistic T is posited to have an asymptotic null distribution that is standard normal. The test is two-tailed. Bickel (1978) proposed a studentising modification of the test statistic as follows,

$$T' = \tilde{\omega}_B^{-1/2}\sum_{i=1}^{n} e_i^2(\hat{y}_i - \bar{t}),, \tag{2.2}$$

where $\tilde{\omega}_B = (n-p)^{-1}\sum_{i=1}^{n}(\hat{y}_i - \bar{t})^2(e_i^2 - \bar{\omega})^2$ and $\bar{\omega} = n^{-1}\sum_{i=1}^{n} e_i^2$ (as defined previously in (1.4)). T' is likewise compared to a standard normal distribution.

[15] A summary of hyperparameters needed for various tests is as follows. Bartlett's M Specification Error Test (BAMSET) (Ramsey 1969) requires the user to partition the data into k subsets on which a Bartlett-style test for equality of variances is then conducted (Bartlett 1937). Harrison and McCabe's (1979) test similarly requires the user to specify an index m based on where change points in the error variance are likely to occur. Bickel's (1978) test is a robust method and requires the user to specify two functions $a(\cdot)$ and $b(\cdot)$, with $b(\cdot)$ corresponding to the derivative function of the M-estimator, $\psi(\cdot)$. Carapeto and Holt's (2003) test and Goldfeld and Quandt's (1965) F test both require the user to specify a proportion c of central observations to remove (after ordering observations by the deflator). Diblasi and Bowman's (1997) test requires a bandwith parameter (either a scalar h, a vector $\boldsymbol{h}$, or a matrix $\boldsymbol{H}$) used in nonparametric regression estimation. Several tests necessarily or optionally involve simulation (bootstrap, MC, or perturbation sampling) thus requiring the user to specify the number of replications and possibly a seed for the pseudorandom number generator (Diblasi and Bowman 1997, Dufour et al. 2004, Godfrey and Orme 1999, Račkauskas and Zuokas 2007, Wilcox and Keselman 2006, Zhou et al. 2015). Evans and King's (1988) Generalised Least Squares (GLS) test requires a parameter $\lambda^{\star}$ representing the degree of severity of heteroskedasticity suspected under the alternative hypothesis. Račkauskas and Zuokas's (2007) test requires a hyperparameter α known as the Hölder exponent. Simonoff and Tsai's (1994) Modified Profile Likelihood Ratio (MPLR) test requires the user to specify initial parameter values for the likelihood maximisation algorithm. Szroeter's (1978) test requires that a nondecreasing function of the indices, $h(i)$, be specified that determines which squared OLS residuals are compared in the RQF. Wilcox and Keselman's (2006) test requires a quantile γ to use in quantile regression estimation.

[16] The authors of most of these tests did not originally suggest the Imhof algorithm as the means of computing exact p-values, probably due to computational limitations at the time of publication.

[17] In the latter case, computation time for exact p-values is prohibitive for $n > 11$.

[18] Note that the tests of Godfrey and Orme (1999) and Dufour et al. (2004) are not separate heteroskedasticity tests, but rather computational methods for obtaining estimated p-values from other heteroskedasticity tests.

2.1.2 Goldfeld-Quandt Tests

Goldfeld and Quandt (1965) propose two heteroskedasticity tests, one parametric and the other nonparametric. As with several other heteroskedasticity tests, the premise of these two tests is that, under the alternative hypothesis, the error variance is monotonically related to some deflator variable.

Having put the observations in increasing order of the deflator, the parametric test procedure proceeds as follows:

1. Remove some proportion c of central observations (chosen such that nc is an integer).
2. Separate OLS regressions are fitted to the first $n(1-c)/2$ observations and to the last $n(1-c)/2$ observations, resulting in residual vectors $\boldsymbol{e}_{\text{first}}$ and $\boldsymbol{e}_{\text{last}}$ respectively.
3. The following variance ratio statistic is computed:

$$T = \frac{\boldsymbol{e}_{\text{last}}'\boldsymbol{e}_{\text{last}}}{\boldsymbol{e}_{\text{first}}'\boldsymbol{e}_{\text{first}}}. \tag{2.3}$$

4. T is compared with its null distribution, which is an F distribution with $n(1-c)/2-p$ degrees of freedom in the numerator and denominator. For the right-tailed test that is implemented by default, the null hypothesis of homoskedasticity is rejected for large values of T.

To implement the nonparametric test the observations are likewise put in increasing order of the deflator variable. The test statistic T' is then the number of 'peaks' in the series of absolute residuals $\{|e_1|, |e_2|, \ldots, |e_n|\}$, where $|e_j|$ is defined as a 'peak,' for $j = 2, 3, \ldots, n$, if $|e_j| \geq |e_i|$ for all $i < j$. (For a graphical representation of peaks in a series, see Figure 4.1). Thus,

$$T' = \sum_{j=2}^{n} 1_{|e_j|\geq \max\left\{|e_1|,|e_2|,\ldots,|e_{j-1}|\right\}}, \tag{2.4}$$

where $1_{\bullet}$ is the indicator function. The test is designed as a right-tailed test, the null hypothesis of homoskedasticity being rejected for large values of T'. The statistic is compared to the distribution of the number of peaks in a series of independent and identically distributed continuous random variables (which is distribution-free).

2.1.3 Glejser's Test

Glejser (1969) had the idea of examining the absolute OLS residuals to detect heteroskedasticity. Their article did not formalise the construction of a hypothesis test, and so different versions of 'Glejser's Test' can be found in the literature. The underlying idea is to fit an auxiliary regression model with response vector $[|e_1|, |e_2|, \ldots, |e_n|]'$ and $n \times p'$ nonstochastic design matrix $\boldsymbol{Z}$. The test described here follows the implementation procedure described in Mittelhammer et al. (2000, p. 541). The test statistic is

$$T = \frac{\sum_{i=1}^{n} e_i^2 - n^{-1}\left(\sum_{i=1}^{n}|e_i|\right)^2 - \sum_{i=1}^{n}\hat{u}_i^2}{(1-2\pi^{-1})\bar{\omega}}, \tag{2.5}$$

where $\hat{u}_i$ is the ith OLS residual from the auxiliary linear regression model. Mittelhammer et al. (2000, p. 537) recommend replacing $\bar{\omega}$ in (2.5) with an estimator computed from the auxiliary model, i.e. $n^{-1}\sum_{i=1}^{n}\hat{u}_i^2$.

The numerator of T can be recognised as the regression sum of squares from the auxiliary model. The asymptotic null distribution of T is $\chi^2(p'-1)$. The test is right-tailed.

2.1.4 Bartlett's M Specification Error Test

Ramsey (1969) developed a test of heteroskedasticity that he called Bartlett's M Specification Error Test (BAMSET). The test entails partitioning the model residuals into $k \geq 2$ subsets and conducting Bartlett's M Test for heterogeneity of variances (Bartlett 1937) using these subsets as its samples. Prior to partitioning, the observations are ordered according to a deflator variable (discussed previously).

Bartlett's M Test requires an assumption of between-sample independence (in this case, between the k subsets of OLS residuals $\boldsymbol{e}$). This assumption does not apply in BAMSET, because under homoskedasticity (when $\boldsymbol{\Sigma} = \omega \boldsymbol{I}_n$), the variance-covariance matrix of $\boldsymbol{e}$ is not diagonal; rather, $\text{Cov}(\boldsymbol{e}) = \omega\boldsymbol{M}$. Hence, in order to satisfy the independence assumption, BAMSET uses the BLUS residuals (Theil 1965, 1968). Computing the BLUS residuals yields only $n-p$ observations rather than n, and so in implementing BAMSET one must specify how to decide which p observations should be omitted. A strategy for deciding this is discussed later in §4.1.2.

Let the BLUS residuals be $\tilde{e}_i$, $i = 1, 2, \ldots, n$, but with $\tilde{e}_i$ undefined for the omitted indices. Further, define ℓ_j to be the set of (non-omitted) indices within the jth subset, $j = 1, 2, \ldots, k$, and let ν_j be the number of observations in the jth subset, from which it follows that $\sum_{j=1}^{k} \nu_j = n - p := \nu$. The BAMSET statistic is written as

$$T = \nu \log s^2 - \sum_{j=1}^{k} \nu_j \log s_j^2, \tag{2.6}$$

where $s^2 = \nu^{-1} \sum_{i=1}^{n-p} \tilde{e}_i^2$ and $s_j^2 = \nu_j^{-1} \sum_{i \in \ell_j} \tilde{e}_i^2$. The asymptotic null distribution of T is $\chi^2(k-1)$ and the test is upper-tailed. The fit to the null distribution can be improved by dividing T by a scaling constant; the statistic then becomes

$$T' = \frac{T}{1 + [3(k-1)]^{-1} \left(\sum_{j=1}^{k} \nu_j^{-1} - \nu^{-1} \right)}. \tag{2.7}$$

Note that Ramsey (1969, p. 368) erroneously includes ν^{-1} within the sum in the formula for the scaling constant.

2.1.5 Harvey's Test

Harvey (1976) constructed a heteroskedasticity test based on an auxiliary regression of the logarithm of the squared OLS residuals on some nonstochastic $n \times p'$ design matrix $\boldsymbol{Z}$. The test statistic, as formally stated in Mittelhammer et al. (2000, p. 540), is

$$T = \frac{\sum_{i=1}^{n} e_i^2 - n^{-1} \left(\sum_{i=1}^{n} |e_i| \right)^2 - \sum_{i=1}^{n} u_i^2}{\psi^{(1)} \left(\frac{1}{2} \right)}, \tag{2.8}$$

where u_i is the ith OLS residual from the auxiliary linear regression model, and $\psi^{(k)}(\cdot)$ is the polygamma function of order k. The numerator of T can be recognised as the regression sum of squares from the auxiliary model, while the denominator is approximately 4.9348. The asymptotic null distribution of T is $\chi^2(p'-1)$ (noting that $\boldsymbol{Z}$ must contain an intercept). The test is right-tailed.

2.1.6 Bickel's Test

Bickel (1978) proposes a robust test of heteroskedasticity that extends the method of Anscombe (1961), replacing the OLS residuals and estimated standard error with robust M-estimators. The first step in implementing the test is to obtain model residuals. These can be obtained via OLS or via robust regression using an M-estimator (to further enhance the robustness of the method). The residuals are denoted as e_i in either case below.

The user must specify a function $a(\cdot)$ to apply to the fitted values and a function $b(\cdot)$ to apply to the residuals to obtain a statistic based on an M estimator. Bickel (1978) suggests $a(\tau) = \tau$. The $b(\cdot)$ function corresponds to the $\psi(\cdot)$ derivative function used to construct M-estimators and must be even, bounded, and twice-differentiable.[19] Two options discussed by Carroll and Ruppert (1981) are Huber's function squared and $b(\tau) = \tanh^2(\tau)$. Huber's function squared is,

$$b(\tau) = \begin{cases} \tau^2 & \text{if } |\tau| \leq k \\ k^2 & \text{if } |\tau| > k \end{cases}, \tag{2.9}$$

where k is a tuning parameter that defaults to 1.345, as is conventional. The test statistic is then

[19] This function $\psi(\cdot)$ is not to be confused with the polygamma function $\psi^{(k)}(\cdot)$ function in (2.8).

$$T = \omega_b^{-1/2} \sum_{i=1}^{n} \left(a(\hat{y}_i) - \bar{a}\right) b(e_i), \tag{2.10}$$

where $\omega_b = (n-p)^{-1} \sum_{i=1}^{n} \left(a(\hat{y}_i) - \bar{a}\right)^2 \sum_{i=1}^{n} \left(b(e_i) - \bar{b}\right)^2$, $\bar{a} = n^{-1} \sum_{i=1}^{n} a(\hat{y}_i)$, and $\bar{b} = n^{-1} \sum_{i=1}^{n} b(e_i)$. Carroll and Ruppert (1981) note that this test statistic is not scale-invariant, and that this can be rectified by replacing $b(e_i)$ in the above expression with $b\left(e_i/\tilde{\omega}^{1/2}\right)$, where $\tilde{\omega}$ is an estimator of ω.

The asymptotic null distribution of T is standard normal and thus the two-sided p-value for the test is computed from the standard normal distribution.

2.1.7 Szroeter's Test

Szroeter (1978) proposes a class of tests for which the test statistic is a RQF in normal random vectors. A prerequisite of the test is that the observations are ordered by a deflator variable. The user must further specify a nondecreasing function $h(i)$ of the indices $i = 1, 2, \ldots, n$. Szroeter (1978) suggests using

$$h(i) = 2\left[1 - \cos\left(\frac{\pi i}{n+1}\right)\right], \; i = 1, 2, \ldots, n. \tag{2.11}$$

The test statistic is

$$T = \frac{\boldsymbol{e}'\boldsymbol{\Delta}\boldsymbol{e}}{\boldsymbol{e}'\boldsymbol{e}}, \tag{2.12}$$

where $\boldsymbol{\Delta} = \operatorname{diag}\left\{h(1), h(2), \ldots, h(n)\right\}$. Certain specifications of $h(\cdot)$ cause the test statistic to reduce to that of another test. For example, if $h(i) = -1$ for the first $n(1-c)/2$ observations, $h(i) = 0$ for the middle nc observations, and $h(i) = 1$ for the last $n(1-c)/2$ observations, then the test statistic is identical to that of the parametric test of Goldfeld and Quandt (1965) (discussed in §2.1.2).

2.1.8 Breusch-Pagan Test and White's Test

One of the better-known heteroskedasticity tests was proposed by Breusch and Pagan (1979). The test is derived using Lagrange multipliers. It requires specification of an $n \times p'$ auxiliary design matrix $\boldsymbol{Z}$ as introduced in §1.1.4. Breusch and Pagan's (1979) procedure is as follows:

1. Compute $\bar{\omega} = n^{-1} \sum_{i=1}^{n} e_i^2$ and $\hat{\boldsymbol{w}}$, an n-vector whose ith element is $\hat{w}_i = e_i^2 - \bar{\omega}$.
2. Fit an auxiliary regression model (using OLS) with response vector $\hat{\boldsymbol{w}}$ and design matrix $\boldsymbol{Z}$ (which must include an intercept column), and obtain the model's fitted values $\tilde{w}_i$, $i = 1, 2, \ldots, n$.
3. Compute the test statistic,

$$T = (2\bar{\omega})^{-1} \sum_{i=1}^{n} \tilde{w}_i^2. \tag{2.13}$$

Koenker (1981) suggested a studentising modification of the statistic that is much more widely used in practice due to its better properties. This statistic is

$$T' = \frac{n \sum_{i=1}^{n} \tilde{w}_i^2}{\sum_{i=1}^{n} \hat{w}_i^2}. \tag{2.14}$$

T' can otherwise be expressed as nr_{aux}^2, where r_{aux}^2 is the multiple coefficient of determination of the auxiliary regression. In either case, the asymptotic null distribution of the statistic is $\chi^2(p'-1)$. The test is right-tailed.

White (1980) proposed a test that is a special case of Breusch and Pagan's (1979) test (with the modification of Koenker (1981)), but has surpassed it in popularity to become arguably the best-known test of heteroskedasticity among practitioners.

The test entails setting the auxiliary design matrix $\boldsymbol{Z}$ to $[\boldsymbol{X} \;\; (\boldsymbol{X} \circ \boldsymbol{X})_{-1}]$, the horizontal concatenation of $\boldsymbol{X}$ with an elementwise-squared version of itself. Note, however, that if $\boldsymbol{X}$ contains a column of ones, this column is not included in $\boldsymbol{X} \circ \boldsymbol{X}$; hence the -1 subscript. One can optionally augment the auxiliary matrix by including all pairwise interaction terms. That is, if X_{ij} is the ith observation on the jth explanatory variable,

$i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, p$ (or $j = 2, 3, \ldots, p$ if an intercept is present), then including the interaction terms entails that the scalar form of the auxiliary regression equation will include terms $\sum_{\substack{j=1 \\ j<k}}^{n} \sum_{k=1}^{n} X_{ij} X_{ik}$.

2.1.9 Harrison-McCabe Test

The next test reviewed, like several others discussed in this review, requires the practitioner to specify a 'deflator' variable. This is Harrison and McCabe's (1979) test. The observations are placed in increasing order of the deflator, and a statistic is then constructed that is an RQF in the error vector:

$$T = \frac{\boldsymbol{e}'\boldsymbol{A}\boldsymbol{e}}{\boldsymbol{e}'\boldsymbol{e}} = \frac{\boldsymbol{\epsilon}'\boldsymbol{M}\boldsymbol{A}\boldsymbol{M}\boldsymbol{\epsilon}}{\boldsymbol{\epsilon}'\boldsymbol{M}\boldsymbol{\epsilon}}, \tag{2.15}$$

where $\boldsymbol{A}$ is an $n \times n$ selector matrix in which the first m diagonal elements are ones and all other elements are zeroes. The index m is determined based on the point at which a breakpoint in the error variance is suspected to occur; in the absence of information about this it would be set as close to $n/2$ as possible. Under the null hypothesis, T will be close to m/n. Under the alternative, if the deflator is positively associated with the error variance, T will be small. Thus, the test as designed is left-tailed.

2.1.10 Horn's Test

A new nonparametric test of heteroskedasticity—in addition to that of Goldfeld and Quandt (1965)—was developed by Horn (1981). This test employs the nonparametric trend statistic D defined by Lehmann (1975, pp. 290-97). It requires specification of a deflator variable believed to be monotonically related to the error variance. The observations are placed in increasing order of the deflator, and the statistic is then

$$D = \sum_{i=1}^{m} (R_i - i)^2, \tag{2.16}$$

where

$$m = \begin{cases} n & \text{if OLS residuals are used} \\ n - p & \text{if BLUS residuals are used} \end{cases},$$

and R_i is the rank of the ith absolute OLS residual, $|e_i|$.[20] Horn (1981) suggested that the BLUS residuals (Theil 1965, 1968) could be used instead of the OLS residuals to minimise the risk of a spurious trend under the null hypothesis. The test, as designed, is two-tailed.

The p-values for Horn's test can be computed from the exact distribution of D for small m, but this becomes computationally infeasible for $m > 10$; in this case a normal approximation based on the Central Limit Theorem can be used.

2.1.11 Cook-Weisberg Test

Cook and Weisberg (1983) propose a score test similar in character to Breusch and Pagan's (1979) test. They assume that the errors $\boldsymbol{\epsilon}$ follow a multivariate normal distribution with mean vector $\boldsymbol{0}$ and diagonal variance-covariance matrix $\boldsymbol{\Omega} = \omega\boldsymbol{\mathfrak{S}}$, with $\boldsymbol{\mathfrak{S}}$ having ith diagonal element $\mathfrak{s}_i = \mathfrak{g}(\boldsymbol{Z}'_{i\cdot}, \boldsymbol{\zeta})$, for some function $\mathfrak{g}(\cdot) : \mathbb{R}^{p'} \to \mathbb{R}$. (Thus, in terms of the notation used elsewhere, $\omega_i = \omega\mathfrak{s}_i$). Here, $\boldsymbol{Z}'_{i\cdot}$ is the ith row of a $n \times p'$ auxiliary design matrix $\boldsymbol{Z}$ as defined in §1.1.4, and $\boldsymbol{\zeta}$ is a $p' - 1$-vector of unknown parameters. $\mathfrak{g}(\cdot)$ is a twice-differentiable, positive real-valued function applied elementwise to $\boldsymbol{Z}'_{i\cdot}$. It is assumed that there is some vector $\boldsymbol{\zeta}_0$ for which $\mathfrak{s}_i = 1$, $i = 1, 2, \ldots, n$, and thus the null hypothesis of homoskedasticity is equivalent to $\boldsymbol{\zeta} = \boldsymbol{\zeta}_0$. The choice of $\mathfrak{g}(\cdot)$ depends on the form of the error variance under the alternative hypothesis. Note that $\mathfrak{g}(\cdot)$ differs from the heteroskedastic function $g(\cdot)$ used elsewhere only by a scaling factor: $g(\boldsymbol{Z}'_{i\cdot}, \boldsymbol{\zeta}) = \omega\mathfrak{g}(\boldsymbol{Z}'_{i\cdot}, \boldsymbol{\zeta})$. Three well-known choices of heteroskedastic model (Cook and Weisberg 1983, Griffiths and Surekha 1986) are

[20] The practice of referring to test statistics as T is altered here due to the convention of using D to denote Lehmann's (1975) nonparametric trend statistic.

$$\mathfrak{g}(\boldsymbol{Z}_{i\cdot}', \boldsymbol{\zeta}) = \left(1 + \sum_{j=1}^{p'-1} \zeta_j Z_{i,j+1}\right)^2 \quad \text{(additive model)}, \tag{2.17}$$

$$\mathfrak{g}(\boldsymbol{Z}_{i\cdot}', \boldsymbol{\zeta}) = \exp\left\{\sum_{j=1}^{p'-1} \zeta_j Z_{i,j+1}\right\} = \prod_{j=1}^{p'-1} \exp\left\{\zeta_j Z_{i,j+1}\right\} \quad \text{(multiplicative model)}, \tag{2.18}$$

and

$$\mathfrak{g}(\boldsymbol{Z}_{i\cdot}', \boldsymbol{\zeta}) = \exp\left\{\sum_{j=1}^{p'-1} \zeta_j \log Z_{i,j+1}\right\} = \prod_{j=1}^{p'-1} Z_{i,j+1}^{\zeta_j} \quad \text{(log-multiplicative model)}. \tag{2.19}$$

In the log-multiplicative model it is required that $Z_{i,j+1} > 0$ for all $i = 1, 2, \ldots, n$, $j = 1, \ldots, p' - 1$. In all three models, $\boldsymbol{\zeta} = \boldsymbol{0}$ implies $\mathfrak{g}(\boldsymbol{Z}_{i\cdot}', \boldsymbol{\zeta}) = 1$. Therefore, $\boldsymbol{\zeta}_0 = \boldsymbol{0}$: a heteroskedasticity test is equivalent to a test of the null hypothesis $\boldsymbol{\zeta} = \boldsymbol{0}$.

Cook and Weisberg's (1983) test entails fitting an auxiliary regression model in which the response vector is the n-vector $\boldsymbol{d}$ whose ith element is $d_i = \bar{\omega}^{-1} e_i^2$, where $\bar{\omega} = n^{-1}\boldsymbol{e}'\boldsymbol{e}$ (as defined previously), and the design matrix is an $n \times p'$ matrix consisting of a column of ones concatenated with $\boldsymbol{J}$, an $n \times (p' - 1)$ Jacobian matrix whose (i, j)th element is $\dfrac{\partial \mathfrak{g}(\boldsymbol{Z}_{i\cdot}', \boldsymbol{\zeta})}{\partial \zeta_j}$, evaluated at $\boldsymbol{\zeta} = \boldsymbol{\zeta}_0$, for $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, p' - 1$. It is easily shown that this derivative term reduces to $2Z_{i,j+1}$ in the additive model, to $Z_{i,j+1}$ in the multiplicative model, and to $\log Z_{i,j+1}$ in the log-multiplicative model. The test statistic is then

$$T = 2^{-1}\left(\boldsymbol{d}'\boldsymbol{d} - \boldsymbol{u}'\boldsymbol{u} - n\right), \tag{2.20}$$

where $\boldsymbol{u}$ is the vector of OLS residuals from the auxiliary regression. T can be interpreted as half the regression sum of squares from the aforementioned auxiliary regression. Under the null hypothesis of homoskedasticity, T has an asymptotic $\chi^2(p' - 1)$ distribution, and the null hypothesis is rejected for large T.

2.1.12 Evans-King Tests

Two new heteroskedasticity tests are described Evans and King (1988), one of which had been less formally suggested in Evans and King (1985). One can be referred to as their GLS test and the other as their Lagrange Multiplier (LM) test.

The test statistic for the GLS method is

$$T = \frac{\boldsymbol{u}'\boldsymbol{\Sigma}(\lambda^\star)^{-1}\boldsymbol{u}}{\boldsymbol{e}'\boldsymbol{e}}, \tag{2.21}$$

where $\boldsymbol{\Sigma}(\lambda^\star) = \operatorname{diag}\left\{(1 + \lambda^\star\tau_1), \ldots, (1 + \lambda^\star\tau_n)\right\}$, $\tau_i = \dfrac{i-1}{n-1}$, $i = 1, 2, \ldots, n$, and $\boldsymbol{u}$ is the residual vector from a GLS regression of $\boldsymbol{y}$ on $\boldsymbol{X}$ with covariance matrix $\boldsymbol{\Sigma}(\lambda^\star)$. As is evident from the expression for $\boldsymbol{\Sigma}(\lambda^\star)$, the parameter $\lambda^\star$ controls the degree of severity of heteroskedasticity. Evans and King (1988) find, based on an empirical study, that $\lambda^\star = 5$ results in the highest power. Evans and King (1985) had earlier observed that (2.21) can be rewritten as a RQF in the error vector $\boldsymbol{\epsilon}$, specifically as

$$T = \frac{\boldsymbol{\epsilon}'\boldsymbol{M}\boldsymbol{R}\boldsymbol{M}^\star\boldsymbol{R}\boldsymbol{M}\boldsymbol{\epsilon}}{\boldsymbol{\epsilon}'\boldsymbol{M}\boldsymbol{\epsilon}},$$

where

$$\begin{aligned}\boldsymbol{R} &= \operatorname{diag}\left\{(1 + \lambda^\star\tau_1)^{-1/2}, (1 + \lambda^\star\tau_2)^{-1/2}, \ldots, (1 + \lambda^\star\tau_n)^{-1/2}\right\}, \\ \boldsymbol{M}^\star &= \boldsymbol{I}_n - \boldsymbol{X}^\star\left(\boldsymbol{X}^{\star\prime}\boldsymbol{X}^\star\right)^{-1}\boldsymbol{X}^{\star\prime},\end{aligned}$$

and $\boldsymbol{X}^\star$ is the matrix formed by dividing the ith row of $\boldsymbol{X}$ by $w_i = (1 + \lambda^\star\tau_i)^{1/2}$.

The LM method, which is a limiting case of the GLS method, also results in a test statistic that can be expressed as a RQF in the random error vector:

$$T' = \frac{\boldsymbol{\epsilon}'\boldsymbol{M}\operatorname{diag}\left\{\frac{n-1}{n-1},\frac{n-2}{n-1},\ldots,\frac{n-n}{n-1}\right\}\boldsymbol{M}\boldsymbol{\epsilon}}{\boldsymbol{\epsilon}'\boldsymbol{M}\boldsymbol{\epsilon}}. \tag{2.22}$$

Evans and King (1985, 1988) do not discuss in detail how to compute critical values or p-values for these tests. This gap is addressed in the current author's software implementation of the methods as discussed in Chapter 4.

2.1.13 Honda's Test

Another deflator-type heteroskedasticity test is that of Honda (1989). The observations are placed in increasing order of the deflator. The test statistic is a RQF in the error vector:

$$T = \frac{\boldsymbol{e}'\operatorname{diag}\{\boldsymbol{Z}_{\cdot j}\}\,\boldsymbol{e}}{\boldsymbol{e}'\boldsymbol{e}} = \frac{\boldsymbol{\epsilon}'\boldsymbol{M}\operatorname{diag}\{\boldsymbol{Z}_{\cdot j}\}\,\boldsymbol{M}\boldsymbol{\epsilon}}{\boldsymbol{\epsilon}'\boldsymbol{M}\boldsymbol{\epsilon}}, \tag{2.23}$$

where $\boldsymbol{Z}_{\cdot j}$ is the deflator variable. If $\boldsymbol{Z}_{\cdot j}$ is positively associated with the error variance, T will tend to be large. Honda (1989) describes the method as a two-tailed test, corresponding to a situation where the direction of the monotonic relationship between the deflator and the error variance (under the alternative hypothesis) is not known. Of course, it can easily be modified into a one-tailed test.

2.1.14 Verbyla's Test

Verbyla (1993) proposes a test that uses the notion of Residual Maximum Likelihood (ReML) and is designed particularly to detect a log-linear dependence of the error variances on some specified predictor variables. The test statistic is a generalisation of that of Breusch and Pagan (1979) and Cook and Weisberg (1983), and like those previous tests it requires an auxiliary design matrix $\boldsymbol{Z}$ to be specified. The statistic's form is

$$T = \frac{1}{2}\boldsymbol{v}'\boldsymbol{Z}\left[\boldsymbol{Z}'(\boldsymbol{M}\circ\boldsymbol{M})\boldsymbol{Z}\right]^{-1}\boldsymbol{Z}'\boldsymbol{v}, \tag{2.24}$$

where $\boldsymbol{v} = \boldsymbol{d} - \operatorname{diag}(\boldsymbol{M})$, $\boldsymbol{d}$ is the n-vector with ith element $\hat{\omega}_{\text{ub}}^{-1}e_i^2$, and $\boldsymbol{Z}$ is an $n \times p'$ auxiliary design matrix, as defined in §1.1.4.

T can be interpreted as half the regression sum of squares for a GLS regression of $(\boldsymbol{M}\circ\boldsymbol{M})^{-1}\boldsymbol{d}/\hat{\omega}_{\text{ub}}$ on $\boldsymbol{Z}$, with covariance matrix $\boldsymbol{M}\circ\boldsymbol{M}$. The asymptotic null distribution of T is $\chi^2(p'-1)$.

2.1.15 Simonoff-Tsai Tests

Two more tests—a likelihood ratio (LR) test using a modified profile likelihood and a score test—are introduced in Simonoff and Tsai (1994). Both tests assume a heteroskedastic model like those described under the Cook-Weisberg test (§2.1.11). The random error vector $\boldsymbol{\epsilon}$ is assumed to be multivariate normal with mean vector $\boldsymbol{0}$ and diagonal variance-covariance matrix $\boldsymbol{\Omega} = \omega\mathfrak{S}$, with $\mathfrak{S}$ having ith diagonal element $\mathfrak{s}_i = \mathfrak{g}(\boldsymbol{Z}'_{i\cdot},\boldsymbol{\zeta})$, as described in §2.1.11. $\mathfrak{g}(\cdot)$ is again assumed to be a real-valued, twice-differentiable function, and $\boldsymbol{Z}$ is again an $n \times p'$ auxiliary design matrix. It is further assumed that there exists some vector $\boldsymbol{\zeta}_0$ such that $\mathfrak{g}(\boldsymbol{Z}'_{i\cdot},\boldsymbol{\zeta}_0) = 1$ for $i = 1, 2, \ldots, n$ (thus, $\boldsymbol{\zeta} = \boldsymbol{\zeta}_0$ corresponds to homoskedasticity). The log-likelihood function for this model can be written as

$$l(\boldsymbol{y};\boldsymbol{\zeta},\omega,\boldsymbol{\beta}) = -\frac{n}{2}\log\omega - \frac{1}{2}\sum_{i=1}^{n}\log\mathfrak{g}(\boldsymbol{Z}'_{i\cdot},\boldsymbol{\zeta}) - (2\omega)^{-1}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})'\,\mathfrak{S}^{-1}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}).$$

As noted by Simonoff and Tsai (1994), the ML estimate of $\boldsymbol{\zeta}$ can be obtained by maximising the profile log-likelihood,

$$l_p(\boldsymbol{y};\boldsymbol{\zeta}) = l(\boldsymbol{y};\boldsymbol{\zeta},\hat{\omega}_{\zeta},\hat{\boldsymbol{\beta}}_{\zeta}),$$

where

$$\hat{\boldsymbol{\beta}}_{\zeta} = (\boldsymbol{X}'\mathfrak{S}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}'\mathfrak{S}^{-1}\boldsymbol{y},$$

and

$$\hat{\omega}_{\zeta} = n^{-1}(\boldsymbol{y}-\boldsymbol{X}\hat{\boldsymbol{\beta}}_{\zeta})'\mathfrak{S}^{-1}(\boldsymbol{y}-\boldsymbol{X}\hat{\boldsymbol{\beta}}_{\zeta}).$$

Simonoff and Tsai (1994) note that inference based on this profile likelihood can be problematic due to the lack of orthogonality between the parameters of interest, $\boldsymbol{\zeta}$, and the nuisance parameters, $(\omega, \boldsymbol{\beta})$. Following on earlier theoretical work, they propose a modification of the profile likelihood to achieve such orthogonality. They derive the modified profile likelihood ratio statistic,

$$T = \frac{n-p-2}{n} L + \log\left\{\frac{\det(\boldsymbol{X}'\boldsymbol{X})}{\det(\hat{\boldsymbol{X}}_m'\hat{\boldsymbol{X}}_m)}\right\}, \tag{2.25}$$

where $L = -2\left\{l_p(\boldsymbol{y};\boldsymbol{\zeta}_0) - l_p(\boldsymbol{y};\hat{\boldsymbol{\zeta}})\right\}$, and $\hat{\boldsymbol{X}}_m = \hat{G}^{-1/2}\boldsymbol{X}$, where $\hat{G}$ is the diagonal $n \times n$ matrix with ith diagonal entry $\dfrac{\mathfrak{g}(\boldsymbol{Z}'_{i\cdot}, \hat{\boldsymbol{\zeta}})}{\left\{\prod\limits_{j=1}^{n} \mathfrak{g}(\boldsymbol{Z}'_{j\cdot}, \hat{\boldsymbol{\zeta}})\right\}^{1/n}}$. The asymptotic null distribution of the statistic is $\chi^2(p'-1)$ and the test is right-tailed.

Ferrari et al. (2004) derive a Bartlett correction for Simonoff and Tsai's (1994) modified profile likelihood ratio test, generalising an earlier proposal (Ferrari and Cribari-Neto 2002). The Bartlett correction improves the fit of the test statistic to the asymptotic null distribution, particularly for small sample sizes. The Bartlett-corrected test statistic is

$$T' = \frac{T}{1 + c_m/(p'-1)},$$

where c_m is a correction factor. The notation in the expression for the correction factor is cumbersome; the reader is referred to Equation (7) in Ferrari et al. (2004, p. 430). The authors derive an explicit expression for c_m only for the multiplicative heteroskedastic model.

Simonoff and Tsai's (1994) score test is designed to be a robust extension either of Cook and Weisberg's (1983) score test or of Koenker's (1981) modification of Breusch and Pagan's (1979) test. The test requires an auxiliary design matrix $\boldsymbol{Z}$. The test statistic of the base test is first computed (call it S) and the test statistic is then computed as

$$T = S + \sum_{j=1}^{p'}\left(\sum_{i=1}^{n} h_{ii}t_{ij}\right)\tau_j, \tag{2.26}$$

where t_{ij} is the (i,j)th element of the Jacobian matrix $\boldsymbol{J}$ (as defined in §2.1.11), and τ_j is the jth element of the $(p'-1)$-vector $\left(\bar{\boldsymbol{J}}'\bar{\boldsymbol{J}}\right)^{-1}\bar{\boldsymbol{J}}'\boldsymbol{d}$. Here, exactly as in the formulation of Cook and Weisberg's (1983) test, $\boldsymbol{d}$ is the n-vector having ith element $\bar{\omega}^{-1}e_i^2$, where $\bar{\omega} = n^{-1}\boldsymbol{e}'\boldsymbol{e}$, and $\bar{\boldsymbol{J}} = \left(\boldsymbol{I}_n - n^{-1}\boldsymbol{1}_{n\times n}\right)\boldsymbol{J}$. The asymptotic null distribution of the score test statistic (2.26) is also $\chi^2(p'-1)$ and the test is likewise right-tailed.

2.1.16 Diblasi-Bowman Test

A test developed by Diblasi and Bowman (1997) involves the use of the kernel method of nonparametric regression to model the relationship between a transformation of the OLS residuals and the explanatory variables. First, define

$$s_i = \sqrt{|e_i|} - \mathrm{E}_0(\sqrt{|e_i|}),\ i = 1, 2, \ldots, n,$$

where E_0 denotes expectation under the null hypothesis of homoskedasticity. The relationship between the s_i and the corresponding observations of the explanatory variable(s) is modelled using the Nadaraya-Watson kernel estimation method of nonparametric regression. Recall the notation $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]'$ for the predictor variable in simple linear regression. In this case, using the normal kernel function,

$$K(x) = (2\pi)^{-1/2}\exp\left\{-\frac{1}{2}x^2\right\},$$

it follows that

$$\tilde{s}(x_i) = \sum_{j=1}^{n} w_j(x_i)s_j, \tag{2.27}$$

where

$$w_j(x_i) = \frac{K\left(\frac{x_i - x_j}{h}\right)}{\sum_{k=1}^{n} K\left(\frac{x_i - x_k}{h}\right)} = \frac{\exp\left\{-\frac{1}{2}\left(\frac{x_i - x_j}{h}\right)^2\right\}}{\sum_{k=1}^{n} \exp\left\{-\frac{1}{2}\left(\frac{x_i - x_k}{h}\right)^2\right\}}, \tag{2.28}$$

and h is a hyperparameter known as the bandwidth.[21] Diblasi and Bowman (1997) do not discuss the extension of their test to the multiple linear regression model, but this is straightforward.[22]

The multivariate normal kernel function is then given by,

$$K(\boldsymbol{x}) = (2\pi)^{-p/2} \exp\left\{-\frac{1}{2}\boldsymbol{x}'\boldsymbol{x}\right\}.$$

If $\boldsymbol{X}_{i\cdot}$ denotes the ith row of $\boldsymbol{X}$ (excluding the intercept column if present), then the kernel weights are now written as

$$\begin{aligned} w_j(\boldsymbol{X}_{i\cdot}) &= \frac{K(\boldsymbol{H}^{-1}(\boldsymbol{X}_{i\cdot} - \boldsymbol{X}_{j\cdot}))}{\sum_{k=1}^{n} K(\boldsymbol{H}^{-1}(\boldsymbol{X}_{i\cdot} - \boldsymbol{X}_{k\cdot}))} \\ &= \frac{\exp\left\{-\frac{1}{2}\left[(\boldsymbol{X}_{i\cdot} - \boldsymbol{X}_{j\cdot})'\boldsymbol{H}^{-1}\boldsymbol{H}^{-1}(\boldsymbol{X}_{i\cdot} - \boldsymbol{X}_{j\cdot})\right]\right\}}{\sum_{k=1}^{n} \exp\left\{-\frac{1}{2}\left[(\boldsymbol{x}_i - \boldsymbol{x}_k)'\boldsymbol{H}^{-1}\boldsymbol{H}^{-1}(\boldsymbol{X}_{i\cdot} - \boldsymbol{X}_{k\cdot})\right]\right\}}. \end{aligned}$$

Letting $\tilde{s}_i = \tilde{s}(x_i)$ (or $\tilde{s}_i = \tilde{s}(X_{i\cdot})$, in the multiple linear regression case), the Diblasi-Bowman test statistic is,

$$T = \frac{\sum_{i=1}^{n}(s_i - \bar{s})^2 - \sum_{i=1}^{n}(s_i - \tilde{s}_i)^2}{\sum_{i=1}^{n}(s_i - \tilde{s}_i)^2}, \tag{2.29}$$

where $\bar{s} = n^{-1}\sum_{i=1}^{n} s_i$. Diblasi and Bowman (1997) observe that the statistic is analogous to the 'lack-of-fit' statistic in parametric regression. Under the null hypothesis, the terms in the numerator will have little difference, but under heteroskedasticity the first term will dominate the second. Thus, the test is right-tailed.

Suppose that the $n \times n$ matrix whose i, jth element is $w_j(x_i)$ is denoted by $\boldsymbol{W}$, $\tilde{\boldsymbol{s}} = \boldsymbol{W}\boldsymbol{s}$, with $\boldsymbol{s} = [s_1, s_2, \ldots, s_n]'$, and the quadratic form $\sum_{i=1}^{n}(s_i - \tilde{s}_i)^2$ can be expressed as $\boldsymbol{s}'\boldsymbol{B}\boldsymbol{s}$, where $\boldsymbol{B} = (\boldsymbol{I}_n - \boldsymbol{W})'(\boldsymbol{I}_n - \boldsymbol{W})$. Then, the test statistic can be rewritten as,

$$T = \frac{\boldsymbol{s}'\boldsymbol{A}\boldsymbol{s} - \boldsymbol{s}'\boldsymbol{B}\boldsymbol{s}}{\boldsymbol{s}'\boldsymbol{B}\boldsymbol{s}} = \frac{\boldsymbol{s}'\boldsymbol{C}\boldsymbol{s}}{\boldsymbol{s}'\boldsymbol{B}\boldsymbol{s}}, \tag{2.30}$$

where $\boldsymbol{A} = \boldsymbol{I}_n - n^{-1}\boldsymbol{1}_n$, $\boldsymbol{C} = \boldsymbol{A} - \boldsymbol{B}$, and $\boldsymbol{1}_n$ is an $n \times n$ matrix of ones. Diblasi and Bowman (1997) propose to compute an approximate p-value for the test by matching cumulants with those of a shifted χ^2 distribution. They first observe that the p-value of the test can be written as follows:

$$\begin{aligned} \Pr\left(T > t_0 | \mathrm{H}_0 \text{ true}\right) &= \Pr\left(\frac{\boldsymbol{s}'\boldsymbol{C}\boldsymbol{s}}{\boldsymbol{s}'\boldsymbol{B}\boldsymbol{s}} > t_0 \middle| \mathrm{H}_0 \text{ true}\right) \\ &= \Pr\left(\boldsymbol{s}'(\boldsymbol{C} - t_0\boldsymbol{B})\boldsymbol{s} > 0 \middle| \mathrm{H}_0 \text{ true}\right). \end{aligned}$$

Specifically, the authors propose to match the first three cumulants of the quadratic form $\boldsymbol{s}'\left(\boldsymbol{C} - t_0\boldsymbol{B}\right)\boldsymbol{s}$ with those of a random variable U of the form $a + bV$ (where $V \sim \chi^2(c)$). For full details of this method, which is quite involved, the reader is referred to their article.

Diblasi and Bowman (1997) also offer a parametric bootstrap procedure for estimating p-values of the test as follows:

[21] Diblasi and Bowman (1997, p. 97) appear to err by omitting the $-\frac{1}{2}$ factor in the numerator and denominator of (2.28).

[22] The bandwidth scalar h is replaced with a $p' \times p'$ symmetric bandwidth matrix $\boldsymbol{H}$ (where p' is the number of explanatory variables excluding an intercept if present)

1. Note the observed value of the test statistic, t_0.
2. Simulate $y_i^{(b)}$, $i = 1, 2, \ldots, n$, from a normal distribution with mean $\hat{y}_i$ and variance $\hat{\omega}_{\text{ub}}$ (defined in (1.5)), for $b = 1$.
3. Refit the model using OLS and obtain bootstrap residuals $e_1^{(b)}, e_2^{(b)}, \ldots, e_n^{(b)}$.
4. Calculate the observed value $t^{(b)}$ of the test statistic T.
5. Repeat steps 2 to 4 for $b = 2, 3, \ldots, B$ and calculate the p-value estimate from the empirical distribution of T:

$$p^\star = B^{-1}(\# t^{(b)} \geq t_0).$$

2.1.17 Carapeto-Holt Test

Carapeto and Holt (2003) propose a test that is similar in its logic to Goldfeld and Quandt's (1965) test (see §2.1.2). Both tests entail partitioning the data into three subgroups, one of which is ignored while the other two are compared. The major difference between the two methods is that, in the Goldfeld-Quandt test, the two subgroups being compared are fitted to separate regressions, yielding mutually independent sets of OLS residuals. By contrast, the Carapeto-Holt test works with subsets of the OLS residuals from the model fit to the full data set.[23]

The Carapeto-Holt test requires a deflator, and, as the authors constructed the test, the observations are assumed to be in *decreasing* order of error variance (thus it is a left-tailed test by design).

The test proceeds by first removing some proportion $c = (n - 2s)/n$ of central observations, leaving two subsets consisting of the first s and last s observations, respectively. The sums of squared residuals of these two subsets are then compared using the following statistic:

$$T = \frac{\boldsymbol{e}'\boldsymbol{I}^\star\boldsymbol{e}}{\boldsymbol{e}'\boldsymbol{I}_\star\boldsymbol{e}} = \frac{\boldsymbol{\epsilon}'\boldsymbol{M}'\boldsymbol{I}^\star\boldsymbol{M}\boldsymbol{\epsilon}}{\boldsymbol{\epsilon}'\boldsymbol{M}'\boldsymbol{I}_\star\boldsymbol{M}\boldsymbol{\epsilon}}, \tag{2.31}$$

where $\boldsymbol{I}^\star$ is an $n \times n$ diagonal matrix whose first s diagonal elements are ones and other diagonal elements are zeroes, and $\boldsymbol{I}_\star$ is an $n \times n$ diagonal matrix whose last s diagonal elements are ones and other diagonal elements are zeroes. $\boldsymbol{M}$ is the usual annihilator matrix. Under the null hypothesis (together with the assumptions of normality and no autocorrelation), T is a RQF in a normal random vector having mean vector $\mathbf{0}$ and covariance matrix $\omega \boldsymbol{I}_n$. The ω cancel in the ratio, so that T is scale-invariant.

2.1.18 Wilcox-Keselman Test

Wilcox and Keselman (2006) propose a test that makes use of quantile regression. Consider first a simple linear regression with response variable y and explanatory variable X. It is assumed that the γ quantile of Y, given X, is given by,

$$y_\gamma = \alpha_\gamma + \beta_\gamma X.$$

Homoskedasticity of y implies that $\beta_\gamma = \beta_{1-\gamma}$ for any $0 < \gamma < \dfrac{1}{2}$. Thus, a quantile regression model can be fitted to test the null hypothesis $\beta_\gamma = \beta_{1-\gamma}$ for some $0 < \gamma < \frac{1}{2}$ and thereby indirectly test for heteroskedasticity. The test statistic takes the form

$$T = \frac{\Delta}{\widehat{\text{SE}}(\Delta)}, \tag{2.32}$$

where $\Delta = \hat{\beta}_\gamma - \hat{\beta}_{1-\gamma}$.

The standard error of Δ under the null hypothesis is intractable, and thus Wilcox and Keselman (2006) propose to use nonparametric bootstrap sampling to estimate it. Thus,

$$\widehat{\text{SE}}(\Delta) = \sqrt{(B-1)^{-1}\sum_{b=1}^{B}(\Delta^{(b)} - \bar{\Delta})^2},$$

where $\Delta^{(b)}$ is the difference between the quantile regression estimates of β_γ and $\beta_{1-\gamma}$ based on the bth bootstrap sample, $b = 1, 2, \ldots, B$, and $\bar{\Delta} = B^{-1}\sum_{b=1}^{B}\Delta^{(b)}$. Applying the Central Limit Theorem, T is compared to the standard normal distribution in what is a two-tailed test.

[23] This distinction in subsetting of residuals anticipates two possible ways of obtaining a test set of residuals in K-fold cross-validation, to be discussed later and diagrammatically represented in Figure 3.6.

2.1.19 Račkauskas-Zuokas Test

Račkauskas and Zuokas (2007) propose a class of tests based on the limit behaviour of the polygonal process constructed from squared residuals. The test is especially designed to detect a 'changed-segment' type of heteroskedasticity where the error variance shifts at one or more specific locations in the data. They propose the statistic

$$T_{n,\alpha} = \max_{1\leq \ell < n} (\ell/n)^{-\alpha} \max_{0\leq k \leq n-\ell} \left| \sum_{j=k+1}^{k+\ell} \left[e_j^2 - n^{-1} \sum_{i=1}^{n} e_i^2 \right] \right|, \tag{2.33}$$

where $0 \leq \alpha < \frac{1}{2}$ is a hyperparameter known as the Hölder exponent. The authors note that 'the question of choosing parameter α remains open,' but suspect that the power of the test for detecting a changed-segment of a particular length varies with α.

The authors show that under certain conditions, $T \xrightarrow{D} T_\alpha$, where

$$T = n^{-1/2} \hat{\delta}_n^{-1} T_{n,\alpha},$$
$$\hat{\delta}_n^2 = n^{-1} \sum_{j=1}^{n} \left[e_j^2 - n^{-1} \sum_{i=1}^{n} e_i^2 \right]^2,$$
$$T_\alpha = \sup_{0<h<1} h^{-\alpha} \sup_{0\leq t<1-h} |B_{t+h} - B_t|,$$

and B_t is a Brownian bridge on $t \in [0,1]$. The test is right-tailed: large values of T provide evidence of heteroskedasticity. However, Račkauskas and Zuokas (2007) do not provide an exact or asymptotic null distribution for T_α but instead resort to a MC simulation scheme to approximate critical values. For the selected Hölder exponent values $\alpha = j/32$, $j = 0, 1, \ldots, 15$, they propose to generate $R = 2^{14}$ replications of approximations for T_α. In each replication, the Brownian bridge is approximated by the partial sum process

$$\xi_m(0) = 0,$$
$$\xi_m(t) = \sum_{j=1}^{[mt]} Z_j + (mt - [mt]) Z_{[mt]+1} - t \sum_{j=1}^{m} Z_j, \; t \in (0,1],$$

where Z_j, $j = 1, 2, \ldots, m$ are generated independent standard normal random variates and $m = 2^{17}$. Empirical quantiles can then be used to compute the critical value for a given significance level.

2.1.20 Yüce's Test

Yüce (2008) proposes two simple tests for heteroskedasticity in which the test statistic is a function of the OLS residuals. As with the tests of Li and Yao (2019), there is no deflator variable or auxiliary design matrix involved. The test is intended as an omnibus test that can detect various kinds of heteroskedasticity, monotonic and nonmonotonic. The two test statistics are denoted T_A and T_B respectively. The asymptotic null distribution of T_A is $\chi^2(n-p)$, while the asymptotic null distribution of T_B is $t(n-p)$. The test statistics are calculated as follows:

$$T_A = \frac{\sum_{i=1}^{n} e_i^2}{\Theta}, \tag{2.34}$$

and

$$T_B = \frac{\hat{\omega}_{\text{ub}} - \Theta}{\sqrt{2(n-p)^{-1}\Theta^2}}, \tag{2.35}$$

where $\Theta = (\pi - 2 + 2n(n-p))^{-1} \pi \left(\sum_{i=1}^{n} |e_i| \right)^2$.

2.1.21 Zhou, Song, and Thompson's Test

Zhou et al. (2015) propose an information ratio (IR) approach to heteroskedasticity testing based on comparisons between sandwich and model-based estimators for the variances of individual regression coefficient estimators. Their test includes a 'covariate-specific' method, a 'pooled' method, and a 'hybrid' method. It allows for a two-step procedure in which the first step is an overall test of heteroskedasticity and the second step—a *post hoc* analysis undertaken only when the first null hypothesis is rejected—facilitates identification of the design or auxiliary design variable(s) associated with the error variance.

The 'covariate-specific' method may be described as follows. Let $\bar{\omega} = n^{-1} \sum_{i=1}^{n} e_i^2$ and let $\boldsymbol{Z}$ be an $n \times p'$ auxiliary design matrix. Then, let $\boldsymbol{H_Z} = \boldsymbol{Z}\left(\boldsymbol{Z}'\boldsymbol{Z}\right)^{-1}\boldsymbol{Z}'$ be the hat matrix of the auxiliary design and, for $j = 1, 2, \ldots, p'$, let

$$\boldsymbol{H}_{\boldsymbol{Z}}^{(-j)} = \boldsymbol{Z}_{(-j)}\left(\boldsymbol{Z}'_{(-j)}\boldsymbol{Z}_{(-j)}\right)^{-1}\boldsymbol{Z}'_{(-j)},$$

where $\boldsymbol{Z}_{(-j)}$ is the auxiliary design matrix with the jth column omitted and $\boldsymbol{H}_{\boldsymbol{Z}}^{(-j)}$ is *its* hat matrix. Now, define the jth IR statistic as

$$\mathrm{IR}_j = \bar{\omega}^{-1} \sum_{i=1}^{n} w_i^{(j)} e_i^2,$$

where

$$w_i^{(j)} = \mathfrak{h}_{ii} - \mathfrak{h}_{ii}^{(-j)},$$

and $\mathfrak{h}_{ii}$ and $\mathfrak{h}_{ii}^{(-j)}$ are, respectively, the diagonal elements of $\boldsymbol{H_Z}$ and $\boldsymbol{H}_{\boldsymbol{Z}}^{(-j)}$. The test statistic pertaining to the jth auxiliary design variable is then,

$$T_j = \sqrt{n}\left(\mathrm{IR}_j - 1\right). \tag{2.36}$$

Zhou et al. (2015) show that the null distribution of T_j is asymptotically normal with 0 mean. However, the variance of this asymptotic null distribution is intractable. They thus propose a perturbation sampling scheme to compute estimated p-values. A perturbed version of T_j may be defined as,

$$T_j^{\star} = \sqrt{n} \sum_{i=1}^{n} \left\{ \left(w_i^{(j)} - n^{-1}\mathrm{IR}_j\right)\left(\bar{\omega}^{-1}e_i^2 - 1\right) - n^{-1}\left(\mathrm{IR}_j - 1\right) \right\} \xi_i, \tag{2.37}$$

where ξ_i, $i = 1, 2, \ldots, n$, are independent and identically distributed (iid) random variables having zero mean and unit variance (e.g., standard normal). Zhou et al. (2015) show that the variance of $T_j^{\star}$, conditioning on the observed data, converges in probability to the variance of the asymptotic null distribution of T_j. Hence, the following procedure applies:

1. Generate B independent values $T_j^{\star(b)}$, $b = 1, 2, \ldots, B$.
2. Compute the empirical p-value for T_j, as $P_j = B^{-1} \sum_{b=1}^{B} 1_{T_j^{\star(b)} \geq T_j}$, where $1_{\bullet}$ is the indicator function.
3. Perform a Bonferroni correction; thus, the reported p-values are $\{p'P_j\}$, $j = 1, 2, \ldots, p'$.

To reject an overall null hypothesis of homoskedasticity one should compare $P_{\text{cs}} = \min\left\{p'P_1, p'P_2, \ldots, p'P_q\right\}$ with the significance level.

The 'pooled' method proceeds in a similar fashion except that there is one overall statistic and p-value that pools the comparisons across all auxiliary covariates. The IR statistic in this case is

$$\mathrm{IR}_{\text{pool}} = \bar{\omega}^{-1} \sum_{i=1}^{n} w_i^{\text{pool}} e_i^2,$$

where $w_i^{\text{pool}} = \mathfrak{h}_{ii}/p'$. The test statistic is then,

$$T_{\text{pool}} = \sqrt{n}\left(\mathrm{IR}_{\text{pool}} - 1\right). \tag{2.38}$$

The perturbation sampling scheme proceeds exactly analogous to that of the covariate-specific method, yielding $T_{\text{pool}}^{\star(b)}$, $b = 1, 2, \ldots, B$, and p-value $P_{\text{pool}} = B^{-1} \sum_{b=1}^{B} 1_{T_{\text{pool}}^{\star(b)} \geq T_{\text{pool}}}$.

Finally, the 'hybrid' method proceeds by computing $P_{\text{hybrid}} = 2\min\left\{P_{\text{cs}}, P_{\text{pool}}\right\}$ (the 2 being a further Bonferroni correction).

2.1.22 Li-Yao Tests

Li and Yao (2019) propose two tests that are distinctive in that no prior information is required about the form of the heteroskedasticity under the alternative hypothesis. The tests are intended to have high power especially in high-dimensional regressions (i.e., when p is large) but also adequate power in low-dimensional regressions. Both tests are upper-tailed, despite having Gaussian asymptotic null distributions.

The first test is called the Approximate Likelihood Ratio Test (ALRT). Its test statistic is,

$$T_1 = \log \left[n^{-1} \sum_{i=1}^{n} e_i^2 \left(\prod_{i=1}^{n} e_i^2 \right)^{-1/n} \right]. \tag{2.39}$$

Li and Yao (2019) show that the asymptotic null distribution of T_1 is normal with a mean of $\log 2 - \psi^{(0)}(1)$ and a variance of $n^{-1}(2^{-1}\pi^2 - 2)$, where $-\psi^{(0)}(1) \approx 0.5772$ is the polygamma function of order 0 evaluated at 1, also known as the Euler constant.

The second test is called the Coefficient of Variation Test (CVT). Its test statistic is

$$T_2 = \frac{n^{-1} \sum_{i=1}^{n} (e_i^2 - \bar{\omega})^2}{\bar{\omega}^2}, \tag{2.40}$$

where $\bar{\omega} = n^{-1} \sum_{i=1}^{n} e_i^2$, as defined previously. Li and Yao (2019) show that the asymptotic null distribution of T_2 is normal with mean 2 and variance $24n^{-1}$.

A possible limitation of both tests is that they rest on the assumption that the design matrix $\boldsymbol{X}$ is stochastic and that the design variables are normally distributed. However, the authors adduce empirical evidence that the tests still perform well in terms of size and power when the design variables are generated from other distributions, or when they are held fixed. In the case of the CVT method, Bai et al. (2016) derive the asymptotic null distribution under the more conventional assumption that the design matrix is nonstochastic. Retaining the normality assumption on the errors, T_2 is shown to converge in distribution to a normal null distribution with mean a and variance b, where

$$a = \frac{3n \operatorname{tr}(\boldsymbol{M} \circ \boldsymbol{M})}{(n-p)^2 + 2(n-p)} - 1,$$

and

$$b = \boldsymbol{\Delta}' \boldsymbol{\Theta} \boldsymbol{\Delta},$$

where

$$\boldsymbol{\Delta}' = \left[\frac{n}{(n-p)^2 + 2(n-p)}, \frac{3n^2 \operatorname{tr}(\boldsymbol{M} \circ \boldsymbol{M})}{\left((n-p)^2 + 2(n-p)\right)^2} \right],$$

$$\boldsymbol{\Theta} = \begin{pmatrix} \Theta_{11} & \Theta_{12} \\ \Theta_{21} & \Theta_{22} \end{pmatrix},$$

$$\Theta_{11} = 72 \operatorname{diag}(\boldsymbol{M})' \boldsymbol{M} \circ \boldsymbol{M} \operatorname{diag}(\boldsymbol{M}) + 24 \operatorname{tr}(\boldsymbol{M} \circ \boldsymbol{M})^2,$$

$$\Theta_{12} = \Theta_{21} = 24(n-p)n^{-1} \operatorname{tr}(\boldsymbol{M} \circ \boldsymbol{M}),$$

and

$$\Theta_{22} = 8n^{-2}(n-p)^3.$$

2.1.23 Computational Extensions of Other Tests

2.1.23.1 Dufour, Khalaf, Bernard, and Genest's MC Method

Dufour et al. (2004) propose a MC procedure for estimating p-values from the exact null distribution of the test statistic of a specified heteroskedasticity test. Not all tests are suitable, because the procedure requires that the test statistic be continuous (which rules out Goldfeld and Quandt's (1965) nonparametric peaks test and the exact version of Horn's (1981) nonparametric test). It further requires that the test statistic is invariant with respect to the nuisance parameters $\boldsymbol{\beta}$ and ω, and that the test statistic can be computed from the OLS residuals $\boldsymbol{e}$, the design matrix $\boldsymbol{X}$, and any other nonstochastic auxiliary variables. This rules out Anscombe's (1961) test and Bickel's (1978) test.

The MC procedure is straightforward and can be described as follows.

1. Compute the value of the test statistic, t_0, from the observed data.
2. Generate MC random error vectors $\boldsymbol{\epsilon}^{(r)}$, $r = 1, 2, \ldots, R$, from a specified continuous distribution with mean 0 and scalar covariance matrix (the scale does not matter, due to the requirement that the test be invariant with respect to ω).[24]
3. Compute R OLS residual vectors $\boldsymbol{e}^{(r)} = \boldsymbol{M}\boldsymbol{\epsilon}^{(r)}$, $r = 1, 2, \ldots, R$, where $\boldsymbol{M}$ is the annihilator matrix defined in §1.1.2.
4. Using $\boldsymbol{e}^{(r)}$ and nonstochastic variables, hyperparameters, etc., compute R MC test statistic values $\boldsymbol{t}^{(r)}$, $r = 1, 2, \ldots, R$.
5. Compute the p-value estimate as,

$$(R+1)^{-1} \sum_{r=1}^{R} 1_{t^{(r)} \geq t_0} + 1,$$

where $1_{\bullet}$ is the indicator function.[25]

2.1.23.2 The Godfrey-Orme Method

In Godfrey and Orme (1999), a nonparametric bootstrap algorithm is offered for estimating p-values from the null distribution of the test statistic of other heteroskedasticity tests. The method is more thoroughly explained in Godfrey et al. (2006). The procedure is straightforward and can be described as follows:

1. Compute the test statistic value t_0 from the observed data.
2. Estimate the unknown CDF F of the random errors with the empirical CDF $\hat{F}_n$ of the OLS residuals, and generate a random sample from $\hat{F}_n$, $\boldsymbol{e}^{(b)} = \left[e_1^{(b)}, e_2^{(b)}, \ldots, e_n^{(b)}\right]'$, for $b = 1$ (which is equivalent to drawing a sample of size n from $\boldsymbol{e}$ with replacement).
3. Compute new response values $\boldsymbol{y}^{(b)} = \boldsymbol{X}\hat{\boldsymbol{\beta}}_{\text{OLS}} + \boldsymbol{e}^{(b)}$, where $\boldsymbol{X}$ is the original design matrix and $\hat{\boldsymbol{\beta}}_{\text{OLS}}$ the original OLS estimator of $\boldsymbol{\beta}$.
4. Fit an OLS regression of $\boldsymbol{y}^{(b)}$ on $\boldsymbol{X}$ and thus compute a bootstrapped test statistic $t^{(b)}$.
5. Repeat steps 2-4 for $b = 2, 3, \ldots, B$, thus obtaining $t^{(1)}, t^{(2)}, \ldots, t^{(B)}$.
6. Compute the p-value of the bootstrap heteroskedasticity test as

$$B^{-1} \sum_{b=1}^{B} 1_{t^{(b)} \geq t_0},$$

where $1_{\bullet}$ is the indicator function.[26]

[24] The error distribution need not be normal; the authors work with t-distributed errors in their simulations, for instance.

[25] This formula is for a right-tailed test. For a left-tailed test, the indicator is instead $1_{t^{(r)} \leq t_0}$. For a two-tailed test, the one-sided p-value is doubled, due to the distribution being an empirical one; thus it is computed as

$$2\,(R+1)^{-1} \left[\min \left\{ \sum_{r=1}^{R} 1_{t^{(r)} \geq t_0}, \sum_{r=1}^{R} 1_{t^{(r)} \leq t_0} \right\} + 1 \right].$$

[26] The above formula is for a right-tailed test. For a left-tailed test, the indicator is $1_{t^{(b)} \leq t_0}$. For a two-tailed test, the one-sided p-value is doubled (due to the distribution being an empirical one); thus it is computed as,

$$2B^{-1} \min \left\{ \sum_{b=1}^{B} 1_{t^{(b)} \geq t_0}, \sum_{b=1}^{B} 1_{t^{(b)} \leq t_0} \right\}.$$

2.1.24 Some Tests of Questionable Value

Before moving on, it is necessary to mention a few tests that have been noticed in the literature but are not discussed in detail here, due to shortcomings that seriously limit their practical value (Kalirajan 1989, Luger 2010, Murteira et al. 2013, Çelik 2017). The test of Kalirajan (1989), extended by Kalirajan and Jayasuriya (1991), requires respecifying the linear model to include two error terms, one symmetric and one asymmetric. This seems too heavy a price to pay for ostensibly improved model diagnostics. Luger (2010) proposes a simulation-based test that employs a range statistic. The test requires the user to fully specify the distribution of the random errors $\boldsymbol{\epsilon}$, which is a very burdensome requirement. Murteira et al. (2013) propose a heteroskedasticity test based on the difference between robust and nonrobust forms of Wald statistics. The user must specify a vector function of auxiliary restrictions and the authors provide no guidance on the choice of this function, which severely limits the test's practical value. Çelik (2017) proposes a Regression on Centered External Variable (RCEV) test, extended by Çelik (2018), designed to detect either monotonic or nonmonotonic heteroskedasticity. The test seems theoretically flawed, relying on an auxiliary regression in which both the response and explanatory variables are functions of the OLS residuals. As a result, the test statistic's putative null distribution seems invalid; a cursory simulation found that the test could achieve an empirical size close to 1 when the nominal size was 0.05.

2.1.25 A Summary of Heteroskedasticity Tests

It is not difficult to see why Breusch and Pagan's (1979) and White's (1980) tests have become and remained among the most popular heteroskedasticity tests for practitioners. They do not require strong assumptions regarding either the error distribution or the form of heteroskedasticity under the alternative hypothesis. Their test statistics are also easy to compute and use, without any hyperparameters to consider. By contrast, tests such as Diblasi and Bowman's (1997) and Račkauskas and Zuokas's (2007) are complicated, computationally expensive, and involve nontrivial hyperparameters. Yet there are other tests among those discussed in this chapter that have some of the good properties of the Breusch-Pagan and White tests and yet remain largely unknown and unused. In §5.1, the empirical performance of some of these tests will be evaluated using a limited simulation experiment.

More fundamental than the question of *which* heteroskedasticity test to use is the question of *whether* heteroskedasticity tests ought to be used at all. As will be discussed in §2.4.3, several empirical studies have argued that a two-stage, adaptive approach to inference on linear model parameters $\boldsymbol{\beta}$ (as described in Figure 2.1) is unwarranted. These authors have argued that it is better simply to use heteroskedasticity-robust inference methods unconditionally. If so, the utility of heteroskedasticity tests is seriously compromised. Of course, it may be of some academic interest to perform heteroskedasticity diagnostics, but in the absence of an adaptive approach to inference, heteroskedasticity testing is hardly an indispensable tool in the linear regression practitioner's toolbox. Despite this, new heteroskedasticity testing methods have continued to proliferate in the literature. The question arises, what other meaningful applications might heteroskedasticity tests have?

2.2 Feasible Parameter Estimation under Heteroskedasticity

2.2.1 Feasible Weighted Least Squares

As noted previously in §1.1.6.2, under heteroskedasticity, the BLUE of $\boldsymbol{\beta}$, $\hat{\boldsymbol{\beta}}_{\text{WLS}}$, is usually infeasible due to the weights $\boldsymbol{\Omega}^{-1}$ being usually unknown. Effective estimation of $\boldsymbol{\Omega}$, however, could provide reasonable weights that result in a WLS estimator of $\boldsymbol{\beta}$ that is more efficient than the OLS estimator.

Feasible Weighted Least Squares (FWLS) refers to an estimation procedure in a heteroskedastic linear regression model whereby the optimal but unknown weight matrix $\boldsymbol{\Omega}^{-1}$ in the WLS estimator (1.7) is replaced with an estimator $\hat{\boldsymbol{\Omega}}^{-1}$, thereby making WLS feasible (Davidson and MacKinnon 2004, p. 264). This section discusses approaches to FWLS for purposes of estimating $\boldsymbol{\beta}$. However, this discussion is also relevant to inference, because the need to estimate $\boldsymbol{\Omega}$ also arises in inference on $\boldsymbol{\beta}$ under heteroskedasticity. This is true regardless of whether an OLS or a WLS estimator is used, because the conditional covariance matrices of both estimators under heteroskedasticity ((1.6) and (1.8)) are functions of $\boldsymbol{\Omega}$.

2.2.1.1 Two-Step Feasible Weighted Least Squares

Fuller and Rao (1978) study the problem of estimation in a linear model in which the design values are divided into subgroups, with the error variance being constant within each subgroup (e.g., replications in a designed experiment). They propose a two-step procedure as follows.

1. Estimate $\boldsymbol{\Omega}$ with some estimator $\hat{\boldsymbol{\Omega}}$.[27]
2. Estimate $\boldsymbol{\beta}$ with $\hat{\boldsymbol{\beta}}_{\text{FWLS}} = \left(\boldsymbol{X}'\hat{\boldsymbol{\Omega}}^{-1}\boldsymbol{X}\right)^{-1}\boldsymbol{X}'\hat{\boldsymbol{\Omega}}^{-1}\boldsymbol{y}$.

For the first step, Fuller and Rao (1978) suggest estimating $\boldsymbol{\Omega}$ with a diagonal matrix with ith diagonal element

$$\hat{\omega}_i = \left(y_i - \boldsymbol{X}_{i\cdot}'\hat{\boldsymbol{\beta}}_{\text{OLS}}\right)^2 = e_i^2.$$

2.2.1.2 Iterative Feasible Weighted Least Squares

Hooper (1993), also in the context of a replication model, discusses an iterative procedure as follows.

1. Obtain an initial estimate of $\boldsymbol{\Omega}$.
2. Estimate $\boldsymbol{\beta}$ with $\hat{\boldsymbol{\beta}}_{\text{FWLS}} = \left(\boldsymbol{X}'\hat{\boldsymbol{\Omega}}^{-1}\boldsymbol{X}\right)^{-1}\boldsymbol{X}'\hat{\boldsymbol{\Omega}}^{-1}\boldsymbol{y}$, where $\hat{\boldsymbol{\Omega}}$ is the most updated estimate of $\boldsymbol{\Omega}$.
3. Update the estimate of $\boldsymbol{\Omega}$ using the most updated estimate of $\boldsymbol{\beta}$.
4. Repeat steps 2 and 3 until some convergence criterion is met.

A simplistic approach to the iterative procedure, discussed by Hooper (1993)—again, in the context of a replication model—entails using the squared OLS residuals as the diagonal elements of the initial covariance estimator, and in step 3, using the squared residuals computed from the $\hat{\boldsymbol{\beta}}$ estimate obtained in step 2. However, Hooper (1993, p. 179) observes that in practice, this iterative procedure is more efficient than the two-step procedure only in cases of extreme heteroskedasticity. He explains as follows:

> ‘The poor performance of the ML estimator [obtained by iteration] seems to be the result of a feedback effect caused by small changes in the [$\hat{\omega}_i$] near 0. Variances for some groups [more generally, observations] are underestimated, and these groups [observations] are given greater weight in the subsequent estimate of $\boldsymbol{\beta}$, tending to produce even smaller variance estimates in the next step.’

One could seek to prevent this feedback loop by including a Winsorisation substep within step 3:

3′. Estimate the ith diagonal element of $\boldsymbol{\Omega}$ with

$$\hat{\omega}_i = \begin{cases} c_1 & \text{if } e_i^2 < c_1 \\ \hat{\omega}_i & \text{if } c_1 \le e_i^2 \le c_2 \\ c_2 & \text{if } e_i^2 > c_2 \end{cases},$$

where e_i^2 is the ith squared residual computed using the $\hat{\boldsymbol{\beta}}$ estimate obtained in step 2, and $0 < c_1 < c_2$ are fences determined using some outlier detection method.[28]

One may consider setting $c_2 = \infty$, since the main concern is to bound ω_i below (thus bound the weights above) and thus prevent a few observations from dominating the WLS estimator of $\boldsymbol{\beta}$ due to extremely large weights.

Such ‘noninformative’ two-step and iterative procedures may be the best option when no other means is available for estimating $\boldsymbol{\Omega}$. However, in practice it may be reasonable to assume that the conditional variances $\omega_i = \text{Var}(\epsilon_i)$ depend on an auxiliary design matrix $\boldsymbol{Z}$ (which, in the simplest case, could be identical to $\boldsymbol{X}$ or consist of some subset of the columns of $\boldsymbol{X}$). Thus, step 1 of the two-step procedure, or step 3 of the iterative procedure, need not consist merely of estimating the ω_i with squared model residuals based on the best available estimate of $\boldsymbol{\beta}$, but could include a substep in which the ω_i are estimated by modelling the relationship between these squared model residuals and the design variables. Such approaches are discussed next.

[27] An intuitive option would be to estimate the error variance of each subgroup with some function of the squared OLS residuals of that subgroup.

[28] Possible methods of computing the fences include Tukey's rule (Tukey 1977), where $c_1 = Q_1 - 1.5(Q_3 - Q_1)$ and $c_2 = Q_3 + 1.5(Q_3 - Q_1)$ (Q_1 and Q_3 being the first and third quartiles of the $\hat{\omega}_i$), and the Hampel filter (Hampel 1974), where $c_1 = Q_2 - 3\text{MAD}$, $c_2 = Q_2 + 3\text{MAD}$ (Q_2 being the second quartile or median of the $\hat{\omega}_i$ and $\text{MAD} = \text{median}(|\hat{\omega}_i - Q_2|)$ being their median absolute deviation (MAD)).

2.2.1.3 Modelling Approaches to Error Variance Estimation for FWLS

An early such modelling approach is that of Robinson (1987), who proposes to estimate the ω_i in the two-step procedure described in §2.2.1.1 using a k-nearest-neighbours approach. Specifically, the estimator of ω_i is a linear combination of OLS residuals for the ith observation's k nearest neighbours in the covariate space. Regression modelling approaches were also developed, and are summarised by Davidson and MacKinnon (2004) as follows. Assume that the error variances are defined by,

$$\omega_i = \exp\left\{\boldsymbol{Z}'_{i\cdot}\boldsymbol{\zeta}\right\},\ i = 1, 2, \ldots, n, \tag{2.41}$$

where $\boldsymbol{\zeta}$ is a p'-vector of parameters and $\boldsymbol{Z}'_{i\cdot}$ is a p'-vector of nonstochastic, observed variables.[29] The exponential function in (2.41) ensures that the ω_i are positive. The advantage of the assumption (2.41) is that one needs to estimate only p' parameters (the elements of $\boldsymbol{\zeta}$) to estimate $\boldsymbol{\Omega}$, whereas the noninformative methods entail estimating n parameters (the individual ω_i).

In the two-step procedure described in §2.2.1.1, step 1 would now consist of the following substeps:

1a. Obtain the OLS residuals $\boldsymbol{e}$.

1b. Obtain parameter estimates $\hat{\boldsymbol{\zeta}}$ by fitting the regression (2.42) using OLS

$$\log e_i^2 = \boldsymbol{Z}'_{i\cdot}\boldsymbol{\zeta} + u_i,\ i = 1, 2, \ldots, n. \tag{2.42}$$

Here, u_i is a random error whose distribution may or may not be specified, but which is assumed to have zero mean.

1c. Estimate the ith diagonal element of $\boldsymbol{\Omega}$ with $\hat{\omega}_i = \exp\left\{\boldsymbol{Z}'_{i\cdot}\hat{\boldsymbol{\zeta}}\right\}$.

Alternatively, this approach could be applied within the iterative procedure, with step 3 (as previously described in §2.2.1.2) replaced by steps 3a-3c:

3a. Obtain residuals $\boldsymbol{e}$ using the most recent estimate of $\boldsymbol{\beta}$.

3b. Obtain estimates $\hat{\boldsymbol{\zeta}}$ by fitting the regression (2.42) using OLS (with the e_i^2 obtained in the previous substep as the responses).

3c. Estimate the ith diagonal element of $\boldsymbol{\Omega}$ with $\hat{\omega}_i = \exp\left\{\boldsymbol{Z}'_{i\cdot}\hat{\boldsymbol{\zeta}}\right\}$.

A more recent approach is that of Miller and Startz (2019), who generalise the conventional approach described in Davidson and MacKinnon (2004) by assuming that the ω_i are related to the observed design variables by some function $g(\cdot)$ (not necessarily linear in the parameters):[30]

$$\omega_i = \exp\left\{g(\boldsymbol{X}'_{i\cdot})\right\},\ i = 1, 2, \ldots, n. \tag{2.43}$$

The function $g(\cdot)$ is then estimated using some regression approach; Miller and Startz (2019) suggest Support Vector Regression (SVR) as the most promising option. Thus, steps 3b-3c in the method of Davidson and MacKinnon (2004) become:[31]

3b′. Estimate $g(\boldsymbol{X}'_{i\cdot})$ by $\hat{g}(\boldsymbol{X}'_{i\cdot})$ using SVR (or another regression method).[32]

3c′. Estimate the ith diagonal element of $\boldsymbol{\Omega}$ with $\hat{\omega}_i = \exp\{\hat{g}(\boldsymbol{X}'_{i\cdot})\}$.

[29] Again, a common special case would be where $\boldsymbol{Z}'_{i\cdot} = \boldsymbol{X}'_{i\cdot}$, or $\boldsymbol{Z}$ is some subset of the columns of $\boldsymbol{X}$ (or perhaps also including *functions* of the columns of $\boldsymbol{X}$).

[30] Miller and Startz (2019) assume the special case where $\boldsymbol{Z} = \boldsymbol{X}$, but their approach easily generalises to any choice of auxiliary design matrix $\boldsymbol{Z}$.

[31] A similar modification of steps 1b-1c would take place if the two-step procedure is being used. Although Miller and Startz (2019) do not discuss the choice of a two-step vs. an iterative procedure, it is clear from the functions in their supplementary R code that their procedure is iterative.

[32] R functions provided by Miller and Startz (2019) allow the user to choose between OLS, WLS, regression trees, random forest regression, k-nearest-neighbours, kernel regression (nonparametric regression), or SVR as the method for estimating $g(\cdot)$.

2.2.2 Adaptive Least Squares

Romano and Wolf (2017) propose an estimation procedure that they call Adaptive Least Squares (ALS). This is a two-stage procedure in which the first stage is a test for heteroskedasticity. If the null hypothesis of homoskedasticity is retained, $\boldsymbol{\beta}$ is estimated using OLS. If, however, the null hypothesis of homoskedasticity is rejected, $\boldsymbol{\beta}$ is estimated using FWLS. The authors state that the benefit of ALS is that it 'sacrifices some efficiency gains of WLS under conditional heteroskedasticity in favor of being closer to the performance of OLS under conditional homoskedasticity' (Romano and Wolf 2017, p. 3).

The efficiency of ALS is obviously dependent on the size and power of the heteroskedasticity test used in the first stage. Romano and Wolf (2017) recommend that the heteroskedasticity test used should be built around the same parametric model that would then be used to estimate $\boldsymbol{\Omega}$ for FWLS if heteroskedasticity is detected.

2.3 Heteroskedasticity-Consistent Covariance Matrix Estimators

Estimation of the random errors' variance-covariance matrix, $\boldsymbol{\Omega}$, was discussed in §2.2.1 as a means to obtain appropriate weights for FWLS estimation of $\boldsymbol{\beta}$. Another benefit of obtaining a good estimate of $\boldsymbol{\Omega}$ concerns inference on the partial slope coefficients β_j. Observe from (1.6) that $\text{Cov}(\hat{\boldsymbol{\beta}}_{\text{OLS}})$ depends on $\boldsymbol{\Omega}$ under heteroskedasticity and from (1.8) that $\text{Cov}(\hat{\boldsymbol{\beta}}_{\text{WLS}})$ does as well. This suggests that feasible and scale-invariant test statistics for inference on $\boldsymbol{\beta}$—regardless of whether they are built around $\hat{\boldsymbol{\beta}}_{\text{OLS}}$ or $\hat{\boldsymbol{\beta}}_{\text{FWLS}}$—require estimation of $\boldsymbol{\Omega}$. Hence, estimation of $\boldsymbol{\Omega}$ facilitates inferences on $\boldsymbol{\beta}$ under heteroskedasticity (see §2.4 below). Undoubtedly, estimation of $\boldsymbol{\Omega}$ is very important in a heteroskedastic linear model, albeit usually as a means to an end rather than an end in itself (since practitioners are typically more interested in the conditional mean of $\boldsymbol{y}$ than in its covariance structure).

Interestingly, although the problem of estimation of $\boldsymbol{\beta}$ and the problem of inference on $\boldsymbol{\beta}$ under heteroskedasticity are closely related and both require estimation of $\boldsymbol{\Omega}$, approaches to estimating $\boldsymbol{\Omega}$ in the literature *for purposes of estimation of* $\boldsymbol{\beta}$ differ markedly from approaches to estimating $\boldsymbol{\Omega}$ *for purposes of inference on* $\boldsymbol{\beta}$. In this section Heteroskedasticity-Consistent Covariance Matrix Estimators (HCCMEs) are introduced, of which many have been proposed in the literature. These are usually expressed as estimators of $\text{Cov}(\hat{\boldsymbol{\beta}}_{\text{OLS}})$ rather than of $\boldsymbol{\Omega}$ (since they are purposed mainly to the end of computing a standard error estimate for a test statistic for inference on $\boldsymbol{\beta}$), but since $\boldsymbol{\Omega}$ is the only unobserved parameter in (1.6), they in fact differ only in the method of estimating $\boldsymbol{\Omega}$. Because these estimators are developed for models in which A3 is retained, they are all diagonal $n \times n$ matrices.

The nomenclature that has developed for the various particular HCCMEs is to denote them by HC#?, where # is an integer (currently between 0 and 7) and ? is an optional letter m, used to denote a minor modification of another HCCME. Here, this nomenclature for denoting the estimators symbolically will also be used; thus $\hat{\boldsymbol{\Omega}}_0$, $\hat{\boldsymbol{\Omega}}_1$, etc.

2.3.1 HC0

The original HCCME, denoted HC0, was proposed by White (1980), who also coined the term HCCME in a seminal paper that also proposed the well-known and eponymous heteroskedasticity test. The estimator entails estimating $\boldsymbol{\Omega}$ with

$$\hat{\boldsymbol{\Omega}}_0 = \text{diag}\,\{\boldsymbol{e} \circ \boldsymbol{e}\}. \tag{2.44}$$

2.3.2 HC1 and HC2

MacKinnon and White (1985) proposed three alternatives, HC1, HC2, and HC3 (the latter to be discussed in §2.3.3), designed to improve on the finite-sample properties of HC0, which are poor. HC1 had actually been suggested already by Hinkley (1977). It entails a degrees-of-freedom multiplicative adjustment of $\hat{\boldsymbol{\Omega}}_0$, the same multiplicative factor $n/(n-p)$ that transforms $\bar{\omega}$ to $\hat{\omega}_{\text{ub}}$:[33]

$$\hat{\boldsymbol{\Omega}}_1 = \frac{n}{n-p}\hat{\boldsymbol{\Omega}}_0. \tag{2.45}$$

[33] See (1.4) and (1.5) in §1.1.5.3 for derivation of $\bar{\omega}$ and definition of $\hat{\omega}_{\text{ub}}$ and see §3.1.1 for discussion of the unbiasedness of $\hat{\omega}_{\text{ub}}$.

MacKinnon and White (1985) propose HC2 based on the earlier work of Horn et al. (1975). The latter authors work with the WLS residuals, and thus with a generalisation of $\boldsymbol{H}$ for WLS ($\boldsymbol{H}_W = \boldsymbol{X}(\boldsymbol{X}'\boldsymbol{W}\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{W}$). Obtaining an expression for $\mathrm{E}(e^2_{\mathrm{WLS},i})$ equivalent to (1.20) (but with the elements of $\boldsymbol{H}$ replaced with those of $\boldsymbol{H}_W$), they show that, if the weights are 'correct', i.e., $\boldsymbol{W} = \boldsymbol{\Omega}^{-1}$, then $\mathrm{E}(e^2_{\mathrm{WLS},i}) = (1 - h_{W,ii})\omega_i$. This leads them to propose an 'almost unbiased estimator,' $\hat{\omega}_{\mathrm{AUE},i} = (1 - h_{W,ii})^{-1} e^2_{\mathrm{WLS},i}$. It is almost unbiased in the sense that, in practice, $\boldsymbol{W} \neq \boldsymbol{\Omega}^{-1}$, since $\boldsymbol{\Omega}$ is usually unknown. MacKinnon and White (1985) retroject this 'almost unbiased estimator' into the OLS case by replacing the $h_{W,ii}$ with h_{ii} and $e^2_{\mathrm{WLS},i}$ with e^2_i. They thus arrive at the HC2 estimator, in which

$$\hat{\boldsymbol{\Omega}}_2 = \operatorname{diag}\left\{(1 - h_{11})^{-1} e_1^2, (1 - h_{22})^{-1} e_2^2, \ldots, (1 - h_{nn})^{-1} e_n^2\right\}. \tag{2.46}$$

Surprisingly, when stating that HC2 is founded on the estimation approach of Horn et al. (1975), MacKinnon and White (1985) do not distinguish between $h_{W,ii}$ and h_{ii} or between $e^2_{\mathrm{WLS},i}$ and e^2_i.[34] This is not inconsequential because, while $\hat{\boldsymbol{\Omega}}_2$ (OLS version) is an unbiased estimator of $\boldsymbol{\Omega}$ under homoskedasticity, the 'almost unbiased' property seems not to apply under heteroskedasticity (see (1.20), where the c_i term *does not* simplify to $h_{ii}\omega_i$ except under homoskedasticity).[35]

2.3.3 HC3

MacKinnon and White (1985) derive HC3 by a very different method from HC2, beginning from a jackknife estimator of $\mathrm{Cov}(\hat{\boldsymbol{\beta}})$, but in the end it is similar in form:

$$\hat{\boldsymbol{\Omega}}_3 = \operatorname{diag}\left\{(1 - h_{11})^{-2} e_1^2, (1 - h_{22})^{-2} e_2^2, \ldots, (1 - h_{nn})^{-2} e_n^2\right\}. \tag{2.47}$$

The idea of a jackknife estimator is to recompute the model estimates n times, each time leaving out one observation, then using the variability between the leave-one-out estimates to estimate the variability of the original estimator. Hence, HC3 is related to $\hat{\boldsymbol{\beta}}_{(-i)}$ introduced previously in the context of externally studentised residuals (§1.1.10.2). Specifically, since $\hat{\boldsymbol{\beta}}_{(-i)}$ can be written as $\hat{\boldsymbol{\beta}} - \dfrac{1}{1 - h_{ii}}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}_{i\cdot} e_i$, $i = 1, 2, \ldots, n$, it follows that $\hat{y}_{i,(-i)} = \boldsymbol{X}'_{i\cdot}\hat{\boldsymbol{\beta}}_{(-i)}$ can be written as

$$\begin{aligned} \hat{y}_{i,(-i)} = \boldsymbol{X}'_{i\cdot}\hat{\boldsymbol{\beta}}_{(-i)} &= \boldsymbol{X}'_{i\cdot}\hat{\boldsymbol{\beta}} - \frac{h_{ii}}{1 - h_{ii}} e_i \\ &= \hat{y}_i - \frac{h_{ii}}{1 - h_{ii}} e_i, \end{aligned}$$

from which it follows that $e_{i,(-i)} = y_i - \hat{y}_{i,(-i)}$ can be written as

$$\begin{aligned} e_{i,(-i)} &= e_i - \frac{h_{ii}}{1 - h_{ii}} e_i \\ &= \frac{e_i}{1 - h_{ii}}. \end{aligned} \tag{2.48}$$

Though MacKinnon and White (1985) do not note this, there may be a stronger case for considering the elements of $\hat{\boldsymbol{\Omega}}_3$ to be 'almost unbiased' than the elements of $\hat{\boldsymbol{\Omega}}_2$. As was pointed out in noted 13, in the absence of extremely high-leverage observations, $\mathrm{E}(e_i^2)$ is dominated by the term $(1 - h_{ii})^2\omega_i$ in (1.18), in which case $\mathrm{E}\left[(1 - h_{ii})^{-2} e_i^2\right] \approx \omega_i$.

MacKinnon and White (1985) conclude from a simulation experiment that HC3 outperforms HC1 and HC2, which in turn outperform HC0. Further empirical corroboration of the good finite-sample properties of HC3 was provided by Long and Ervin (2000). Without any other notable HCCME proposals appearing for nearly two decades, HC3 became established in practice. It remains the default HCCME in the **sandwich** R package (Zeileis 2006).

[34] (MacKinnon and White 1985, pp. 307-308) state that Horn et al. (1975) 'suggest using' the estimator with h_{ii} and e_i^2, whereas Horn et al. (1975) refer only to the WLS version throughout.

[35] Li et al. (2017) incorrectly state that Cribari-Neto and Zarkos (1999) show that the HC2 estimator is 'almost unbiased.'

2.3.4 HC4

A new HCCME, dubbed HC4, is designed by Cribari-Neto (2004) specifically to adjust for the effect of high-leverage design points on quasi-t tests for significance of β_j parameters. The pattern in HC2 and HC3 was to adjust for the biasing effect of influential design points on the e_i^2 by dividing by some power of $1 - h_{ii}$; call it δ_i. HC3 thus 'discounts' the effect of the h_{ii} more than HC2 ($\delta_i = 2$ with HC3 whereas $\delta_i = 1$ with HC2). This notion of discounting is repeated in a more nuanced way in HC4, since the power of $1 - h_{ii}$ now varies with i: $\delta_i = \min\left\{\frac{h_{ii}}{\bar{h}}, 4\right\}$, where $\bar{h} = p/n$ is the mean of the h_{ii}. Thus the level of discounting is proportional to the leverage score, but truncated at a maximum of 4. A possible criticism of this approach is that the choice of 4 as the truncation point seems somewhat arbitrary.

Most of the HCCMEs can be expressed in the form,

$$\hat{\mathbf{\Omega}}_{\#} = \operatorname{diag}\left\{(1-h_{11})^{-\delta_1}e_1^2, (1-h_{22})^{-\delta_2}e_2^2, \ldots, (1-h_{nn})^{-\delta_n}e_n^2\right\}. \tag{2.49}$$

This allows one to distinguish between HCCMEs by describing only the power δ_i, $i = 1, 2, \ldots, n$. This notation will be used to describe the remaining HCCMEs, with the exception of HC6, which does not have the form of (2.49).

2.3.5 HC5

HC5, introduced in Cribari-Neto et al. (2007), fine-tunes the truncation point of linear discounting that had been used in HC4. Instead of $\delta_i = \min\left\{\frac{h_{ii}}{\bar{h}}, 4\right\}$, this estimator uses $\delta_i = \min\left\{\frac{h_{ii}}{\bar{h}}, \max\left\{4, \frac{kh_{\max}}{\bar{h}}\right\}\right\}$, where $0 \leq k \leq 1$ is a tuning parameter and $h_{\max} = \max_i \{h_{ii}\}$. Thus, the truncation point is the larger of 4 and some fraction of the ratio of largest leverage score to mean leverage score. This procedure therefore allows for heavier discounting than HC4 in cases of extremely high leverage. Cribari-Neto et al. (2007) note that if $k = 0$, or if $kh_{\max}/\bar{h} \leq 4$, HC5 reduces to HC4. Based on empirical analysis, the authors suggest using $k = 0.7$.

2.3.6 HC4m

A modified version of HC4, called HC4m, is suggested in Cribari-Neto and da Silva (2011). In terms of (2.49), the exponent is $\delta_i = \min\left\{\gamma_1, \frac{h_{ii}}{\bar{h}}\right\} + \min\left\{\gamma_2, \frac{h_{ii}}{\bar{h}}\right\}$, where $\gamma_1, \gamma_2 > 0$. These tuning parameters establish a *minimal*, rather than maximal, discounting level. The idea is to make discounting heavier than that of HC4 for low-leverage observations. The authors propose to set $\gamma_1 = 1$ and, using empirical analysis, arrive at 1.5 as the best choice of γ_2. HC4m thus in a sense entails the heaviest discounting of all, since all $\delta_i \geq 2.5$.

2.3.7 HC5m

Li et al. (2017) argue that HC4 is a good estimator, but that HC5 improves on it when the data are strongly leveraged, while HC4m is preferable in the absence of high-leverage points. This, they argue, implies a weakness in the generality of HC4m and HC5: each one will perform relatively poorly under certain design conditions. They therefore propose the estimator HC5m, which combines HC4m and HC5. The discounting parameter is $\delta_i = k_1 \min\left\{\gamma_1, \frac{h_{ii}}{\bar{h}}\right\} + k_2 \min\left\{\gamma_2, \frac{h_{ii}}{\bar{h}}\right\} + k_3 \min\left\{\frac{h_{ii}}{\bar{h}}, \max\left\{4, \frac{kh_{\max}}{\bar{h}}\right\}\right\}$. They suggest using the same values of k, γ_1, and γ_2 as proposed for HC5 and HC4m, and for the new tuning parameters, they suggest $k_1 = k_3 = 1$ and $k_2 = 0$ (thus the second term falls away). Effectively, the discounting factor δ_i is being Winsorised both below (at 1, for this choice of parameters) and above (at the larger of 4 and $\frac{0.7h_{\max}}{\bar{h}}$, for this choice of parameters). It is claimed that this estimator gives the best of both worlds, performing well both in the presence and absence of high-leverage points. An estimator with no less than six tuning parameters seems overcomplicated, however.

2.3.8 HC6

Aftab and Chand (2016) develop an HCCME, HC6, that differs from HC2-HC5(m) in that the ith diagonal element of $\hat{\mathbf{\Omega}}_6$ cannot be written in the form (2.49), and more specifically, in that the e_i^2 are adjusted by a stochastic, rather than deterministic (or conditioned-on) factor:

$$\hat{\boldsymbol{\Omega}}_6 = \text{diag}\left\{\sqrt{D_1}e_1^2, \sqrt{D_2}e_2^2, \ldots, \sqrt{D_n}e_n^2\right\}, \tag{2.50}$$

where D_i is the Cook's Distance for observation i (see (1.50), (1.51), and (1.52)), $i = 1, 2, \ldots, n$. The logic of HC6 is that the squared OLS residual is adjusted by a factor "that reflects how well the model is fitted to the ith observation y_i and a component that measures the distance of the ith observation from the rest of the data." A questionable feature of this estimator is that it is effectively built around $|e_i^3|$ rather than e_i^2, and Aftab and Chand (2016) do not show that the estimator is consistent. Moreover, Cook's Distance is a metric designed for the homoskedastic scenario (hence the $\hat{\omega}_{\text{ub}}$ factor in the denominator), and may therefore not be an effective influence metric under heteroskedasticity. Another observation is that, if the HCCME is to be used for a hypothesis test of an individual element β_j, HC6 should arguably use $D_i(\beta_j)$ (see (1.53)) rather than D_i.

2.3.9 HC7

Yet another HCCME, called HC7 here,[36] is developed by Aftab and Chand (2018). They critique HC4 for poor asymptotic behaviour and HC4m and HC5 for requiring the user to specify appropriate values of tuning parameters. They thus propose a new discounting parameter, $\delta_i = \min\left\{\frac{h_{ii}}{\bar{h}}, \left(\frac{h_{\max}}{2\bar{h}}\right)^{1/2}\right\}$, that aims to achieve the leverage-oriented refinements of HC4m and HC5 without requiring user inputs.

2.3.10 A Wild Bootstrap Heteroskedasticity-Consistent Covariance Matrix Estimator

Cribari-Neto and Zarkos (1999) propose an HCCME that makes use of the 'wild bootstrap.'[37] The wild bootstrap circumvents a problem with using the basic nonparametric bootstrap in linear regression with heteroskedasticity of unknown form, namely that the heteroskedasticity cannot be mimicked in the bootstrap distribution (Davidson and Flachaire 2008, p. 163).

The procedure may be described as follows:

1. Independently draw values $r_i^{(b)}$, $i = 1, 2, \ldots, n$, from a distribution with zero mean and unit variance, for $b = 1$.
2. Compute bootstrap responses $y_i^{(b)} = \boldsymbol{X}_{i\cdot}'\hat{\boldsymbol{\beta}}_{\text{OLS}} + f_i(e_i)r_i^{(b)}$, $i = 1, 2, \ldots, n$, where $f_i(\cdot)$ is some transformation of the ith OLS residual.
3. Fit the bootstrap responses to the original design matrix using OLS to obtain $\hat{\boldsymbol{\beta}}^{(b)} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{y}^{(b)}$.
4. Repeat steps 1-3 for $b = 2, 3, \ldots, B$, thereby obtaining B bootstrap regression models.
5. Compute the sample standard deviations of the B bootstrap estimates of β_j for any $j \in \{1, 2, \ldots, p\}$ on which inference is to be performed:

$$\text{SE}(\hat{\boldsymbol{\beta}}) = \left[(B-1)^{-1}\sum_{b=1}^{B}\left(\hat{\beta}_j^{(b)} - \bar{\hat{\beta}}_j\right)^2\right]^{1/2},$$

where $\bar{\hat{\beta}}_j = B^{-1}\sum_{b=1}^{B}\hat{\beta}_j^{(b)}$.

Davidson and Flachaire (2008) note that a nonparametric version of this bootstrap procedure is obtained if the $r_i^{(b)}$ are resampled values from the finite population $a_1, a_2, \ldots, a_n$, where a_i is as defined in (2.51) (noting that $\bar{e} = 0$ if the model has an intercept):

$$a_i = \frac{e_i - \bar{e}}{\sqrt{n^{-1}\sum_{j=1}^{n}(e_j - \bar{e})^2}}, \; i = 1, 2, \ldots, n. \tag{2.51}$$

[36] Aftab and Chand (2016) and Aftab and Chand (2018) propose two different HCCMEs but name both of them HC6! In referencing their work, Salem et al. (2019) denote the HCCME from Aftab and Chand (2016) HC7 and the HCCME from Aftab and Chand (2018) HC6. Without seeking to add to the confusion, this work orders the HCCMEs chronologically by publication date, and so denotes the proposal of Aftab and Chand (2016) HC6 and that of Aftab and Chand (2018) HC7.

[37] A similar proposal is made subsequently, and apparently independently, by Zimmermann et al. (2017).

Davidson and Flachaire (2008) recommend drawing the $r_i^\star$ from the lattice distribution, i.e.,

$$r_i^{(b)} = \begin{cases} -1 & \text{with probability } 1/2 \\ 1 & \text{with probability } 1/2 \end{cases} . \tag{2.52}$$

Concerning the choice of the transformation function $f_i(\cdot)$, Davidson and Flachaire (2008) aver that the best options are either the identity transformation $f_i(e_i) = e_i$ or one of the HCCME transformations, i.e. $f_i(e_i) = e_i(1-h_{ii})^{-\delta_i/2}$. Cribari-Neto and Zarkos (1999) focus specifically on the HC2 case (see Table 2.2). Davidson and Flachaire (2008) mention that another 'variant' of the wild bootstrap is obtained by using $f_i(|e_i|)$ rather than $f_i(e_i)$. However, if the $r_i^{(b)}$ are generated from the lattice distribution in (2.52), this is no variant at all, because the original sign of e_i will still either be retained or changed with probability 1/2.

The wild bootstrap will be revisited in §3.4.1, as one of two bootstrap methods that may be used to construct approximate confidence intervals for error variances ω_i.

2.3.11 Summary of Heteroskedasticity-Consistent Covariance Matrix Estimators

Table 2.2 summarises how most of the existing HCCMEs can have the diagonal elements of their estimator of $\boldsymbol{\Omega}$ written in the form $e_i^2 c_i$, where $c_i = (1-h_{ii})^{-\delta_i}$.

Table 2.2: Heteroskedasticity-Consistent Covariance Matrix Estimator Powers δ_i

HCCME	δ_i, $i = 1, 2, \ldots, n$
HC0	0
HC1	N/A (see (2.45))
HC2	1
HC3	2
HC4	$\min\left\{\frac{h_{ii}}{\bar{h}}, 4\right\}$ (where $\bar{h} = n^{-1}\sum_{i=1}^n h_{ii} = pn^{-1}$)
HC5	$\frac{1}{2}\min\left\{\frac{h_{ii}}{\bar{h}}, \max\{4, kh_{\max}\}\right\}$ (where $k = 0.7$ and $h_{\max} = \max\{h_{11}, h_{22}, \ldots, h_{nn}\}$)
HC4m	$\min\left\{\gamma_1, \frac{h_{ii}}{\bar{h}}\right\} + \min\left\{\gamma_2, \frac{h_{ii}}{\bar{h}}\right\}$ (where $\gamma_1 = 1.0, \gamma_2 = 1.5$ are tuning constants)
HC5m	$k_1 \min\left\{\gamma_1, \frac{h_{ii}}{\bar{h}}\right\} + k_2 \min\left\{\gamma_2, \frac{h_{ii}}{\bar{h}}\right\} + k_3 \min\left\{\frac{h_{ii}}{\bar{h}}, \max\left\{4, \frac{kh_{\max}}{\bar{h}}\right\}\right\}$ (where k_1, k_2, k_3 are tuning constants)
HC6	N/A (see (2.50))
HC7	$\min\left\{\frac{h_{ii}}{\bar{h}}, \left(\frac{h_{\max}}{2\bar{h}}\right)^{1/2}\right\}$
Wild Bootstrap HCCME	N/A (see §2.3.10)

A number of means of robustifying HCCMEs are also proposed in the literature. A review of such methods can be found in Salem et al. (2019). These methods typically entail using a trimmed or weighted set of residuals rather than the OLS residuals. This places them outside the scope of the current study, which builds methods based on the statistical properties of the OLS residuals and their squares.

2.4 Inference on Model Parameters under Heteroskedasticity

2.4.1 Test Statistics Based on Ordinary Least Squares Estimators

Because the covariance of $\hat{\boldsymbol{\beta}}_{\text{OLS}}$ changes under heteroskedasticity, the estimator $\hat{\omega}_{\text{ub}}$ defined in (1.5) no longer applies (there is no longer just a single ω parameter to estimate). Therefore, the denominator of (1.33) is no longer the standard error of $\hat{\beta}_j$, and the standard errors of the interval estimator (1.34) are likewise no longer valid. Thus, a test of parameter significance using (1.33) will no longer hold its nominal size, and an interval estimator using (1.34) will no longer achieve the nominal coverage probability.

In terms of distributional results, under heteroskedasticity, $\boldsymbol{\epsilon} \sim N(\mathbf{0}, \boldsymbol{\Omega})$ and $\boldsymbol{y} \sim N(\boldsymbol{X\beta}, \boldsymbol{\Omega})$. Using the MGF argument as in §1.1.8, $M_{\boldsymbol{y}}(\boldsymbol{t}) = \exp\{\boldsymbol{\beta}'\boldsymbol{X}'\boldsymbol{t} + \boldsymbol{t}'\boldsymbol{\Omega t}\}$. Accordingly,

$$M_{\hat{\boldsymbol{\beta}}_{\text{OLS}}}(\boldsymbol{t}) = \exp\left\{\boldsymbol{\beta}'\boldsymbol{t} + \frac{1}{2}\boldsymbol{t}'(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{\Omega}\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{t}\right\}. \tag{2.53}$$

Thus, under A1 and A3-A5, $\hat{\boldsymbol{\beta}}_{\text{OLS}} \sim N(\boldsymbol{\beta}, (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{\Omega}\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1})$. Next,

$$\begin{aligned} M_{\hat{\boldsymbol{y}}}(\boldsymbol{t}) &= \exp\left\{\boldsymbol{\beta}'\boldsymbol{X}'\boldsymbol{H}'\boldsymbol{t} + \frac{1}{2}\boldsymbol{t}'\boldsymbol{H}\boldsymbol{\Omega}\boldsymbol{H}'\boldsymbol{t}\right\} \\ &= \exp\left\{(\boldsymbol{X\beta})'\boldsymbol{t} + \frac{1}{2}\boldsymbol{t}'\boldsymbol{H}\boldsymbol{\Omega}\boldsymbol{H}\boldsymbol{t}\right\} \ \text{(since } \boldsymbol{HX} = \boldsymbol{X}\text{)}. \end{aligned} \tag{2.54}$$

Thus, under A1 and A3-A5, $\hat{\boldsymbol{y}} \sim N(\boldsymbol{X\beta}, \boldsymbol{H\Omega H})$. Finally,

$$\begin{aligned} M_{\boldsymbol{e}}(\boldsymbol{t}) &= \exp\left\{\boldsymbol{\beta}'\boldsymbol{X}'\boldsymbol{M}'\boldsymbol{t} + \frac{1}{2}\boldsymbol{t}'\boldsymbol{M}\boldsymbol{\Omega}\boldsymbol{M}'\boldsymbol{t}\right\} \\ &= \exp\left\{(\mathbf{0}'\boldsymbol{t} + \frac{1}{2}\boldsymbol{t}'\boldsymbol{M}\boldsymbol{\Omega}\boldsymbol{M}\boldsymbol{t}\right\}. \end{aligned} \tag{2.55}$$

Thus, under A1 and A3-A5, $\boldsymbol{e} \sim N(\mathbf{0}, \boldsymbol{M\Omega M})$. This is the covariance matrix derived directly earlier (without requiring A5) in (1.14).

Despite these distributional results, under heteroskedasticity (A1 and A3-A5 without A2) an exact t-statistic analogous to (1.33) cannot be constructed by updating the standard error formula in the denominator based on (1.6) and then replacing $\boldsymbol{\Omega}$ with an estimator. The reason is that under the square root there is no longer a chi-square-distributed random variable divided by its degrees of freedom; no analogue of (B.3) is now available.

In what follows, approaches to inference on the linear regression partial slope coefficient parameter β_j under heteroskedasticity are discussed that are built on the OLS estimator. These results could be extended to the problem of inference on the significance of the whole regression (analogues to the F test), where the null hypothesis to be tested is $\beta_2 = \beta_3 = \cdots = \beta_p = 0$. Attention herein will be confined to the problem of inference on an individual coefficient β_j, $j \in \{1, 2, \ldots, p\}$.

2.4.1.1 Quasi-t-Tests for Inference on Individual Parameters

MacKinnon and White (1985) discuss the use of a quasi-t-test to test hypotheses about the values of individual elements of $\boldsymbol{\beta}$. The test statistic is as defined in (2.56), with the standard error estimate constructed by replacing $\boldsymbol{\Omega}$ in (1.6) with an HCCME, $\hat{\boldsymbol{\Omega}}_{\#}$:

$$T_q = \frac{\hat{\beta}_j}{\sqrt{\left((\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\hat{\boldsymbol{\Omega}}_{\#}\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\right)_{jj}}}. \tag{2.56}$$

According to MacKinnon and White (1985), this statistic has an asymptotic standard normal distribution (presumably by the Central Limit Theorem), while acknowledging that exact finite-sample results are difficult to obtain analytically. Subsequent studies of inference based on HCCMEs have similarly relied on simulation experiments more than analytical results (Long and Ervin 2000, Ng and Wilcox 2011). Apparently on *ad hoc* grounds, Ng and Wilcox (2011) propose that the statistic T_q should be compared to the $t(n-p)$ distribution rather than to the standard normal. Davidson and Flachaire (2008) state, however, that the HCCME-based quasi-t-statistics are asymptotically pivotal under the null hypothesis ($\beta_j = 0$) provided that the error variances ω_i are all positive and finite.

2.4.1.2 Wild Bootstrap Test for Significance of a Single Parameter

Cribari-Neto (2004) and Davidson and Flachaire (2008) propose to estimate p-values for the quasi-t-statistic using the wild bootstrap procedure. The test procedure consists of steps 1-4 of the wild bootstrap HCCME procedure described in §2.3.10, except that in the second step, the parameter estimates are computed from a restricted model in which $\beta_j = 0$ (thus $\boldsymbol{X}_{\cdot j}$ is omitted in the fit). The restricted design matrix can be denoted by $\boldsymbol{X}_{\cdot(-j)}$; the parameter estimates and residuals from this restricted model can be denoted by $\hat{\boldsymbol{\beta}}_{(-j)}$ and $\boldsymbol{e}_{(-j)}$, respectively. The final step is then modified as follows.

5′. Compute the empirical p-value $p^\star = \dfrac{1 + \#\left\{|T_q^{(b)}| \geq |T_q|\right\}}{B+1}$, where $T_q^{(b)}$ is the bth bootstrap value of the quasi-t-statistic T_q from (2.56).[38]

Cribari-Neto (2004) propose a double bootstrap test to achieve greater precision. This entails performing a second level of bootstrap nested within each original bootstrap replication. The inner procedure is as follows, where C is the number of replications in the inner bootstrap and b is the index of the outer bootstrap.

(i) Independently draw values $r_i^{(b,c)}$, $i = 1, 2, \ldots, n$, from a distribution with zero mean and unit variance.

(ii) Compute the bootstrap response $y_i^{(b,c)} = \boldsymbol{X}_{i\cdot}'\hat{\beta}_{(-j)}^{(b)} + f_i(e_{(-j),i}^{(b)})r_i^{(b,c)}$ for $i = 1, 2, \ldots, n$, where $\hat{\boldsymbol{\beta}}_{(-j)}^{(b)}$ and $\boldsymbol{e}_{(-j)}^{(b)}$ are the parameter estimates and associated residuals from a restricted regression of $y^{(b)}$ on $\boldsymbol{X}_{\cdot(-j)}$.

(iii) Fit the bootstrap responses $\boldsymbol{y}^{(b,c)}$ to the original design matrix using OLS to obtain $\hat{\boldsymbol{\beta}}^{(b,c)} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{y}^{(b,c)}$, and associated quasi-$t$-statistic $T_q^{(b,c)}$ (which has the form of (2.56)).

(iv) For each c, $c \in \{1, 2, \ldots, C\}$, compute the bootstrap p-value,

$$p^{(b)} = (C+1)^{-1}\left[1 + \#\left\{|T_q^{(b,c)} \geq T_q^{(b)}\right\}\right].$$

(v) Hence, compute the overall bootstrap p-value $\dfrac{1 + \#\left\{p^{(b)} \geq p^\star\right\}}{B+1}$, where $p^\star$ is the p-value of the 'single bootstrap' test described above.

Godfrey et al. (2006) find empirical evidence that this double-bootstrap procedure is the best option for inference under heteroskedasticity for very small sample sizes. A systematic review of methods for heteroskedasticity-robust inference in linear regression is given by MacKinnon (2013), who also discusses bootstrap methods for interval estimation under heteroskedasticity.

2.4.2 Test Statistics Based on Feasible Weighted Least Squares Estimators

The foregoing shows that, for several decades, much of the emphasis in scholarship on inference in heteroskedastic linear regression has focused on developing tests based on the OLS parameter estimates. Romano and Wolf (2017), however, propose to 'resurrect WLS,' that is, to construct tests that combine FWLS with HCCMEs. They assume that there exists some nonstochastic 'skedastic function' $g : \mathbb{R}^p \to \mathbb{R}$ such that $\mathrm{E}(\epsilon_i^2) = g(\boldsymbol{X}_{i\cdot}')$. This implies that $\mathrm{diag}(\boldsymbol{\Omega}) = [g(\boldsymbol{X}_{1\cdot}'), g(\boldsymbol{X}_{2\cdot}'), \ldots, g(\boldsymbol{X}_{n\cdot}')]'$ and consequently that $\boldsymbol{\Omega}$ is estimated by $\mathrm{diag}(\hat{\boldsymbol{\Omega}}) = [\hat{g}(\boldsymbol{X}_{1\cdot}', \hat{g}(\boldsymbol{X}_{2\cdot}'), \ldots, \hat{g}(\boldsymbol{X}_{n\cdot}')]'$, where $\hat{g}(\cdot)$ is an estimator of $g(\cdot)$. These authors argue that even if $\hat{g}(\cdot)$ is an inconsistent estimator of $g(\cdot)$, 'WLS can result in large efficiency gains over OLS in the presence of noticeable conditional heteroskedasticity' (Romano and Wolf 2017, p. 3). The authors recommend estimating $g(\cdot)$ using an auxiliary regression approach analogous to, but not identical to, (2.41), namely,[39]

$$g_\gamma(\boldsymbol{X}_{i\cdot}) = \exp\left\{\gamma_0 + \gamma_1 \log|X_{i1}| + \cdots + \gamma_{p-1}\log|X_{i,p-1}|\right\}, \; i = 1, 2, \ldots, n. \tag{2.57}$$

The actual auxiliary regression proposed by Romano and Wolf (2017) is (2.58), which utilises a Winsorisation approach that is necessary for their asymptotic results:

$$\log\left[\max(\delta^2, e_i^2)\right] = \gamma_0 + \gamma_1 \log|X_{i1}| + \cdots + \gamma_{p-1}\log|X_{i,p-1}|, \; i = 1, 2, \ldots, n, \tag{2.58}$$

where $\delta > 0$ is some small constant.

Their approach to inference is simply to apply the transformation,

$$\tilde{y}_i = \frac{y_i}{\sqrt{\hat{\omega}_i(\boldsymbol{X}_{i\cdot})}} \text{ and } \tilde{X}_{ij} = \frac{X_{ij}}{\sqrt{\hat{\omega}_i(\boldsymbol{X}_{i\cdot})}}, \; i = 1, 2, \ldots, n; \; j = 1, 2, \ldots, n, \tag{2.59}$$

to fit OLS to these transformed data, where $\hat{\omega}_i(\boldsymbol{X}_{i\cdot})$ is an estimate of the ith observation's error variance, assumed to be a function of the ith design point, $\boldsymbol{X}_{i\cdot}$. The quasi-t-statistic (2.56) is then compared to a $t(n-p)$ distribution to arrive at an inference.[40]

[38] For an alternative way of constructing the p-value, see MacKinnon (2013).

[39] Romano and Wolf (2017) argue that (2.57) is a more intuitive formulation than (2.41).

[40] The transformation (2.59) is equivalent to FWLS, since it entails weighting the individual observations.

The standard errors of the test statistic are in fact based on the asymptotic covariance matrix,

$$\widehat{\mathrm{Cov}}(\hat{\boldsymbol{\beta}}_{\mathrm{FWLS}}) = (\boldsymbol{X}'\hat{\boldsymbol{\Omega}}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}'\hat{\boldsymbol{\Sigma}}\boldsymbol{X}(\boldsymbol{X}'\hat{\boldsymbol{\Omega}}^{-1}\boldsymbol{X})^{-1}, \tag{2.60}$$

where $\hat{\boldsymbol{\Sigma}}$ is a diagonal matrix with ith diagonal element $\left[\dfrac{f_i(e_{\mathrm{FWLS},i})}{\hat{\omega}(\boldsymbol{X}_{i\cdot})^2}\right]^2$, and $f_i(\cdot)$ is a transformation along the lines of the HCCMEs.[41]

Miller and Startz (2019) propose a modification of the standard error of the FWLS test statistic to better account for the variability of the $\hat{\omega}(\cdot)$ estimator, which in their case uses the SVR approach described in §2.2.1.3.

2.4.3 Adaptive vs. Robust Inference on Parameters

White (1980) proposed both a new test for heteroskedasticity (described in §2.1.8) and the first HCCME, HC0 (described in §2.3.1). This study takes it for granted that these two tools ought to be used together in a two-stage approach to arrive at inferences on parameters of the linear regression model. Suppose one wishes to conduct a test of the null hypothesis $\beta_j = 0$ for some element of $\boldsymbol{\beta}$. In the first stage, one conducts a test of heteroskedasticity. As illustrated in Figure 2.1, if no heteroskedasticity is detected, a t-test is then conducted using the homoskedastic standard error estimate based on (1.3). If heteroskedasticity is detected, however, then a quasi-t-test is conducted using a heteroskedasticity-robust standard error estimate based on (1.6) (with $\boldsymbol{\Omega}$ replaced by an HCCME).

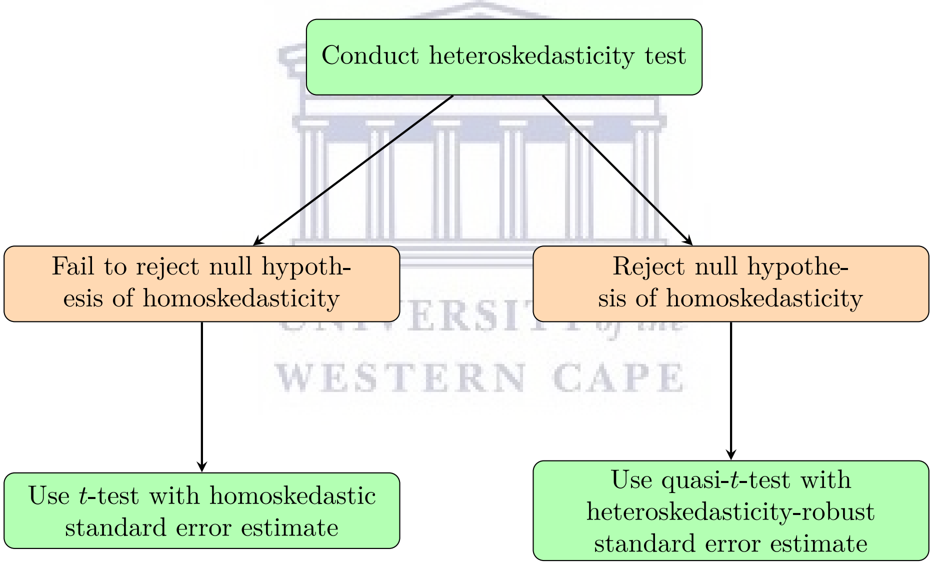

Conduct heteroskedasticity test
Fail to reject null hypothesis of homoskedasticity
Reject null hypothesis of homoskedasticity
Use t-test with homoskedastic standard error estimate
Use quasi-t-test with heteroskedasticity-robust standard error estimate

Figure 2.1: Adaptive OLS-Based Procedure for Inference on Parameter β_j

White (1980) was an extraordinarily influential study, garnering over 32 000 citations by the time of writing (5 November 2022), according to Google Scholar. The same author suggested just five years later, when proposing new HCCMEs, that ‘it may be wise to use HC3 in preference to the usual OLS covariance estimator, even when there is little evidence of heteroskedasticity’ (MacKinnon and White 1985, p. 12). This was due to a finding that the OLS covariance estimator ‘can be very seriously misleading in the presence of heteroskedasticity,’ whereas HC3 ‘does not seem to be much less reliable’ than OLS under homoskedasticity (MacKinnon and White 1985, p. 8). However, the ship may have already sailed from a practitioner’s point of view. MacKinnon and White (1985) has under 1800 citations on Google Scholar (as of 5 November 2022). Fifteen years later, Long and Ervin (2000) pointed out that HC0 was the most used HCCME in statistical software, more so than HC3 which had improved upon it from White’s point of view. Meanwhile, they found

[41] This is based on Romano and Wolf’s (2017) result (3.14). The implications for the test statistic standard errors are not clearly stated by Romano and Wolf (2017) but are mentioned in Miller and Startz (2019).

that the two-stage procedure represented in Figure 2.1 was frequently used in applied research. However, they conducted simulation studies that led them to conclude that,

> *a test for heteroscedasticity should not be used to determine whether HCCM-based tests should be used*. Far better results are obtained by using HC3 all of the time (Long and Ervin 2000, p. 223) (emphasis original).

They found in their simulations that if a quasi-t-test were conducted immediately (without the screening heteroskedasticity test), estimating $\boldsymbol{\Omega}$ in (1.6) using HC3, the test was only mildly over-sized regardless of sample size. If a screening heteroskedasticity test were introduced, the lower power of such tests when sample size is small resulted in the standard t-test being used too often, which further inflated the size. As the sample size grew large, the size of the test under the two-stage approach converged to that of the immediate quasi-t-test approach. There was thus nothing to be gained by conducting the heteroskedasticity test.

Another empirical study a decade later reached a similar conclusion: ‘performing a test of heteroscedasticity prior to applying a heteroscedastic robust test can lead to poor control over Type I errors’ (Ng and Wilcox 2011, p. 244). Again, the main source of the problem was found to be ‘the lack of power of the various tests of heteroscedasticity’ (Ng and Wilcox 2011, p. 256). Like Long and Ervin (2000), these authors recommended using HCCME-based methods unconditionally, as they ‘offer reasonable control over Type I errors under both homoscedasticity and heteroscedasticity’ (Ng and Wilcox 2011, p. 256). Another empirical study by Rosopa et al. (2018) provides further corroboration of these earlier findings. Both Ng and Wilcox (2011) and Rosopa et al. (2018) preferred HC4 as the HCCME to use unconditionally, in contrast to Long and Ervin (2000), who advocated for HC3 but whose study was published before HC4 appeared.

Romano and Wolf (2017) object to the two-stage, adaptive OLS-based approach to inference illustrated in Figure 2.1 for the same reasons argued by Long and Ervin (2000) and Ng and Wilcox (2011). However, they advocate a different adaptive approach analogous to their ALS approach to estimation. First, conduct a heteroskedasticity test. Then, use a heteroskedasticity-consistent test statistic either based on OLS (if heteroskedasticity is not detected) or based on FWLS (if heteroskedasticity is detected). They point out that under this approach, ‘the pretest...decides between two inference methods that are *both* valid under conditional heteroskedasticity’ (Romano and Wolf 2017, p. 8).

2.5 Implementation of Existing Methods in Statistical Software

2.5.1 Heteroskedasticity Tests

Of the many heteroskedasticity tests published in the past six decades, relatively few have been implemented in standard statistical software (see Table 2.3).[42]

Table 2.3: Implementation of Heteroskedasticity Tests in Statistical Software

Software	Heteroskedasticity Test						
	Goldfeld and Quandt	Glejser	Harvey	Harrison and McCabe	Breusch and Pagan	White	Cook and Weisberg
SAS					✓	✓	
R	✓			✓	✓		✓
SPSS					✓	✓	
Stata		✓	✓		✓	✓	✓
StatPlus		✓	✓		✓	✓	
EViews		✓	✓		✓	✓	
SHAZAM		✓	✓		✓	✓	✓

The inaccessibility of more recent heteroskedasticity diagnostic methods may explain why practitioners continue to prefer older methods. For example, if one considers Google Scholar citation counts since 2000 of

[42] Note: SAS features White’s (1980) Test in `PROC REG` and both Breusch and Pagan’s (1979) Test and White’s (1980) Test in `PROC MODEL`. R features Goldfeld and Quandt’s (1965) Test, Harrison and McCabe’s (1979) Test and Breusch and Pagan’s (1979) Test in **lmtest** (Zeileis and Hothorn 2002) and Cook and Weisberg’s (1983) Test in **car** (Fox and Weisberg 2019). A table similar to Table 2.3 appears in Uyanto (2019, p. 2).

publications proposing heteroskedasticity tests, White (1980) has been cited about 29200 times, Breusch and Pagan (1979) about 5640 times, and Goldfeld and Quandt (1965) 613 times. By contrast, searches of Google Scholar identified no such publication published after 1983 that has been cited even 200 times since 2000.[43]

2.5.2 Feasible Weighted Least Squares

FWLS is easily implemented in most statistical software. In R, for example, the `lm` function in the basic **stats** package allows the user to fit FWLS by specifying the weights using the `weights` argument. A leaner, faster FWLS fit can be obtained using the `lm.wfit` function, where weights are specified using the `w` argument.

2.5.3 Heteroskedasticity-Consistent Covariance Matrix Estimators

In R software, the **sandwich** package (Zeileis 2004, Zeileis et al. 2020) can be used to implement HCCMEs. Specifically, the `vcovHC` function can be used to compute an HCCME for a given linear model. The user can specify the HCCME type according to the HC# notation using the `type` argument. HC3 is the default; other options are the homoskedastic estimator, HC0, HC1, HC2, HC4, HC4m, and HC5. Notably, HC5m, HC6, HC7, and the wild bootstrap HCCME (as described in §2.3) are not included as options in this function, though one can use the `omega` argument to specify a customised function for computing the covariance matrix estimator. The name **sandwich** comes from the notion of $\text{Cov}(\hat{\boldsymbol{\beta}}_{\text{OLS}})$ as a sandwich estimator due to $\hat{\Omega}$ being 'sandwiched' by $\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}$ and its transpose (see (1.6)). Following on this metaphor, the estimation of $\boldsymbol{\Omega}$ in **sandwich** is actually done by a function called `meatHC`. The `sandwich` argument in `vcovHC` (which defaults to `TRUE`) controls whether the entire sandwich estimate is returned or just the 'meat,' $\hat{\boldsymbol{\Omega}}$.

2.5.4 Inference on Parameters

The **sandwich** package can be used in conjunction with the **lmtest** package (Zeileis and Hothorn 2002) to perform heteroskedasticity-robust inference on linear regression coefficients in R. Specifically, the `coeftest` function in **lmtest** performs quasi-t tests for significance of coefficients of a linear model object, using an HCCME specific by the `vcov` argument. `vcov` could either be a matrix estimate of (1.6) or a function (such as `vcovHC` from **sandwich**) that can compute such a matrix estimate.

2.6 Chapter Summary

This chapter offered a review of existing methods of dealing with heteroskedasticity in linear regression. First, hypothesis testing methods for detecting heteroskedasticity were reviewed. It is safe to say that this is the most thorough literature review of heteroskedasticity tests that has been produced to date. Methods of feasible parameter estimation under heteroskedasticity, Heteroskedasticity-Consistent Covariance Matrix Estimators, and approaches to inference on linear regression parameters under heteroskedasticity were also reviewed. Finally, implementations of all of these existing methods in statistical software were catalogued.

A couple of noteworthy findings from the literature review should be highlighted here. Heteroskedasticity testing methods abound in the literature, and continue to proliferate more than 50 years after they first began to appear. However, a body of literature has arisen alongside these methods that argues that they ought not to be used as part of a procedure for making inferences on the linear regression coefficients β_j. What essential purpose, then, if any, do heteroskedasticity tests play in handling heteroskedasticity in linear regression?

A second finding is that heteroskedasticity-robust methods of estimation and of inference in linear regression both involve estimation of the variance-covariance matrix of the errors, $\boldsymbol{\Omega}$. Despite this, there is a divergence in methods of estimating $\boldsymbol{\Omega}$ when the end goal is to estimate $\boldsymbol{\beta}$ as opposed to when the end goal is to make inferences on $\boldsymbol{\beta}$. In the former case, iterative and modelling approaches are popular. In the latter case, HCCMEs are preferred. This raises the question of whether handling of heteroskedasticity in linear regression can be simplified by developing a unified approach to estimation of $\boldsymbol{\Omega}$ that is appropriate *both* for the end goal of estimation *and* for the end goal of inference. Indeed, such a unified approach ought also to be useful if estimating the variances of the responses y_i is an end goal in itself.

[43] These Google Scholar searches were conducted on 5 November 2022.

3 Methodology

In the first chapter, the linear regression model and its classical assumptions were introduced, and the problems of estimation and inference on the coefficient vector $\boldsymbol{\beta}$ were introduced, including how they are affected by violation of the homoskedastic assumption, A2. Quantities useful for characterising heteroskedasticity, such as the leverage scores h_{ii} and the OLS residuals e_i, were also introduced and their properties discussed at some length.

In the second chapter, existing approaches and methods for detecting and correcting for heteroskedasticity in the linear regression model were reviewed, and certain problems and gaps were highlighted (see §2.6).

The introduction and literature review have set the stage for the development of new methods for estimating the error variances ω_i and thus enabling effective estimation of and inferences on $\boldsymbol{\beta}$ under heteroskedasticity. Such methods are developed in this chapter. However, before proposing them, some additional theoretical groundwork is needed, focusing mainly on the OLS residuals and especially their squares, e_i^2. This theory is foundational to the new methods because the squared OLS residual vector $\boldsymbol{e} \circ \boldsymbol{e}$ will be the response variable in the new models to be proposed. It is included here, rather than in Chapter 1, because it does not involve merely the restatement of well-known results, but ventures into lesser-known or even uncharted theoretical territory.[44] This new theory is the subject of §3.1.

The main methodological contribution of this study is found in §3.2, which proposes two new classes of auxiliary regression models that can be used to estimate the error variances $\boldsymbol{\omega}$ in a linear regression model.

Section 3.3 delves into the finer details of applying the new models, proposing methods for estimating the models' parameters, tuning their hyperparameters (where applicable), and selecting features to include in the models. This subsection also offers a brief discussion of the statistical properties of the new estimators. It closes by explaining how the variance estimates obtained from the new models can be used as part of a FWLS routine for estimating $\boldsymbol{\beta}$ or as part of a heteroskedasticity-robust quasi-t-test for significance of elements of $\boldsymbol{\beta}$.

Section 3.4 tackles the problem of obtaining interval estimates for the error variances ω_i. Given the intractability of analytical distributional results, bootstrap methods offer the simplest solution. A discussion of how to bootstrap a heteroskedastic linear regression model in a way that leaves intact the heteroskedastic variance structure is thus provided before delving into how to construct confidence intervals from the bootstrapped regressions.

Finally, §3.5 briefly proposes a new heteroskedasticity test based on the auxiliary variance models introduced earlier in the chapter.

3.1 Further Statistical Results on Residuals

3.1.1 Squared Ordinary Least Squares Residuals under Homoskedasticity

It was noted above in (1.11) that, under A1-A4, $\mathrm{Var}(e_i) = \mathrm{E}(e_i^2) = \omega m_{ii}$. Thus, even under homoskedasticity, the individual squared OLS residuals are biased estimators of ω. As for the sum of squared residuals $\boldsymbol{e}'\boldsymbol{e}$ (already encountered in the ML estimator of ω, (1.4)), $\mathrm{E}(\boldsymbol{e}'\boldsymbol{e}) = \sum_{i=1}^{n} \mathrm{E}(e_i^2) = \omega\,\mathrm{tr}(\boldsymbol{M})$. As noted earlier in §1.1.2, by the commutative property of the trace operator, $\mathrm{tr}(\boldsymbol{H}) = p$. Thus, $\mathrm{tr}(\boldsymbol{M}) = \mathrm{tr}(\boldsymbol{I}_n - \boldsymbol{H}) = n - p$, resulting in

$$\mathrm{E}(\boldsymbol{e}'\boldsymbol{e}) = \sum_{i=1}^{n} \omega m_{ii} = (n-p)\omega. \tag{3.1}$$

(3.1) implies that $\mathrm{E}(\bar{\omega}) = \dfrac{n-p}{n}\omega$, and that $\hat{\omega}_{\text{ub}} = (n-p)^{-1}\boldsymbol{e}'\boldsymbol{e}$ is an unbiased estimator of ω under homoskedasticity (which was already implied by (1.32) under the stronger assumption of normality).

The marginal distribution of the squared OLS residuals e_i^2, conditioning on $\boldsymbol{X}$, can also be given under A1-A5. Now, (1.31) entails that in scalar form, $e_i \sim N(0, \omega m_{ii})$. This implies, using two elementary statistical

[44] The author would be surprised if any of the theoretical results described in §3.1.1 and §3.1.2 are derived here for the first time, though they are certainly not found in standard graduate-level texts on linear regression and were not, in fact, found in their entirety in any literature consulted by the author. The distributional results on squared OLS residuals in §3.1.3 and the treatment of bias in §3.1.4 may have a stronger claim to being original.

results (Miller and Miller 2019), that $\dfrac{e_i}{\sqrt{\omega m_{ii}}} \sim N(0,1)$ and that $\dfrac{e_i^2}{\omega m_{ii}} \sim \chi^2(1)$. Expressed differently (since the chi-square distribution is a special case of the Gamma distribution), $\dfrac{e_i^2}{\omega m_{ii}} \sim \text{Gamma}\left(\alpha = \frac{1}{2}, \beta = \frac{1}{2}\right)$, where α is the shape parameter and β the rate parameter.[45] It follows by the scalability property of the Gamma distribution that,

$$e_i^2 \sim \text{Gamma}\left(\alpha = \frac{1}{2}, \beta = \frac{1}{2\omega m_{ii}}\right). \tag{3.2}$$

Since a Gamma(α, β)-distributed random variable has expectation $\alpha\beta^{-1}$ and variance $\alpha\beta^{-2}$, (3.2) aligns with the more general result established previously in (1.11) (without the normality assumption) that $\text{E}(e_i^2) = \omega m_{ii}$ and also implies that $\text{Var}(e_i^2) = 2\omega^2 m_{ii}^2$, which is proven (*with* the normality assumption) below in (C.1).

The variances and covariances of the squared OLS residuals can be derived under A1-A5 directly, without recourse to their marginal or joint distributions. The covariance of any two squared OLS residuals e_i^2, e_j^2, $i \neq j$, is given by,

$$\text{Cov}(e_i^2, e_j^2) = 2\omega^2 m_{ij}^2. \tag{3.3}$$

A proof of (3.3) is given in Appendix C.1.1. This result leads directly to an expression for the variance-covariance matrix of $\boldsymbol{e} \circ \boldsymbol{e}$:

$$\text{Cov}(\boldsymbol{e} \circ \boldsymbol{e}) = 2\omega^2(\boldsymbol{M} \circ \boldsymbol{M}). \tag{3.4}$$

From (3.3) it is evident that any two squared residuals have a positive relationship. Indeed, the correlation between any two squared OLS residuals e_i^2, e_j^2 is,

$$\text{Corr}(e_i^2, e_j^2) = \frac{m_{ij}^2}{m_{ii} m_{jj}} = \rho_{ij}^2 \text{ (where } \rho_{ij} \text{ is as in (1.13)).} \tag{3.5}$$

It turns out that if the normality assumption A5 is relaxed but A1-A4 are retained and A6′ introduced, most of the simplifications in the variance-covariance derivation in Appendix C.1.1 still hold. Unfortunately, $3\omega^2$ can no longer be substituted for $\text{E}(\epsilon_k^4)$. However, if the excess kurtosis of the errors is written as $\phi_k = \omega^{-2}\,\text{E}\left(\epsilon_k^4\right) - 3$ and $\boldsymbol{\Phi} = \text{diag}\left\{\phi_1, \phi_2, \ldots, \phi_n\right\}$, the variances and covariances can be expressed in scalar notation as follows:

$$\text{Var}(e_i^2) = 2\omega^2 m_{ii}^2 + \omega^2 \sum_{k=1}^{n} \phi_k m_{ik}^4 \tag{3.6}$$

and

$$\text{Cov}(e_i^2, e_j^2) = 2\omega^2 m_{ij}^2 + \omega^2 \sum_{k=1}^{n} \phi_k m_{ik}^2 m_{jk}^2. \tag{3.7}$$

In matrix notation, the variance-covariance matrix can be written,

$$\text{Cov}(\boldsymbol{e} \circ \boldsymbol{e}) = 2\omega^2 \boldsymbol{M} \circ \boldsymbol{M} + \omega^2 \left(\boldsymbol{M} \circ \boldsymbol{M}\right) \boldsymbol{\Phi} \left(\boldsymbol{M} \circ \boldsymbol{M}\right). \tag{3.8}$$

Significantly, the variances and covariances of the squared OLS residuals are unaffected by skewness in the error distribution. If assumption A5′ is also made, the ϕ_k can be replaced with common excess kurtosis ϕ in equations (3.6) and (3.7), and the second term in (3.8) can be rewritten as $\phi\omega^2 \left(\boldsymbol{M} \circ \boldsymbol{M}\right)\left(\boldsymbol{M} \circ \boldsymbol{M}\right)$. In this case, $\text{Cov}(\boldsymbol{e} \circ \boldsymbol{e})$ is a function of only two unknown scalar parameters, ω and ϕ.

[45] This should not be confused with the alternate parametrisation of the Gamma distribution where the second parameter is a scale parameter, the reciprocal of the rate parameter given here.

3.1.2 Squared Ordinary Least Squares Residuals under Heteroskedasticity

The marginal distribution of e_i^2 can be derived under A1 and A3-A5 using the same method used under A1-A5 in §3.1.1. First, (1.37) entails that, in scalar form, $e_i \sim N(0, \sum_{k=1}^{n} \omega_k m_{ik}^2)$. This implies that $\left[\sum_{k=1}^{n} \omega_k m_{ik}^2\right]^{-1} e_i^2 \sim \text{Gamma}(\alpha = 1/2, \beta = 1/2)$. Therefore,

$$e_i^2 \sim \text{Gamma}\left(\frac{1}{2}, \frac{1}{2}\left[\sum_{k=1}^{n} \omega_k m_{ik}^2\right]^{-1}\right). \tag{3.9}$$

Thus, under A1 and A3-A5, the marginal distribution of the OLS squared residuals given $\boldsymbol{X}$ is a Gamma distribution. Result (3.9) aligns with the previous result (1.15), established without the normality assumption, that $\mathrm{E}(e_i^2) = \sum_{k=1}^{n} \omega_k m_{ik}^2$. Result (3.9) also implies that $\mathrm{Var}(e_i^2)$ is as given in (3.12) (this also follows from (3.11), discussed below).

It follows immediately from (1.14) that

$$\mathrm{E}(\boldsymbol{e} \circ \boldsymbol{e}) = \mathrm{diag}(\boldsymbol{M\Omega M}) = (\boldsymbol{M} \circ \boldsymbol{M})\,\boldsymbol{\omega}. \tag{3.10}$$

Moreover, using the same method as in the derivation of (3.4) (see Appendix C.1.1), it can be shown that—under A1 and A3-A5—the conditional variance-covariance matrix of the squared OLS residuals is given by (3.11):

$$\mathrm{Cov}(\boldsymbol{e} \circ \boldsymbol{e}) = 2\,(\boldsymbol{M\Omega M}) \circ (\boldsymbol{M\Omega M})\,. \tag{3.11}$$

In scalar form, the variances, covariances, and correlations of the squared residuals can be expressed as in (3.12), (3.13), and (3.14):

$$\mathrm{Var}(e_i^2) = 2\left(\sum_{k=1}^{n} \omega_k m_{ik}^2\right)^2, \; i \in \{1, 2, \ldots, n\}\,, \tag{3.12}$$

$$\mathrm{Cov}(e_i^2, e_j^2) = 2\left(\sum_{k=1}^{n} m_{ik} m_{jk} \omega_k\right)^2, \; i \in \{1, 2, \ldots, n\}\,, j \in \{1, 2, \ldots, n\}\,, i \neq j, \tag{3.13}$$

and

$$\mathrm{Corr}(e_i^2, e_j^2) = \frac{\left(\sum_{k=1}^{n} \omega_k m_{ik} m_{jk}\right)^2}{\sum_{k=1}^{n} \omega_k m_{ik}^2 \sum_{\ell=1}^{n} \omega_\ell m_{j\ell}^2} = \rho_{ij}^2, \tag{3.14}$$

that is, the square of ρ_{ij}, the correlation between e_i and e_j given in (1.17). Notice that, as in the homoskedastic case, the covariances between squared OLS residuals are strictly positive.

Again, if A5 is relaxed but A6′ introduced, it is necessary to introduce the excess kurtosis, now $\phi_k = \omega_k^{-2}\,\mathrm{E}\left(\epsilon_k^4\right) - 3$. The conditional variances and covariances of the squared OLS residuals still simplify considerably—notably, they do not depend on the third moment ('skewness')—and can be expressed in scalar form as

$$\mathrm{Var}(e_i^2) = 2\left(\sum_{k=1}^{n} \omega_k m_{ik}^2\right)^2 + \sum_{k=1}^{n} \omega_k^2 \phi_k m_{ik}^4, \; i \in \{1, 2, \ldots, n\}\,, \tag{3.15}$$

and

$$\mathrm{Cov}(e_i^2, e_j^2) = 2\left(\sum_{k=1}^{n} m_{ik} m_{jk} \omega_k\right)^2 + \sum_{k=1}^{n} \omega_k^2 \phi_k m_{ik}^2 m_{jk}^2, \; i \in \{1, 2, \ldots, n\}\,, j \in \{1, 2, \ldots, n\}\,, i \neq j, \tag{3.16}$$

and, in matrix form, as

$$\mathrm{Cov}(\boldsymbol{e} \circ \boldsymbol{e}) = 2\left(\boldsymbol{M\Omega M}\right) \circ \left(\boldsymbol{M\Omega M}\right) + \left(\boldsymbol{M} \circ \boldsymbol{M}\right) \boldsymbol{\Omega\Phi\Omega} \left(\boldsymbol{M} \circ \boldsymbol{M}\right). \tag{3.17}$$

If assumption A5′ is also introduced, it implies that $\phi_k = \phi, k = 1, 2, \ldots, n$. Thus, ϕ_k can be replaced with ϕ in equations (3.15) and (3.16), and (3.17) becomes

$$\begin{aligned}\mathrm{Cov}(\boldsymbol{e} \circ \boldsymbol{e}) &= 2\left(\boldsymbol{M\Omega M}\right) \circ \left(\boldsymbol{M\Omega M}\right) + \phi\left(\boldsymbol{M} \circ \boldsymbol{M}\right) \boldsymbol{\Omega\Omega} \left(\boldsymbol{M} \circ \boldsymbol{M}\right) \\ &= 2\left(\boldsymbol{M\Omega M}\right) \circ \left(\boldsymbol{M\Omega M}\right) + \phi\left(\boldsymbol{M} \circ \boldsymbol{M}\right) \mathrm{diag}\left\{\boldsymbol{\omega} \circ \boldsymbol{\omega}\right\} \left(\boldsymbol{M} \circ \boldsymbol{M}\right). \end{aligned} \tag{3.18}$$

The simplification of $\boldsymbol{\Omega\Omega}$ to $\mathrm{diag}\left\{\boldsymbol{\omega} \circ \boldsymbol{\omega}\right\}$ is a consequence of $\boldsymbol{\Omega}$ being diagonal (due to A3). Equation (3.18) implies that, under A1, A3-A4, and A5′-A6′, $\mathrm{Cov}(\boldsymbol{e} \circ \boldsymbol{e})$ is a function of $n+1$ unknown parameters: the n-vector $\boldsymbol{\omega}$ and the scalar ϕ.

3.1.3 Joint Distribution of Squared Ordinary Least Squares Residuals

It has been shown in §3.1.1 that, under A1-A5, the squared OLS residuals e_i^2 have marginal Gamma distributions. It has likewise been shown in §3.1.2 that, under A1 and A3-A5, the squared OLS residuals e_i^2 have marginal Gamma distributions (albeit with different rate parameters than under homoskedasticity).

If the joint distribution of the squared OLS residuals were known under these two cases, this could pave the way for the development of likelihood-based methods constructed on the likelihood function of the squared OLS residuals, taken as a function of the common error variance ω (under homoskedasticity) or of the error variance vector $\boldsymbol{\omega}$ (under heteroskedasticity). This is the motivation behind the distributional results presented in this section.

Let $(U_1, V_1), \ldots, (U_m, V_m)$ be an independent random sample of size m from a bivariate normal distribution with mean $\mathbf{0}$ and covariance matrix $\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1^2 & \rho_0\sigma_1\sigma_2 \\ \rho_0\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix}$. Thus, $\mathrm{Var}(U_i) = \sigma_1^2$ and $\mathrm{Var}(V_i) = \sigma_2^2$ for $i = 1, 2, \ldots, m$ and $\mathrm{Corr}(U_i, V_j) = \rho_0,\ i \neq j$. Define $U = \dfrac{1}{2\sigma_1^2}\displaystyle\sum_{i=1}^{m} U_i^2$ and $V = \dfrac{1}{2\sigma_2^2}\displaystyle\sum_{i=1}^{m} V_i^2$. It can easily be shown, using the relationship between the zero-mean normal distribution and Gamma distribution and the scalability property of the Gamma distribution,[46] that $U, V \sim \mathrm{Gamma}\left(\alpha = \dfrac{m}{2}, \beta = 1\right)$.

Then, the joint PDF of U and V is given by

$$p(u, v; \alpha, \rho) = \frac{\rho^{-(\alpha-1)/2}}{\Gamma(\alpha)(1-\rho)} \exp\left\{-\frac{u+v}{1-\rho}\right\} (uv)^{(\alpha-1)/2} I_{\alpha-1}\left(\frac{2\sqrt{\rho uv}}{1-\rho}\right), u, v, \geq 0, \tag{3.19}$$

where $\alpha = m/2$, $\rho = \rho_0^2$, and $I_\nu(\cdot)$ is the modified Bessel function of the first kind and order ν (Kibble 1941, Balakrishnan and Lai 2009).[47] The MGF of U, V is given by

$$M(s,t) = \left[(1-s)(1-t) - \rho st\right]^{-\alpha}, 0 < \rho < 1. \tag{3.20}$$

This is known as the Kibble Bivariate Gamma Distribution, or sometimes as the Kibble-Wicksell Bivariate Gamma Distribution.

Now, using the foregoing results in §1.1.8, §1.1.9, §3.1.1, and §3.1.2, and taking $m = 1$ above, it follows immediately that, under A1-A5, for any two squared OLS residuals $e_i^2, e_j^2,\ i, j \in \{1, 2, \ldots, n\}, i \neq j,\ U = \dfrac{e_i^2}{2\sigma_i^2}$ and $V = \dfrac{e_j^2}{2\sigma_j^2}$ have the Kibble Bivariate Gamma Distribution with $\alpha = 1/2$ and where $\sigma_i^2 = \mathrm{Var}(e_i) = \omega m_{ii}$ and where $\rho_0 = \mathrm{Corr}(e_i, e_j) = \dfrac{m_{ij}}{\sqrt{m_{ii}m_{jj}}}$. The same result holds if A2 is relaxed, but now with $\sigma_i^2 = \displaystyle\sum_{k=1}^{n} \omega_k m_{ik}^2$ and ρ_0 as given in (1.17).

[46] The scalability is that if $X \sim \mathrm{Gamma}\left(\alpha, \beta\right)$ then $X/\beta \sim \mathrm{Gamma}\left(\alpha, 1\right)$. This property is easily demonstrated using the MGF.

[47] Note that the expression (3.19) in (Balakrishnan and Lai 2009, p. 304) is missing a negative sign in the exponent of (uv).

Moreover, the joint PDF of e_i^2, e_j^2 can be derived from (3.19) using the transformation technique:

$$q(e_i^2, e_j^2) = p\left(\frac{e_i^2}{2\sigma_i^2}, \frac{e_j^2}{2\sigma_j^2}\right) \begin{vmatrix} \dfrac{\partial u}{\partial e_i^2} & \dfrac{\partial u}{\partial e_j^2} \\ \dfrac{\partial v}{\partial e_i^2} & \dfrac{\partial v}{\partial e_j^2} \end{vmatrix} = \left(4\sigma_i^2\sigma_j^2\right)^{-1} p\left(\frac{e_i^2}{2\sigma_j^2}, \frac{e_j^2}{2\sigma_j^2}\right). \tag{3.21}$$

Figure 3.1 shows an example of the joint PDF (3.21) for two squared OLS residuals from a linear regression model with strong multiplicative heteroskedasticity. The error variance for one observation is more than three times as large as for the other, and this, combined with the hat matrix structure, results in a strong positive correlation of 0.73.

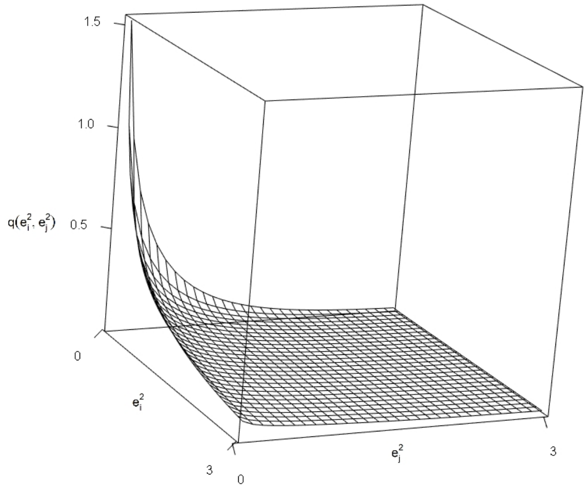


Figure 3.1: Joint PDF of Two Squared OLS Residuals from a Heteroskedastic DGP with $\rho_0 = 0.73$

Krishnamoorthy and Parthasarathy (1951) extend the Kibble distribution to the d-variate case. They give an expression for the MGF but not the joint PDF. Expressions in matrix notation for the MGF and characteristic function are found in Royen (2007) and Kotz et al. (2000).

Let $\{\boldsymbol{U}_1, \dots, \boldsymbol{U}_k\}$ be a random sample of size k from a d-variate $N(\mathbf{0}, \boldsymbol{\Sigma})$ distribution, where $\boldsymbol{\Sigma}$ is nonsingular, and define the correlation matrix as $\boldsymbol{R} = \sqrt{\operatorname{diag}(\boldsymbol{\Sigma})^{-1}}\boldsymbol{\Sigma}\sqrt{\operatorname{diag}(\boldsymbol{\Sigma})^{-1}}$, where the square root is applied elementwise. Then, $\boldsymbol{U} = \left(\frac{1}{2\sigma_1^2}\sum_{i=1}^{k} U_{1i}^2, \dots, \frac{1}{2\sigma_d^2}\sum_{i=1}^{k} U_{di}^2\right)'$, where $\operatorname{Var}(U_{ji}) = \sigma_j^2$, has the d-variate Kibble Gamma distribution with parameters $\alpha = k/2$ and $\boldsymbol{R}$. The MGF and characteristic function of $\boldsymbol{U}$ are given, respectively, by

$$M_{\boldsymbol{U}}(\boldsymbol{t}) = \mathrm{E}\left[e^{\boldsymbol{t}'\boldsymbol{U}}\right] = |\boldsymbol{I}_d - \boldsymbol{R}\boldsymbol{T}|^{-\alpha} \tag{3.22}$$

and

$$\phi_{\boldsymbol{U}}(\boldsymbol{t}) = \mathrm{E}\left[e^{i\boldsymbol{t}'\boldsymbol{U}}\right] = |\boldsymbol{I}_d - i\boldsymbol{R}\boldsymbol{T}|^{-\alpha}, \tag{3.23}$$

where $\boldsymbol{I}_d$ is the $d \times d$ identity matrix and $\boldsymbol{T} = \operatorname{diag}\{t_1, t_2, \dots, t_d\}$.

Royen (2007) notes that the PDF of the d-variate Kibble distribution is difficult to compute for $d \geq 4$. He proposes integral representations for the PDF. These integrals are $\binom{m+1}{2}$-dimensional, where m is the rank of $\boldsymbol{B}$ in a decomposition $\boldsymbol{R}^{-1} = \boldsymbol{\mathcal{D}} - \boldsymbol{B}\boldsymbol{B}'$ with a diagonal $d \times d$ matrix $\boldsymbol{\mathcal{D}}$ (which may be real-valued or complex).[48] The expression for the PDF of $\boldsymbol{U}$ is

[48] For further details of this decomposition, see section 3 of Royen (2007).

$$p(u_1, u_2, \ldots, u_d; \alpha, \boldsymbol{R}) = |\boldsymbol{R}|^{-\alpha} \prod_{j=1}^{d} \left(\frac{\exp\{-\boldsymbol{\mathcal{D}}_j u_j\}\, u_j^{\alpha-1}}{\Gamma(\alpha)} \right) \times \mathrm{E}\left[\prod_{j=1}^{d} {}_0F_1\left(\alpha; \frac{1}{2} u_j \boldsymbol{B}_j \boldsymbol{S} \boldsymbol{B}_j'\right) \right], \tag{3.24}$$

where $\boldsymbol{B}_j$ are the rows of $\boldsymbol{B}$, the expectation refers to a $\boldsymbol{W}_m(2\alpha, \boldsymbol{I}_m)$-Wishart matrix $\boldsymbol{S}$, ${}_0F_1(\cdot)$ is the generalised hypergeometric function, and $\boldsymbol{R}$ is the correlation matrix defined above. Royen (2007) offers various techniques for obtaining numerical approximations for (3.24), but it remains that the computational cost increases rapidly with m.

The n-variate Kibble distribution is, in principle, the distribution of the squared OLS residuals under A1-A5 (or even under A1 and A3-A5, i.e., heteroskedasticity), at least up to a scale factor (which could be addressed by multiplying by a Jacobian determinant depending only on parameters, as in (3.21)).[49] However, aside from the difficulties with evaluating (3.24) (especially for large n), the singularity of $\boldsymbol{\Sigma} = \mathrm{Cov}(\boldsymbol{e})$ in this case (and therefore of $\boldsymbol{R} = \mathrm{Corr}(\boldsymbol{e})$) implies that the distribution is, like that of $\boldsymbol{e}$, degenerate.[50] The degeneracy problem can be circumvented by sacrificing p observations and beginning from an $n-p$-vector of residuals, the covariance matrix of which is nonsingular. This would, in principle, allow ML methods to be used on an auxiliary model with only $n-p$ observations. However, the loss of p observations, coupled with the difficulty of computing (3.24) (never mind maximising the associated likelihood function) combine to make this approach to variance estimation undesirable. A distribution-based approach to modelling the error variances is thus not pursued further in this study.

Appendix C.2 offers some approximate statistical results on the moments of the logarithms of the squared OLS residuals, both under homoskedasticity and under heteroskedasticity, based on Taylor expansions. These results have been relegated to an appendix because they ultimately played no role in the new variance estimation methods proposed herein.

3.1.4 Squared Ordinary Least Squares Residuals as Estimators of the Error Variance(s)

In practice, the OLS residuals e_i are often thought of as predictions of, or even as proxies for, the unobserved errors ϵ_i. As a result, the squared OLS residuals e_i^2 are often taken as estimators of the common error variance ω (under homoskedasticity) or of the individual error variances ω_i (under heteroskedasticity). Indeed, this approach provides the rationale for the original HCCME, HC0, of which most subsequent HCCMEs are transformations (see §2.3). In this subsection, it will be shown that treating the e_i^2 as estimators of the corresponding ω_i under heteroskedasticity is problematic from a bias standpoint. This will motivate a new approach to error variance estimation to be introduced in §3.2.

Consider the properties of the e_i^2 as estimators. Under homoskedasticity, it is clear from (1.11) that the e_i^2 are biased estimators of ω, and that the bias is given by,

$$\mathrm{Bias}(e_i^2) = \omega(1 - h_{ii}) - \omega = -h_{ii}\omega. \tag{3.25}$$

Thus, under homoskedasticity the squared residuals have strictly negative biases; that is, they tend to *underestimate* ω. Provided that A1-A5 hold, the variance of the estimator is $2\omega^2 m_{ii}^2$ (see (C.1)). From this, it follows that the mean squared error of the estimator is as follows:

$$\begin{aligned}
\mathrm{MSE}(e_i^2) &= \mathrm{Var}(e_i^2) + \left[\mathrm{Bias}(e_i^2)\right]^2 \\
&= 2\omega^2(1 - h_{ii})^2 + (-h_{ii}\omega)^2 \\
&= \omega^2\left(2 - 4h_{ii} + 3h_{ii}^2\right) &(3.26) \\
&= \omega^2\left(3m_{ii}^2 - 2m_{ii} + 1\right). &(3.27)
\end{aligned}$$

The derivative of the Mean Squared Error (MSE) with respect to h_{ii} is given by,

[49] More specifically, the scale factor would be $\det(\boldsymbol{J}) = \mathrm{tr}(\boldsymbol{J}) = 2^{-n} \prod_{i=1}^{n} [\mathrm{Var}(e_i)]^{-1}$, since the transformation $(e_1^2, e_2^2, \ldots, e_n^2) = (2\,\mathrm{Var}(e_1)u_1, 2\,\mathrm{Var}(e_2)u_2, \ldots, 2\,\mathrm{Var}(e_n)u_n)$, where $U_i \sim \mathrm{Gamma}(\alpha = 1/2, \beta = 1)$, has a diagonal Jacobian.

[50] The same problem occurs under both homoskedasticity and heteroskedasticity, since the covariance matrices (1.10) and (1.14) are both singular.

$$\frac{\partial}{\partial h_{ii}}\text{MSE}(e_i^2) = \omega^2\left(-4+6h_{ii}\right). \tag{3.28}$$

It is evident that the MSE of the estimator is minimised when $h_{ii} = \frac{2}{3}$. Since $0 \leq h_{ii} \leq 1$, it also follows from evaluating (3.26) at 0 and 1 that the MSE is maximised when $h_{ii} = 0$. Since the derivative is negative for all $0 \leq h_{ii} < \frac{2}{3}$, the performance of a particular e_i^2 as an estimator of the homoskedastic variance ω improves as the leverage of the ith design point increases, up to the value 2/3 (which is an extremely large leverage value that will seldom be achieved by any observation in practice). This is illustrated in Figure 3.2.

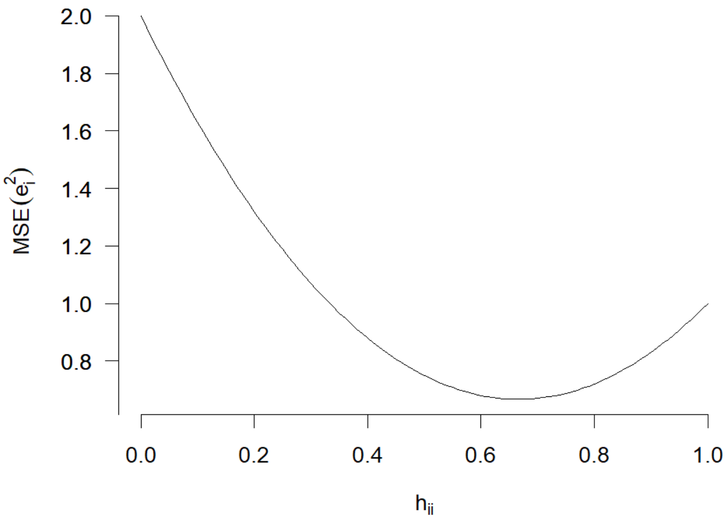


Figure 3.2: $\text{MSE}(e_i^2)$ under homoskedasticity plotted as a function of h_{ii} for $\omega = 1$

Under heteroskedasticity, it follows from (1.15) that the bias of the variance estimator e_i^2 is given by,

$$\begin{aligned}\text{Bias}(e_i^2) &= \sum_{k=1}^{n}\omega_k m_{ik}^2 - \omega_i \\ &= \omega_i(1-h_{ii})^2 + \sum_{k\neq i}\omega_k h_{ik}^2 - \omega_i \qquad (3.29)\\ &= -\omega_i h_{ii}(2-h_{ii}) + \sum_{k\neq i}\omega_k h_{ik}^2 \\ &= -2h_{ii}\omega_i + \sum_{k=1}^{n}\omega_k h_{ik}^2. \qquad (3.30)\end{aligned}$$

Unlike under homoskedasticity, the direction of the bias is ambiguous and depends on the relative magnitudes of the error variances and leverage scores. The partial derivatives of (3.30) with respect to ω_i and h_{ii} are given by

$$\frac{\partial}{\partial \omega_i}\text{Bias}(e_i^2) = -h_{ii}(2-h_{ii}) \tag{3.31}$$

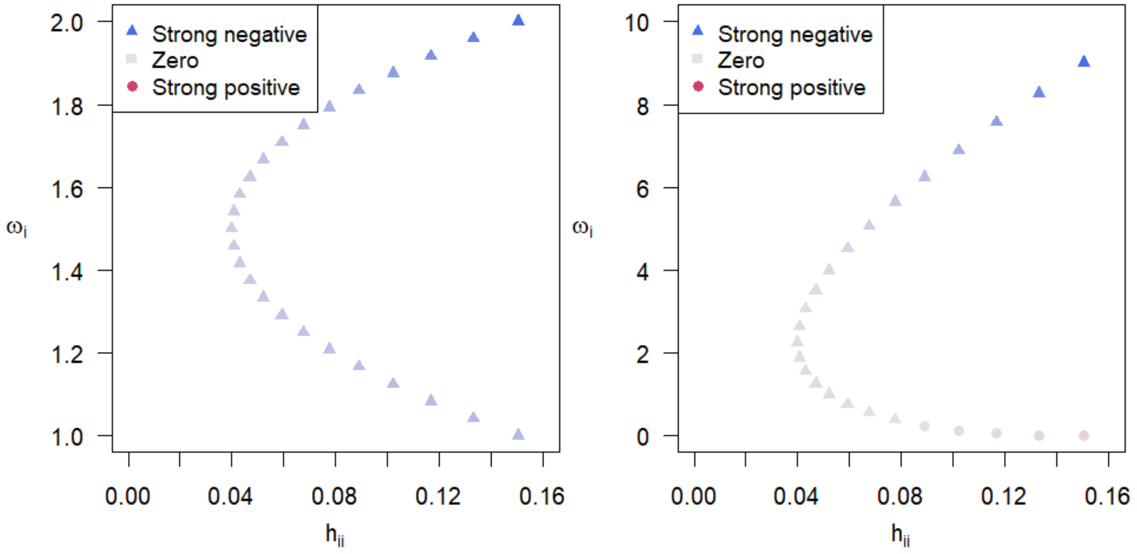
and

$$\frac{\partial}{\partial h_{ii}}\text{Bias}(e_i^2) = -2\omega_i(1-h_{ii}). \tag{3.32}$$

Equations (3.31) and (3.32) are both strictly negative; thus the bias strictly decreases with ω_i when h_{ii} is held constant, and with h_{ii} when ω_i is held constant. Using the quadratic formula to find roots of (3.29) with respect to h_{ii}, it is also evident that the bias is negative for any observation(s) with leverage $h_{ii} > 1 - \sqrt{1 - \sum_{k \neq i} \frac{\omega_k}{\omega_i} h_{ik}^2}$. Positive biases tend to occur for observations that have very small error variances (relative to those of other observations), especially if they also have very small leverage scores.[51]

It was previously observed in (1.41) that the leverage h_{ii} increases with the squared standardised (Mahalanobis) distance of the ith design point from the centre of the design points. If the error variances ω_i are related to the design points $\boldsymbol{X}_{i\cdot}'$ by a function that—like the Mahalanobis distance—takes on small values for 'central' design points and large values for 'outlying' design points, then the h_{ii} and ω_i will have a strong positive correlation, with the result that the bias is positive for 'central' design points and negative for 'outlying' design points.

Figure 3.3 illustrates the relationship between leverage h_{ii}, error variance ω_i, and bias, in four different scenarios involving linear regression with a single covariate $\boldsymbol{x}$. $\omega_i = g(x_i)$ here represents the heteroskedastic function. The upper left frame (a) shows that, with mild heteroskedasticity (as with homoskedasticity), all squared residuals are negatively biased. The upper right frame (b) shows that, with uniformly distributed data[52] and more severe heteroskedasticity, some squared residuals can have a slightly positive bias. The bottom left frame (c) shows that, if there is one outlier having a very large error variance, the corresponding squared residual is negatively biased but all others have a slight positive bias. The bottom right frame (d) shows that, if there is one outlier having a very small error variance (relative to other error variances), the corresponding squared residual may have a moderate positive bias while all others are negatively biased.

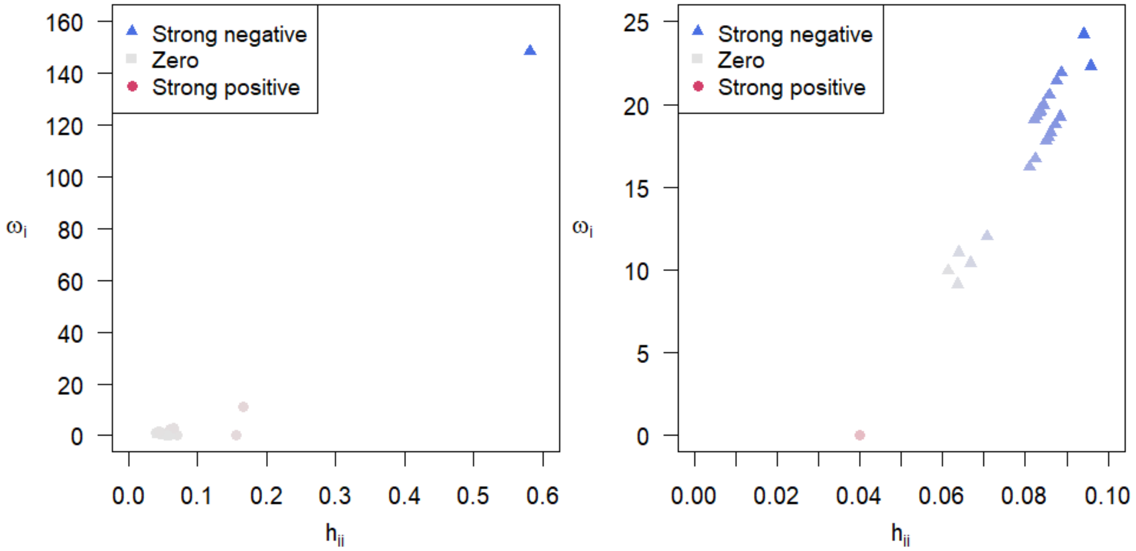


(a) $g(x) = 1 + x/3$, $x \sim \mathrm{U}(0, 3)$

(b) $g(x) = x^2$, $x \sim \mathrm{U}(0, 3)$

[51] Bear in mind that, in models with an intercept, all $h_{ii} \geq \frac{1}{n}$.

[52] To be more precise, the uniformly distributed data in frames (a) and (b) of Figure 3.3 was a discrete sequence of equally spaced values on the interval $[0, 3]$.



(c) $g(x) = e^x$, $x \sim \mathrm{N}(0, 1)$ with outlier $x_i = 5$

(d) $g(x) = x^2$, $x \sim \mathrm{U}(-5, -3) \cup \mathrm{U}(3, 5)$ with outlier $x_i = 10^{-3}$

Figure 3.3: Bias(e_i^2) under Four Scenarios

A brief further discussion of the properties of the MSE of the squared OLS residuals, taken as estimators of the error variances ω_i, is provided in Appendix C.1.2.

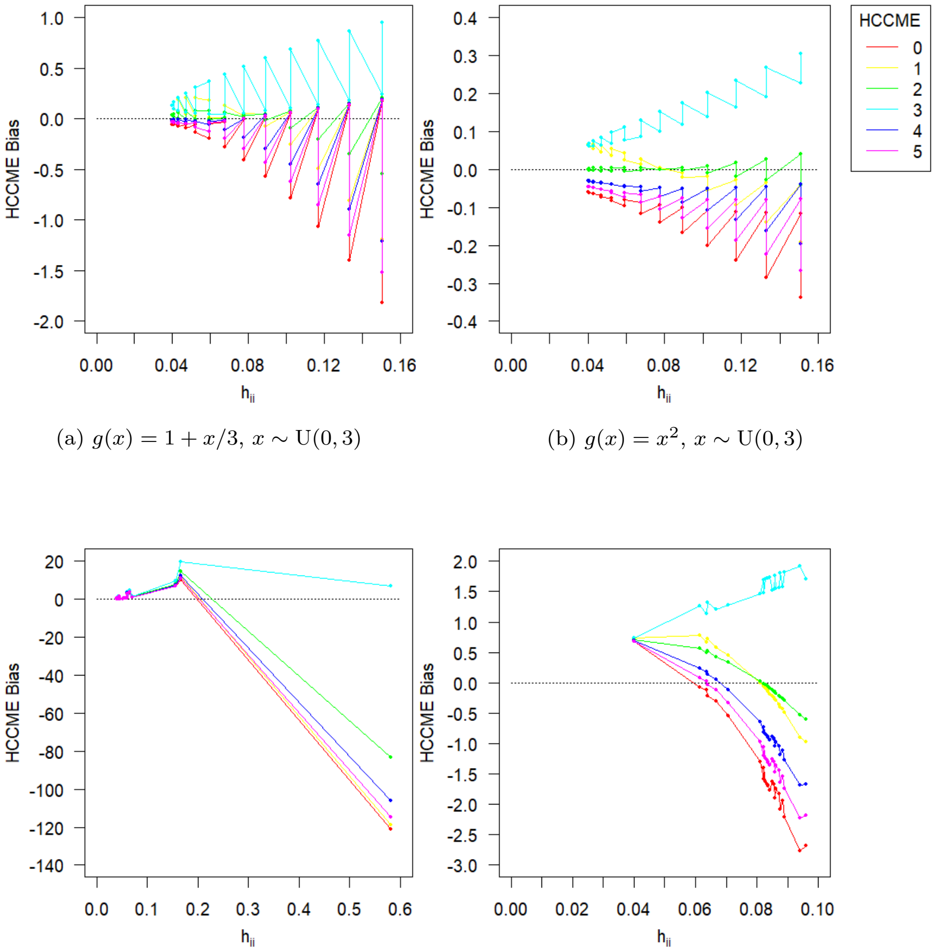
All of the HCCMEs discussed in §2.3 entail estimating each error variance ω_i by multiplying the corresponding OLS squared residual e_i^2 by some constant factor c_i;[53] in most cases this factor is of the form $c_i = (1 - h_{ii})^{-\delta_i}$, where $\delta_i \geq 1$.[54] Thus, most of the HCCMEs have the effect of *inflating* the e_i^2 by multiplying them by factors larger than 1. Consequently, the biases of negatively biased squared residuals may shrink toward 0, but the biases of positively biased squared residuals *strictly increase*.

Figure 3.4 shows the effect on squared residual biases of six HCCMEs for each of the examples illustrated in Figure 3.3.[55] It is evident that none of the HCCMEs are really successful in eliminating the bias across different leverage (or error variance) values over all of these scenarios.

[53] The factor is stochastic in the case of HC6, but this HCCME can be disregarded due to its poor properties, still to be discussed.

[54] Only in HC5 and HC7 can δ_i in principle take on values less than 1.

[55] HC4m and HC5m are not shown due to being identical to HC2 in all four cases. HC6 is not shown because the magnitude of its bias is enormous relative to the other HCCMEs. HC7 is not shown due to being identical to HC4 in all four cases.



(a) $g(x) = 1 + x/3$, $x \sim \mathrm{U}(0, 3)$

(b) $g(x) = x^2$, $x \sim \mathrm{U}(0, 3)$

(c) $g(x) = e^x$, $x \sim \mathrm{N}(0, 1)$ with outlier $x_i = 5$

(d) $g(x) = x^2$, $x \sim \mathrm{U}(-5, -3) \cup \mathrm{U}(3, 5)$ with outlier $x_i = 10^{-3}$

Figure 3.4: Bias of HCCMEs under Four Scenarios

3.2 New Auxiliary Regression Models for Estimating Error Variances in Heteroskedastic Linear Regression Models

3.2.1 Model Motivation and Description

In §2.3, various HCCMEs were introduced, most of them constructed by multiplying the squared OLS residuals e_i^2 by some bias correction factor c_i. In §2.2.1, several FWLS techniques were introduced that use the squared OLS residual vector $\boldsymbol{e} \circ \boldsymbol{e}$ as the response in an auxiliary regression model (Davidson and MacKinnon 2004, Miller and Startz 2019, e.g.,). Both of these broad approaches to handling heteroskedasticity in linear regression rest on a common premise: that the squared OLS residuals $\boldsymbol{e} \circ \boldsymbol{e}$ can be used as proxies for the unknown error variances $\boldsymbol{\omega}$. However, this premise is intrinscally problematic. In fact, there are two problems with it. One is that e_i^2 is a biased estimator of ω_i. Indeed, as has just been highlighted in §3.1.4, the bias correction factors used in the HCCMEs may in fact *worsen* the bias in certain instances. A second problem is that the expectation of e_i^2 depends not only on ω_i but on all the other $\omega_j, j = 1, 2, \ldots, n, j \neq i$ (see (1.15)).

The theoretical results on $\boldsymbol{e} \circ \boldsymbol{e}$ given in §3.1.1 suggest an alternative approach: an auxiliary regression model *that makes use of the true relationship* between the moments of $\boldsymbol{e} \circ \boldsymbol{e}$ and $\boldsymbol{\omega}$, as given in (3.10) and (3.11), under assumptions A1 and A3-A5. Such an auxiliary regression model, unlike those in §2.2.1, has its conditional mean function correctly specified, by definition. It is a regression model with an n-vector of observed responses $\boldsymbol{e} \circ \boldsymbol{e}$, an $n \times n$ design matrix $\boldsymbol{M} \circ \boldsymbol{M}$, and an n-vector of unknown parameters, $\boldsymbol{\omega}$. (The obvious problem that the number of parameters equals the number of observations will be attended to shortly).

Consider, therefore, the following auxiliary regression model:

$$e_i^2 = \sum_{k=1}^{n} \omega_k m_{ik}^2 + u_i, i = 1, 2, \ldots, n, \tag{3.33}$$

or, equivalently,

$$\boldsymbol{e} \circ \boldsymbol{e} = (\boldsymbol{M} \circ \boldsymbol{M})\, \boldsymbol{\omega} + \boldsymbol{u}, \boldsymbol{\omega} \succ \boldsymbol{0}, \tag{3.34}$$

where $\boldsymbol{u}$ is a random error satisfying $\mathrm{E}(\boldsymbol{u}) = \boldsymbol{0}$ and $\mathrm{Cov}(\boldsymbol{u})$ as given in (3.11),[56] and where $\succ$ in the restriction on $\boldsymbol{\omega}$ denotes $>$ applied elementwise. This restriction is necessary because $\boldsymbol{\omega}$ is a vector of variance parameters. It must also be assumed that the distribution of $\boldsymbol{u}$ is such that each element of the event $\boldsymbol{u} \prec -(\boldsymbol{M} \circ \boldsymbol{M})\boldsymbol{\omega}$ has zero probability, since the response vector $\boldsymbol{e} \circ \boldsymbol{e}$ is elementwise nonnegative.[57]

It may be asked whether the right side of (3.34) qualifies as linear in the parameters, given that $\boldsymbol{u}$ depends on $\boldsymbol{\omega}$ through the restriction $\boldsymbol{u} \succeq -(\boldsymbol{M} \circ \boldsymbol{M})\boldsymbol{\omega}$. To say that $f(\boldsymbol{\omega}) = (\boldsymbol{M} \circ \boldsymbol{M})\boldsymbol{\omega} + \boldsymbol{u}$ is linear in the parameters is equivalent, in algebraic terms, to saying that it is an affine function, i.e.,

$$f\left(c\boldsymbol{\omega}_1 + (1-c)\boldsymbol{\omega}_2\right) = cf(\boldsymbol{\omega}_1) + (1-c)f(\boldsymbol{\omega}_2) \text{ for all } \boldsymbol{\omega}_1, \boldsymbol{\omega}_2 \in \mathbb{R}^n \text{ and all } c \in \mathbb{R}.$$

Now, it is easy to show that $f(\boldsymbol{\omega}) = (\boldsymbol{M} \circ \boldsymbol{M})\boldsymbol{\omega} + \boldsymbol{u}$ is an affine function. Moreover, since the restriction on $\boldsymbol{u}$ can be written as $f(\boldsymbol{\omega}) \succeq \boldsymbol{0}$, it is clear that the restriction is also affine since both sides of the inequality are affine functions with respect to $\boldsymbol{\omega}$. The same is true of both sides of the inequality $\boldsymbol{\omega} \succ \boldsymbol{0}$. Consequently, the restrictions on the model in (3.34) are linearity-respecting.

Observe that in this model the conditional mean $(\boldsymbol{M} \circ \boldsymbol{M})\boldsymbol{\omega}$ is correctly specified (see (3.10)), while the response variables e_i^2 and explanatory variables m_{ik}^2 can be computed from the observed data ($\boldsymbol{y}$ and $\boldsymbol{X}$). By fitting the model one obtains an estimate of $\boldsymbol{\omega}$ or, equivalently, of $\boldsymbol{\Omega}$.

It is important to note that, because $\mathrm{tr}(\boldsymbol{M}) = n - p$, the matrix $\boldsymbol{M} \circ \boldsymbol{M}$ becomes less important as the sample size n increases relative to p and in the absence of high-leverage points. For $n \gg p$, $\boldsymbol{M} \circ \boldsymbol{M}$ resembles the identity matrix. For instance, in a simulated example with $n = 10^4$ where $\boldsymbol{X}$ consists of a column of ones and three columns of independent $U(0,1)$ random variables, the diagonal elements of $\boldsymbol{M} \circ \boldsymbol{M}$ all ranged betwen 0.998 and 0.9998, while the off-diagonal elements ranged between 0 and 10^{-6}. In this kind of scenario, the model equation reduces to $\boldsymbol{e} \circ \boldsymbol{e} \approx \boldsymbol{\omega} + \boldsymbol{u}$, and using $\boldsymbol{e} \circ \boldsymbol{e}$ as a proxy for $\boldsymbol{\omega}$—as the FWLS methods described in §2.2.1 do—is quite reasonable. Hence, the degree of improvement of the new modelling approach over other FWLS methods is posited to be meaningful only if there are high-leverage points and/or n is not too large relative to p.

An alternative way of specifying the auxiliary regression model is to take a log transform of the response e_i^2 in (3.34). In this case, the conditional mean and covariance in the model are no longer known exactly in terms of $\boldsymbol{\Omega}$, but can be expressed in terms of $\boldsymbol{\Omega}$ up to a second-order Taylor series approximation. This method is discussed further in Appendix C.2, but is not pursued further in the body of this thesis.

Fitting of the regression model (3.34) leads immediately to estimates of the error variances $\boldsymbol{\omega}$. The model is estimated by solving

$$\begin{gathered} \arg\min_{\boldsymbol{\omega}} \left|\left|\boldsymbol{e} \circ \boldsymbol{e} - (\boldsymbol{M} \circ \boldsymbol{M})\, \boldsymbol{\omega}\right|\right|_2^2, \\ \text{subject to } \boldsymbol{\omega} \succeq \boldsymbol{0}. \end{gathered} \tag{3.35}$$

In practice, the right side of the constraint $\boldsymbol{\omega} \succeq \boldsymbol{0}$ is set to a vector consisting of a very small positive real number, which can be denoted 0^+. The reason for this is computational: if any $\hat{\omega}_i$ is numerically too close to zero, the weights in the FWLS estimator will not be computationally finite and the estimator then cannot

[56] If the normality assumption A5 is replaced with the weaker assumptions A5′ and A6′, the model remains the same except that $\mathrm{Cov}(\boldsymbol{u})$ is now given by (3.18) and is thus dependent on one additional scalar kurtosis parameter.

[57] Under A1, A3-A5, $\boldsymbol{u}$ would have a multivariate Gamma distribution location-shifted by $(\boldsymbol{M} \circ \boldsymbol{M})\boldsymbol{\omega}$ (see §3.1.3), and would thus satisfy this assumption by definition, due to the support of each variable being limited to the positive domain.

be computed. Empirical work suggests that $0^+ = 10^{-10}$ works well in practice, being large enough to avoid singularities in FWLS computation but small enough not to meaningfully alter the estimates of elements of $\boldsymbol{\omega}$ that lie off the constraint boundary.

An appropriate name for the model in (3.34) is Auxiliary Linear Variance Model (ALVM): 'auxiliary,' because it supplements the original linear regression model (1.1); 'linear,' because it is linear in its parameters, $\boldsymbol{\omega}$; 'variance,' because the purpose of fitting the model is to estimate the error variances, $\mathrm{Var}(\epsilon_i) = \omega_i, i = 1, 2, \ldots, n$.

The ALVM in (3.34) can indeed be fitted (see §3.3.1 for details). An obvious difficulty, however, is that the dimensionality of the parameter vector (n) equals the number of observations. It can hardly be expected that such a model will yield precise variance estimates. Therefore, for the model in (3.34) to be practicable, the parameter dimensionality must be reduced. The ALVM fitted exactly as in (3.34), without parameter reduction, can be referred to as the 'basic' or 'naïve' ALVM.

A natural way of reducing the parameter dimensionality of the auxiliary regression model is by assuming that, just as $\mathrm{E}(\boldsymbol{y})$ is a function of the covariate matrix $\boldsymbol{X}$ in the original linear regression model, so $\mathrm{Var}(y_i) = \mathrm{Var}(\epsilon_i) = \omega_i$ is a function of some covariate vector $\boldsymbol{Z}_{i\cdot}$.

Specifically, assume that,

$$\omega_i = g(\boldsymbol{Z}_{i\cdot}; \boldsymbol{\gamma}), i = 1, 2, \ldots, n,$$

where $\boldsymbol{Z}_{i\cdot}$ is the ith row of an $n \times p'$ predictor matrix $\boldsymbol{Z}$, $\boldsymbol{\gamma}$ is a q-vector of unknown parameters ($q < n$), and $g : \mathbb{R} \to \mathbb{R}^+$ is a continuous, differentiable function. In the absence of prior information on the covariates to include in $\boldsymbol{Z}$, one could either set $\boldsymbol{Z} = \boldsymbol{X}$ or use some feature selection routine to select a subset of columns of $\boldsymbol{X}$ to include in $\boldsymbol{Z}$.

How should the heteroskedastic function $g(\cdot)$ be chosen? Three approaches will be explored herein. The first is simply to specify the form of $g(\cdot)$ explicitly by assumption. The second is to estimate the form of $g(\cdot)$ within certain boundaries—that is, making a weaker assumption about its form. The third is to use a clustering procedure that obviates specification or estimation of $g(\cdot)$. This approach requires the weakest assumptions and is nonparametric in the sense that a parametric form of $g(\cdot)$ need not be specified.

3.2.2 Explicit Specification of the Heteroskedastic Function

Let $g(\boldsymbol{Z}_{k\cdot}; \boldsymbol{\gamma})$ be some known function (continuous, differentiable, and positive real-valued, as indicated above). Then, the auxiliary regression model (3.34) can be rewritten as,

$$e_i^2 = \sum_{k=1}^{n} g(\boldsymbol{Z}_{k\cdot}; \boldsymbol{\gamma}) m_{ik}^2 + u_i, i = 1, 2, \ldots, n,$$

or, equivalently,

$$\boldsymbol{e} \circ \boldsymbol{e} = (\boldsymbol{M} \circ \boldsymbol{M})\, \boldsymbol{g}(\boldsymbol{Z}; \boldsymbol{\gamma}) + \boldsymbol{u} = \boldsymbol{f}(\boldsymbol{\Xi}; \boldsymbol{\gamma}) + \boldsymbol{u}, \tag{3.36}$$

where $\boldsymbol{g}(\boldsymbol{Z}; \boldsymbol{\gamma}) = [g(\boldsymbol{Z}_1; \boldsymbol{\gamma}), g(\boldsymbol{Z}_2; \boldsymbol{\gamma}), \ldots, g(\boldsymbol{Z}_n; \boldsymbol{\gamma})]$, and $\boldsymbol{\Xi}$ is the union of columns of $\boldsymbol{X}$ and $\boldsymbol{Z}$ (since any columns of $\boldsymbol{X}$ not in $\boldsymbol{Z}$ will still influence the model through $\boldsymbol{M}$). By replacing the n-vector of parameters $\boldsymbol{\omega}$ with the q-vector of parameters $\boldsymbol{\gamma}$, the dimensionality of the parameter space has been reduced from n to q.

Natural choices of $g(\cdot)$ might include the following:

$$g(\boldsymbol{Z}_{k\cdot}; \boldsymbol{\gamma}) = \boldsymbol{Z}'_{k\cdot}\boldsymbol{\gamma}, \tag{3.37}$$

$$g(\boldsymbol{Z}_{k\cdot}; \boldsymbol{\gamma}) = \left(\boldsymbol{Z}'_{k\cdot}\boldsymbol{\gamma}\right)^2, \tag{3.38}$$

or

$$g(\boldsymbol{Z}_{k\cdot}; \boldsymbol{\gamma}) = \exp\left\{\boldsymbol{Z}'_{k\cdot}\boldsymbol{\gamma}\right\}. \tag{3.39}$$

In the case (3.37), the auxiliary regression model equation can be written as,

$$\boldsymbol{e} \circ \boldsymbol{e} = (\boldsymbol{M} \circ \boldsymbol{M})\boldsymbol{Z}\boldsymbol{\gamma} + \boldsymbol{u}. \tag{3.40}$$

The model (3.40) is still linear in its parameters, $\boldsymbol{\gamma}$ (a $q = p'$-vector). It is, therefore, still an ALVM. It will be termed the *linear* ALVM, with the prefixed 'linear' referring not to its linearity in the parameters $\boldsymbol{\gamma}$ (since all ALVMs are linear in this sense) but to its linearity in the auxiliary covariate matrix $\boldsymbol{Z}$.

The estimation problem, for the linear ALVM, becomes

$$\begin{aligned} &\arg\min_{\boldsymbol{\gamma}} \left|\left|\boldsymbol{e}\circ\boldsymbol{e} - \left(\boldsymbol{M}\circ\boldsymbol{M}\right)\boldsymbol{Z}\boldsymbol{\gamma}\right|\right|_2^2 , \\ &\text{subject to } \boldsymbol{Z}\boldsymbol{\gamma} \succeq \mathbf{0}^+ . \end{aligned} \tag{3.41}$$

Specifying (3.38) or (3.39) as the heteroskedastic function results in a model that is not linear in its parameter vector, $\boldsymbol{\gamma}$. Therefore, in these cases the model is not an ALVM but an Auxiliary Nonlinear Variance Model (ANLVM), and will require a different estimation method, as discussed in §3.3.1.

3.2.3 Estimating the Heteroskedastic Function

A second approach to the problem of choosing the heteroskedastic function $g(\cdot)$ is to estimate it. This still requires some assumptions about the form of $g(\cdot)$, but the assumptions are weaker than under the previous approach described in §3.2.2. This reduces the risk of poor model performance resulting from misspecification of $g(\cdot)$.

Two broad approaches to estimating $g(\cdot)$ are discussed here. The first is to use a polynomial function that is penalised in a way analogous to Least Absolute Shrinkage and Selection Operator (LASSO) regression or Ridge Regression (RR). The second is to use a spline function. Both approaches result in a model that is linear in its parameters; that is, an ALVM. Thus, a general form of the ALVM—applicable to the linear ALVM discussed previously, the penalised polynomial and spline ALVMs discussed in this section, and the clustering ALVM discussed below in §3.2.4—is as follows:

$$\boldsymbol{e}\circ\boldsymbol{e} = (\boldsymbol{M}\circ\boldsymbol{M})\boldsymbol{L}\boldsymbol{\gamma} + \boldsymbol{u} = \boldsymbol{D}\boldsymbol{\gamma} + \boldsymbol{u}, \tag{3.42}$$

where $\boldsymbol{L}$ is a known $n \times q$ linear predictor matrix that maps the parameters $\boldsymbol{\gamma}$ onto $\boldsymbol{\omega}$, and $\boldsymbol{D} = (\boldsymbol{M}\circ\boldsymbol{M})\boldsymbol{L}$ is an $n \times q$ auxiliary design matrix that maps the parameters $\boldsymbol{\gamma}$ onto the mean response $\mathrm{E}(\boldsymbol{e}\circ\boldsymbol{e})$.

3.2.3.1 Penalised Polynomial ALVMs

It was assumed previously that $g(\boldsymbol{Z}_{k\cdot};\boldsymbol{\gamma})$ is continuous and differentiable. If this assumption is strengthened to assert that $g(\boldsymbol{Z}_{k\cdot};\boldsymbol{\gamma})$ is d-times differentiable at the point $\boldsymbol{z} = \boldsymbol{a}$, then it follows from the Taylor series expansion about $\boldsymbol{Z}_{k\cdot} = \boldsymbol{a}$ that, in the neighbourhood of point $\boldsymbol{a}$,

$$g(\boldsymbol{Z}_{k\cdot};\boldsymbol{\gamma}) \approx P^{(d)}(\boldsymbol{Z}_{k\cdot};\boldsymbol{\gamma}) = \sum_{|\boldsymbol{\alpha}|\leq d} \frac{D^{\boldsymbol{\alpha}}P(\boldsymbol{a})}{\boldsymbol{\alpha}!}\left(\boldsymbol{Z}_{k\cdot} - \boldsymbol{a}\right)^{\boldsymbol{\alpha}} , \tag{3.43}$$

where

$$\begin{aligned} \boldsymbol{\alpha}! &= \alpha_1!\alpha_2!\cdots\alpha_{p'}!, \\ \left(\boldsymbol{Z}_{k\cdot} - \boldsymbol{a}\right)^{\boldsymbol{\alpha}} &= \left(Z_{k1\cdot} - a_1\right)^{\alpha_1}\left(Z_{k2\cdot} - a_2\right)^{\alpha_2}\cdots\left(Z_{ks\cdot} - a_{p'}\right)^{\alpha_{p'}}, \\ D^{\boldsymbol{\alpha}}P(\boldsymbol{a}) &= \frac{\partial^{|\boldsymbol{\alpha}|}P(\boldsymbol{a})}{\partial Z_{k1\cdot}^{\alpha_1}\partial Z_{k2\cdot}^{\alpha_2}\cdots\partial Z_{kp'\cdot}^{\alpha_{p'}}}, \end{aligned}$$

and

$$|\boldsymbol{\alpha}| \leq d \text{ denotes } \left\{(\alpha_1,\alpha_2,\ldots,\alpha_{p'}) : \alpha_1+\alpha_2+\cdots+\alpha_{p'} \leq d\right\}.$$

Observe—setting $\boldsymbol{a} = \mathbf{0}$ makes it easier to see—that $P^{(d)}(\boldsymbol{Z}_{k\cdot})$ is a $(p'-1)$-variate polynomial function of degree d (p'-variate if $\boldsymbol{Z}_1$ is not a vector of ones). One can therefore reparametrise $P^{(d)}(\cdot)$ as follows:

$$P^{(d)}(\boldsymbol{Z}_{k\cdot};\boldsymbol{\gamma}) = \sum_{j_2=0}^{d}\sum_{j_3=0}^{d-j_1}\cdots\sum_{j_{p'}=0}^{d-j_1-j_2-\cdots-j_{p'-1}} \gamma_{j_1,j_2,\ldots,j_{p'}} Z_{k2\cdot}^{j_2} Z_{k3\cdot}^{j_3}\cdots Z_{kp'\cdot}^{j_{p'}}, \tag{3.44}$$

or, alternatively,

$$P^{(d)}(\boldsymbol{Z}_{k\cdot};\boldsymbol{\gamma}) = \boldsymbol{Z}^{(d)}\boldsymbol{\gamma}, \tag{3.45}$$

where $\boldsymbol{Z}^{(d)}$ is the $n \times q$ matrix whose elements in the kth row are all terms in the expansion (3.44) and, in this instance, $q = \binom{p'-1+d}{d}$.[58]

An advantage of approximating $g(\cdot)$ with $P^{(d)}(\boldsymbol{Z}_{k\cdot};\boldsymbol{\gamma})$ is that it is linear in the parameters. However, one disadvantage is that Taylor series theory only guarantees the approximation in the neighbourhood of point $\boldsymbol{a}$, so (3.44) entails the risky further assumption that the polynomial coefficients are fixed for different values of $\boldsymbol{a}$. A second disadvantage is that q, the dimensionality of $\boldsymbol{\gamma}$, increases rapidly with d (which must also be specified).

This second disadvantage can be mitigated by including a penalty term in the model. This penalty term would penalise against terms of excessive degree but also against overspecification of covariates in $\boldsymbol{Z}$. Consider first an L_2-norm penalty, as used in RR. The linear predictor matrix (per (3.42)) is $\boldsymbol{L} = \boldsymbol{Z}^{(d)}$ and the estimation problem (3.35) becomes

$$\begin{aligned}&\underset{\boldsymbol{\gamma}}{\arg\min} \left|\left|\boldsymbol{e} \circ \boldsymbol{e} - (\boldsymbol{M} \circ \boldsymbol{M})\, \boldsymbol{Z}^{(d)}\boldsymbol{\gamma}\right|\right|_2^2 + \lambda\boldsymbol{\gamma}'\boldsymbol{P}\boldsymbol{\gamma},\\ &\text{subject to } \boldsymbol{Z}^{(d)}\boldsymbol{\gamma} \succeq \boldsymbol{0}^+,\end{aligned} \tag{3.46}$$

where $\lambda \geq 0$ is a penalty parameter and $\boldsymbol{P}$ is a penalty matrix; in this case $\boldsymbol{P}$ is a $q \times q$ identity matrix modified so that $P_{11} = 0$ if the model includes an intercept (so that the intercept is not penalised). The minimisation problem (3.46) is an instance of Inequality-Constrained Ridge Regression (ICRR) (Toker et al. 2013). This model will be referred to as the L_2-norm penalised polynomial ALVM, or the polynomial RR ALVM.

Alternatively, an L_1-norm penalty can be used, as in LASSO regression. The sparsity properties of the LASSO model tend to result in some coefficients being shrunk to 0, making it useful for feature selection (Tibshirani 1996). In this case, the penalty term in (3.46) changes so that the problem becomes,

$$\begin{aligned}&\underset{\boldsymbol{\gamma}}{\arg\min} \left|\left|\boldsymbol{e} \circ \boldsymbol{e} - (\boldsymbol{M} \circ \boldsymbol{M})\, \boldsymbol{Z}^{(d)}\boldsymbol{\gamma}\right|\right|_2^2 + \lambda\boldsymbol{p}\left|\left|\boldsymbol{\gamma}\right|\right|_1,\\ &\text{subject to } \boldsymbol{Z}^{(d)}\boldsymbol{\gamma} \succeq \boldsymbol{0}^+.\end{aligned} \tag{3.47}$$

Here, $\boldsymbol{p}$ is a q-vector of ones, with the first element $p_1 = 0$ if the model includes an intercept (again, to avoid penalising the intercept). This model will be referred to as the L_1-norm penalised polynomial ALVM, or the polynomial LASSO ALVM.

3.2.3.2 Regression Spline ALVMs

Another approach to estimating $g(\cdot)$ is to use regression splines, which are piecewise functions typically stitched together at particular knots with a certain degree of smoothness. For simplicity, consider first the case where $\boldsymbol{Z} = [\boldsymbol{1}_n \;\; \boldsymbol{z}]$, the n-vector $\boldsymbol{z}$ being a single covariate. Let

$$g(z;\boldsymbol{\gamma}) = \sum_{\ell=1}^{q} b_\ell(z)\gamma_\ell, \tag{3.48}$$

where $b_\ell(z), \ell = 1, 2, \ldots, q$, are basis functions for $g(z)$. (3.36) now becomes

$$\boldsymbol{e} \circ \boldsymbol{e} = (\boldsymbol{M} \circ \boldsymbol{M})\, \boldsymbol{B}\boldsymbol{\gamma} + \boldsymbol{u}, \tag{3.49}$$

where $\boldsymbol{\gamma}$ is a q-vector of unknown parameters, and $\boldsymbol{B}(= \boldsymbol{L})$ is an $n \times q$ basis matrix with k, ℓth element $b_\ell(z_k)$. Perperoglou et al. (2019) gives the recursive formulas used to calculate the basis functions.

Model (3.49) may be termed the B-spline ALVM. Its estimation problem is

$$\begin{aligned}&\underset{\boldsymbol{\lambda}}{\arg\min} \left|\left|\boldsymbol{e} \circ \boldsymbol{e} - (\boldsymbol{M} \circ \boldsymbol{M})\, \boldsymbol{B}\boldsymbol{\gamma}\right|\right|_2^2,\\ &\text{subject to } \boldsymbol{B}\boldsymbol{\gamma} \succeq \boldsymbol{0}^+.\end{aligned} \tag{3.50}$$

[58] Observe that $P^{(d)}(\boldsymbol{Z}_{k\cdot};\boldsymbol{\gamma})$ is equivalent to (3.38) if $d = 2$, although parametrised differently.

Wood (2017) notes that cubic splines are most commonly used in practice, i.e. where each piece of the spline is a degree 3 polynomial. The critical settings for a B-spline are the number and locations of the knots. In a cubic spline with intercept, the degrees of freedom q is equal to $k + 3 + 1$, where k is the number of knots. B-spline fitting procedures, such as the `bs` function in the R package **splines** (R Core Team 2022), will by default set the knot locations at suitable quantiles of the predictor variable.

A second univariate regression spline technique is the smoothing spline (Wood 2017, Perperoglou et al. 2019). In this model, the knot locations are the n ordered observations of $\boldsymbol{z}$; this includes two boundary knots and so the parameter dimensionality is $n + 2$ for a cubic smoothing spline. An ordinary cubic spline with the observations as knots would completely interpolate the data and be grossly overfitted. With a smoothing spline, however, the objective function includes a smoothness penalty imposed on the squared second derivative of the spline. This is captured in a $q \times q$ symmetric penalty matrix $\boldsymbol{P}$ and the estimation problem becomes

$$\begin{aligned} &\arg\min_{\boldsymbol{\lambda}} ||\boldsymbol{e} \circ \boldsymbol{e} - (\boldsymbol{M} \circ \boldsymbol{M})\, \boldsymbol{B}\boldsymbol{\gamma}||_2^2 + \lambda\boldsymbol{\gamma}'\boldsymbol{P}\boldsymbol{\gamma}, \\ &\text{subject to } \boldsymbol{B}\boldsymbol{\gamma} \succeq \boldsymbol{0}^+, \end{aligned} \tag{3.51}$$

where $\lambda \geq 0$ is a hyperparameter controlling the intensity of the smoothness penalty. The basis matrix $\boldsymbol{B}$ is again the linear predictor matrix $\boldsymbol{L}$, with reference to (3.42). Smoothing splines can be fit in R using the `smooth.spline` function of the **stats** package (R Core Team 2022).

The B-spline and smoothing spline ALVMs described above can only be used with a single covariate $\boldsymbol{z}$. There are other spline methods, however, such as thin-plate splines and tensor product smooths, that can be used in a multi-dimensional setting (Wood 2017). Only the thin-plate spline is considered in this research project. In general, a thin-plate spline seeks to estimate g from n observations $(y_k, \boldsymbol{X}_{k\cdot}), k = 1, 2, \ldots, n$, by finding

$$\arg\min_{g} ||\boldsymbol{y} - \boldsymbol{g}||_2^2 + \lambda J_{m,p'}(g), \tag{3.52}$$

where $\boldsymbol{g} = [g(\boldsymbol{X}_{1\cdot}), g(\boldsymbol{X}_{2\cdot}), \ldots, g(\boldsymbol{X}_{n\cdot})]'$ and $J_{m,p'}(g)$ is a 'wiggliness' penalty defined as

$$J_{md} = \int_{\mathbb{R}^{p'}} \sum_{\nu_1+\nu_2+\cdots+\nu_{p'}=m} \frac{m!}{\nu_1!\nu_2!\cdots\nu_{p'}!} \left(\frac{\partial^m g}{\partial x_1^{\nu_1}\partial x_2^{\nu_2}\cdots\partial x_{p'}^{\nu_{p'}}}\right)^2 dx_1 dx_2 \cdots dx_{p'}. \tag{3.53}$$

Here, p' is the number of covariates and m is the order of derivatives in the penalty, which would be 2 in the default case of a cubic spline.

The computational cost of solving (3.52) increases rapidly with p', so Wood (2003) proposes a dimension-reduction approach leading to an approximate solution. The reduced problem can still be expressed in the usual linear form,

$$\begin{aligned} &\arg\min_{\boldsymbol{\gamma}} ||\boldsymbol{e} \circ \boldsymbol{e} - (\boldsymbol{M} \circ \boldsymbol{M})\boldsymbol{L}\boldsymbol{\gamma}||_2^2 + \lambda\boldsymbol{\gamma}'\boldsymbol{P}\boldsymbol{\gamma}, \\ &\text{subject to } \boldsymbol{L}\boldsymbol{\gamma} \succeq \boldsymbol{0}^+. \end{aligned} \tag{3.54}$$

However, the procedure for computing $\boldsymbol{L}$ and $\boldsymbol{P}$ for the thin-plate spline ALVM is highly technical. The theory is summarised in Appendix C.3. In practice, these matrices can be computed with the help of the functions in the R package **mgcv** (Wood 2003).

3.2.4 A Nonparametric Approach to the Heteroskedastic Function Using Clustering

Another approach is possible that is nonparametric in the sense that it does not require either specification or estimation of the functional form of $g(\cdot)$. If two points $\boldsymbol{Z}_{j\cdot}$ and $\boldsymbol{Z}_{k\cdot}$ are near to each other in $\mathbb{R}^{p'}$ (that is, the distance from $\boldsymbol{Z}_{j\cdot}$ to $\boldsymbol{Z}_{k\cdot}$ is small, in terms of some distance metric), it follows from the differentiability of $g(\cdot)$ that the values of $g(\boldsymbol{Z}_{j\cdot})$ and $g(\boldsymbol{Z}_{k\cdot})$ are also similar:

$$\boldsymbol{Z}_{j\cdot} \approx \boldsymbol{Z}_{k\cdot} \implies g(\boldsymbol{Z}_{j\cdot}; \boldsymbol{\gamma}) \approx g(\boldsymbol{Z}_{k\cdot}; \boldsymbol{\gamma}) \Leftrightarrow \omega_j \approx \omega_k. \tag{3.55}$$

Hence, the dimensionality of (3.34) can be reduced by making the simplifying—and intuitively appealing—assumption that $\omega_j = \omega_k$ for observations sufficiently close to each other in $\mathbb{R}^{p'}$. To determine which observations are 'sufficiently close' in an empirical setting, a clustering algorithm can be used, with observations assigned to the same cluster assumed to have equal error variances.

Specifically, assume that the n rows of $\boldsymbol{Z}$—which is assumed to be standardised and not to include a column of ones—can be assigned to n_c groups ($n_c < n$). The group assignments are made in such a way that $\omega_k = \gamma_{C(k)}, k = 1, 2, \ldots, n$, where $C(k) \in \{1, 2, \ldots, n_c\}$ denotes the group number to which the kth observation (or variance) belongs, $k = 1, 2, \ldots, n$. $\boldsymbol{\gamma}$ is an n_c-vector of unknown parameters (hence $q = n_c$ in this case).

In practice, agglomerative hierarchical clustering can be used to assign the $\boldsymbol{Z}_{k\cdot}$ observations to groups (clusters) that are sufficiently compact to make the assumption of within-group homoskedasticity plausible. An agglomerative hierarchical clustering procedure runs as follows (Liu 2016):

1. Start with n clusters, each containing a single observation, and therefore an $n \times n$ symmetric matrix of distances;
2. Search the distance matrix for the nearest pair of clusters. Let this distance between the 'most similar' clusters U and V be denoted by d_{UV};
3. Merge clusters U and V, forming a new cluster, labelled UV. Update the distance matrix, deleting distances involving U and V and including distances involving UV;
4. Repeat steps 2 and 3 a total of $n - 1$ times. This will result in all variables being in a single cluster at the end of the algorithm.

In steps 2 and 3, a distance function is required to evaluate between-cluster distances. The distance function is chosen according to the desired 'linkage rule.' Common choices include the average linkage rule, single linkage rule (a.k.a. 'nearest neighbour'), and complete linkage rule (a.k.a. 'furthest neighbour'). A fourth option is to use Ward, Jr.'s (1963) method, which at each step combines the two clusters such that the resulting increase in the overall within-cluster sum of squared distances is minimised. Liu (2016) asserts that the average linkage and Ward rules perform best empirically, while James et al. (2013) express a preference for either complete or average linkage. The complete linkage rule is attractive in the present context, because the maximum distance between two would-be merged clusters represents the least plausible instance of the equality-of-variance assumption, per (3.55). Hence, the complete linkage rule is the default option considered here.[59]

Define $s(j)$ as the index set of observations belonging to the jth cluster, $j = 1, 2, \ldots, n_c$. Hence, for example, if $C(1) = C(4) = C(11) = 2$ and $C(k) \neq 2$ for all other $k \in \{1, 2, \ldots, n\}$, then $s(2) = \{1, 4, 11\}$. The distance between the kth and ℓth clusters, using the complete linkage rule, can be written as,

$$D_{k\ell} = \max_{i \in s(k), j \in s(\ell)} d(\boldsymbol{Z}_{i\cdot}, \boldsymbol{Z}_{j\cdot}), \tag{3.56}$$

where $d(\boldsymbol{Z}_{i\cdot}, \boldsymbol{Z}_{j\cdot})$ denotes the Euclidean distance between points $\boldsymbol{Z}_{i\cdot}$ and $\boldsymbol{Z}_{j\cdot}$.[60]

In practice, of course, the agglomerative hierarchical clustering procedure is normally stopped ('cut') at some point so that the final number of clusters is not 1 but some integer $n_c \in \{1, 2, \ldots, n\}$. The problem of selecting n_c is addressed in §3.3.2.2. In principle, however, there is a trade-off in that a smaller n_c means fewer parameters to be estimated whereas a larger n_c means observations within a cluster are closer together (and thus equality-of-variance assumptions are more reasonable).

Figure 3.5 shows an animation of the agglomerative hierarchical clustering algorithm on a two-dimensional data set with $n = 20$ observations and $n_c = 8$ clusters. Click the ▷ icon beneath the plot to play the animation. (Note that the animation should work in Adobe Acrobat but may not work in all PDF readers).

[59] Preliminary simulations suggested that the choice between average, complete, or Ward linkage does not drastically affect the results.

[60] Euclidean distance is appropriate if $\boldsymbol{Z}$ has been standardised. Otherwise, Mahalanobis distance may be preferable. If there are categorical predictors in the data, a hybrid distance metric such as Gower distance (Bruce and Bruce 2017) may be used.

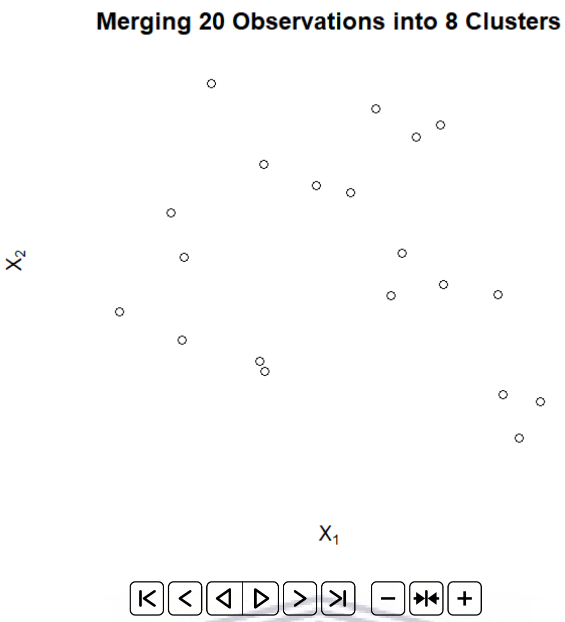


Figure 3.5: Animation of Agglomerative Hierarchical Clustering with Complete Linkage

By substituting $\omega_k = \gamma_{C(k)}$ into (3.33), the auxiliary model equation becomes,

$$e_i^2 = \sum_{k=1}^{n} \gamma_{C(k)} m_{ik}^2 + u_i. \tag{3.57}$$

This can be referred to as the clustering ALVM. By introducing further notation, however, the model equation can be written in the form of (3.42). Define $\boldsymbol{\mathcal{I}}$ as the $n \times n_c$ matrix with (i, j)th element

$$\mathcal{I}_{ij} = \begin{cases} 1 & \text{if } i \in s(j) \\ 0 & \text{otherwise} \end{cases}. \tag{3.58}$$

(3.57) can then be rewritten as

$$e_i^2 = \sum_{j=1}^{n_c} \gamma_j \sum_{k \in s(j)} m_{ik}^2 + u_i, i = 1, 2, \ldots, n,$$

or, in matrix form,

$$\boldsymbol{e} \circ \boldsymbol{e} = (\boldsymbol{M} \circ \boldsymbol{M}) \boldsymbol{\mathcal{I}} \boldsymbol{\gamma} + \boldsymbol{u}. \tag{3.59}$$

In terms of (3.42), the linear predictor matrix is $\boldsymbol{L} = \boldsymbol{\mathcal{I}}$. The estimation problem can be stated as

$$\begin{aligned} &\arg\min_{\boldsymbol{\gamma}} ||\boldsymbol{e} \circ \boldsymbol{e} - (\boldsymbol{M} \circ \boldsymbol{M}) \boldsymbol{\mathcal{I}} \boldsymbol{\gamma}||_2^2 \,, \\ &\text{subject to } \boldsymbol{I}_{n_c} \boldsymbol{\gamma} \succeq \boldsymbol{0}^+, \end{aligned} \tag{3.60}$$

where $\boldsymbol{I}_{n_c}$ is the $n_c \times n_c$ identity matrix.[61]

The clustering approach to dimensionality reduction can alternatively be used to construct a nonlinear model (ANLVM). Replace $\boldsymbol{\gamma}$ in (3.59) with $\boldsymbol{\gamma} \circ \boldsymbol{\gamma}$. The kth error variance ω_k is thus modelled as $\gamma_{C(k)}^2$, and no inequality constraint is required. The price of eliminating the constraint, however, is nonlinearity in the

[61] If $\boldsymbol{\mathcal{I}}$ were used instead of $\boldsymbol{I}_{n_c}$ in the constraint, the same constraint would be repeated for every row corresponding to a given cluster.

parameters. The model therefore becomes an ANLVM; indeed, it can be written in the form of (3.36), with $g(\boldsymbol{Z}_{k\cdot};\gamma) = (\boldsymbol{Z}'_{k\cdot}\gamma)^2$ and $\boldsymbol{Z} = \boldsymbol{\mathcal{I}}$. Thus, the model equation is,

$$e_i^2 = \sum_{j=1}^{n_c} \gamma_j^2 \sum_{k \in s(j)} m_{ik}^2 + u_i = \sum_{k=1}^{n} \left(\boldsymbol{\mathcal{I}}'_k \gamma\right)^2 m_{ik}^2 + u_i. \tag{3.61}$$

3.3 Applying the Auxiliary Variance Models: Fitting, Tuning, and Feature Selection

3.3.1 Fitting the Auxiliary Variance Models

The method used to fit the variance model will depend on whether or not the dimension-reduced model is linear in the parameters (an ALVM) or nonlinear (an ANLVM). In the case of an ALVM, the method further depends on whether or not the model includes a penalty term, and if so, what form that penalty takes.

3.3.1.1 Inequality-Constrained Least Squares

The estimation problem (3.35) for the basic or naïve ALVM in (3.34) is a least squares problem with a linear inequality constraint; that is, an Inequality-Constrained Least Squares (ICLS) problem.

An ICLS problem is of the form

$$\begin{gathered} \arg\min_{\boldsymbol{b}} ||\boldsymbol{y} - \boldsymbol{X}\boldsymbol{b}||_2^2, \\ \text{subject to } \boldsymbol{A}\boldsymbol{b} \succeq \boldsymbol{c}, \end{gathered} \tag{3.62}$$

where $\boldsymbol{b}$ is a q-vector, $\boldsymbol{y}$ is an n-vector, and $\boldsymbol{c}$ is an m-vector.[62] Note that nonnegative least squares (NNLS) is a special case of ICLS where $\boldsymbol{A} = \boldsymbol{I}_q$ and $\boldsymbol{c} = \boldsymbol{0}$. NNLS is not applicable to any of the ALVMs herein due to $\boldsymbol{A}$ (apart from the clustering model) not being the identity matrix and due to $\boldsymbol{c}$ being $\boldsymbol{0}^+$ rather than $\boldsymbol{0}$ (as discussed in connection with (3.35)).

For the basic ALVM, if the ICLS estimator does not touch the boundary of the constraint—in other words, if the OLS estimator of $\boldsymbol{\omega}$ from the model (3.34) has all positive elements—and there are no high-leverage points with $h_{ii} \geq 1/2$ (which usually holds),[63] then the ALVM is trivial in the sense that its predicted response vector $\widehat{\boldsymbol{e} \circ \boldsymbol{e}}$ is exactly its response vector $\boldsymbol{e} \circ \boldsymbol{e}$. To see this, observe that the model's projection matrix—the matrix $\boldsymbol{H}$ that satisfies $\widehat{\boldsymbol{e} \circ \boldsymbol{e}} = \boldsymbol{H}(\boldsymbol{e} \circ \boldsymbol{e})$—is $(\boldsymbol{M} \circ \boldsymbol{M})\left[(\boldsymbol{M} \circ \boldsymbol{M})'(\boldsymbol{M} \circ \boldsymbol{M})\right]^{-1}(\boldsymbol{M} \circ \boldsymbol{M})'$. But any positive definite symmetric matrix $\boldsymbol{A}$ has the property that $\boldsymbol{A}(\boldsymbol{A}'\boldsymbol{A})^{-1}\boldsymbol{A}' = \boldsymbol{I}$.[64]

It remains only to determine under what conditions $\boldsymbol{M} \circ \boldsymbol{M}$ is positive definite. A matrix is said to be diagonally dominant if the absolute value of each diagonal element is strictly greater than the sum of the absolute values of the off-diagonal elements in its row. A diagonally dominant matrix with positive diagonal entries is positive definite (Greenbaum 1997).[65]

Now, all elements of $\boldsymbol{M} \circ \boldsymbol{M}$ are nonnegative by definition, so it is not necessary to work with absolute values. Moreover, the diagonal elements $m_{ii}^2 = (1 - h_{ii})^2$ are strictly positive apart from a rare trivial case discussed in §1.1.10.1. Moreover, the identity $m_{ii} = \sum_{j=1}^{n} m_{ij}^2$ implies that the sum of off-diagonal elements in the ith row of $\boldsymbol{M} \circ \boldsymbol{M}$ is $m_{ii} - m_{ii}^2$. Thus, $\boldsymbol{M} \circ \boldsymbol{M}$ is diagonally dominant if $m_{ii}^2 > m_{ii} - m_{ii}^2$, which simplifies

[62] The use of $\boldsymbol{X}$ and $\boldsymbol{y}$ here are per convention and are not intended to refer to $\boldsymbol{X}$ and $\boldsymbol{y}$ in the classical linear regression model.

[63] Since $\operatorname{tr}(\boldsymbol{H}) = p$, the h_{ii} values average to p/n, and thus a value greater than $1/2$ would be extreme if $n \gg p$.

[64] Proof:

$$\begin{aligned} \boldsymbol{A}(\boldsymbol{A}'\boldsymbol{A})^{-1}\boldsymbol{A}' &= \boldsymbol{A}(\boldsymbol{A}\boldsymbol{A})^{-1}\boldsymbol{A} \\ &= \boldsymbol{A}\boldsymbol{A}^{-1}\boldsymbol{A}^{-1}\boldsymbol{A} \\ &= \boldsymbol{I}, \end{aligned}$$

provided that $\boldsymbol{A}^{-1}$ exists, which it does if $\boldsymbol{A}$ is positive definite.

[65] Note that this is a sufficient but not a necessary condition for positive definiteness; thus $\boldsymbol{M} \circ \boldsymbol{M}$ *may* be positive definite even if it is not diagonally dominant.

to the condition that $m_{ii} > 1/2$, or equivalently, $h_{ii} < 1/2$, for $i = 1, 2, \ldots, n$. Thus, in the absence of any observations for which $h_{ii} \geq 1/2$, $\boldsymbol{M} \circ \boldsymbol{M}$ is positive definite, and the OLS estimator of (3.34) is trivial in the aforementioned sense.

Empirical results suggest that the ICLS estimator (3.35) very often lies on the constraint boundary, in which case the estimator is nontrivial. The estimation problems (3.41) (for the linear ALVM model) and (3.60) (for the clustering ALVM) are also ICLS problems.

The `lsqlincon` function in the R package **pracma** (Borchers 2022) is one example of an ICLS solver in R. Since ICLS can be expressed as a Quadratic Programming (QP) problem, QP solvers such as those mentioned in §3.3.1.3 can also be used. Closed-form expressions for the ICLS estimator are given in Paula (1999) and Toker et al. (2013), but these do not allow circumvention of an algorithmic solver, since the closed form expressions require knowledge either of the Lagrangian vector (itself the solution to the dual problem, a QP problem) or at least knowledge of which constraints are satisfied at the boundary.

3.3.1.2 Inequality-Constrained Ridge Regression (ICRR)

An Inequality-Constrained Ridge Regression (ICRR) problem is of the form

$$\begin{aligned} &\arg\min_{\boldsymbol{b}} ||\boldsymbol{y} - \boldsymbol{X}\boldsymbol{b}||_2^2 + \lambda \boldsymbol{b}'\boldsymbol{P}\boldsymbol{b}, \\ &\text{subject to } \boldsymbol{A}\boldsymbol{b} \succeq \boldsymbol{c}, \end{aligned} \tag{3.63}$$

where $\boldsymbol{b}$ is a q-vector, $\boldsymbol{P}$ is a $q \times q$ penalty matrix, $\boldsymbol{y}$ is an n-vector, and $\boldsymbol{c}$ is an m-vector. The estimation problem (3.46) for the L_2-norm penalised polynomial ALVM is of the form (3.63), as are the estimation problems (3.51) and (3.54), for the smoothing spline and thin-plate spline ALVMs, respectively.

In the usual definition of Ridge Regression such as that found in Hastie et al. (2009), $\boldsymbol{P}$ is an identity matrix. However, $\boldsymbol{P}$ is not the identity matrix in the polynomial RR ALVM as parametrised here (since the upper left element is $P_{11} = 0$, due to the non-penalised intercept being the first element of $\boldsymbol{\gamma}$), and is also not an identity matrix in the smoothing spline ALVM or the thin-plate spline ALVM.

Toker et al. (2013) give a closed-form expression for the ICRR estimator. However, as with the ICLS estimator, computing the expression depends on knowledge of which constraints (if any) are satisfied at the boundary, and thus an algorithmic solver is still required. There does not seem to be an R function specifically tailored to solving (3.63), but since it is a special case of a QP problem, any of the quadratic programming solvers mentioned below in §3.3.1.3 can be used.

3.3.1.3 Quadratic Programming (QP)

Quadratic Programming (QP) problems are a well-studied class of optimisation problem (Boyd and Vandenberghe 2004, Best 2017). A QP problem can be written as

$$\begin{aligned} &\arg\min_{\boldsymbol{b}} \frac{1}{2}\boldsymbol{b}'\boldsymbol{Q}\boldsymbol{b} - \boldsymbol{d}'\boldsymbol{b}, \\ &\text{subject to } \boldsymbol{A}\boldsymbol{b} \succeq \boldsymbol{c}, \end{aligned} \tag{3.64}$$

where $\boldsymbol{Q}$ is a real-valued $q \times q$ symmetric positive definite or semi-definite matrix, $\boldsymbol{d}$ is a real-valued q-vector, and the parameter vector $\boldsymbol{b}$, constraint matrix $\boldsymbol{A}$ and constraint vector $\boldsymbol{c}$ are as defined previously.

It will be shown that all of the ALVMs discussed in §3.2.1, §3.2.2, and §3.2.3 (basic, linear, L_2- and L_1-norm penalised polynomials, B-spline, smoothing spline, thin-plate spline, clustering model) can be written as QP problems. First, it was noted already that, for all of these models, the model equation can be written in the form of (3.42),

$$\boldsymbol{e} \circ \boldsymbol{e} = \boldsymbol{D}\boldsymbol{\gamma} + \boldsymbol{u},$$

where $\boldsymbol{L}$ is an $n \times q$ ‘linear predictor matrix’ that projects the parameter $\boldsymbol{\gamma}$ onto the error variances $\boldsymbol{\omega}$, and $\boldsymbol{D} = (\boldsymbol{M} \circ \boldsymbol{M})\boldsymbol{L}$. The estimation problem for all of these models, with the exception of the polynomial model with L_1-norm penalty, can be written as

$$\begin{aligned} &\arg\min_{\boldsymbol{\gamma}} ||\boldsymbol{e} \circ \boldsymbol{e} - \boldsymbol{D}\boldsymbol{\gamma}||_2^2 + \lambda \boldsymbol{\gamma}' \boldsymbol{P} \boldsymbol{\gamma}, \\ &\text{subject to } \boldsymbol{A}\boldsymbol{\gamma} \succeq \boldsymbol{0}^+, \end{aligned} \tag{3.65}$$

where, for models without a penalty term, $\boldsymbol{P}$ is a zero matrix. Note that, for all ALVMs except the clustering ALVM, $\boldsymbol{A} = \boldsymbol{L}$, the linear predictor matrix. For the clustering ALVM, as noted earlier, $\boldsymbol{A} = \boldsymbol{I}_q$.[66] The objective function can be expanded and simplified as follows:

$$(\boldsymbol{e} \circ \boldsymbol{e} - \boldsymbol{D}\boldsymbol{\gamma})' (\boldsymbol{e} \circ \boldsymbol{e} - \boldsymbol{D}\boldsymbol{\gamma}) + \lambda \boldsymbol{\gamma}' \boldsymbol{P} \boldsymbol{\gamma} = (\boldsymbol{e} \circ \boldsymbol{e})'(\boldsymbol{e} \circ \boldsymbol{e}) - 2(\boldsymbol{e} \circ \boldsymbol{e})' \boldsymbol{D}\boldsymbol{\gamma} + \boldsymbol{\gamma}'(\boldsymbol{D}'\boldsymbol{D} + \lambda \boldsymbol{P})\boldsymbol{\gamma}.$$

Dropping the $(\boldsymbol{e} \circ \boldsymbol{e})'(\boldsymbol{e} \circ \boldsymbol{e})$ term and dividing the expression by 2 (neither of which affect minimisation with respect to $\boldsymbol{\gamma}$), the problem becomes

$$\begin{aligned} &\arg\min_{\gamma} \frac{1}{2} \gamma'(\boldsymbol{D}'\boldsymbol{D} + \lambda \boldsymbol{P})\gamma - (\boldsymbol{e} \circ \boldsymbol{e})' \boldsymbol{D}\boldsymbol{\gamma}, \\ &\text{subject to } \boldsymbol{A}\boldsymbol{\gamma} \succeq \boldsymbol{0}^+, \end{aligned} \tag{3.66}$$

which can be recognised as a QP problem with $\boldsymbol{b} = \gamma$, $\boldsymbol{Q} = \boldsymbol{D}'\boldsymbol{D} + \lambda \boldsymbol{P}$, $\boldsymbol{d} = \boldsymbol{D}'(\boldsymbol{e} \circ \boldsymbol{e})$, $\boldsymbol{A} = \boldsymbol{A}$, and $\boldsymbol{c} = \boldsymbol{0}^+$.

The L_1-norm penalised polynomial or LASSO ALVM does not follow the form in (3.65). This too can be expressed as a QP problem, but it requires a reparametrisation to accommodate the absolute value form, discussed in Gaines et al. (2018). Let $\boldsymbol{\gamma} = \boldsymbol{\gamma}^+ - \boldsymbol{\gamma}^-$, where $\boldsymbol{\gamma}^+$ and $\boldsymbol{\gamma}^-$ are, respectively, the positive and negative parts of $\boldsymbol{\gamma}$.[67] It follows that the L_1 norm of γ is $||\gamma||_1 = \gamma^+ + \gamma^-$. The objective function in (3.46) (but with L_1 norm penalty) is then modified as follows and written in terms of the $2q$-vector $\boldsymbol{\gamma}^{+-} = \begin{bmatrix} \gamma^+ \\ \gamma^- \end{bmatrix}$:

$$\begin{aligned} &||\boldsymbol{e} \circ \boldsymbol{e} - \boldsymbol{D}\boldsymbol{\gamma}||_2^2 + \lambda \boldsymbol{p}\, ||\gamma||_1 \\ &= \left|\left| \boldsymbol{e} \circ \boldsymbol{e} - \boldsymbol{D}\left(\gamma^+ - \gamma^-\right)\right|\right|_2^2 + \lambda \boldsymbol{p} \left(\gamma^+ + \gamma^-\right) \\ &= \left|\left| \boldsymbol{e} \circ \boldsymbol{e} - [\boldsymbol{D} \;\; -\boldsymbol{D}] \begin{bmatrix} \gamma^+ \\ \gamma^- \end{bmatrix} \right|\right|_2^2 + \lambda \left[\boldsymbol{p}', \boldsymbol{p}'\right] \begin{bmatrix} \gamma^+ \\ \gamma^- \end{bmatrix} \\ &= \left(\boldsymbol{e} \circ \boldsymbol{e} - [\boldsymbol{D} \;\; -\boldsymbol{D}]\, \gamma^{+-}\right)' \left(\boldsymbol{e} \circ \boldsymbol{e} - [\boldsymbol{D} \;\; -\boldsymbol{D}]\, \gamma^{+-}\right) + \lambda \left[\boldsymbol{p}', \boldsymbol{p}'\right] \gamma^{+-} \\ &= (\boldsymbol{e} \circ \boldsymbol{e})' (\boldsymbol{e} \circ \boldsymbol{e}) - \left(2\, (\boldsymbol{e} \circ \boldsymbol{e})' [\boldsymbol{D} \;\; -\boldsymbol{D}] + \lambda \left[\boldsymbol{p}', \boldsymbol{p}'\right]\right) \gamma^{+-} + \gamma^{+-\prime} \begin{bmatrix} \boldsymbol{D}' \\ -\boldsymbol{D}' \end{bmatrix} [\boldsymbol{D} \;\; -\boldsymbol{D}]\, \gamma^{+-}. \end{aligned}$$

Dropping the first term and dividing by 2 yields,

$$-\left((\boldsymbol{e} \circ \boldsymbol{e})' [\boldsymbol{D} \;\; -\boldsymbol{D}] + \frac{\lambda}{2} \left[\boldsymbol{p}', \boldsymbol{p}'\right]\right) \gamma^{+-} + \frac{1}{2} \gamma^{+-\prime} \begin{bmatrix} \boldsymbol{D}'\boldsymbol{D} & -\boldsymbol{D}'\boldsymbol{D} \\ -\boldsymbol{D}'\boldsymbol{D} & \boldsymbol{D}'\boldsymbol{D} \end{bmatrix} \gamma^{+-}.$$

Minimising this objective function with respect to γ^{+-} is thus a QP problem with

$$\begin{aligned} \boldsymbol{Q} &= \begin{bmatrix} \boldsymbol{D}'\boldsymbol{D} & -\boldsymbol{D}'\boldsymbol{D} \\ -\boldsymbol{D}'\boldsymbol{D} & \boldsymbol{D}'\boldsymbol{D} \end{bmatrix}, \\ \boldsymbol{d} &= \begin{bmatrix} \boldsymbol{D}' \\ -\boldsymbol{D}' \end{bmatrix} (\boldsymbol{e} \circ \boldsymbol{e}) + \frac{\lambda}{2} \begin{bmatrix} \boldsymbol{p} \\ \boldsymbol{p} \end{bmatrix}, \end{aligned}$$

and inequality constraint

$$\begin{bmatrix} \boldsymbol{A} & -\boldsymbol{A} \\ \boldsymbol{I}_{2q} & \end{bmatrix} \succ \begin{bmatrix} \boldsymbol{0}_n^+ \\ \boldsymbol{0}_{2q}. \end{bmatrix}. \tag{3.67}$$

[66] One could use the linear predictor matrix $\boldsymbol{L} = \boldsymbol{\mathcal{I}}$ as the constraint matrix $\boldsymbol{A}$ in the clustering ALVM as well, but this would result in some of the scalar inequalities being repeated.

[67] Thus, for example, if $\boldsymbol{\gamma} = [2, 0, -3]'$, then $\boldsymbol{\gamma}^+ = [2, 0, 0]'$ and $\boldsymbol{\gamma}^- = [0, 0, 3]'$.

The upper n rows of the $(n + 2q) \times 2q$ constraint matrix are derived through the reparametrisation as follows:

$$\boldsymbol{A}\boldsymbol{\gamma} = \boldsymbol{A}\left(\boldsymbol{\gamma}^{+} - \boldsymbol{\gamma}^{-}\right) = [\boldsymbol{A} \;\; -\boldsymbol{A}]\,\boldsymbol{\gamma}^{+-}.$$

The upper n elements of the constraint vector $\boldsymbol{c}$ are $\boldsymbol{0}_n^{+}$, representing the requirement that the variance estimates $\hat{\boldsymbol{\omega}}$ be positive. The last $2q$ rows of the constraint, $\boldsymbol{I}_{2q} \succ \boldsymbol{0}_{2q}$, reflect that the elements of $\boldsymbol{\gamma}^{+-} = \left[\boldsymbol{\gamma}^{+\prime}, \boldsymbol{\gamma}^{-\prime}\right]^{\prime}$ are by definition nonnegative.

As was noted previously, there is an absence of solvers specifically for ICRR in R software; thus, for the models that include L_2-norm penalties the best option is to use quadratic programming solvers. These are numerous in R software. There is the `solve.QP` function of the **quadprog** package (Turlach et al. 2019), the `QP.solve` function of the **quadprogpp** package (Noorian 2015), the `qpOASES` plugin (Schwendinger 2020) for the **ROI** package (Theußl et al. 2020), and the `solve_osqp` function of the **osqp** package (Stellato et al. 2021), which is an R implementation of the Operator Splitting Quadratic Program solver (Stellato et al. 2020).

As for the L_1-norm-penalised polynomial model, the `buildQP` function in the R package **quadprogXT** (Harlow 2020) automates the reparametrisation process to convert the estimation problem into a QP problem. The `lasso.ineq` function in the R package **PACLasso** (Paulson 2019) solves Inequality-Constrained LASSO (ICLASSO) regression problems directly, without requiring reparametrisation into a quadratic programming problem, using a method described in James et al. (2020).

3.3.1.4 Maximum Quasi-Likelihood Estimation

As stated above, (3.36) would be nonlinear in $\boldsymbol{\gamma}$—thus a ANLVM, not an ALVM—if $g(\cdot)$ is assumed to take a form like (3.38) or (3.39). The cluster model (3.59) is also an ANLVM if $\boldsymbol{\gamma}$ is replaced with $\boldsymbol{\gamma} \circ \boldsymbol{\gamma}$ as expressed in (3.61).

One method of fitting such ANLVMs is Maximum Quasi-Likelihood (MQL) estimation, as discussed in §2.3 of Seber and Wild (2003), following McCullagh (1983). The quasi-likelihood estimator has properties akin to those of a ML estimator but is based on less stringent conditions. MQL estimation makes no distributional assumptions but requires both the conditional mean vector and the conditional variance-covariance matrix of the response to be known in terms of the parameters to be estimated. This makes it particularly suitable for estimating an ANLVM, because the conditional mean vector and variance-covariance matrix of $\boldsymbol{e} \circ \boldsymbol{e}$ *are* known in terms of $\boldsymbol{\omega}$ under assumptions A1 and A3-A5 (see (3.34) and (3.11)),[68] and thus also in terms of $\boldsymbol{\gamma}$ after reparametrisation in terms of the heteroskedastic function $g(\cdot)$.

For simplicity, write $\boldsymbol{f}(\boldsymbol{\gamma})$ for the conditional mean function $\boldsymbol{f}(\boldsymbol{\Xi}; \boldsymbol{\gamma})$ in (3.36). Further, write $\boldsymbol{V}(\boldsymbol{\gamma})$ for the conditional covariance function in (3.11), $\mathrm{Cov}(\boldsymbol{u}) = \mathrm{Cov}(\boldsymbol{e} \circ \boldsymbol{e})$. This variance-covariance matrix is a function of $\boldsymbol{\gamma}$ in this case since $\boldsymbol{\Omega} = \mathrm{diag}\left\{\boldsymbol{\omega}\right\}$ and $\boldsymbol{\omega} = \boldsymbol{g}(\boldsymbol{Z}; \boldsymbol{\gamma})$.

The log-quasi-likelihood function of $\boldsymbol{\gamma}$, $\ell(\boldsymbol{\gamma})$, is defined by the system of partial differential equations,

$$\frac{\partial \ell(\boldsymbol{\gamma})}{\partial \boldsymbol{\gamma}} = \boldsymbol{V}^{-1}(\boldsymbol{\gamma})\left(\boldsymbol{e} \circ \boldsymbol{e} - \boldsymbol{f}(\boldsymbol{\gamma})\right). \tag{3.68}$$

The MQL estimator $\hat{\boldsymbol{\gamma}}_{MQL}$ is obtained by setting (3.68) equal to $\boldsymbol{0}$ and solving for $\boldsymbol{\gamma}$, which is equivalent to solving the system,

$$\boldsymbol{F}_{\bullet}^{\prime}(\boldsymbol{\gamma})\boldsymbol{V}^{-1}(\boldsymbol{\gamma})\left[\boldsymbol{e} \circ \boldsymbol{e} - \boldsymbol{f}(\boldsymbol{\gamma})\right] = \boldsymbol{0}, \tag{3.69}$$

where $\boldsymbol{F}_{\bullet}(\boldsymbol{\gamma}) = \dfrac{\partial \boldsymbol{f}(\boldsymbol{\gamma})}{\partial \boldsymbol{\gamma}^{\prime}}$.[69]

This is also equivalent to minimising the weighted sum of squares $S(\boldsymbol{\gamma}, \boldsymbol{V}(\boldsymbol{\gamma})) = \left[\boldsymbol{e} \circ \boldsymbol{e} - \boldsymbol{f}(\boldsymbol{\gamma})\right]^{\prime} \boldsymbol{V}^{-1}(\boldsymbol{\gamma})\left[\boldsymbol{e} \circ \boldsymbol{e} - \boldsymbol{f}(\boldsymbol{\gamma})\right]$. An approximate solution to (3.69) can be obtained using a Gauss-Newton method based on the linear Taylor expansion

$$\boldsymbol{f}(\hat{\boldsymbol{\gamma}}_{MQL}) \approx \boldsymbol{f}(\boldsymbol{\gamma}^{(a)}) + \boldsymbol{F}_{\bullet}(\boldsymbol{\gamma}^{(a)})\left(\hat{\boldsymbol{\gamma}}_{MQL} - \boldsymbol{\gamma}^{(a)}\right), \tag{3.70}$$

[68] Even if A5 is relaxed to A5′, the covariance matrix is known in terms of $\boldsymbol{\omega}$ and one additional scalar parameter ϕ; see §3.1.2.

[69] The prime in $\boldsymbol{F}_{\bullet}^{\prime}(\boldsymbol{\gamma})$ denotes matrix transpose, not a derivative.

for $\hat{\gamma}_{MQL}$ about an approximation $\boldsymbol{\gamma}^{(a)}$. By substituting (3.70) into (3.69), and approximating $\boldsymbol{F}_{\bullet}(\boldsymbol{\gamma})$ and $\boldsymbol{V}(\boldsymbol{\gamma})$ by, respectively, $\boldsymbol{F}_{\bullet}(\boldsymbol{\gamma}^{(a)})$ and $\boldsymbol{V}(\boldsymbol{\gamma}^{(a)})$, one arrives at an updating equation:

$$\boldsymbol{\gamma}^{(a+1)} - \boldsymbol{\gamma}^{(a)} \approx \left(\boldsymbol{F}_{\bullet}(\boldsymbol{\gamma}^{(a)})'\boldsymbol{V}(\boldsymbol{\gamma}^{(a)})^{-1}\boldsymbol{F}_{\bullet}(\boldsymbol{\gamma}^{(a)})\right)^{-1}\boldsymbol{F}_{\bullet}(\boldsymbol{\gamma}^{(a)})'\boldsymbol{V}(\boldsymbol{\gamma}^{(a)})^{-1}\left[\boldsymbol{e}\circ\boldsymbol{e} - \boldsymbol{f}\left(\boldsymbol{\gamma}^{(a)}\right)\right]. \tag{3.71}$$

Seber and Wild (2003, §2.8.8) alternatively suggest a ‘nested’ updating procedure:

$$\boldsymbol{\gamma}^{(a,b+1)} - \boldsymbol{\gamma}^{(a,b)} = \left(\boldsymbol{F}_{\bullet}(\boldsymbol{\gamma}^{(a,b)})'\boldsymbol{V}(\boldsymbol{\gamma}^{(a)})^{-1}\boldsymbol{F}_{\bullet}(\boldsymbol{\gamma}^{(a,b)})\right)^{-1}\boldsymbol{F}_{\bullet}(\boldsymbol{\gamma}^{(a,b)})'\boldsymbol{V}(\boldsymbol{\gamma}^{(a)})^{-1}\left[\boldsymbol{e}\circ\boldsymbol{e} - \boldsymbol{f}\left(\boldsymbol{\gamma}^{(a,b)}\right)\right]. \tag{3.72}$$

This scheme entails using $\boldsymbol{\gamma}^{(a,1)} = \boldsymbol{\gamma}^{(a)}$ and iterating (3.72) until convergence. One then iterates (3.71), but with $\boldsymbol{\gamma}^{(a)}$ replaced with $\lim_{b\to\infty}\boldsymbol{\gamma}^{(a,b)}$.

For a model with an assumed form of $g(\cdot)$ such as (3.38) or (3.39), $\boldsymbol{f}(\boldsymbol{\gamma})$ is an n-vector with ith element $f_i(\boldsymbol{\gamma}) = \sum_{k=1}^{n} g(\boldsymbol{Z}_k'\boldsymbol{\gamma})m_{ik}^2$. It follows that $\boldsymbol{F}_{\bullet}(\boldsymbol{\gamma})$ is an $n \times q$ matrix with (i,j)th element $\dfrac{\partial f_i(\boldsymbol{\gamma})}{\partial \gamma_j} = \sum_{k=1}^{n}\dfrac{\partial g(\boldsymbol{Z}_k'\boldsymbol{\gamma})}{\partial \gamma_j}m_{ik}^2$.

For the cluster model (3.59), $\boldsymbol{f}(\boldsymbol{\gamma})$ is an n-vector with ith element $f_i(\boldsymbol{\gamma}) = \sum_{j=1}^{r}\gamma_j^2\sum_{k\in s(j)} m_{ik}^2$. It follows that $\boldsymbol{F}_{\bullet}(\boldsymbol{\xi})$ is an $n \times r$ matrix with (i,j)th element $\dfrac{\partial f_i(\boldsymbol{\gamma})}{\partial \gamma_j} = 2\gamma_j\sum_{k\in s(j)} m_{ik}^2$.

In practice, $\boldsymbol{\gamma}$ is estimated by iterating (3.71) until some convergence criterion is reached. In some instances, the algorithm may not converge, or may not converge to the global minimum of $\boldsymbol{S}(\boldsymbol{\gamma},\boldsymbol{V}(\boldsymbol{\gamma}))$. Seber and Wild (2003, §15.2.1) note that, for classical Gauss-Newton methods, a good initial value is crucial to achieving convergence (and, one might add, a global optimum). Accordingly, it is suggested here to implement the estimation procedure over a grid of initial parameter vectors or a set of initial parameter vectors generated randomly from the uniform distribution.

3.3.1.5 Generalised Estimation Procedures

The MQL estimation procedure for ANLVMs made use of the fact that Cov($\boldsymbol{u}$) is known in terms of $\boldsymbol{\gamma}$. However, the methods used to fit ALVMs (ICLS, ICRR, and QP) do not make use of any information about Cov($\boldsymbol{u}$). Knowledge of the form of Cov($\boldsymbol{u}$) is thus wasted in the QP estimation method for ALVMs outlined in §3.3.1.3.

Recall from §1.1.6.2 that the WLS estimator is the BLUE of $\boldsymbol{\beta}$ under heteroskedasticity. The same argument implies that the Generalised Least Squares (GLS) estimator is the BLUE of $\boldsymbol{\beta}$ under heteroskedasticity and autocorrelation (the only difference between WLS and GLS being that, in the latter case, $\boldsymbol{W} = \boldsymbol{\Omega}^{-1}$ is not diagonal).

Why is this relevant here? It follows from (3.11) that Cov($\boldsymbol{u}$) (also denoted by $\boldsymbol{V}(\boldsymbol{\gamma})$) is non-diagonal: the ALVM errors $\boldsymbol{u}$ are both heteroskedastic and autocorrelated. Although the inequality constraint complicates derivation of the BLUE of $\boldsymbol{\gamma}$ in an ALVM, the argument of §1.1.6.2 at least implies that a GLS approach to fitting an ALVM—which exploits the known form of Cov($\boldsymbol{u}$), as the MQL procedure does—might improve the precision of estimation of $\boldsymbol{\gamma}$.

This motivates the use of Inequality-Constrained Generalised Least Squares (ICGLS). Since $\boldsymbol{V}(\boldsymbol{\gamma})$ is known only in terms of unknown parameters $\boldsymbol{\gamma}$, ICGLS is infeasible, just like WLS in a heteroskedastic linear regression model. What is really needed is Feasible Inequality-Constrained Generalised Least Squares (FICGLS), where $\boldsymbol{V}(\boldsymbol{\gamma})$ is replaced with an estimator, $\widehat{\boldsymbol{V}(\boldsymbol{\gamma})}$. If a penalty term is present, the model should be estimated by FICGRR (feasible inequality-constrained generalised ridge regression).

A natural choice for an estimator of $\boldsymbol{V}(\boldsymbol{\gamma})$ is the plug-in estimator,

$$\widehat{\boldsymbol{V}(\boldsymbol{\gamma})} = \boldsymbol{V}(\hat{\boldsymbol{\gamma}}). \tag{3.73}$$

Methods for ICGLS estimation are discussed in Werner (1990). The squared-error loss function in this instance changes from $||\boldsymbol{e}\circ\boldsymbol{e} - \boldsymbol{D}\boldsymbol{\gamma}||_2^2$ to

$$(\boldsymbol{e}\circ\boldsymbol{e}-\boldsymbol{D}\boldsymbol{\gamma})'\,\boldsymbol{V}^{-1}(\boldsymbol{\gamma})\,(\boldsymbol{e}\circ\boldsymbol{e}-\boldsymbol{D}\boldsymbol{\gamma})\,. \tag{3.74}$$

For the ALVMs that have been introduced (other than the L_1-norm penalty polynomial model), replacing $\boldsymbol{V}^{-1}(\boldsymbol{\gamma})$ in the loss function with an estimator $\widehat{\boldsymbol{V}^{-1}(\boldsymbol{\gamma})}$ results in a QP problem in $\boldsymbol{\gamma}$. The problem is like (3.66) but with $\boldsymbol{Q}=\boldsymbol{D}'\widehat{\boldsymbol{V}^{-1}(\boldsymbol{\gamma})}\boldsymbol{D}+\lambda\boldsymbol{P}$ and $\boldsymbol{d}=\boldsymbol{D}'\widehat{\boldsymbol{V}^{-1}(\boldsymbol{\gamma})}(\boldsymbol{e}\circ\boldsymbol{e})$.

If the plug-in estimator (3.73) is to be used, a multi-step procedure is required to estimate $\boldsymbol{\gamma}$, along the lines of the FWLS procedures for estimating $\boldsymbol{\beta}$ discussed in §2.2.1. A two-step FICGLS procedure would be:

1. Obtain a preliminary estimate of $\boldsymbol{\gamma}$, $\hat{\boldsymbol{\gamma}}^{(0)}$, using ICLS, ICRR, or QP as appropriate.
2. Compute the covariance matrix estimate $\widehat{\boldsymbol{V}(\boldsymbol{\gamma})}^{(1)}=\boldsymbol{V}(\hat{\boldsymbol{\gamma}}^{(0)})$.
3. Obtain an updated parameter estimate $\hat{\boldsymbol{\gamma}}^{(1)}$ by solving the QP (3.66) as modified above.

An iterative version of the procedure would entail repeatedly updating the covariance matrix estimate and re-estimating $\boldsymbol{\gamma}$ until some convergence criterion is reached (or the maximum allowable number of iterations is exhausted):

1. Obtain a preliminary estimate of $\boldsymbol{\gamma}$, $\hat{\boldsymbol{\gamma}}^{(0)}$, using ICLS, ICRR, or QP as appropriate.
2. Compute the covariance matrix estimate $\widehat{\boldsymbol{V}(\boldsymbol{\gamma})}^{(j)}=\boldsymbol{V}(\hat{\boldsymbol{\gamma}}^{(j-1)})$, for $j=1$.
3. Obtain an updated estimate $\hat{\boldsymbol{\gamma}}^{(j)}$, for $j=1$, by solving the QP (3.66) as modified above.
4. Repeat steps (2) and (3), for $j=2,3,\ldots$, updating the parameter estimate and covariance matrix estimate in turn until some convergence criterion is satisfied.

The function `orgls` in the R package **goric** (Gerhard and Kuiper 2021) fits ICGLS models directly.

Other generalisation approaches could incorporate a weight matrix in the first estimation step. For example, (3.4) implies that, under A1-A4 (homoskedasticity), $\mathrm{Cov}(\boldsymbol{e}\circ\boldsymbol{e})\propto\boldsymbol{M}\circ\boldsymbol{M}$. Thus, in the absence of information regarding the presence or degree of heteroskedasticity, one could use $(\boldsymbol{M}\circ\boldsymbol{M})^{-1}$ as a weight matrix in the loss function (3.74) in place of $\boldsymbol{V}(\boldsymbol{\gamma})^{-1}$ already in the first estimation step. Since $\boldsymbol{D}=(\boldsymbol{M}\circ\boldsymbol{M})\boldsymbol{L}$, in this case the matrix and vector of the quadratic programming problem simplify to $\boldsymbol{Q}=\boldsymbol{L}'(\boldsymbol{M}\circ\boldsymbol{M})\boldsymbol{L}$ and $\boldsymbol{d}=\boldsymbol{L}'(\boldsymbol{e}\circ\boldsymbol{e})$. Notice that the diagonal elements of $\boldsymbol{M}\circ\boldsymbol{M}$ are $(1-h_{ii})^2$, $i=1,2,\ldots,n$. Thus, disregarding the effect of the off-diagonal elements (most of which will usually be close to 0), this generalisation approach is equivalent to weighted least squares with weights $(1-h_{ii})^{-2}$—the adjustment factor used in the HC3 HCCME.

Yet another weighting approach can be proposed for the case where the purpose of estimating $\boldsymbol{\omega}$ is specifically to perform inference on one element of $\boldsymbol{\beta}$, say β_j, $j\in\{1,2,\ldots,p\}$. Consider the deviation of a sandwich estimator (based on an HCCME) from (1.6),

$$\widehat{\mathrm{Cov}}(\hat{\boldsymbol{\beta}})-\mathrm{Cov}(\hat{\boldsymbol{\beta}})=\left(\boldsymbol{X}'\boldsymbol{X}\right)^{-1}\boldsymbol{X}'\boldsymbol{\Delta}_{\hat{\boldsymbol{\Omega}}}\boldsymbol{X}\left(\boldsymbol{X}'\boldsymbol{X}\right)^{-1}, \tag{3.75}$$

where $\boldsymbol{\Delta}_{\hat{\boldsymbol{\Omega}}}=\hat{\boldsymbol{\Omega}}-\boldsymbol{\Omega}$. The matrix $\boldsymbol{X}\left(\boldsymbol{X}'\boldsymbol{X}\right)^{-1}$ is sometimes referred to as the 'bread' of the sandwich estimator. If the bread matrix is denoted $\boldsymbol{\mathcal{B}}$, with (i,j)th element b_{ij}, then the jth diagonal element of the deviation matrix (3.75) is given by $\displaystyle\sum_{i=1}^{n}b_{ij}^2\delta_i$, where $\boldsymbol{\delta}$ is the diagonal of $\boldsymbol{\Delta}_{\hat{\boldsymbol{\Omega}}}$. Thus, the extent to which the sandwich estimator based on $\hat{\boldsymbol{\Omega}}$ deviates from the true $\mathrm{Var}(\hat{\beta}_j)$—the square root of which is the standard error used in a quasi-t-test statistic for inference on β_j—depends not only on the δ_i but also on the squares of the jth column of the bread matrix $\boldsymbol{\mathcal{B}}$. This implies that, to achieve precise inference on β_j, it is more important to accurately estimate the error variances for observations with large b_{ij}^2. To prioritise these observations in the estimation procedure, the auxiliary model can be fitted using inequality-constrained weighted least squares, with weight matrix $\boldsymbol{W}=\mathrm{diag}\left\{b_{1j}^2,b_{2j}^2,\ldots,b_{nj}^2\right\}$.

Evaluation of the performance of these weighted estimation procedures is left for further research.

3.3.2 Tuning of Hyperparameters in an Auxiliary Linear Variance Model

3.3.2.1 The Penalty or Wiggliness Parameter λ

The penalised polynomial ALVM (3.46) (either with L_2-norm penalty or L_1-norm penalty) has a hyperparameter $\lambda\geq 0$ that governs the degree of shrinkage, that is, the degree of penalty imposed on the magnitude of $\boldsymbol{\gamma}$

(excluding the first element, i.e., the intercept). The univariate smoothing spline (3.51) and the multivariate thin-plate spline (3.54) likewise have a hyperparameter $\lambda \geq 0$ that imposes a penalty on the 'smoothness' or 'wiggliness' of the spline, as measured by the magnitude of second and/or higher-order derivatives.

An appropriate choice of λ is critical to the performance of these models. An excessively small penalty results in overfitting, whereas an excessively large penalty results in underfitting.

Two methods that are widely used to tune such continuous penalty parameters—both shrinkage penalties in regression models and smoothness penalties in spline models—are K-fold Cross-Validation (CV) and Generalised Cross-Validation (GCV) (Hastie et al. 2009, Wood 2017). The former authors remark that the shrinkage penalty parameter in RR 'should be adaptively chosen to minimise an estimate of expected prediction error' (Hastie et al. 2009, p. 69)

K-Fold Cross-Validation

K-fold CV entails randomly partitioning the data into K subsets of roughly equal size. For each $k = 1, 2, \ldots, K$, the model is trained on the kth *training fold*, consisting of all observations *not* in the kth subset. A performance metric (usually the squared-error loss) is computed using this model but using responses obtained from the kth subset (called the kth *test fold*). The performance metric is then averaged across all K folds. In this way, the model is being evaluated only on its predictive performance on out-of-sample data, without needing to set aside any portion of the data as 'test data.' It is generally considered best to use $K = 5$ or $K = 10$ folds (Hastie et al. 2009) to achieve a balanced bias/variance trade-off, although the special case of 'leave-one-out' CV ($K = n$) does have some useful applications.

Cross-Validation Modelling Procedure

How to compute the observed and predicted test response values for the ALVMs is nontrivial and requires some discussion. First, some new notation is needed. Let $\boldsymbol{y}_{\text{train}}$ and $\boldsymbol{X}_{\text{train}}$ be subsets of $\boldsymbol{y}$ and $\boldsymbol{X}$, respectively, corresponding to the observations in some arbitrary training fold. Similarly, let $\boldsymbol{y}_{\text{test}}$ and $\boldsymbol{X}_{\text{test}}$ be subsets of $\boldsymbol{y}$ and $\boldsymbol{X}$, respectively, corresponding to the observations in the test fold that is the counterpart of the training fold just mentioned. Let $\boldsymbol{e}_{\text{train}}$ and $\boldsymbol{e}_{\text{test}}$ be subsets of $\boldsymbol{e}$ analogous to $\boldsymbol{y}_{\text{train}}$ and $\boldsymbol{y}_{\text{test}}$. Define $\boldsymbol{Z}_{\text{train}}$ and $\boldsymbol{Z}_{\text{test}}$ in a similar manner, and let $\boldsymbol{M}_{\text{train}} = \boldsymbol{I} - \boldsymbol{X}_{\text{train}}(\boldsymbol{X}'_{\text{train}}\boldsymbol{X}_{\text{train}})^{-1}\boldsymbol{X}'_{\text{train}}$ and $\boldsymbol{M}_{\text{test}} = \boldsymbol{I} - \boldsymbol{X}_{\text{test}}(\boldsymbol{X}'_{\text{test}}\boldsymbol{X}_{\text{test}})^{-1}\boldsymbol{X}'_{\text{test}}$.

Further, let $\breve{\boldsymbol{\beta}}_{\text{train}} = (\boldsymbol{X}'_{\text{train}}\boldsymbol{X}_{\text{train}})^{-1}\boldsymbol{X}'_{\text{train}}\boldsymbol{y}_{\text{train}}$ and $\breve{\boldsymbol{\beta}}_{\text{test}} = (\boldsymbol{X}'_{\text{test}}\boldsymbol{X}_{\text{test}})^{-1}\boldsymbol{X}'_{\text{test}}\boldsymbol{y}_{\text{test}}$ be estimates of $\boldsymbol{\beta}$ computed from applying OLS to only the training data and only the test data, respectively. Similarly, let $\breve{\boldsymbol{e}}_{\text{train}} = \boldsymbol{y}_{\text{train}} - \boldsymbol{X}_{\text{train}}\breve{\boldsymbol{\beta}}_{\text{train}}$ and $\breve{\boldsymbol{e}}_{\text{test}} = \boldsymbol{y}_{\text{test}} - \boldsymbol{X}_{\text{test}}\breve{\boldsymbol{\beta}}_{\text{test}}$ be the OLS residuals computed from the models fitted to the training data and test data, respectively.

In terms of the above notation, the procedure for fitting an ALVM to a training fold is straightforward.

1. Form the predictor matrix $\boldsymbol{Z}_{\text{train}}$ by partitioning $\boldsymbol{Z}$ or, in the case that $\boldsymbol{Z}$ was formed by applying a feature selection procedure to $\boldsymbol{X}$, by applying this feature selection procedure to $\boldsymbol{X}_{\text{train}}$.[70]

2. Fit the ALVM,
$$\breve{\boldsymbol{e}}_{\text{train}} \circ \breve{\boldsymbol{e}}_{\text{train}} = (\boldsymbol{M}_{\text{train}} \circ \boldsymbol{M}_{\text{train}})\,\boldsymbol{L}_{\text{train}}\boldsymbol{\gamma} + \breve{\boldsymbol{u}}_{\text{train}}, \tag{3.76}$$
where $\boldsymbol{L}_{\text{train}}$ is a linear predictor matrix and $\breve{\boldsymbol{u}}_{\text{train}}$ is a random error vector. $\boldsymbol{L}_{\text{train}}$ would be computed from the training set of auxiliary covariates, $\boldsymbol{Z}_{\text{train}}$, using one of the methods discussed in §3.2.2-§3.2.4.

Fitting the ALVM (3.76) yields $\breve{\boldsymbol{\gamma}}_{\text{train}}$, an estimate of $\boldsymbol{\gamma}$, which is then used to predict the ALVM response vector for the test fold.

Obtaining predicted responses from the test fold is less straightforward than it first appears, however. A crucial principle in K-fold CV emphasised by Hastie et al. (2009) is that the training of the model must not be influenced in any way by the observations in the test fold. Consequently, the entire modelling procedure, including pre-processing or feature selection steps, must be performed on each training fold as part of the CV algorithm.

Two distinct procedures suggest themselves for computing the value of a loss function using the test fold. The first method may be termed the 'partitioning of residuals' technique and uses $\boldsymbol{e}_{\text{test}} \circ \boldsymbol{e}_{\text{test}}$, a subset of the squared residuals obtained from the OLS fit on the full data set of n observations. In this instance the steps to compute the loss function are as follows:

[70] The feature selection procedure may entail fitting of ALVMs for best subset selection, in which case this step is contained within the next step.

1. Use the training estimate of $\boldsymbol{\gamma}$, $\breve{\boldsymbol{\gamma}}_{\text{train}}$, to predict the ALVM responses for the full data set of n observations, $\widehat{\boldsymbol{e} \circ \boldsymbol{e}} = (\boldsymbol{M} \circ \boldsymbol{M})\, \boldsymbol{L} \breve{\boldsymbol{\gamma}}_{\text{train}}$.
2. Take a subset of $\widehat{\boldsymbol{e} \circ \boldsymbol{e}}$, $\left(\widehat{\boldsymbol{e} \circ \boldsymbol{e}}\right)_{\text{test}}$, using the indices of the test fold.
3. Compute the total squared-error loss for this test fold, $\left[\boldsymbol{e}_{\text{test}} \circ \boldsymbol{e}_{\text{test}} - \left(\widehat{\boldsymbol{e} \circ \boldsymbol{e}}\right)_{\text{test}}\right]' \left[\boldsymbol{e}_{\text{test}} \circ \boldsymbol{e}_{\text{test}} - \left(\widehat{\boldsymbol{e} \circ \boldsymbol{e}}\right)_{\text{test}}\right]$.
4. Aggregate the result of the previous step across all K test folds and average across the n observations, obtaining

$$\text{Loss}_{\text{CV}}(\lambda) = n^{-1} \sum_{i=1}^{n} \left(e_i^2 - \hat{e}_i^2\right)^2, \tag{3.77}$$

where $\hat{e}_i^2$ is the predicted value of the ith squared residual extracted from $\left(\widehat{\boldsymbol{e} \circ \boldsymbol{e}}\right)_{\text{test}}$ for the test fold containing the ith observation.

The main advantage of the partitioning of residuals technique is that it is simple to implement and allows the test folds to be arbitrarily small; even leave-one-out CV is possible. Its main shortcoming is that, because the squared OLS residuals are autocorrelated under both homoskedasticity and heteroskedasticity (see (3.5) and (3.14)), and because the training observations $\boldsymbol{y}_{\text{train}}$ and $\boldsymbol{X}_{\text{train}}$ are used both in the training model (to obtain $\breve{\boldsymbol{e}}_{\text{train}} \circ \breve{\boldsymbol{e}}_{\text{train}}$) and in the full model (of which $\boldsymbol{e}_{\text{test}} \circ \boldsymbol{e}_{\text{test}}$ are a subset of the squared residuals), the training and test responses are not mutually independent. Indeed, the $\boldsymbol{e}_{\text{test}} \circ \boldsymbol{e}_{\text{test}}$ are functions of both the $\boldsymbol{y}_{\text{train}}$ and $\boldsymbol{X}_{\text{train}}$ observations. Yet, importantly, Hastie et al.'s (2009) cardinal rule has not been violated: the test responses have, in a sense, 'seen' the training data, but the training responses have not in any sense 'seen' the test data.

The second method of obtaining predicted ALVM responses for the test fold may be called the 'test fold OLS' technique. It entails the following steps:

1. Apply OLS to the test fold only, thus obtaining the squared OLS residuals $\breve{\boldsymbol{e}}_{\text{test}} \circ \breve{\boldsymbol{e}}_{\text{test}}$.
2. Obtain predictions of $\breve{\boldsymbol{e}}_{\text{test}} \circ \breve{\boldsymbol{e}}_{\text{test}}$ by computing $\widehat{\breve{\boldsymbol{e}}_{\text{test}} \circ \breve{\boldsymbol{e}}_{\text{test}}} = (\boldsymbol{M}_{\text{test}} \circ \boldsymbol{M}_{\text{test}})\, \boldsymbol{L}_{\text{test}} \breve{\boldsymbol{\gamma}}_{\text{train}}$, where $\boldsymbol{L}_{\text{test}}$ is a linear predictor matrix computed from $\boldsymbol{Z}_{\text{test}}$.[71]
3. Compute the total squared-error loss for this test fold, $\left(\breve{\boldsymbol{e}}_{\text{test}} \circ \breve{\boldsymbol{e}}_{\text{test}} - \widehat{\breve{\boldsymbol{e}}_{\text{test}} \circ \breve{\boldsymbol{e}}_{\text{test}}}\right)' \left(\breve{\boldsymbol{e}}_{\text{test}} \circ \breve{\boldsymbol{e}}_{\text{test}} - \widehat{\breve{\boldsymbol{e}}_{\text{test}} \circ \breve{\boldsymbol{e}}_{\text{test}}}\right)$.
4. Aggregate the result of the previous step across all K test folds and average across the n observations, obtaining

$$\text{Loss}_{\text{CV}}(\lambda) = n^{-1} \sum_{i=1}^{n} \left(\breve{e}_i^2 - \hat{\breve{e}}_i^2\right)^2, \tag{3.78}$$

where $\breve{e}_i^2$ is the residual for the ith observation computed from the OLS fitted to its test fold, and $\hat{\breve{e}}_i^2$ is the predicted value thereof.

The main advantage of the test fold OLS technique is that there is complete mutual independence between the trained model and the test fold responses. Unlike the residual partitioning technique, the $\breve{e}_i^2$ have not in any sense 'seen' the training data.

A shortcoming with the test fold OLS technique is that the number of observations used to fit each test fold OLS may be small. This will result in high variances of the OLS parameter estimators, and is one reason why $K = 5$ folds may be preferable to $K = 10$. Indeed, if n is not very large relative to p, the number of observations in some test folds may be $\leq p$, in which case OLS cannot be fit.

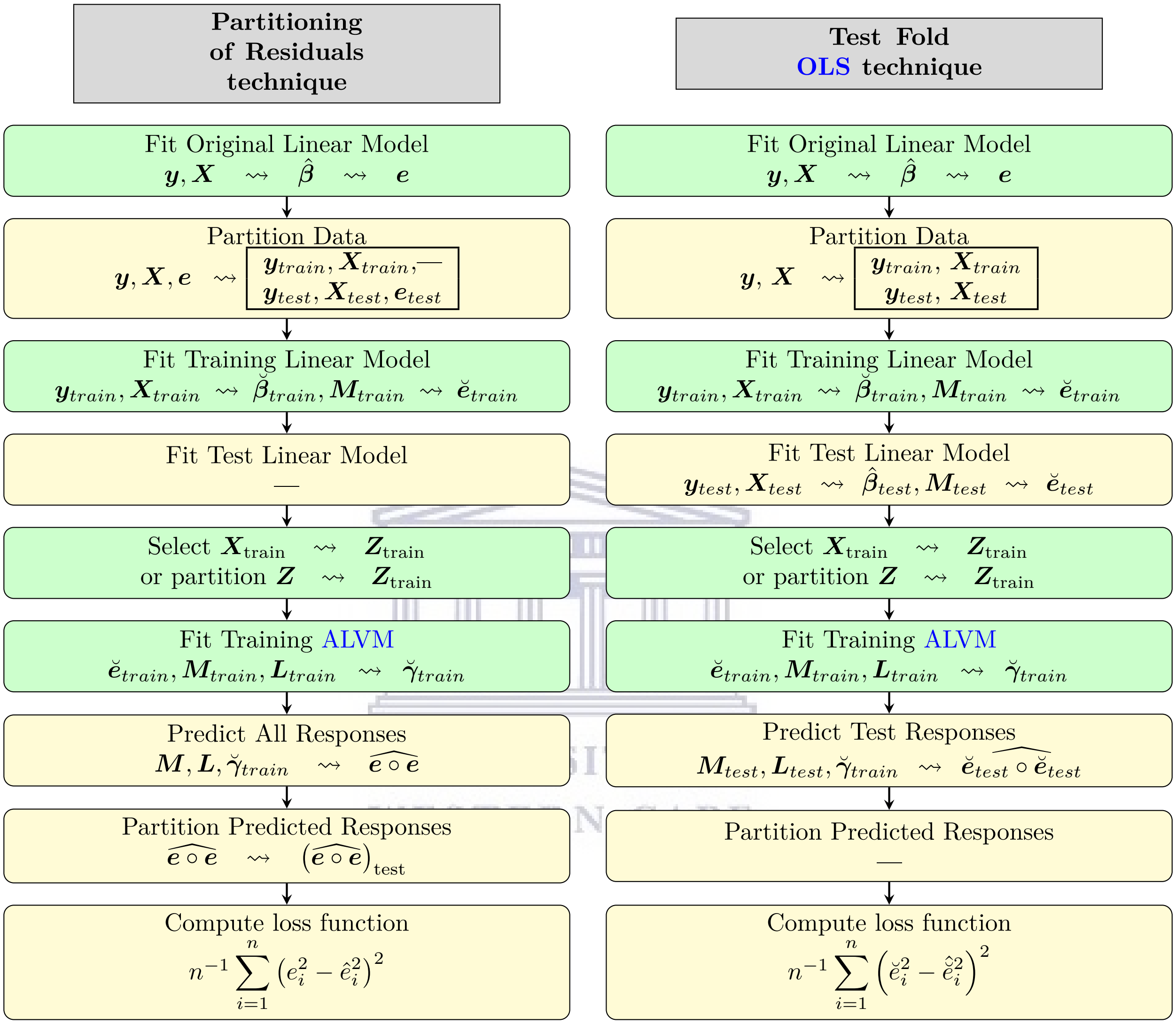
Under either of the two techniques, the tuned value of λ is,

$$\lambda_{\text{tuned}} = \arg\min_{\lambda} \text{Loss}_{\text{CV}}(\lambda). \tag{3.79}$$

The two techniques for computing the CV loss function are summarised diagrammatically in Figure 3.6. $a \rightsquigarrow b$ denotes that b is calculated from (is a function of) a. Steps highlighted in green are identical in both techniques, while steps highlighted in yellow differ.

[71] $\boldsymbol{L}_{\text{test}}$ is straightforward in the case of the polynomial model. With spline models, its computation is more complicated, since splines do not handle interpolation predictions and extrapolation predictions in the same way. The `predict` methods in the relevant R packages can handle construction of $\boldsymbol{L}$, however. If CV is being used to tune n_c in the clustering model, computation of $\boldsymbol{L}$ is also complicated, because a clustering procedure cannot be conducted separately on the test set. Rather, the $\boldsymbol{Z}_{i\cdot}$ observations in the test set must be assigned to the n_c existing clusters created from the training set in such a way as to optimise the linkage criterion being used.

The results in Chapter 5 all use the test fold OLS technique for CV, which was deemed to be the more conceptually sound of the two methods. Further research may be used to compare the two methods in terms of bias/variance trade-off.



Partitioning of Residuals technique
Test Fold OLS technique
Fit Original Linear Model
$\boldsymbol{y}, \boldsymbol{X} \rightsquigarrow \hat{\boldsymbol{\beta}} \rightsquigarrow \boldsymbol{e}$
Partition Data
$\boldsymbol{y}, \boldsymbol{X}, \boldsymbol{e} \rightsquigarrow \boldsymbol{y}_{train}, \boldsymbol{X}_{train}, — \quad \boldsymbol{y}_{test}, \boldsymbol{X}_{test}, \boldsymbol{e}_{test}$
$\boldsymbol{y}, \boldsymbol{X} \rightsquigarrow \boldsymbol{y}_{train}, \boldsymbol{X}_{train} \quad \boldsymbol{y}_{test}, \boldsymbol{X}_{test}$
Fit Training Linear Model
$\boldsymbol{y}_{train}, \boldsymbol{X}_{train} \rightsquigarrow \breve{\boldsymbol{\beta}}_{train}, \boldsymbol{M}_{train} \rightsquigarrow \breve{\boldsymbol{e}}_{train}$
Fit Test Linear Model
—
$\boldsymbol{y}_{test}, \boldsymbol{X}_{test} \rightsquigarrow \hat{\boldsymbol{\beta}}_{test}, \boldsymbol{M}_{test} \rightsquigarrow \breve{\boldsymbol{e}}_{test}$
Select $\boldsymbol{X}_{\text{train}} \rightsquigarrow \boldsymbol{Z}_{\text{train}}$
or partition $\boldsymbol{Z} \rightsquigarrow \boldsymbol{Z}_{\text{train}}$
Fit Training ALVM
$\breve{\boldsymbol{e}}_{train}, \boldsymbol{M}_{train}, \boldsymbol{L}_{train} \rightsquigarrow \breve{\boldsymbol{\gamma}}_{train}$
Predict All Responses
$\boldsymbol{M}, \boldsymbol{L}, \breve{\boldsymbol{\gamma}}_{train} \rightsquigarrow \widehat{\boldsymbol{e} \circ \boldsymbol{e}}$
Predict Test Responses
$\boldsymbol{M}_{test}, \boldsymbol{L}_{test}, \breve{\boldsymbol{\gamma}}_{train} \rightsquigarrow \widehat{\breve{\boldsymbol{e}}_{test} \circ \breve{\boldsymbol{e}}_{test}}$
Partition Predicted Responses
$\widehat{\boldsymbol{e} \circ \boldsymbol{e}} \rightsquigarrow \left(\widehat{\boldsymbol{e} \circ \boldsymbol{e}}\right)_{\text{test}}$
—
Compute loss function
$n^{-1} \sum_{i=1}^{n} \left(e_i^2 - \hat{e}_i^2\right)^2$
$n^{-1} \sum_{i=1}^{n} \left(\breve{e}_i^2 - \hat{\breve{e}}_i^2\right)^2$

Figure 3.6: Illustration of Cross-Validation Modelling Procedures

Minimising the Cross-Validated Squared-Error Loss Function

Minimising $\text{Loss}_{\text{CV}}(\lambda)$ with respect to λ is not a trivial optimisation problem, as the function may have multiple local minima. Wu and Lange (2008) recommend the use of a grid search combined with algorithmic search methods. A grid search entails evaluating $\text{Loss}_{\text{CV}}(\lambda)$ at a variety of different λ chosen randomly or systematically to cover a desired range of magnitudes. It may be desirable to search across several orders of magnitude (e.g., powers of 10), incrementing logarithmically rather than by equal steps.

In terms of algorithmic searches, since a closed form expression for $\text{Loss}_{\text{CV}}(\lambda)$ is not available, a method is needed that does not require computation of the function's derivative. Wu and Lange (2008) recommend combining bracketing and Golden Section Search (GSS). The bracketing method proceeds as follows. Begin with a large value λ_0 for which all or nearly all of the $\boldsymbol{\gamma}$ elements have been shrunk to 0 (in the polynomial model) or all second-order derivatives have been shrunk to 0 (in the spline model). Then:

1. Calculate $\lambda_{j+1} = r\lambda_j$, where $r \in (0, 1)$ (thus reducing λ by a fixed proportion), beginning with $j = 0$.
2. Compare $\text{Loss}_{\text{CV}}(\lambda_{j+1})$ with $\text{Loss}_{\text{CV}}(\lambda_j)$. If $\text{Loss}_{\text{CV}}(\lambda_{j+1}) \leq \text{Loss}_{\text{CV}}(\lambda_j)$, increment j and repeat from previous step. If $\text{Loss}_{\text{CV}}(\lambda_{j+1}) > \text{Loss}_{\text{CV}}(\lambda_j)$, perform a GSS to minimise $\text{Loss}_{\text{CV}}(\lambda)$ on the interval $[\lambda_{j+1}, \lambda_{j-1}]$.

An instance of the bracketing procedure is illustrated in Figure 3.7. In this instance, the function is evaluated at λ_0, then $\lambda_1 = 0.5\lambda_0$, and then $\lambda_2 = 0.5^2\lambda_0$. Since $\text{Loss}_{\text{CV}}(\lambda_1) < \text{Loss}_{\text{CV}}(\lambda_0)$ and $\text{Loss}_{\text{CV}}(\lambda_1) < \text{Loss}_{\text{CV}}(\lambda_2)$, a local minimum is bracketed by the interval $[\lambda_2, \lambda_0]$ and a GSS would therefore be conducted in that interval.

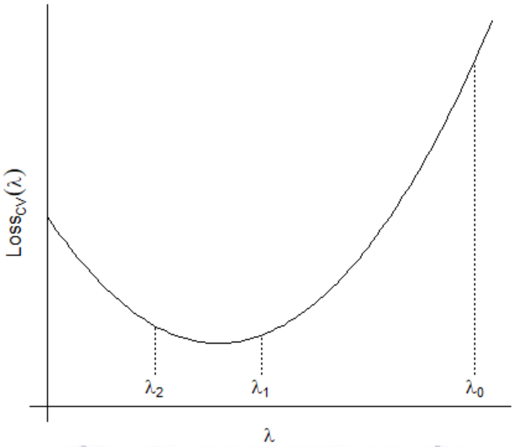


Figure 3.7: Bracketing Procedure for Identifying Golden Section Search Interval

Golden Section Search (GSS) (Lange 2010, Fox 2021) is a simple algorithm for minimising a univariate function $f(x)$ within a specified interval $[a, b]$. The method assumes that there is one local minimum in this interval. The solution produced is also an interval, the width of which can be made arbitrarily small by setting a tolerance value. Since the desired solution is a single point,

$$\underset{a<x<b}{\arg\min} f(x),$$

the midpoint of the final interval can be taken as the best approximation to the solution.

The premise of the method is to shorten the interval $[a, b]$ by comparing the values of the function at two test points, $x_1 < x_2$, where,

$$x_1 = a + r(b - a) \tag{3.80}$$

$$x_2 = b - r(b - a), \tag{3.81}$$

and $1/2 < r < 1$. The algorithm works by comparing $f(x_1)$ to $f(x_2)$ and updating the search interval accordingly, as follows:

- If $f(x_1) < f(x_2)$, the solution cannot lie in $[x_2, b]$; thus, update the search interval to $[a_{\text{new}}, b_{\text{new}}] = [a, x_2]$.
- If $f(x_1) > f(x_2)$, the solution cannot lie in $[a, x_1]$; thus, update the search interval to $[a_{\text{new}}, b_{\text{new}}] = [x_1, b]$.

One then chooses two new test points by replacing a and b with a_{new} and b_{new} in (3.80) and (3.81), and iterating until $b - a < \tau$, where τ is the desired tolerance.

The name 'golden section search' comes from the optimal choice of r in (3.80) and (3.81). The boundary values $r = 1/2$ and $r = 1$ represent the cases where $x_1 = x_2$ and $x_1 = a$, $x_2 = b$, respectively, at which the algorithm breaks down. It can be shown that the optimal choice of r is $\dfrac{\sqrt{5}-1}{2} \approx 0.618$, which is the reciprocal of the golden ratio.

The combined bracketing-and-golden-section-search method is highly effective if the function has only one local minimum. If there are multiple local minima, however, then since bracketing moves from right to left, the algorithm is biased toward arriving at the largest one (in terms of λ value, not necessarily function value). To increase the chances of finding the global minimum, therefore, this method is combined with a grid search as follows:

1. Evaluate $\text{Loss}_{\text{CV}}(\lambda)$ at values $\lambda \in \left\{10^m, 10^{m+1}, \ldots, 10^M\right\}$, where m is a small integer (e.g., -3) and M is an integer large enough that $\lambda = 10^M$ results in all coefficients (for polynomial model) or second-order derivatives being penalised to 0.
2. If ℓ is the exponent of 10 that minimises $\text{Loss}_{\text{CV}}(\lambda)$ among the values in the sequence, apply the bracketing procedure with $\lambda_0 = 10^{\ell+1}$.
3. Apply the GSS algorithm on the interval identified by the bracketing procedure. If the bracketing procedure failed, apply the GSS algorithm on the interval $\left[0, 10^{\ell+1}\right]$.

Quasi-Generalised Cross-Validation

Hastie et al. (2009) define a linear fitting method as a method for which one can write $\hat{\boldsymbol{y}} = \boldsymbol{S}\boldsymbol{y}$, where $\boldsymbol{y}$ is the vector of responses, $\hat{\boldsymbol{y}}$ is the vector of fitted values, and $\boldsymbol{S}$ is an $n \times n$ matrix depending on the covariate matrix but not on the response vector.[72] For any linear fitting method, GCV provides a computationally efficient approximation to the cross-validated squared-error loss function under leave-one-out CV. The formula, adapted to the variance models and the λ hyperparameter under consideration here, is

$$\text{GCV}_{\text{ICRR}}(\lambda) = n^{-1} \sum_{i=1}^{n} \left[\frac{e_i^2 - \hat{e}_i^2}{1 - \text{tr}(\boldsymbol{S})/n} \right]^2, \tag{3.82}$$

where $\hat{e}_i^2$ is the predicted response from the ALVM, and $\text{tr}(\boldsymbol{S})$ is a measure of effective number of parameters or degrees of freedom. Now, (3.77) has a higher variance under leave-one-out CV than under five- or ten-fold CV, due to the similarity of the n training sets (Hastie et al. 2009). Thus, obtaining an approximation to (3.77) under leave-one-out CV involves moving to a less-than-optimal point along the bias-variance trade-off spectrum in exchange for greatly reduced computation time.

Escobar and Skarpness (1984) derive a closed-form expression for the ICLS estimator, while Paula (1993) write it in a more convenient form, which Toker et al. (2013) extend to the ICRR case. With notation adapted to the present situation, write

$$\tilde{\gamma}_{\text{ICRR}} = \hat{\gamma}_{\text{RR}} + \left(\boldsymbol{D}'\boldsymbol{D} + \lambda\boldsymbol{P}\right)^{-1} \boldsymbol{A}'_{\mathcal{R}} \hat{\boldsymbol{\eta}}_{\mathcal{R}}, \tag{3.83}$$

where $\hat{\gamma}_{\text{RR}} = (\boldsymbol{D}'\boldsymbol{D} + \lambda\boldsymbol{P})^{-1}\boldsymbol{D}'(\boldsymbol{e} \circ \boldsymbol{e})$ is the unconstrained RR estimator of $\boldsymbol{\gamma}$, $\boldsymbol{D}$, $\boldsymbol{A}$, and $\boldsymbol{P}$ are as specified in (3.65), $\hat{\boldsymbol{\eta}}$ is the Lagrangian vector (of length m), $\mathcal{R}$ is the set of cardinality s containing indices of nonzero elements of $\hat{\boldsymbol{\eta}}$ (corresponding to constraints satisfied at equality), $\boldsymbol{A}_{\mathcal{R}}$ is the $s \times q$ sub-matrix of $\boldsymbol{A}$ consisting only of rows $\{i : i \in \mathcal{R}\}$, and $\hat{\boldsymbol{\eta}}_{\mathcal{R}}$ is the sub-vector of $\hat{\boldsymbol{\eta}}$ consisting only of elements $\{i : i \in \mathcal{R}\}$. The ICLS estimator, $\tilde{\gamma}_{\text{ICLS}}$, can be obtained from (3.83) by setting $\boldsymbol{P}$ to a zero matrix (or fixing λ at 0), and replacing $\hat{\gamma}_{\text{RR}}$ with $\hat{\gamma}_{\text{OLS}} = (\boldsymbol{D}'\boldsymbol{D})^{-1}\boldsymbol{D}'(\boldsymbol{e} \circ \boldsymbol{e})$.

A closed-form expression for $\hat{\boldsymbol{\eta}}_{\mathcal{R}}$ is given by

$$\hat{\boldsymbol{\eta}}_{\mathcal{R}} = -\left(\boldsymbol{A}_{\mathcal{R}}(\boldsymbol{D}'\boldsymbol{D} + \lambda\boldsymbol{P})^{-1}\boldsymbol{A}'_{\mathcal{R}}\right)^{-1} \boldsymbol{A}_{\mathcal{R}} \hat{\gamma}_{\text{RR}}. \tag{3.84}$$

Notice that the ICRR problem cannot be solved by evaluating (3.83), because one cannot evaluate (3.83) or (3.84) without first knowing $\mathcal{R}$, which cannot be known prior to solving the ICRR problem.

Extending the argument of Paula (1993) and Paula (1999) from the ICLS case to ICRR case, it can be shown that, if $\boldsymbol{b}_{\mathcal{R}} = \boldsymbol{0}$ (where $\boldsymbol{b}_{\mathcal{R}}$ is a subvector of the right side of the inequality constraint),[73] the ICRR fitted values $\widetilde{\boldsymbol{e} \circ \boldsymbol{e}}_{\text{ICRR}} = \boldsymbol{D}\tilde{\gamma}_{\text{ICRR}}$ can be written as

$$\widetilde{\boldsymbol{e} \circ \boldsymbol{e}}_{\text{ICRR}} = (\boldsymbol{H}_{\boldsymbol{D},\lambda} - \boldsymbol{G}_{\lambda})\,(\boldsymbol{e} \circ \boldsymbol{e}), \tag{3.85}$$

where $\boldsymbol{H}_{\boldsymbol{D},\lambda} = \boldsymbol{D}\left(\boldsymbol{D}'\boldsymbol{D} + \lambda\boldsymbol{P}\right)^{-1}\boldsymbol{D}'$ and $\boldsymbol{G}_{\lambda} = \boldsymbol{U}\left(\boldsymbol{A}_{\mathcal{R}}(\boldsymbol{D}'\boldsymbol{D} + \lambda\boldsymbol{P})^{-1}\boldsymbol{A}'_{\mathcal{R}}\right)^{-1}\boldsymbol{U}'$, $\boldsymbol{U} = \boldsymbol{D}\left(\boldsymbol{D}'\boldsymbol{D} + \lambda\boldsymbol{P}\right)^{-1}\boldsymbol{A}'_{\mathcal{R}}$. Again, these expressions can be reduced to the ICLS case by fixing $\lambda = 0$

[72] For instance, $\boldsymbol{S} = \boldsymbol{H}$ in the case of linear regression (see §1.1.2).

[73] This 'nearly' holds in the auxiliary variance models discussed herein, per the notation $\boldsymbol{0}^{+}$ introduced in connection with (3.35).

or $\boldsymbol{P}$ to a zero matrix. Notice further that $\boldsymbol{G}_\lambda$ falls away if no constraints are met at the boundary, since $\boldsymbol{U}$ is in that case an $n \times 0$ matrix.

From (3.85) it follows that $\boldsymbol{S}_{\text{ICRR}} = \boldsymbol{H}_{\boldsymbol{D},\lambda} - \boldsymbol{G}_\lambda$ is the projection matrix of the ICRR model. Importantly, however, because the set $\mathcal{R}$ depends on the response $\boldsymbol{e} \circ \boldsymbol{e}$, ICLS and ICRR are not linear fitting methods as defined by Hastie et al. (2009), which introduces potentially significant bias into the use of (3.82) as an approximation for the leave-one-out cross-validated loss function.

One possible way around this is to perform GCV on the *unconstrained* versions of the models—which do qualify as linear fitting methods—and extrapolate the optimal tuning parameter to the constrained situation. The GCV loss function (in the RR case) is then

$$\text{GCV}_{\text{RR}}(\lambda) = n^{-1} \sum_{i=1}^{n} \left[\frac{e_i^2 - \hat{e}_i^2}{1 - \text{tr}(\boldsymbol{H}_{\boldsymbol{D},\lambda})/n} \right]^2, \tag{3.86}$$

where $\hat{e}_i^2$ is the predicted value of the ith squared OLS residual based on $\hat{\boldsymbol{\gamma}}_{\text{OLS}}$ or $\hat{\boldsymbol{\gamma}}_{\text{RR}}$, and $\boldsymbol{H}_{\boldsymbol{D}} = \boldsymbol{D}(\boldsymbol{D}'\boldsymbol{D})^{-1}\boldsymbol{D}'$. However, this involves exchanging one form of bias for another, since the GCV function based on the unconstrained model could behave quite differently than that based on the constrained model, leading to an unsuitable choice of λ. Thus, minimising either (3.82) (with projection matrix $\boldsymbol{S}_{\text{ICRR}}$) or (3.86) with respect to λ is not a true GCV method. This approach will therefore be referred to as Quasi-Generalised Cross-Validation (QGCV).

The QGCV approaches to tuning λ described above are applicable to the L_2-norm penalised polynomial model and the smoothing spline and thin-plate spline models, but not to the L_1-norm (LASSO-type) penalised polynomial model. The unconstrained LASSO estimator does not have a closed form expression. However, by introducing an additional layer of approximation, a QGCV approach for this model is also possible.

As Tibshirani (1996) explains, an approximate closed form of the (unconstrained) LASSO estimator can be expressed (using the notation of the auxiliary variance model) as

$$\hat{\boldsymbol{\gamma}}_{\text{lasso}} = \left(\boldsymbol{D}'\boldsymbol{D} + \lambda \boldsymbol{\mathcal{W}}^{-} \right)^{-1} \boldsymbol{D}'(\boldsymbol{e} \circ \boldsymbol{e}), \tag{3.87}$$

where $\boldsymbol{\mathcal{W}}^{-}$ is the generalised inverse of $\boldsymbol{\mathcal{W}} = \text{diag}\left\{0, |\hat{\gamma}_2|, \ldots, |\hat{\gamma}_q|\right\}$,[74] $\hat{\gamma}_j$ being the jth element of the unconstrained OLS estimator of $\boldsymbol{\gamma}$. Hence, a QGCV procedure for the LASSO model can be outlined as follows:

(i) Compute the unconstrained LASSO estimator $\hat{\boldsymbol{\gamma}}_{\text{lasso}}$ using QP and so determine $\boldsymbol{\mathcal{W}}^{-}$.

(ii) Compute the Inequality-Constrained LASSO (ICLASSO) estimator $\tilde{\boldsymbol{\gamma}}_{\text{lasso}}$ using QP and so determine the predicted squared residual values.

(iii) Evaluate the GCV loss function (3.82), using the projection matrix $\boldsymbol{H}_{\boldsymbol{D},\lambda} - \boldsymbol{G}_\lambda$ but with $\boldsymbol{P}$ replaced by $\boldsymbol{\mathcal{W}}^{-}$.

This QGCV procedure requires two QP problems to be solved for each GCV loss function evaluation, compared with five QP problems (with fewer observations in each) for five-fold CV. Thus, the savings in computation time may not justify the multiple layers of approximation that have been introduced.

A simpler and faster approach is to follow the observation of Efron et al. (2004) that, in the LASSO (as parametrised here), $\text{tr}(\boldsymbol{S}) \approx \#\left(j \in \{2, 3, \ldots, q\} : \hat{\gamma}_j \neq 0\right)$. This approximation should hold true in the ICLASSO as well, and obviates step (i) in the procedure above; thus only one QP problem needs to be solved to evaluate the QGCV loss.

A final note about the QGCV approach concerns the issue that, technically, if a feature selection method is applied as a pre-processing step in the model, e.g., using a heteroskedasticity test (as discussed below in §3.3.3), this pre-processing step needs to be performed separately on each training fold. This pre-processing is not taken into account in the QGCV approximation. However, because GCV approximates leave-one-out CV, each training set contains $n-1$ observations, and it is therefore reasonable to assume that the features that would have been selected for each training set are identical to those that are selected for the full set of n observations.

[74] The 0 is due to the first element of $\boldsymbol{\gamma}$ not being penalised in the LASSO model as parametrised herein.

3.3.2.2 Number of Clusters n_c

There are a large number of metrics attested in the literature for identifying the relevant number of clusters in a data set (Charrad et al. 2014). One of the simpler methods involves identifying the elbow point (sometimes called knee point) of a curve measuring some important criterion. Here, two such criteria are considered that measure the compactness of the clusters, which is important given the assumption that all observations in the same cluster are close enough to each other to have the same error variance. These are the Sum of Within-Cluster Distances (SWD) and Maximum Within-Cluster Distance (MWD) criteria:

$$\mathrm{SWD}(n_c) = \sum_{k=1}^{n_c} \sum_{\substack{i,j \in s(k) \\ i<j}} d(\boldsymbol{Z}_{i\cdot,-1}, \boldsymbol{Z}_{j\cdot,-1}) \tag{3.88}$$

and

$$\mathrm{MWD}(n_c) = \max_{k \in \{1,2,\ldots,n_c\}} \max_{\substack{i,j \in s(k) \\ i<j}} d(\boldsymbol{Z}_{i\cdot,-1}, \boldsymbol{Z}_{j\cdot,-1}), \tag{3.89}$$

where $\boldsymbol{Z}_{i\cdot,-1}$ is the ith row of $\boldsymbol{Z}_{-1}$, which is $\boldsymbol{Z}$ with a column of ones removed if present.

The elbow points of these two functions of n_c are determined numerically using the Unit Invariant Knee (UIK) technique implemented in the `uik` function of the R package **inflection** (Christopoulos 2019). Both $\mathrm{SWD}(n_c)$ and $\mathrm{MWD}(n_c)$ decrease with n_c, and in both cases a smaller value is desirable. However, the elbow point allows one to find a point that trades off optimally, in some sense, between a smaller value of the criterion and a smaller dimensionality of the parameter $\boldsymbol{\gamma}$.

How far apart two points must be before the equal variance assumption becomes problematic depends, of course, on the function $g(\cdot)$, and particularly on the magnitude of its first derivative function, $|g'(\cdot)|$. In a neighbourhood where this is large, observations that are quite close together in the $\boldsymbol{Z}_{-1}$ space may nonetheless have highly unequal error variances. Conversely, in a neighbourhood where $|g'(\cdot)|$ is very small ($g(\cdot)$ is very flat), observations that are quite far apart may nonetheless have nearly equal error variances. Thus, the elbow methods have a shortcoming in that they are informed only by the distribution of the $\boldsymbol{Z}_{i\cdot,-1}$ covariate points and not by any information about $|g'(\cdot)|$.

CV is an alternative approach to choosing n_c that attends more directly to its impact on the performance of the auxiliary regression model. Here, the aim is to minimise the same loss function as in (3.78) (or (3.77), depending on the CV technique used), now taken as a function of n_c rather than of λ.

Since n_c is an integer, the search for the optimal value is much simpler; an exhaustive search of all $t \in \{1, 2, \ldots, n\}$ can be carried out. Since computation time for fitting the model increases with n_c, an early stop rule can be applied, as follows: if $t \geq 3$ and $\mathrm{Loss}_{\mathrm{CV}}(t) > \mathrm{Loss}_{\mathrm{CV}}(t-1)$ and $\left[\dfrac{\mathrm{Loss}_{\mathrm{CV}}(t) - \mathrm{Loss}_{\mathrm{CV}}(n_{c,\mathrm{opt}})}{\mathrm{Loss}_{\mathrm{CV}}(n_{c,\mathrm{opt}})}\right] > 1$, where $n_{c,\mathrm{opt}} = \arg\min_{s \in \{1,2,\ldots,t\}} \{\mathrm{Loss}_{\mathrm{CV}}(s)\}$, stop the search and set $n_c = n_{c,\mathrm{opt}}$. That is, the search will continue up to at least $t = 3$. Thereafter, if the cross-validated loss function at $n_c = t$ is greater than at $n_c = t - 1$, and if the current value of the loss function is more than 100% greater than the minimum value seen so far, the search is stopped.

The implementation of the clustering routine in conjunction with CV requires some comment. The observations in the kth training fold, $k = 1, 2, \ldots, K$, are assigned to $n_c = t$ clusters as explained previously and the ALVM is fitted. A completely separate clustering routine cannot be used with the kth test fold, because it would then not be obvious how to map the coefficient estimates from the training fold's clusters onto the test fold's clusters. Instead, each observation in the kth test fold is assigned to the nearest cluster of the corresponding training fold, with point-to-cluster distance computed using the applicable linkage rule, as discussed in §3.2.4. This allows computation of a matrix $\boldsymbol{L}_{test} = \boldsymbol{\mathcal{I}}_{test}$ after the manner of (3.58); predicted test responses (squared residuals) can then be computed under the test fold OLS technique (Figure 3.6) using $\widehat{\breve{\boldsymbol{e}}_{test} \circ \breve{\boldsymbol{e}}_{test}} = (\boldsymbol{M}_{test} \circ \boldsymbol{M}_{test})\, \boldsymbol{L}_{test} \breve{\boldsymbol{\gamma}}_{train}$. The process of assigning test fold observations to training fold clusters is illustrated in Figure 3.8. The circular points represent 40 training observations, which were assigned to six clusters. The triangular points represent 10 test observations, each of which has been assigned to the nearest cluster using the complete linkage rule.

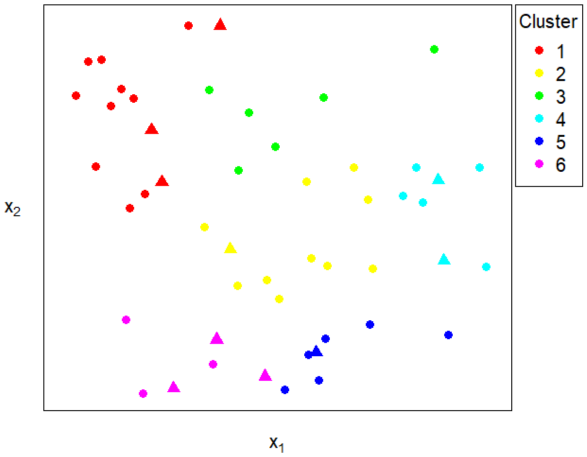


Figure 3.8: Allocation of Test Fold Observations to Nearest Training Fold Cluster

B-Spline Number of Interior Knots

If the univariate B-spline ALVM (3.49) is used, the number of interior knots in the spline is an important hyperparameter. This, too, is a nonnegative integer. The CV procedure just described for tuning n_c in the cluster model can also be applied here, as can the QGCV loss function.

3.3.3 Feature Selection in Auxiliary Variance Models

In principle, the error variances—that is, the response variances—might depend on different variables than the mean response. The $\boldsymbol{Z}$ matrix could include covariates that are not part of $\boldsymbol{X}$, but may not include all covariates that that are part of $\boldsymbol{X}$. In the absence of any prior knowledge of additional variables that are not part of $\boldsymbol{X}$ that might influence the error variances, however, the covariates in $\boldsymbol{X}$ effectively become the candidates for the covariates in $\boldsymbol{Z}$. It is, however, not safe to assume that all columns of $\boldsymbol{X}$ belong in $\boldsymbol{Z}$ and set $\boldsymbol{Z} = \boldsymbol{X}$. Overspecification of $\boldsymbol{Z}$ could negatively affect the performance of the variance model. This motivates the use of variable selection techniques to choose which columns of $\boldsymbol{X}$ to include in $\boldsymbol{Z}$.

Three feature selection methods are discussed in this section: a shrinkage penalty, heteroskedasticity testing, and best subset selection. The shrinkage penalty is proposed due to its being already built into some of the ALVMs. The heteroskedasticity testing approach is proposed due to its simplicity and computational efficiency. The best subset selection approach is proposed due to its meticulousness.

3.3.3.1 Feature Selection by a Shrinkage Penalty

L_2-norm and L_1-norm penalties on the parameters have already been considered as part of the polynomial ALVM (see (3.46)). While the motivation for the penalty was partly to shrink unnecessary higher-degree terms or cross-terms, it could also shrink all terms associated with a particular covariate that does not in fact influence the error variances $\boldsymbol{\omega}$. Hence, if one sets $\boldsymbol{Z} = \boldsymbol{X}$ and uses a penalised polynomial model, variable selection will arguably take care of itself. Indeed, the acronym LASSO coined by Tibshirani (1996) for L_1-norm-penalised regression stands for Least Absolute Shrinkage and Selection Operator, and was designed in part as a tool for feature selection.

An advantage of the shrinkage approach to feature selection is that, being built into the fitting of the ALVM, it does not require an additional model-building step. This is particularly significant given that, as was highlighted in §3.3.2, the fitting of training models in K-fold CV for tuning of hyperparameters must include all modelling steps. Thus, a pre-processing feature selection step in the polynomial and thin-plate spline methods would be computationally expensive, because it would have to be implemented separately on each of the K training folds.

The sparseness properties of the LASSO make it ideal for feature selection, since LASSO tends to shrink unimportant coefficients *to* zero. By contrast, RR tends only to shrink them *towards* zero, and thus does not truly 'deselect' unimportant features.

The penalty used in the thin-plate spline ALVM also offers some support for feature selection. In this case, it is not the parameter magnitudes that are penalised but the magnitude of second-order derivatives of the spline, the 'wiggliness.' In principle, if a particular covariate in $\boldsymbol{Z}$ does not contribute to the error variance, the magnitude of the spline's second derivative in that direction should be shrunk to 0, resulting in a straight line that could have an estimated gradient of 0.

The linear ALVM fitted by solving (3.41) does not have a penalty term. However, if it is desired to use an L_2- or L_1-norm shrinkage penalty on a linear ALVM, the penalised polynomial model can be used with degree $d = 1$.

Therefore, of the ALVMs proposed herein that are conducive to use in multiple linear regression, it is only the clustering ALVM for which a shrinkage approach to feature selection is infeasible.

If shrinkage is used for feature selection, the hyperparameter λ must still be tuned, using one of the methods discussed in §3.3.2.1.

3.3.3.2 Feature Selection by Heteroskedasticity Testing

This feature selection technique is a pre-processing step performed before fitting the variance model. The idea is to use a hypothesis test for heteroskedasticity to identify covariates in $\boldsymbol{X}$ that are, at some significance level α, implicated in heteroskedasticity. Particularly appropriate are the 'deflator'-type tests discussed in §2.1 (see especially Table 2.3), where the alternative hypothesis posits heteroskedasticity linked to a particular predictor variable. 'Auxiliary design' tests such as Breusch and Pagan's (1979) could also be used, provided that the auxiliary design matrix contains only one predictor variable. Previous empirical investigations of the power of different heteroskedasticity tests (e.g., Griffiths and Surekha 1986, Evans 1992, Lyon and Tsai 1996, Godfrey and Orme 1999, Adamec 2017, Uyanto 2019), as well as that undertaken in this study (see §5.1.1) may inform the choice of testing method.

The procedure runs like this:

1. Perform a heteroskedasticity test at significance level α with $\boldsymbol{X}_{\cdot 2}$ as the 'deflator' variable.[75] If the null hypothesis of homoskedasticity is rejected, include $\boldsymbol{X}_{\cdot 2}$ in $\boldsymbol{Z}$; otherwise, exclude it.
2. Repeat step (1) for $\boldsymbol{X}_{\cdot 3}, \boldsymbol{X}_{\cdot 4}, \ldots, \boldsymbol{X}_{\cdot p}$.

Even if a powerful heteroskedasticity test is used, there remains a trade-off between the risks of Type I and Type II errors. In this case, a Type I error results in overspecification (inclusion of a feature that is *not* related to the error variance(s)) while a Type II error results in underspecification (exclusion of a feature that *is* related to the error variance(s)). The choice of significance level α is thus another important consideration with this feature selection technique; in effect, α is a hyperparameter. Intuitively, the cost to variance estimation of underspecification seems higher than that of overspecification. Thus, a significance level higher than the typical 0.05 used in inference is suggested, such as 0.1, especially when n is small and power is consequently low. It should also be borne in mind that the family-wise Type I error rate increases with p, which may motivate a Bonferroni correction.

This feature selection technique can be used with any of the ALVMs introduced earlier. However, it may be computationally intensive to use with a penalised polynomial or thin-plate spline model where λ is being tuned using CV. The aforementioned requirement to conduct feature selection on each training fold would not only increase computation time; the power of the heteroskedasticity test will be lower with the training folds than on the full data set, due to their smaller number of observations. Thus, if the same significance level is used throughout, one can expect under-specification to occur more frequently in the CV procedure than on the full data set.[76] To avoid these complications, it seems best to rely on the shrinkage approach for models requiring hyperparameter tuning using CV.

3.3.3.3 Feature Selection Using Best Subset Selection

Best Subset Selection (BSS) is a classical feature selection method in statistics (Hastie et al. 2020). It entails finding the subset of candidate features that minimises some loss function.

Several metrics were considered to use for BSS for feature selection in the ALVMs, including Mallows' C_p (discussed in Hastie et al. 2009) and the .632 estimator (Efron and Tibshirani 1993) and its improved

[75] This assumes that $\boldsymbol{X}_{\cdot 1}$ is a column of ones; otherwise one would start with $\boldsymbol{X}_{\cdot 1}$.

[76] One could perhaps adjust the significance level upward to compensate for this; but it is unclear by how much to adjust it, and this technically violates the principle that pre-processing steps done when training the main model must be performed identically when training the model on each CV fold.

.632+ version (Efron and Tibshirani 1997). However, the loss functions already introduced for hyperparameter tuning purposes—namely, the CV error metric (3.77) and the QGCV error metric (3.86)—proved in preliminary simulations to work just as well, and (particularly QGCV) faster to compute. Note that BSS using K-fold CV with a model that already uses CV for hyperparameter tuning (such as the penalised polynomial ALVMs or thin-plate spline ALVM) would be complicated, as a nested CV step would need to be performed to choose λ for each feature subset. Thus, to conserve on computation time, BSS is only considered in conjunction with the linear or clustering ALVM, and in the latter case only when the number of clusters n_c is chosen using an elbow method.

Another issue with BSS is that, if there are $p-1$ candidate features (p being the number of columns in $\boldsymbol{X}$, including the intercept), there are 2^{p-1} candidate subsets to consider (assuming that an intercept will be included unconditionally; otherwise 2^p). An exhaustive search of these subsets quickly becomes computationally prohibitive as p increases. Thus a greedy algorithm, akin to forward selection, may be used if p is too large to run all subsets. Such an algorithm can be outlined as follows:

1. Fit the null model (with only an intercept) and evaluate the loss function.
2. Fit all models consisting of one feature and an intercept, and identify the model (and associated feature subset) that minimises the loss function.
3. If the minimal loss function value is less than that of the null model, select this feature. Otherwise, adopt the null model and break out of the procedure.
4. Repeat steps (2) and (3), but each time comparing all models with $k+1$ features to the best k-feature model, until the best $k+1$-feature model does not improve on the best k-feature model.

3.3.4 Statistical Results on the Variance Estimator

Liew (1976) derives the variance-covariance matrix of the ICLS estimator, conditional on knowing which inequality constraints are met at the boundary and which are not. Toker et al. (2013) extend this result to ICRR. However, as Geweke (1986) and Knottnerus (2016) point out, one generally does *not* know ahead of time which constraints will be met at the boundary, rendering this variance-covariance matrix approach potentially seriously misleading in practice. Geweke (1986) proposes a Bayesian approach to exact inference on inequality-constrained linear models, and Knottnerus (2016) proposes an approximation for the variance-covariance matrix of the ICLS estimator based on censored and truncated normal distributions. Unfortunately, both of their methods are valid only under the assumption of normality on the constrained linear model, which is not a reasonable assumption for the ALVM errors. It has been shown in §3.1.1, for instance, that under A1-A5 the marginal distributions of the squared OLS residuals e_i^2, conditioning on $\boldsymbol{X}$, are Gamma distributions.

Knottnerus (2016) shows that the ICLS estimator is biased (regardless of distributional assumptions) but that the variance of the ICLS estimator is less than or equal to that of the OLS estimator. Thus, in terms of mean squared error there is a bias-variance trade-off between the two estimators. Even if the trade-off favours the OLS estimator, however, this would still be a price worth paying to avoid negative variance estimates that cannot be used for HCCME or FWLS purposes.

Since no analytical results are possible for the standard errors or distribution of the ICLS or ICRR estimators, bootstrap methods are the most promising approach to obtaining interval estimates for the error variances based on ALVMs. Such bootstrap methods will be discussed further in §3.4.

Seber and Wild (2003), following McCullagh (1983), discuss statistical properties of quasi-likelihood estimators, including a central limit theorem that allows for asymptotic inference on, or approximate interval estimation of, the parameters $\boldsymbol{\gamma}$. Specifically,

$$\sqrt{n}\left(\hat{\boldsymbol{\gamma}}-\boldsymbol{\gamma}\right) \xrightarrow{D} \boldsymbol{N}\left(\mathbf{0}, \left[n^{-1}\mathfrak{I}(\boldsymbol{\gamma})\right]^{-1}\right), \tag{3.90}$$

where $\mathfrak{I}(\boldsymbol{\gamma}) = \boldsymbol{F}_{\bullet}'(\boldsymbol{\gamma})\boldsymbol{V}^{-1}(\boldsymbol{\gamma})\boldsymbol{F}_{\bullet}(\boldsymbol{\gamma})$, and the latter notation is as introduced previously in §3.3.1.4. Of course, $\mathfrak{I}(\boldsymbol{\gamma})$ would in practice need to be replaced with an estimator, effectively 'studentising' the approximation. These statistical results will play no further role in this thesis, but could be a starting point for further research.

3.3.5 Use of an Auxiliary Variance Model for Feasible Weighted Least Squares and Standard Error Estimation

An immediate consequence of fitting any of the ALVMs or ANLVMs discussed in this section is that one is able to obtain point estimates $\hat{\boldsymbol{\omega}}$ of the error variances $\boldsymbol{\omega}$, either by $\hat{\boldsymbol{\omega}} = \boldsymbol{L}\hat{\boldsymbol{\gamma}}$ (for an ALVM) or by $\hat{\boldsymbol{\omega}} = \boldsymbol{g}(\boldsymbol{Z}; \hat{\boldsymbol{\gamma}})$ (for an ANLVM).

An estimate of $\boldsymbol{\omega}$ immediately leads to an estimate of $\boldsymbol{\Omega} = \operatorname{diag}\{\boldsymbol{\omega}\}$, which can be used to compute a FWLS estimate of $\boldsymbol{\beta}$ (as discussed in §2.2.1), with weights $\boldsymbol{W} = \hat{\boldsymbol{\Omega}}^{-1}$. The effectiveness of such a FWLS estimator can be measured using a MSE metric, as discussed below in §5.2.2.3.

Equally, an estimate of $\boldsymbol{\Omega}$ can be plugged into (1.6) in the manner of an HCCME (as discussed in §2.3) and thus used to compute a quasi-t statistic for inference on individual linear model parameters β_j, $j = 1, 2, \ldots, p$. The performance of such a quasi-t-test depends on how close the resulting standard error estimate in the denominator is to the true standard error of $\hat{\beta}_j$. This motivates another MSE metric, as discussed below in §5.2.2.4.

3.4 Constructing Bootstrap Confidence Intervals for the Error Variances

Practitioners using the ALVMs and ANLVMs may wish to have Confidence Interval (CI) estimates of the individual error variances ω_i, $i = 1, 2, \ldots, n$. Because of the multivariate nature of the estimation problem, it is not advisable to obtain CIs for the elements of $\boldsymbol{\gamma}$ and then map the vector of lower limits and upper limits respectively onto $\boldsymbol{\omega}$ using the relation $\boldsymbol{\omega} = \boldsymbol{L}\boldsymbol{\gamma}$. Instead, interval estimates will be obtained for the individual error variances ω_i directly.

Bootstrap methods are an ingenious technique for obtaining approximate CIs with good properties without requiring analytical results on the distribution of the estimator or obtaining additional data.

3.4.1 Nonparametric Bootstrap Resampling Methods Suitable for Heteroskedastic Linear Regression Models

In the context of heteroskedastic linear regression, parametric bootstrap methods are not appropriate if the true form of the heteroskedastic function $g(\cdot)$ is unknown. Efron and Tibshirani (1993) and Chernick (2008) discuss two nonparametric bootstrap methods for linear regression models that they call *bootstrapping residuals* and *bootstrapping pairs*.

Both methods entail drawing a random sample of size n with replacement and with uniform probability from $\{1, 2, \ldots, n\}$, B times. Let $\{b\}$ denote the bth set of sampled indices, $b = 1, 2, \ldots, B$, and let $\boldsymbol{y}_{\{b\}}$, $\boldsymbol{X}_{\{b\}}$, and $\boldsymbol{e}_{\{b\}}$ denote the resampled response vector, design matrix, and OLS residual vector corresponding to the observations with indices $\{b\}$.

Bootstrapping residuals proceeds by computing bootstrap responses $\boldsymbol{y}^{(b)}$ from the original design matrix $\boldsymbol{X}$, the original OLS parameter estimate $\hat{\boldsymbol{\beta}}$, and the resampled residual vector $\boldsymbol{e}_{\{b\}}$, as per (3.91):

$$\boldsymbol{y}^{(b)} = \boldsymbol{X}\hat{\boldsymbol{\beta}} + \boldsymbol{e}_{\{b\}}, b = 1, 2, \ldots, B. \tag{3.91}$$

The bootstrap OLS parameter estimator, fitted values vector and residual vector are then

$$\hat{\boldsymbol{\beta}}^{(b)} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{y}^{(b)}, \tag{3.92}$$

$$\hat{\boldsymbol{y}}^{(b)} = \boldsymbol{X}\hat{\boldsymbol{\beta}}^{(b)}, \tag{3.93}$$

and

$$\boldsymbol{e}^{(b)} = \boldsymbol{y}^{(b)} - \hat{\boldsymbol{y}}^{(b)}. \tag{3.94}$$

Chernick (2008, p. 79) states that 'Bootstrapping the residuals requires that the residuals be independent and identically distributed (or at least exchangeable).' Technically, the OLS residuals are *never* independent and identically distributed, even under A1-A5 (see §1.1.7.2). More to the point, when heteroskedasticity is present and is related to at least one of the covariates in $\boldsymbol{X}$, the residuals are not exchangeable; each residual e_i must remain associated with the corresponding covariate observation $\boldsymbol{X}_{i\cdot}$. Thus, bootstrapping residuals can be ruled out for purposes of estimating standard errors of heteroskedastic variance estimators.

Bootstrapping pairs simply entails resampling the response-covariate pairs so that the bootstrap responses are $\boldsymbol{y}^{(b)} = \boldsymbol{y}_{\{b\}}$ and the bootstrap design matrix is $\boldsymbol{X}^{(b)} = \boldsymbol{X}_{\{b\}}$. The bootstrap OLS parameter estimator, fitted values, and residuals are then computed as

$$\hat{\boldsymbol{\beta}}^{(b)} = (\boldsymbol{X}^{(b)\prime}\boldsymbol{X}^{(b)})^{-1}\boldsymbol{X}^{(b)\prime}\boldsymbol{y}^{(b)}, \tag{3.95}$$

$$\hat{\boldsymbol{y}}^{(b)} = \boldsymbol{X}^{(b)}\hat{\boldsymbol{\beta}}^{(b)}, \tag{3.96}$$

and

$$\boldsymbol{e}^{(b)} = \boldsymbol{y}^{(b)} - \hat{\boldsymbol{y}}^{(b)}. \tag{3.97}$$

The bootstrap annihilator matrix can also be computed as $\boldsymbol{M}^{(b)} = \boldsymbol{I}_n - \boldsymbol{X}^{(b)}(\boldsymbol{X}^{(b)\prime}\boldsymbol{X}^{(b)})\boldsymbol{X}^{(b)\prime}$.

Chernick (2008) notes that some statisticians consider bootstrapping pairs to be philosophically inappropriate in that the predictor observations are being treated as fixed and yet are being resampled. However, from a practical point of view, bootstrapping pairs is less sensitive to model assumptions and leads to standard error estimates with good properties (Efron and Tibshirani 1993). This method preserves the link between the error variances (captured in the observed response) and the predictor variables, if present, and is therefore appropriate under heteroskedasticity.

The bootstrap ALVM becomes

$$\boldsymbol{e}^{(b)} \circ \boldsymbol{e}^{(b)} = \left(\boldsymbol{M}^{(b)} \circ \boldsymbol{M}^{(b)}\right) \boldsymbol{L}^{(b)}\boldsymbol{\gamma} + \boldsymbol{u}^{(b)}, \tag{3.98}$$

where $\boldsymbol{L}^{(b)}$ is a linear predictor matrix generated from $\boldsymbol{Z}^{(b)}$ (which may be formed from a subset of $\boldsymbol{X}^{(b)}$ or resampled from $\boldsymbol{Z}$). In the nonlinear case the model would be constructed analogously.

By fitting the bootstrap auxiliary variance models one obtains parameter estimates $\hat{\boldsymbol{\gamma}}^{(b)}$, $b = 1, 2, \ldots, B$.

An alternative to bootstrapping pairs is to use the *wild bootstrap* technique that was introduced in §2.3.10. For example, define $\boldsymbol{r}^{(b)}, b = 1, 2, \ldots, B$, to be a random vector drawn independently from a distribution with zero mean and unit variance. Then compute bootstrap responses,

$$\boldsymbol{y}^{(b)} = \boldsymbol{X}\hat{\boldsymbol{\beta}} + [\boldsymbol{F}(\boldsymbol{e})]\,\boldsymbol{r}^{(b)}, \tag{3.99}$$

where $\boldsymbol{F}(\boldsymbol{e})$ is a diagonal matrix with diagonal $[f_1(e_1), f_2(e_2), \ldots, f_n(e_n)]'$ and $f_i(e_i)$ is some function of the ith OLS residual depending on no other stochastic variables. The OLS parameter estimator $\hat{\boldsymbol{\beta}}^{(b)}$, fitted values $\hat{\boldsymbol{y}}^{(b)}$, and residuals $\boldsymbol{e}^{(b)}$ can then be computed using (3.92)-(3.94).

Cribari-Neto and Zarkos (1999) use $f_i(e_i) = e_i(1 - h_{ii})^{-1/2}$, but it is proposed here to simply use the identity transformation $f_i(e_i) = e_i$ due to the straightforward expression for $\mathrm{E}\left(\boldsymbol{e}^{(b)} \circ \boldsymbol{e}^{(b)}\right)$ that results, namely

$$\mathrm{E}\left(\boldsymbol{e}^{(b)} \circ \boldsymbol{e}^{(b)}\right) = (\boldsymbol{M} \circ \boldsymbol{M})\,(\boldsymbol{M} \circ \boldsymbol{M})\,\boldsymbol{\omega}, \tag{3.100}$$

which enables construction of the ALVM

$$\boldsymbol{e}^{(b)} \circ \boldsymbol{e}^{(b)} = (\boldsymbol{M} \circ \boldsymbol{M})\,(\boldsymbol{M} \circ \boldsymbol{M})\,\boldsymbol{L}\boldsymbol{\gamma} + \boldsymbol{u}^{(b)}. \tag{3.101}$$

This takes the same form as the original model equation (3.34) but with an extra matrix factor of $(\boldsymbol{M} \circ \boldsymbol{M})$ on the conditional mean term. A derivation of (3.100) is given in Appendix C.4.

A practical question that must be answered is whether it is necessary to run feature selection and retune the hyperparameter λ (where applicable) as part of the ALVM fitting procedure with each bootstrapped regression model. Retuning λ is very computationally expensive, so it would be much faster to use the λ value tuned on the full sample for every bootstrap ALVM. But is this approach reasonable? Wang and Wahba (1994) describe a bootstrap procedure for smoothing spline models that *does* entail re-tuning λ for each bootstrap sample, and this is also recommended by Kauermann et al. (2009). On the other hand, Sartori (2010) and Laurin et al. (2016) describe bootstrap procedures for penalised models where the parameter value optimised from the full sample is reused in all the bootstrap samples. Clearly, the optimal settings for the full data are not necessarily going to be optimal for each bootstrap sample, so some additional variation is introduced by the second approach. It comes down to a trade-off between computational cost and precision.

3.4.2 Computation of Bootstrap Confidence Intervals

3.4.2.1 Naïve Normal Interval

A simple, naïve way of obtaining a bootstrap confidence interval for ω_i, $i = 1, 2, \ldots, n$, is the normal interval,

$$\hat{\omega}_i \pm z_{\alpha/2}\widehat{\mathrm{SE}}_{\mathrm{boot}}(\hat{\omega}_i), \tag{3.102}$$

where $\widehat{\mathrm{SE}}_{\mathrm{boot}}(\hat{\omega}_i)$ is the empirical standard deviation of the bootstrap ALVM estimates $\hat{\omega}_i^{(b)}, b = 1, 2, \ldots, B$. While simple to compute, this interval will not perform well in cases where the distribution of $\hat{\omega}_i$ is strongly skewed, platykurtic, or leptokurtic. Moreover, one should actually 'correct' the lower limit of the interval defined in (3.102) by taking $\max\left\{0, \hat{\omega}_i - z_{\alpha/2}\widehat{\mathrm{SE}}_{\mathrm{boot}}(\hat{\omega}_i)\right\}$, since 0 forms a lower bound for the error variances.

3.4.2.2 Percentile Interval

Another way of obtaining an approximate $(1-\alpha)100\%$ confidence interval for ω_i, $i = 1, 2, \ldots, n$, is the percentile interval method. If $\hat{G}_i$ is the empirical CDF of $\hat{\omega}_i$, then the $(1-\alpha)100\%$ percentile interval for ω_i is defined as

$$[\hat{\omega}_{i,\mathrm{lo}}, \hat{\omega}_{i,\mathrm{up}}] = \left[\hat{G}_i^{-1}(\alpha/2), \hat{G}_i^{-1}(1-\alpha/2)\right]. \tag{3.103}$$

The bootstrap approximation of the percentile interval is

$$[\hat{\omega}_{i,\mathrm{lo}}, \hat{\omega}_{i,\mathrm{up}}] \approx \left[\hat{\omega}_{i,(\alpha/2)}^{\{B\}}, \hat{\omega}_{i,(1-\alpha/2)}^{\{B\}}\right], \tag{3.104}$$

where $\hat{\omega}_{i,(p)}^{B}$ is the lower p-quantile of the bootstrap estimates $\left\{\hat{\omega}_i^{(b)}\right\}$, $b = 1, 2, \ldots, B$.

The percentile interval method is first-order correct, whereas modifications can be made to the quantile probabilities to arrive at a second-order correct method (Hesterberg 2011).

3.4.2.3 Modifications to the Percentile Interval

Two such modification techniques are the Bias-Corrected and accelerated (BCa) technique (Efron and Tibshirani 1993) and the expansion technique (Hesterberg 1999). These techniques are not mutually exclusive; it is possible to use them separately or together.

The BCa method modifies the quantile probabilities in (3.104) as follows, for $i = 1, 2, \ldots, n$:

$$[\hat{\omega}_{i,\mathrm{lo}}, \hat{\omega}_{i,\mathrm{up}}] \approx \left[\hat{\omega}_{i,(\alpha_{1,i})}^{\{B\}}, \hat{\omega}_{i,(\alpha_{2,i})}^{\{B\}}\right], \tag{3.105}$$

where

$$\alpha_{1,i} = \Phi\left(\hat{z}_{0,i} + \frac{\hat{z}_{0,i} + z_{(\alpha/2)}}{1 - \hat{a}_i(\hat{z}_{0,i} + z_{(\alpha/2)})}\right)$$

and

$$\alpha_{2,i} = \Phi\left(\hat{z}_{0,i} + \frac{\hat{z}_{0,i} + z_{(1-\alpha/2)}}{1 - \hat{a}_i(\hat{z}_{0,i} + z_{(1-\alpha/2)})}\right). \tag{3.106}$$

Here, $\Phi(\cdot)$ is the CDF of the standard normal distribution, $z_{(p)}$ is the lower p-quantile of the standard normal distribution, $\hat{z}_{0,i}$ is the bias-correction, and $\hat{a}$ is the acceleration statistic. If $\hat{a}$ and $\hat{z}_{0,i}$ are 0, then (3.105) reduces to (3.104).

The bias-correction $\hat{z}_{0,i}$ is computed by comparing the bootstrap estimates of ω_i to the full-sample estimate, using

$$\hat{z}_{0,i} = \Phi^{-1}\left(\frac{\#\left\{\hat{\omega}_i^{(b)} < \hat{\omega}_i\right\}}{B}\right). \tag{3.107}$$

The acceleration statistic $\hat{a}$ estimates the rate of change of $\mathrm{SE}(\hat{\omega}_i)$ with respect to the true parameter ω_i. Efron and Tibshirani (1993) suggest using the formula

$$\hat{a}_i = \frac{\sum_{j=1}^{n}\left(\hat{\omega}_{i,(\cdot)} - \hat{\omega}_{i,(j)}\right)^3}{6\left[\sum_{j=1}^{n}\left(\hat{\omega}_{i,(\cdot)} - \hat{\omega}_{i,(j)}\right)^2\right]^{3/2}}, \tag{3.108}$$

where $\hat{\omega}_{i,(j)}$ is the leave-one-out (jackknife) estimate of ω_i based on the original sample with the jth observation omitted, and $\hat{\omega}_{i,(\cdot)} = n^{-1}\sum_{j=1}^{n}\hat{\omega}_{i,(j)}$. Like the percentile interval, the BCa interval is transformation-respecting for monotonic transformations (Efron and Tibshirani 1993).

Efron and Tibshirani (1993) note that at least $B = 1000$ replications are required to sufficiently reduce the MC sampling error in the BCa interval estimates. Efron and Tibshirani (1993) propose another improvement on the percentile interval called the Approximate Bootstrap Confidence (ABC) method that is computationally less expensive than the BCa method. It approximates the bootstrap random sampling results by Taylor series expansions. However, this method requires that the statistical estimator be a smooth function of the data, which is not the case here due to the inequality constraints on $\boldsymbol{\omega}$.

The expansion technique (Hesterberg 1999, 2015) is a simple coverage-level adjustment that can be applied to a percentile interval or a BCa interval. Suppose—leaving aside bootstrap notation for the moment—that one wanted to obtain a $(1-\alpha)100\%$ CI for the mean μ of a normal population with unknown variance σ^2. Let $\bar{x}$ be the mean of a random sample of size n and let $\hat{\sigma}^2 = n^{-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$ and $s^2 = (n-1)^{-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$ be the ML estimator of σ^2 and the sample variance, respectively. A plausible approximate CI for μ would be $\bar{x} \pm z_{\alpha/2}\hat{\sigma}$, but replacing $z_{\alpha/2}$ with $t_{\alpha/2,n-1}$ and $\hat{\sigma}$ with s would give an exact CI. The first interval can be transformed into the second by multiplying its margin of error (half-width) by $a_{\alpha/2,n} = \frac{t_{\alpha/2,n-1}}{z_{\alpha/2}}\frac{s}{\hat{\sigma}} = \frac{t_{\alpha/2,n-1}}{z_{\alpha/2}}\sqrt{\frac{n}{n-1}}$.

The logic of the expansion technique is that, if the bootstrap statistics are approximately normally distributed, a similar improvement can be achieved by applying such an adjustment to the percentile (or BCa) interval. Since, however, it would not be transformation-invariant to multiply the limits of the percentile interval by $a_{\alpha/2,n}$, a modified probability α' can be found such that $z_{\alpha'/2} = t_{\alpha/2,n-1}\sqrt{\frac{n}{n-1}}$, namely,

$$\alpha'/2 = \Phi\left(t_{\alpha/2,n-1}\sqrt{\frac{n}{n-1}}\right). \tag{3.109}$$

Hence, the expansion technique applied to the percentile interval involves replacing the probability $\alpha/2$ in (3.104) with $\alpha'/2$ from (3.109). The expansion technique applied to the BCa interval involves changing the probabilities $\alpha_{1,i}$ and $\alpha_{2,i}$ to $\alpha'_{1,i}$ and $\alpha'_{2,i}$, respectively, by replacing $\alpha/2$ in (3.106) with $\alpha'/2$ from (3.109).

Of course, the premise that the bootstrap statistics are approximately normally distributed may not hold, but Hesterberg (2015) points out that even for heavy-tailed distributions, the expansionary adjustment is still in the right direction.

3.4.3 Dependent vs. Independent Interval Estimates

If the percentile interval (3.104) or BCa interval (3.105) is computed elementwise for ω_i, $i = 1, 2, \ldots, n$, the impression may be given that there is approximate $(1-\alpha)100\%$ confidence that all of the ω_i fall within their respective intervals. This is not the case, however, because the point estimates $\hat{\omega}_i, \hat{\omega}_j$ and bootstrap estimates $\hat{\omega}_i^{(b)}, \hat{\omega}_j^{(b)}$ (for any $i \neq j$) are obviously correlated, since they are computed from a single parameter estimate $\hat{\boldsymbol{\gamma}}$ (or $\hat{\boldsymbol{\gamma}}^{(b)}$) calculated from the same data set. Due to these dependencies, the CIs are only valid elementwise (approximately).

One means of obtaining CIs (whether percentile or BCa) that are (approximately) familywise-valid is to independently draw n separate sets of B bootstrap samples (whether using the pairs or wild bootstrap technique). The individual bootstrap variance estimates $\hat{\omega}_i^{(b)}$ (and the $\hat{z}_{0,i}$, in the case of BCa) will thus be computed from an independently drawn bootstrap data set for each $i = 1, 2, \ldots, n$—albeit drawn from the same original data set (which mimics the 'population'). This method of achieving approximately familywise-valid

CIs comes at a formidable computational cost, as it entails fitting nB bootstrap linear regressions and then fitting nB ALVMs, as opposed to just B. This will only be feasible if n and B are relatively small, unless a faster ALVM such as the linear or clustering model is used.

3.4.4 Constructing a Bootstrap Confidence Region for the Error Variance Vector

Although perhaps of less value to the applied practitioner, it may be of interest to obtain an approximate $(1-\alpha)100\%$ n-dimensional confidence *region* for $\boldsymbol{\omega}$. Olive (2018) proposes a multivariate extension of the bootstrap percentile interval to compute an approximate hyperellipsoidal confidence region using the Mahalanobis distance metric $\mathrm{MH}^2(\boldsymbol{X}_{i\cdot}; \bar{\boldsymbol{\mathcal{X}}}, \boldsymbol{S})$ (Mahalanobis 1936), which was introduced previously in §1.1.10.1 (in a different context, and with slightly different notation).

Olive's (2018) bootstrap confidence region, applied to the present problem, is a set of points $\boldsymbol{w}$ satisfying,

$$\left\{\boldsymbol{w} : \mathrm{MH}^2\left(\boldsymbol{w}; \hat{\boldsymbol{\omega}}, \boldsymbol{S}^{\star}_{\hat{\boldsymbol{\omega}}}\right) \le D^2_{(U_B)}\right\}, \tag{3.110}$$

where $\hat{\boldsymbol{\omega}}$ is the full-sample ALVM estimate of $\boldsymbol{\omega}$, $\boldsymbol{S}^{\star}_{\hat{\boldsymbol{\omega}}}$ is the $n \times n$ empirical covariance matrix of the bootstrap estimates $\hat{\boldsymbol{\omega}}^{(1)}, \hat{\boldsymbol{\omega}}^{(2)}, \ldots, \hat{\boldsymbol{\omega}}^{(B)}$, and $D^2_{(U_B)}$ is a Mahalanobis distance empirical quantile computed as follows. Set

$$q_B = \begin{cases} \min\left(1-\alpha+0.05, 1-\alpha+n/B\right) & \alpha > 0.1 \\ \min\left(1-\alpha/2, 1-\alpha+10\alpha n/B\right) & \alpha \le 0.1 \end{cases}. \tag{3.111}$$

If $1-\alpha < 0.999$ and $q_B < 1-\alpha+0.001$, Olive (2018) proposes changing q_B to $1-\alpha$. Then, $D^2_{(U_B)}$ is the $U_B = \lceil B q_B \rceil$th order statistic of the squared Mahalanobis distances of the bootstrap variance vector estimates, $\mathrm{MH}^2\left(\hat{\boldsymbol{\omega}}^{(1)}; \hat{\boldsymbol{\omega}}, \boldsymbol{S}^{\star}_{\hat{\boldsymbol{\omega}}}\right), \mathrm{MH}^2\left(\hat{\boldsymbol{\omega}}^{(2)}; \hat{\boldsymbol{\omega}}, \boldsymbol{S}^{\star}_{\hat{\boldsymbol{\omega}}}\right), \ldots, \mathrm{MH}^2\left(\hat{\boldsymbol{\omega}}^{(B)}; \hat{\boldsymbol{\omega}}, \boldsymbol{S}^{\star}_{\hat{\boldsymbol{\omega}}}\right)$.

Once computed, the confidence region is defined by its hyperellipsoidal centroid, $\hat{\boldsymbol{\omega}}$, by the scalar $D^2_{(U_B)}$, representing the squared Mahalanobis distance from the hyperellipsoidal centroid to the boundary, and by the covariance matrix estimate computed from the bootstrap data, $\boldsymbol{S}^{\star}_{\hat{\boldsymbol{\omega}}}$. The squared Mahalanobis distance $\mathrm{MH}^2\left(\boldsymbol{w}; \hat{\boldsymbol{\omega}}, \boldsymbol{S}^{\star}_{\hat{\boldsymbol{\omega}}}\right)$ of any point $\boldsymbol{w}$ can then be computed and compared to $D^2_{(U_B)}$ to determine whether it falls within the approximate $(1-\alpha)100\%$ confidence region. One of the drawbacks of this confidence region method is that it is n-dimensional and the 'curse of dimensionality' will loom large unless n is small.

3.5 A New Heteroskedasticity Test

A test of the null hypothesis of homoskedasticity (A2) can be constructed based on a fitted ALVM (or ANLVM) as follows. Recall that, under A2, the expectation of the ith squared OLS residual is given by (1.11). Under heteroskedasticity, the expectation of the ith squared OLS residual is given by (1.15). Consider the quotient Q given in (3.112):

$$Q = \frac{\displaystyle\sum_{i=1}^{n}\left(e_i^2 - \omega m_{ii}\right)^2}{\displaystyle\sum_{i=1}^{n}\left(e_i^2 - \sum_{k=1}^{n}\omega_i m_{ik}^2\right)^2}. \tag{3.112}$$

Under A2, where $\omega_i = \omega, i = 1, 2, \ldots, n$, the denominator simplifies to the numerator and $Q = 1$. The strategy of the heteroskedasticity test entails replacing ω in the numerator of (3.112) with its unbiased (under A2) estimate, $\hat{\omega}_{\mathrm{ub}}$, from (1.5), and the ω_i in the denominator of (3.112) with the estimates $\hat{\omega}_i$ from an ALVM or ANLVM:

$$\hat{Q} = \frac{\displaystyle\sum_{i=1}^{n}\left(e_i^2 - \hat{\omega}_{\mathrm{ub}} m_{ii}\right)^2}{\displaystyle\sum_{i=1}^{n}\left(e_i^2 - \sum_{k=1}^{n}\hat{\omega}_i m_{ik}^2\right)^2}. \tag{3.113}$$

Under homoskedasticity, $\hat{\omega}_{\text{ub}}$ will tend to be a better estimator of ω than the ALVM/ANLVM estimators $\hat{\omega}_i$ are. Thus, the numerator of (3.113) will tend to be smaller than the denominator, and $\hat{Q}$ will tend to take on values less than 1. Under heteroskedasticity, $\hat{\omega}_{\text{ub}} m_{ii}$ will be a less effective estimator of $\mathrm{E}(e_i^2)$, and—provided that the ω_i are estimated better by the ALVM or ANLVM than by the homoskedastic estimator—$\hat{Q}$ will tend to take on values larger than 1. Admittedly, the test statistic is constructed on *ad hoc* intuition rather than any firm theoretical grounds. Moreover, the power of the test will depend on how well the ω_i are modelled by the particular choice of auxiliary variance model.

The test is easily adapted to be either of the 'deflator' type (as that term was used in §2.1), by fitting the ALVM/ANLVM using only one covariate, or of the omnibus variety, by fitting the auxiliary variance model using all available information. Feature selection could be incorporated into the fitting of the variance model, but it would be circular logic to use heteroskedasticity testing as the feature selection method (as discussed in §3.3.3) if the purpose of fitting the auxiliary variance model is to use the resulting variance estimates in a heteroskedasticity test. Rather, the covariates included in the ALVM should be chosen based on assumptions about the kind of heteroskedasticity posited under the alternative hypothesis.

What of the null distribution of $\hat{Q}$? (3.113) has certain resemblances to a ratio of two variances, and at first glance, one thinks of dividing the numerator and denominator by appropriate degrees of freedom (perhaps $n-p$ and q, respectively) and comparing $\hat{Q}$ to F distribution quantiles. However, (3.113) is plainly not F-distributed, even approximately, because of the obvious dependency between the numerator and denominator. Moreover, any attempt to arrive at analytical results is complicated by the difficulty of obtaining analytical results on the ALVM or ANLVM variance estimators (see §3.3.4).

Hence, the problem of deriving an exact or asymptotic null distribution of (3.113) seems intractable. Fortunately, the bootstrap method of Godfrey and Orme (1999) (§2.1.23.2) and the MC method of Dufour et al. (2004) (§2.1.23.1) allow one to obtain approximate p-values from any heteroskedasticity test. Hence, a right-tailed test using the statistic $\hat{Q}$ can be performed in practice by computing p-values from one of these two computational methods.

The power of this new heteroskedasticity test will be investigated empirically in §5.1.1.

3.6 Chapter Summary

This chapter opened with some derivations of theoretical results on the squared OLS residuals under homoskedasticity (A1-A4) and under heteroskedasticity (A1, A3-A4). The expectation and variance-covariance matrix of $\boldsymbol{e} \circ \boldsymbol{e}$ were given for both cases (with and without the normality assumption A5), along with the marginal distributions of the e_i^2 under A5, which were shown to be Gamma distributions under both homoskedasticity and heteroskedasticity. The joint distribution of any pair of squared residuals e_i^2, e_j^2 under A5 was also derived, being an instance of Kibble's (1941) bivariate Gamma distribution. It was posited that the joint distribution of all n squared residuals under A5 is a multivariate Kibble Gamma distribution. This result could in principle be used to develop likelihood-based estimation and inference methods for the error variances. However, not only is the joint PDF difficult to compute but it is in fact degenerate. These complications motivated methods that make use of the moments of the squared OLS residuals but not their joint distribution.

A discussion of the bias of the e_i^2 as estimators of the ω_i followed. It was shown that the e_i^2 have a strictly negative bias when treated as estimators of the homoskedastic error variance ω. However, under heteroskedasticity the e_i^2 can be either negatively or positively biased, depending on the relative magnitudes of the ω_i as well as the leverage structure of the design matrix. That e_i^2 is a biased estimator of ω_i means that it may be problematic to use e_i^2 as a proxy for ω_i in a modelling approach to FWLS, such as those discussed in §2.2.1. Moreover, the fact that some e_i^2 may be *positively* biased suggests an inherent problem with most existing HCCMEs, which entail multiplying the e_i^2 by some factor that is strictly greater than 1, which would increase a positive bias. This motivated a new approach to modelling of heteroskedastic error variances based on the true expectation of the e_i^2.

The first model introduced is called an Auxiliary Linear Variance Model (ALVM), with the general model equation

$$e_i^2 = \sum_{k=1}^{n} \omega_k m_{ik}^2 + u_i, i = 1, 2, \dots, n,$$

or, equivalently,

$$\boldsymbol{e} \circ \boldsymbol{e} = (\boldsymbol{M} \circ \boldsymbol{M})\,\boldsymbol{\omega} + \boldsymbol{u}.$$

This model is linear in the parameter $\boldsymbol{\omega}$, but entails estimating an n-vector of parameters from only n observations. It was thus proposed to reparametrise the model by assuming that the error variances ω_i are related to some covariates $\boldsymbol{Z}_{i\cdot}$ by a continuous, differentiable, positive real-valued function $g(\boldsymbol{Z}_{i\cdot};\boldsymbol{\gamma})$, where $\boldsymbol{\gamma}$ is a q-vector of parameters. Various strategies for choosing this unknown heteroskedastic function $g(\cdot)$ were proposed, including specifying its form explicitly by assumption, estimating it using a penalised polynomial or spline function, or a nonparametric approach involving agglomerative hierarchical clustering. In some cases, the reparametrised model is linear in $\boldsymbol{\gamma}$, and thus still an ALVM. In other cases, it is nonlinear in $\boldsymbol{\gamma}$, and thus an Auxiliary Nonlinear Variance Model (ANLVM).

Strategies for fitting the ALVMs and ANLVMs were proposed. In the former case, these involved Inequality-Constrained Least Squares (ICLS), Inequality-Constrained Ridge Regression (ICRR), or more generally, Quadratic Programming (QP). In the latter case, Maximum Quasi-Likelihood (MQL) estimation was the method of choice. The problems of tuning of hyperparameters (for models involving hyperparameters), and feature selection (to arrive at an appropriate specification of $\boldsymbol{Z}$) were also addressed in detail.

A brief discussion of the possibility of deriving statistical properties of the ALVM and ANLVM error variance estimators $\hat{\boldsymbol{\omega}}$ ensued, but was not very fruitful. As a result, the best option for producing interval estimates of the error variances ω_i based on an ALVM or ANLVM seems to be bootstrap methods. Therefore, two methods for nonparametric bootstrapping of heteroskedastic linear regression models—bootstrapping pairs and the wild bootstrap—were discussed, along with suitable methods for arriving at a bootstrap confidence interval for the ω_i or even a bootstrap confidence region for $\boldsymbol{\omega}$.

Finally, a new test of heteroskedasticity was proposed that makes use of the variance estimates from an ALVM or ANLVM as compared to the homoskedastic error variance estimator $\hat{\omega}_{\text{ub}}$.

A logical question arising at this point in the research is, 'do these new methods work?' A substantial set of Monte Carlo (MC) simulations will be used to answer this question empirically. However, before the methods can be evaluated empirically, they must be programmed. Therefore, the Results and Discussion chapter will be preceded by a chapter looking at an R package that has been created specially for this research study. The package includes functions that implement existing methods (especially heteroskedasticity tests and HCCMEs), but also—and more importantly, for this study—functions that implement the new models and methods introduced in this chapter.

4 Software Implementation: The skedastic R Package

Statistical methods are of limited practical value if they are inaccessible to practitioners. R (R Core Team 2022), an open-source language and environment for statistical software, provides statisticians with a means to deploy statistical methods to potential users thereof. It was noted in §2.5.1 that relatively few heteroskedasticity tests from the literature have been deployed in standard statistical software, including R.

To address this gap in making existing methods accessible to practitioners, and to make the new methods developed in §3 accessible to practitioners, a new R package has been developed as an integral part of this research project. The package is named **skedastic**. What follows is a description of the package's functionality. A link to the Comprehensive R Archive Network (CRAN) page for the package, from which the package documentation can be accessed and the package downloaded, can be found in Appendix D.

Section 4.1 describes 25 different functions, each of which implements a particular heteroskedasticity test from the literature. Functions that were written to support the implementation of some of the heteroskedasticity tests described in Chapter 2, but which are exported with the **skedastic** package due to potentially being useful in other applications, are described in §4.2. These functions represent a research contribution in their own right. Section 4.3 describes a single function, `hccme`, that implements the various HCCMEs described in §2.3. The three most important functions in the **skedastic** package, in terms of methodological novelty and contribution. These are `alvm.fit`, which implements an ALVM, `anlvm.fit`, which implements an ANLVM, and `avm.ci`, which computes bootstrap CIs for the error variances based on either of these auxiliary variance models, are explained in §4.4.

4.1 Functions That Implement Heteroskedasticity Tests

The **skedastic** package features 25 distinct functions that implement heteroskedasticity tests from the literature.[77] The naming convention used for these functions is based on the surname(s) of the author(s) of the publication where the test was first proposed. If the publication had one author, the function name is the author's surname in lowercase. If the publication had two authors, the function name is both their surnames separated by an underscore. For three or more authors, it is the first author's surname followed by `_etal`.[78]

This section focuses on the software implementation of the various heteroskedasticity tests in the **skedastic** package that was developed as one of the contributions of this research project. For a description of the statistical methods themselves, refer to §2.1. All of the functions that implement heteroskedasticity tests have two arguments in common. The first argument, `mainlm`, is a linear model object as generated by the `lm` function in the **stats** package. Alternatively, `mainlm` can be a `list` object containing two or three named objects: a response vector `y`, a design matrix `X`, and an OLS residual vector `e`. The rationale for allowing the user to specify `mainlm` as a list is that the model could then have been fitted using the faster `lm.fit` function rather than `lm`, thus reducing computation time.

The second argument that is common to all the heteroskedasticity testing functions is `statonly`. This is a logical argument that defaults to `FALSE`, but if true, causes the function to return only the value of the test statistic, and not (for example) to compute a p-value as well. This saves on computation time in instances where a p-value is not required (for example, when passing the test statistic value to Godfrey and Orme's (1999) method or Dufour et al.'s (2004) method). If `statonly` is `FALSE`, the value returned by the function is a list object of class `"htest"`. This is a class of lists widely used in hypothesis testing functions in R, and includes elements such as `statistic` (the value of the test statistic), `p.value` (the p-value), `parameter` (a relevant parameter, if applicable), `null.value` (the value of the parameter under the null hypothesis, if applicable), and `alternative`, a character (either `"greater"`, `"less"`, or `"two.sided"`) denoting the tailed-ness of the test.

Other arguments that are common to several (but not all) of the heteroskedasticity testing functions include `deflator`, `auxdesign`, and `restype`. In the case of the deflator-type heteroskedasticity tests (Goldfeld and Quandt (1965), Ramsey's (1969) BAMSET, etc.), the `deflator` argument specifies which column of the design matrix is the deflator, the covariate believed to be related to the error variances ω_i under the alternative hypothesis. In the case of the omnibus-type heteroskedasticity tests that use an auxiliary design matrix, denoted $\boldsymbol{Z}$ in §2.1 (Harvey (1976), Breusch and Pagan (1979), etc.), the `auxdesign` argument specifies the auxiliary

[77] Some functions implement more than one kind of test depending on the arguments. For instance, `goldfeld_quandt` runs either a parametric F test or a nonparametric peaks test, `evans_king` runs either a LM test or a GLS test, `li_yao` runs either an ALRT or a CVT, and `simonoff_tsai` runs either a MPLR test or a score test.

[78] An exception is `bamset`, the function that implements BAMSET (Ramsey 1969). Ramsey proposed a number of different tests in this article and gave the name BAMSET to his heteroskedasticity test, which has stuck.

design matrix. If the argument is set to `NA` (the default), the original design matrix $\boldsymbol{X}$ serves as the auxiliary design matrix.

In certain tests such as Goldfeld and Quandt (1965) and Horn (1981), the `restype` argument is used to control which residuals (OLS or BLUS) are used for the test. The remainder of this section provides details of the implementation of the functions in **skedastic** that implement these various existing heteroskedasticity tests.

4.1.1 Anscombe's Test (`anscombe`)

Anscombe's (1961) test has been described in §2.1.1 and is implemented by the function `anscombe`. The only special argument of this function is `studentise`, a logical variable that controls whether or not to apply Bickel's studentising modification. It defaults to `TRUE`.

4.1.2 BAMSET (`bamset`)

Ramsey's (1969) BAMSET, a Bartlett-type test of homogeneity of variances across k subsets (Bartlett 1937), has been described in §2.1.4 and is implemented by the function `bamset`. When calling `bamset`, one must specify the number of subsets k to be compared (using the argument `k`, which defaults to 3), and a logical `correct` argument determining whether a scaling correction should be made to improve the fit to the chi-squared distribution.

The function orders the observations by the deflator variable (specified by the `deflator` argument as discussed above), and then partitions them into k subsets that are as near as possible to equal in size. Since BAMSET uses BLUS residuals, computation of which yields only $n - p$ residuals from an original set of n observations, one must specify which p observations should 'sacrifice' their residuals. By setting the `omitatmargins` argument to `TRUE` when calling `bamset` (which is the default), one indicates that the p omitted observations are those nearest to the breaks between subsets (after the observations have been ordered by the deflator). The advantage of omitting at the margins between subsets is that, under the alternative hypothesis, this could accentuate the heterogeneity in variances between groups, and therefore increase the power of the test. If `omitatmargins` is set to `FALSE`, the function passes the `omit` argument to `blus` to determine which observations to sacrifice (see §4.2.1).

4.1.3 Bickel's Test (`bickel`)

Bickel's (1978) test has been described in §2.1.6 and is implemented by the `bickel` function. The model residuals used for Bickel's test can either be OLS residuals or residuals obtained from a robust regression using an M estimator (to further enhance the robustness of the method). The choice of model residuals is controlled by the `fitmethod` argument, which can be set to `"lm"` (the default) to use OLS residuals, or to `"rlm"`. In the latter case, the model is fitted using robust regression by calling the `rlm` function of **MASS** (Venables and Ripley 2002).

The user must specify a function $a(\cdot)$ to apply to the fitted values (using argument `a`) and a function $b(\cdot)$ to apply to the residuals (using argument `b`) to obtain a statistic based on an M estimator. The argument `a` can be any function that takes one argument and returns a `numeric` value of `length` 1. Alternatively, the argument `a` can be a `numeric` value of `length` 1, in which case the function is taken to be $a(\tau) = \tau^q$, where q is the value passed for argument `a`. The default $a(\cdot)$ function is $a(\tau) = \tau$, as suggested by Bickel (1978, p. 274), represented in R by `identity`. The $b(\cdot)$ function corresponds to the $\psi(\cdot)$ function used to construct M estimators and must be even, bounded, and twice-differentiable. The `bickel` function currently supports only two choices of $b(\cdot)$: Huber's function squared (as defined in (2.9)) and $b(\tau) = \tanh(\tau)^2$. These are called by passing either of the characters `"hubersq"` or `"tanhsq"`, respectively, for the `b` argument. Huber's function squared is the default, as suggested by Carroll and Ruppert (1981).

The `scale_invariant` argument controls whether to use the modified form of the test statistic proposed by Carroll and Ruppert (1981) to make Bickel's test statistic scale-invariant. If `scale_invariant` is set to `TRUE` (the default), the estimator used in `bickel` is $\tilde{\omega}^{1/2} = \text{median}\{|e_1|, |e_2|, \ldots, |e_n|\}/\Phi^{-1}(0.75)$ (where $\Phi(\cdot)$ is the standard normal CDF).

4.1.4 Breusch-Pagan Test (`breusch_pagan`)

Breusch and Pagan's (1979) test has been described in §2.1.8 and is implemented by the `breusch_pagan` function. The only special argument for this function is `koenker`, a logical variable that controls whether to apply Koenker's (1981) studentising modification. It defaults to `TRUE`.

4.1.5 Carapeto-Holt Test (`carapeto_holt`)

Carapeto and Holt's (2003) test has been described in §2.1.17 and is implemented by the function `carapeto_holt`. By default, this test assumes that the deflator variable is *negatively* associated with the error variance, and is right-tailed. If, as is more common with the deflator-type heteroskedasticity tests, the deflator is assumed to be positively associated with error variance, the `alternative` argument should be set to `"less"`, making the test left-tailed.

The proportion c of observations to remove (i.e., those with 'central' values of the deflator) is specified using the argument `prop_central`, with default value $\frac{1}{3}$ (rounded, if necessary, to ensure that s is an integer). The `group1prop` argument allows the user to specify the proportion of remaining observations allocated to the first subset, in case it is desired to use subsets of unequal size.

The `carapeto_holt` function computes p-values on the RQF-type test statistic using the `pRQF` function, discussed in §4.2.5. If the two-sided version of the test is used, p-values are then adjusted using the `twosidedpval` function, discussed in §4.2.2.

4.1.6 Cook-Weisberg Test (`cook_weisberg`)

Cook and Weisberg's (1983) test has been described in §2.1.11 and is implemented by the `cook_weisberg` function. The form of the heteroskedastic function $w(\cdot)$ assumed under the alternative hypothesis is specified using the `hetfun` argument. It can take one of three character values, `"add"`, `"mult"`, or `"logmult"`, corresponding to the three heteroskedastic models described in §2.1.11.[79]

4.1.7 Diblasi-Bowman Test (`diblasi_bowman`)

Diblasi and Bowman's (1997) test, implemented by the `diblasi_bowman` function, is one of the more complicated heteroskedasticity testing methods in the literature (see description in §2.1.16). The bandwidth parameter of the kernel function is specified using the `H` argument. This can be a scalar numeric value, in which case the bandwidth matrix is this scalar multiplied by the identity matrix. Alternatively, `H` can be a vector—in which case it is taken as the diagonal of a diagonal matrix—or as a matrix. It is intended that a future version of the package will incorporate automated tuning of the bandwidth (using CV, for instance).

The `diblasi_bowman` function calls the `adaptIntegrate` function from **cubature** (Narasimhan et al. 2020) to evaluate the required double integrals. The `ignorecov` argument is a logical which, if `TRUE`, leads the variance-covariance matrix of the transformed residual vector $\boldsymbol{s}$ to be treated as diagonal. This hugely reduces computation time and is the default setting.

The method used to compute the p-value is controlled using the `distmethod` argument, which can be set to `"moment.match"` (the default) or `"bootstrap"`, corresponding to the moment matching and bootstrap methods discussed in §2.1.16. If the bootstrap method is used, the `B` argument represents the number of bootstrap replications.

4.1.8 Dufour, Khalaf, Bernard, and Genest's Monte Carlo Test (`dufour_etal`)

Dufour et al.'s (2004) method has been described in §2.1.23.1 and is implemented by the `dufour_etal` function. The function takes as one of its arguments `hettest`, a character corresponding to the name of the function that implements one of the other heteroskedasticity tests implemented in the **skedastic** package (excluding Godfrey and Orme's (1999)). The function will call that auxiliary heteroskedasticity test function with the `statonly` argument set to `TRUE` in order to compute the value of its test statistic for each MC replication.

The number of MC replications R is specified using the `R` argument. This defaults to 1000, although Dufour et al. (2004) report that power improvements are not noticeable beyond $R = 99$ MC replications.

By default, each replication of the random error vector, $\boldsymbol{\epsilon}^{(j)}$, used in simulating the null distribution, is generated from $N(\mathbf{0}, \boldsymbol{I}_n)$. The distribution can, however, be changed using the `errorgen` argument. The `errorparam` argument can be used to pass distributional parameters to `errorgen` to ensure that the error distribution has zero mean.

[79] Note that this test produces identical results for the additive and multiplicative models.

The user must always specify the tailed-ness of the auxiliary heteroskedasticity test using the `alternative` argument (which defaults to `"greater"`), even if the auxiliary test function does not have an `alternative` argument.

4.1.9 Evans-King Tests (`evans_king`)

Evans and King's (1988) two heteroskedasticity tests have been described in §2.1.12 and are both implemented by the `evans_king` function. The test to apply is controlled using the `method` character argument: `"GLS"` for the GLS test or `"LM"` for the LM test. The hyperparameter $\lambda^{\star}$, controlling the degree of heteroskedasticity under the alternative hypothesis, is set using the `lambda_star` argument. It defaults to 5, since, according to the simulations conducted by Evans and King (1988), this results in the highest power.

Evans and King (1985, 1988) do not discuss in detail how to compute critical values or p-values for their test statistics. Thus, the `evans_king` function does more than merely implement their existing theory. It offers a significant new contribution, by using the `pRQF` function (described in §4.2.5) to compute p-values, since the test statistics are ratios of quadratic forms. Both tests are implemented in `evans_king` as left-tailed tests (as originally designed).

4.1.10 Glejser's Test (`glejser`)

The test implemented in `glejser` follows the procedure previously described in §2.1.3.

Mittelhammer et al. (2000, p. 537) recommend using a ω estimator from the auxiliary model, i.e. $\hat{\omega}_a = n^{-1}\sum_{i=1}^{n} u_i^2$. A more conventional approach would be to use $\bar{\omega} = n^{-1}\sum_{i=1}^{n} e_i^2$ (as, for instance, is done in SHAZAM software). The `sigmaest` argument allows the user to implement either of these two approaches: if it is set to `"main"` (the default), the OLS residuals from the main model are used, while if it is set to `"auxiliary"`, the OLS residuals from the auxiliary model are used.

4.1.11 Godfrey-Orme Test (`godfrey_orme`)

Godfrey and Orme's (1999) nonparametric bootstrap method for estimating p-values for a heteroskedasticity test was discussed in §2.1.23.2. The `godfrey_orme` function implements this method as though a separate heteroskedasticity test, but the function takes as one of its arguments `hettest`, which is the name of one of the other heteroskedasticity test functions in the package.

`godfrey_orme` passes `TRUE` for the `statonly` argument when calling function `hettest`, so that `hettest` only computes the test statistic value. The number of bootstrap samples B is specified using the `B` argument and defaults to 1000.[80]

The user *must* specify the tailed-ness of the test using the `alternative` argument (which defaults to `"greater"`), even if the corresponding test function passed as `hettest` does not require this as an argument.

4.1.12 Goldfeld-Quandt Tests (`goldfeld_quandt`)

Goldfeld and Quandt's (1965) parametric F test and nonparametric 'peaks' test, discussed previously in §2.1.2, are both implemented by calling the same function, `goldfeld_quandt`. The method is specified by setting the `method` argument to `"parametric"` and `"nonparametric"`, respectively.

By default, it is assumed that the error variance is *positively* related to the deflator variable. However, this can be reversed by setting the `alternative` argument to `"less"` rather than `"greater"`. Otherwise, if no prior information is available on the suspected direction of monotonic dependency, `alternative` can be set to `"two.sided"` for a two-tailed test.

The proportion c of observations to remove (i.e., those with 'central' values of the deflator) is specified using the argument `prop_central`, with default value $\frac{1}{3}$ (rounded, if necessary, to ensure that nc is an integer). The `group1prop` argument allows the user to specify the proportion of remaining observations allocated to the first subset, in case it is desired to use subsets of unequal size. By default, the two subsets each contain an equal number of observations, $n(1-c)/2$.[81]

[80] This is slightly more conservative than the 400 suggested in Godfrey et al. (2006).

[81] Changing this proportion alters the form of the F statistic as the degrees of freedom in the numerator and the denominator no longer cancel.

If the nonparametric 'peaks' test method is used, p-values are computed from the exact CDF of the number of peaks in an iid sequence of continuous random variables by calling `ppeak`, which in turn calls `dpeak` to compute probability mass values.[82] More details on `ppeak` and `dpeak` are given below in §4.2.4. Because `dpeak` is computationally slow for large n, the probability masses of this distribution for $n = 1, 2, \ldots, 1000$ are stored in a dataset called `dpeakdat` that is exported with **skedastic**.[83] The `restype` character argument controls which residuals are used in the nonparametric test: `"ols"`—for OLS residuals—or `"blus"`—for BLUS residuals (Theil 1965, 1968).[84] If `alternative` is set to `"two.sided"`, the argument `twosidedmethod` allows the user to specify the method by which a two-sided p-value should be computed in `twosidedpval`. For more details on the supporting `dpeak`, `ppeak`, and `twosidedpval` functions, see §4.2.2 and §4.2.4.

4.1.13 Harrison-McCabe Test (harrison_mccabe)

Users can call `harrison_mccabe` to apply Harrison and McCabe's (1979) test, which was introduced in §2.1.9 The user specifies the 'breakpoint index' m for the test using the `m` argument, which can either be an `integer` representing index m or a `double` representing $\frac{m}{n}$. `m` defaults to 0.5. The `harrison_mccabe` function by default conducts a left-tailed test, but this can be changed using the `alternative` argument. The test's p-values are obtained from the CDF of a RQF in a normal random vector using the `pRQF` function (see §4.2.5).

4.1.14 Harvey's Test (harvey)

The `harvey` function implements Harvey's (1976) heteroskedasticity test, as discussed in §2.1.5. This function has no special arguments.

4.1.15 Honda's Test (honda)

Honda's (1989) test, which was described in §2.1.13, can be applied by calling the `honda` function. The test is two-tailed by default, but this can be adjusted to a left-tailed or right-tailed test using the `alternative` argument. p-values are computed from the CDF of a RQF in a normal random vector using the `pRQF` function (see §4.2.5); the two-sided p-value is computed using the `twosidedpval` function (see §4.2.2).

4.1.16 Horn's Test (horn)

The `horn` function implements Horn's (1981) nonparametric heteroskedasticity test, as described in §2.1.10. The `restype` argument controls which type of residuals to use (`"ols"` for OLS or `"blus"` for BLUS). The test is by default two-tailed, but this can be adjusted using the `alternative` argument. p-values are computed from the distribution of Lehmann's (1975) nonparametric trend statistic. The `exact` logical argument controls whether the exact distribution is used or a normal approximation. By default, `exact` is `TRUE` if the length of the residual vector is ≤ 10, and `FALSE` otherwise. Computation time increases rapidly with the number of residuals. The exact probability mass function (PMF) and CDF of the nonparametric trend statistic are computed by the `dDtrend` and `pDtrend` functions, respectively, which are discussed in §4.2.6.

4.1.17 Li-Yao Tests (li_yao)

The `li_yao` function implements the two tests of Li and Yao (2019) as described in §2.1.22, namely the ALRT and the CVT. Which test to apply is controlled by setting the `method` argument to `"cvt"` or to `"alrt"`. The `baipanyin` logical argument, which defaults to `TRUE`, controls whether or not to apply the distribution derived by Bai et al. (2016) to compute the p-value of the CVT; this argument is ignored for the ALRT.

[82] Passing the vector of probabilities for the required n to `goldfeld_quandt` as the `prob` argument enables the nonparametric test to be implemented much more rapidly. Otherwise, if `prob` is set to `NA` (the default), the probabilities are computed.

[83] The implication is that the nonparametric test is extremely slow for $n > 1000$.

[84] `"ols"` is the default only because they are used by Goldfeld and Quandt (1965), who were probably not yet aware of Theil's procedure. However, `"blus"` may be preferable, because, while p residuals are lost in the process of computing the BLUS residuals, the BLUS residuals under homoskedasticity do constitute an iid sequence of random variables, which the OLS residuals do not.

4.1.18 Račkauskas-Zuokas Test (`rackauskas_zuokas`)

Račkauskas and Zuokas's (2007) test, which has been described in §2.1.19, can be conducted by calling the `rackauskas_zuokas` function. The `alpha` argument specifies the hyperparameter $\alpha \in [0, 1/2)$ (not to be confused with the significance level). The `pvalmethod` argument controls how to compute the p-value. If set to `"data"`, the p-value is computed from 2^{14} pre-generated MC replicates from the asymptotic null distribution of the test statistic. These replicates are stored in the `T_alpha` data object in **skedastic**, and cover the α values $i/32$, $i = 0, 1, \ldots, 15$, with $m = 2^{17}$. Alternatively, the user can set the `pvalmethod` argument to `"sim"` to run one's own MC simulation. One would then specify the number of replications R (`R`), the sample size m (`m`) to use when generating the Brownian Bridge for each replicate, and the pseudorandom number generating seed value `seed` (for reproducibility).

4.1.19 Simonoff-Tsai Tests (`simonoff_tsai`)

The MPLR and score tests of Simonoff and Tsai (1994), reviewed in §2.1.15, can be implemented using the `simonoff_tsai` function. The choice of test is controlled by setting the `method` argument to `"mlr"` or `"score"`.

The three forms of $w(\cdot)$ implemented in `simonoff_tsai` are the additive, multiplicative, and log-multiplicative and these are specified using the `hetfun` argument just as with `cook_weisberg` (`"mult"` is the default). In all three cases, $\boldsymbol{\lambda}_0$ is the zero vector, and so the null hypothesis of homoskedasticity can be expressed as $\boldsymbol{\lambda} = \mathbf{0}$.

The ML estimate of $\boldsymbol{\lambda}$ is computed using the `optim` function of **stats**. The optimisation algorithm can be specified using the `optmethod` argument, which corresponds to the `method` argument of `optim` and defaults to `"Nelder-Mead"`.[85] By default, the initial values of $\boldsymbol{\lambda}$ are the q-vector $\left[10^{-3}, 10^{-3}, \ldots, 10^{-3}\right]$.[86] The MPLR test is computationally slow due to the need to maximise the modified profile likelihood function numerically.

The Bartlett correction is activated in `simonoff_tsai` by default, but can be suppressed by setting `bartlett` to `FALSE`.[87]

For the score test, the 'base test' to use is specified in `simonoff_tsai` by the `basetest` argument, which can be set to `"koenker"` (the default) or `"cook_weisberg"`. The form of the function $\mathfrak{g}(\boldsymbol{Z}'_{i\cdot}, \boldsymbol{\zeta})$ (used in the computation of the Jacobian matrix) is specified using the argument `hetfun` exactly as for the MPLR test.

4.1.20 Szroeter's Test (`szroeter`)

One may call the function `szroeter` to implement Szroeter's (1978) heteroskedasticity test. The user must specify a nondecreasing function $h(i)$ of the indices $i = 1, 2, \ldots, n$, as discussed in §2.1.7. This is done using the argument `h`. The default in `szroeter` is `h = SKH`, corresponding to

$$h(i) = 2\left[1 - \cos\left(\frac{\pi i}{n+1}\right)\right], \; i = 1, 2, \ldots, n.$$

The p-values in `szroeter` are computed using the `pRQF` function (see §4.2.5).

4.1.21 Verbyla's Test (`verbyla`)

The `verbyla` function implements Verbyla's (1993) heteroskedasticity test, as reviewed in §2.1.14. The function has no special arguments and is thus straightforward to use.

4.1.22 White's Test (`white`)

The `white` function implements the famous heteroskedasticity test of White (1980), which was discussed in §2.1.8 as an extension of Breusch and Pagan's (1979) test. The `interactions` argument controls whether

[85] The `...` argument allows the user to pass other arguments to `optim`, such as `par` (the initial values of $\boldsymbol{\lambda}$), `maxit` (the maximum number of iterations to use in the optimisation procedure), `trace` (which can be used to display detailed output from the optimisation procedure), etc.

[86] Particularly where `hetfun` is `"mult"` (the multiplicative model), the user should ensure that the initial parameter values are sufficiently small that the initial function evaluation of l_p in the optimisation algorithm returns a computationally finite value.

[87] Note that the Bartlett correction is not currently implemented for the additive heteroskedastic model, due to the extremely complicated expression for c_m that results in this case.

interaction terms are included in the auxiliary design; if `FALSE` (the default), only the covariates and their squares are included.

4.1.23 Wilcox-Keselman Test (`wilcox_keselman`)

One can implement the heteroskedasticity test of Wilcox and Keselman (2006) by calling the `wilcox_keselman` function, described in §2.1.18. This test requires computation of quantile regression estimates, which is done using the Barrodale-Roberts method in the `rq.fit` function of **quantreg** (Koenker 2020).

Wilcox and Keselman (2006, p. 707) note that the test does not hold its size well for non-normal error distributions and thus propose an *ad hoc* size correction method. This method, which they call N2, is not implemented in `wilcox_keselman` as it does not enable computation of a p-value. Rather, the function implements the method that they call N1.

Wilcox and Keselman (2006) do not propose a generalisation of the test to multiple linear regression. However, Wilcox (2020) has written an R package **WRS** featuring a function `qhomtv2` that implements the test using a simple quantile regression model with each explanatory variable. The `qhomtv2` function thus generates $p-1$ p-values, where $p-1$ is the number of explanatory variables (excluding intercept) in the model.[88] The values of the test statistic and corresponding adjusted p-values are displayed in the order of the explanatory variables in the design matrix.

The user must decide on the quantile γ to use; this is passed using the argument `gammapar`, which defaults to 0.2 as recommended in Wilcox and Keselman (2006).[89] The user also specifies the number of bootstrap samples B (using `B`, which defaults to 500). For reproducibility of results, the user may pass an argument `seed` to be used in `set.seed` to set the pseudorandom number generator seed. Finally, the logical argument `matchWRS` allows the bootstrap sampling algorithm and seed to be aligned exactly to those of Wilcox's (2020) function `qhomtv2`.[90]

4.1.24 Yüce's Test (`yuce`)

The `yuce` function implements either of the two heteroskedasticity tests proposed in Yüce (2008), and described above in §2.1.20. The only special argument that the user must specify is `method`, which is set to `"A"` for the chi-squared test or `"B"` for the t-test.

4.1.25 Zhou, Song, and Thompson's Test (`zhou_etal`)

`zhou_etal` is a function that implements the heteroskedasticity test method of Zhou et al. (2015), reviewed in §2.1.21. The B perturbation samples are generated from the normal distribution, with B specified using the `Bperturbed` argument. Using the `method` argument, the user can implement either an omnibus or 'pooled' test (`"pooled"`), a covariate-specific deflator-type test (`"covariate-specific"`), or the hybrid approach (`"hybrid"`) as described in §2.1.21.

If the covariate-specific method is used, it is applied to each covariate separately, and the function's output is a `tibble` object containing the test statistic and corresponding p-value for each deflator. The `seed` argument can be used to set the pseudorandom number generator seed for reproducibility of the perturbation sampling.

The test statistic returned by `zhou_etal` in the hybrid case is either T_{pool} (if $P_{\text{pool}} < P_{\text{cs}}$) or T_r (if $P_{\text{cs}} < P_{\text{pool}}$), where r is the index such that $P_r = \min\{P_1, P_2, \ldots, P_q\}$.

4.2 Supporting Functions for Heteroskedasticity Testing

Each of the functions discussed in §4.1 implements one or more heteroskedasticity testing methods. Many of these functions need to perform complicated computations to calculate the test statistic and its p-value. In keeping with the principles of functional programming, separate functions were created for these supporting computations. Some of these functions are exported with the **skedastic** package, because they are deemed to

[88] `qhomtv2` performs no adjustment to control the familywise error rate; however, `wilcox_keselman` allows the user to pass an argument `p.adjust.method` which will be passed to `p.adjust` (in **stats**) as its `method` argument, thus adjusting the p-values. By default, no adjustment is made, in order to align with Wilcox and Keselman's (2006) method.

[89] This quantile parameter is denoted γ following the notation of Wilcox and Keselman (2006) and should not be confused with the parameter $\boldsymbol{\gamma}$ used in the ALVMs and ANLVMs in this study.

[90] Note that the default number of bootstrap samples in `qhomtv2` is 100.

have value and applicability beyond their use in applying certain heteroskedasticity tests. These functions therefore also constitute an original research contribution in their own right, and for that reason are described in this section.

The `hetplot` function is not related to any particular heteroskedasticity test but produces diagnostic plots for detecting heteroskedasticity. The `blus` function computes BLUS residuals. Several functions described in this section assist with computation of p-values, either by computing probabilities from a certain distribution (`dpeak`, `ppeak`, `dDtrend`, `pDtrend`, `pRQF`), or by computing two-sided p-values from a given asymmetric distribution (`twosidedpval`).

4.2.1 Computation of Best Linear Unbiased Scalar-Covariance-Matrix Residuals (`blus`)

Theil's (1965) BLUS residuals were introduced in §1.1.7.5, and can be (or are, by default) used instead of OLS residuals in some heteroskedasticity tests, such as Goldfeld and Quandt's (1965), Ramsey's (1969) BAMSET, and Horn's (1981).

Theil's (1968) algorithm for computing the BLUS residuals, which is implemented by the `blus` function in the **skedastic** R package, can be outlined as follows.

1. Choose which p observations will be 'lost' when computing the BLUS residuals and reorder the observations so that these p observations are first.
2. Partition the model as follows:

$$\begin{bmatrix} \boldsymbol{y}_0 \\ \boldsymbol{y}_1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{X}_0 \\ \boldsymbol{X}_1 \end{bmatrix} \boldsymbol{\beta} + \begin{bmatrix} \boldsymbol{\epsilon}_0 \\ \boldsymbol{\epsilon}_1 \end{bmatrix},$$

   where

   $\boldsymbol{y}_0$ consists of the first p observations,

   $\boldsymbol{y}_1$ consists of the last $n-p$ observations,

   and similarly for $\boldsymbol{X}$ and $\boldsymbol{\epsilon}$. It is assumed that $\boldsymbol{X}_0$ is nonsingular. $\boldsymbol{e}$ is partitioned into $\begin{bmatrix} \boldsymbol{e}_0 \\ \boldsymbol{e}_1 \end{bmatrix}$ in the same manner.
3. Compute the BLUS residuals as,

$$\boldsymbol{e}_{\text{BLUS}} = \boldsymbol{e}_1 - \boldsymbol{X}_1 \boldsymbol{X}_0^{-1} \left[\sum_{j=1}^{p} \frac{\lambda_j}{1+\lambda_j} \boldsymbol{q}_j \boldsymbol{q}_j' \right] \boldsymbol{e}_0, \tag{4.1}$$

   where λ_j^2, $j = 1, 2, \ldots, p$ are the eigenvalues of $\boldsymbol{X}_0 (\boldsymbol{X}'\boldsymbol{X})^{-1} \boldsymbol{X}_0'$ and $\boldsymbol{q}_j$ are the corresponding eigenvectors.

This is, as far as the author knows, the first implementation of BLUS residuals in an R package available on CRAN, although an R procedure for computing BLUS residuals was described previously in Vinod (2014). The `omit` argument of `blus` controls which p observations are not represented in the BLUS residual vector. This argument can be a numeric vector of length p specifying the indices to omit. Alternatively, it can be a character value, either `"first"` (indicating that the first p observations should be omitted), `"last"` (indicating that the last p observations should be omitted), or `"random"` (indicating that p randomly chosen observations should be omitted). Sometimes the algorithm fails due to $\boldsymbol{X}_0$ or $\boldsymbol{X}'\boldsymbol{X}$ being numerically singular. If such singularity occurs for the chosen subset of indices to omit, `blus` instead chooses a random subset of observations to omit. If this also results in a singular $\boldsymbol{X}_0$, another random subset is attempted, and so on until a subset is found for which the BLUS residuals can be computed.[91] The `seed` argument can be used to set the seed of the pseudorandom number generator to make the randomisation of omitted indices reproducible.

The `keepNA` logical argument controls the structure of the BLUS residual vector $\boldsymbol{e}_{\text{BLUS}}$ returned by `blus`. If `TRUE` (the default), an n-vector is returned, with `NA_real_` as the value for the p indices that were omitted. If `FALSE`, an $(n-p)$-vector is returned with no `NA` values.

[91] The user can specify how many random subsets to attempt using the `exhaust` argument. The default value, `NA`, results in all $\binom{n}{p}$ possible subsets being attempted, if necessary, provided that $\binom{n}{p} \leq 10^4$. Otherwise, up to 10^4 different random subsets are attempted. If `exhaust` is set to an integer value, the maximum number of subsets attempted will be the smaller of `exhaust` and $\binom{n}{p}$.

4.2.2 Computing Two-Sided p-Values from Asymmetric Distributions (`twosidedpval`)

There is no generally accepted method of obtaining a two-sided p-value from an asymmetric null distribution in statistical inference. Some of the tests implemented in **skedastic** have such null distributions, either continuous (Carapeto and Holt 2003, Honda 1989) or discrete (Goldfeld and Quandt's (1965) nonparametric test). A common way of obtaining a two-sided p-value from an asymmetric null distribution is to simply double the one-sided p-value. One of the weaknesses of this approach is that it can result in a p-value greater than 1. Another method sometimes used is to compute the probability over all values (from both tails) with probability mass or density less than or equal to that of the observed value. A weakness of this approach is that, particularly with multimodal distributions, there may be values between the null value and the observed value that are less likely than the observed value.

Kulinskaya (2008) proposes a new method of defining two-sided p-values that she refers to as 'conditional two-sided p-values' and denoted by P_C. The conditional two-sided p-value is intuitively similar to the doubled p-value, but the p-values on each of the two tails are weighted inversely according to the probability of falling on that tail. The result is that p-values are 'inflated' on the thinner tail and 'deflated' on the thicker tail (relative to the doubled p-value).

Computation of P_C, both in the continuous and discrete cases, requires one to specify a generic location parameter A used to separate the two tails of the null distribution, of which particular examples include the mean, mode, and median.[92] It is required that $0 < F(A) < 1$, where F is the cumulative distribution function of the null distribution.

Let T be a test statistic with observed value q and null distribution F. For the continuous case, Kulinskaya (2008, p. 5) defines the weighted two-sided p-value centred at A as follows:[93]

$$P_w^A(q) = \min\left\{1, \frac{F(q)}{w_L}1_{q\leq A} + \frac{1-F(q)}{w_R}1_{q>A}\right\}, \tag{4.2}$$

where $1_\bullet$ is the indicator function and w_L and w_R are positive weights satisfying $w_L + w_R = 1$. If $w_L = w_R = \dfrac{1}{2}$, then $P_w^A(q)$ is a version of the doubled one-sided p-value that will never exceed 1. The conditional two-sided p-value centered at A is obtained by setting $w_L = F(A)$ and $w_R = 1 - F(A)$; thus

$$P_C^A(q) = \frac{F(q)}{F(A)}1_{q\leq A} + \frac{1-F(q)}{1-F(A)}1_{q>A}. \tag{4.3}$$

This is a smooth function of q (except at A), with a maximum of 1 at $q = A$.[94] The function strictly increases for $x < A$ and strictly decreases for $x > A$.

For the discrete case, the definition of the conditional two-sided p-value centred at A, as per Kulinskaya (2008, p. 11), depends on whether A is attainable, i.e. belongs to the support of T:

$$P_C^A(q) = \begin{cases} \dfrac{\Pr(T\leq q)}{\Pr(T<A)}1_{q<A} + \dfrac{\Pr(T\geq q)}{\Pr(T>A)}1_{q>A} & \text{if } A \text{ is not attainable} \\[2ex] \dfrac{\Pr(T\leq q)}{\Pr(T\leq A)/\left(1+\Pr(T=A)\right)}1_{q<A} & \\ +\ 1_{q=A} + \dfrac{\Pr(T\geq q)}{\Pr(T\geq A)/\left(1+\Pr(T=A)\right)}1_{q>A} & \text{if } A \text{ is attainable} \end{cases}. \tag{4.4}$$

The A parameter is specified in `twosidedpval` using the `Aloc` argument. Two plausible choices are the null value of the parameter being tested and the null distribution mean. If `Aloc` is not specified, `twosidedpval`

[92] If the median is selected for A, P_C is identical to the doubled one-sided p-value.

[93] The notation here has been slightly altered from that used by Kulinskaya (2008). Moreover, her notation had a $< A$ indicator and a $> A$ indicator, with neither term including the value of A; the result is that the weighted two-sided p-value would take on a value of 0 at A. This is untidy, even if the probability of a continuous random variable equalling a particular value is vanishing.

[94] The definition of P_C^A given in Kulinskaya (2008, p. 6) is confusing and technically incorrect. It expresses P_w^A with weights $w_L = F(A)$ and $w_R = 1 - F(A)$ as a sum of conditional probabilities involving the test statistic and an independent random variable (call it T') having the same distribution. This is invalid because if T and T' are independent, the conditional distribution of $T|T'$ is simply the marginal distribution of T. In fact, the two-sided p-value proposed by Kulinskaya (2008) does not technically involve a conditional probability and thus might be better named 'tail-weighted two-sided p-value'.

attempts to compute the distribution mean from `CDF` (the user-specified cumulative distribution function) and sets `Aloc` to this value.[95] If `CDF` corresponds to the cumulative distribution function $F(\cdot)$ and the null distribution is continuous, the distribution mean is computed by evaluating the following integral using `quadinf` in **pracma** (Borchers 2022):

$$\mathrm{E}_0(T) = \int_0^{\infty} (1 - F(t))dt - \int_{-\infty}^{0} F(t)dt.$$

If the null distribution is discrete, its distribution mean is computed by evaluating the following expression:[96]

$$\mathrm{E}_0(T) = \sum_{t=0}^{\infty}(1 - F(t)) - \sum_{t=-\infty}^{-1} F(t).$$

Distribution parameters for `CDF` may be specified using the `...`, the ellipsis argument in R. Optionally, the user may specify the minimum and maximum values in the support of a discrete distribution using the `supportlim` argument; this will improve computational efficiency if the `"minlikelihood"` method is used or if the function must compute the distribution mean to use as the `Aloc` value. The user specifies whether the null distribution is continuous or discrete using the logical argument `continuous`. Finally, the `method` argument is used to specify which of three methods should be used to compute the two-sided p-value. The value `"doubled"` corresponds to the doubled one-sided p-value, `"kulinskaya"` to the conditional two-sided p-value (Kulinskaya 2008), and `"minlikelihood"` corresponds to the sum of probabilities of all values with probability less than or equal to that of the observed value.

4.2.3 Scatter Plots for Heteroskedasticity Diagnostics (`hetplot`)

Graphical methods, and in particular residual plots, provide a useful diagnostic and visualisation tool for heteroskedasticity in linear regression models. Most practitioners are familiar with the use of scatter plots for heteroskedasticity diagnostics, such as a plot of the OLS residuals e_i vs. the OLS fitted values $\hat{y}_i$ or one of the explanatory variables. Cook and Weisberg (1983) suggest several ways to improve on this basic plot. First, they suggest that the squared residuals e_i^2 are a better choice for the variable plotted on the vertical axis, because this doubles the sample size in a visual sense. Moreover, they point out that even under homoskedasticity, the variances of the OLS residuals is not constant but equals $\sigma^2 m_{ii}$. Thus, to reduce the risk of a spurious pattern appearing, it would be better to consider $e_i/\sqrt{m_{ii}}$, an observable variable that *does* have constant variance under homoskedasticity.

The `hetplot` function incorporates these and other possibilities to offer a customisable heteroskedasticity plotting tool built around the basic scatter plot functionality of the `plot` function in base R. The three key arguments that define the plot(s) are `horzvar`, `vertvar`, and `vertfun`. `horzvar` is a character argument specifying the variable(s) to plot on the horizontal axis. Possible values and the variables they represent are displayed in Table 4.1.

If one wants to plot only one explanatory variable, one can pass the `names` element of the `data.frame` corresponding to that variable; by concatenating `"log"` with the `names` element, one can plot the natural

Table 4.1: Possible Values for `horzvar`

`horzvar` Value	Variable Plotted
`"index"`	i
`"fitted.values"`	$\hat{y}_i$
`"fitted.values2"`	$m_{ii}\hat{y}_i$
`"explanatory"`	X_{ij} for all j
`"log_explanatory"`	$\log X_{ij}$ for all j

[95] This has been tested for the cumulative distribution functions of well-known distributions included in **stats** (e.g., `pchisq`, `pbinom`), but may fail or yield unexpected results for other choices of `CDF`, especially user-defined functions.

[96] Since `sum` cannot take a vector of infinite length, the function truncates the vector at `1e6` rather than `Inf`, even if the support continues to infinity.

Table 4.2: Possible Values for `vertvar`

`vertvar` Value	Variable Plotted
`"res"`	e_i
`"res_blus"`	$\tilde{e}_i$ (BLUS residuals)
`"res_stand"`	$\frac{e_i}{s}$, $s^2 = n^{-1} \sum_{i=1}^{n} e_i^2$
`"res_constvar"`	$\frac{e_i}{\sqrt{m_{ii}}}$
`"res_stud"`	$\frac{e_i}{\hat{\sigma}\sqrt{m_{ii}}}$, $\hat{\sigma}^2 = (n-p)^{-1} \sum_{i=1}^{n} e_i^2$

logarithm of the specified explanatory variable. If the argument corresponds to more than one variable to plot on the horizontal axis (e.g. `"explanatory"` in a model with multiple explanatory variables), multiple plots will be produced.

`vertvar` is a character argument specifying the residual variable(s) to plot on the vertical axis. Possible values and the variables they represent are displayed in Table 4.2.

The user may specify a character vector of length > 1 with multiple values, in which case multiple plots are produced.

`vertfun` is a character argument specifying the name(s) of one or more functions to apply to the residual variable indicated by `vertvar`. A number passed as a character, such as `"2"`, is interpreted as a power to be applied to the vertical axis variable. Other functions to consider include `"identity"` (to plot the `vertvar` variable as is) and `"abs"` (for the absolute value function).

Since one can pass more than one value for both the `vertvar` and `vertfun` arguments, and since some of the `horzvar` arguments entail multiple horizontal variable arguments (e.g. `"explanatory"`, representing all explanatory variables), the total number of plots to be produced by one call of `hetplot` may be large. Accordingly, there are two ways to output the plot(s), which are specified using the `filetype` character argument. If `filetype` is set to `NA` (the default), all required plots are passed to a single device where they are displayed in a matrix structure using the `mfrow` graphical parameter.[97] Alternatively, the `filetype` argument can be one of `"png"`, `"bmp"`, `"jpeg"`, or `"tiff"`, which results in each individual plot being written to an image file of that type. In order to comply with CRAN's Repository Policy, these image files are written to a subfolder called `hetplot` within the R session's temporary directory. The path of the temporary directory can be obtained within the session using `tempdir()`. The filename of each image file names the horizontal and vertical variable for that plot and also includes a timestamp. If the image files are needed after the R session is ended, the user should copy them to a permanent directory. Besides these arguments, the user may pass other arguments such as graphical parameters to use in plotting. Examples of plots generated using `hetplot` can be found in §5.6.

4.2.4 Computing Probabilities of Number of Peaks in an iid Random Sequence (`dpeak`, `ppeak`)

Let $\{Q_1, Q_2, \ldots, Q_n\}$ be a sequence of independent and identically distributed continuous random variables. A random variable in the sequence is a 'peak' if its value exceeds the values of all previous random variables in the series. Q_1 is not considered a peak. Thus, the number of peaks P in the sequence is defined as follows:

$$P = \sum_{i=2}^{n} 1_{Q_i \geq \max\left\{Q_1, Q_2, Q_{i-1}\right\}}, \tag{4.5}$$

where $1_{\bullet}$ is the indicator function. The support for the number of peaks consists of the integers $\{0, 1, \ldots, n-1\}$.

[97] The function will attempt to find an attractive dimensionality for the required number of plots; thus for instance if the number of plots required is 6, they will be displayed in a 2×3 structure. If the number of rows or columns in the plotting structure exceeds 4, a warning is produced.

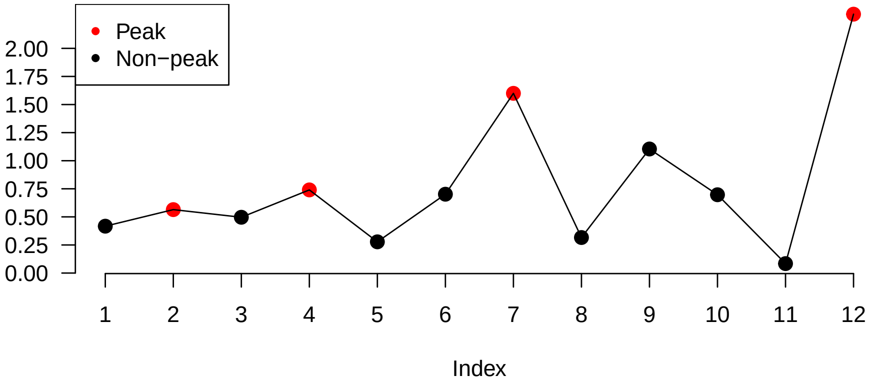


Figure 4.1: Illustration of Peaks in a Sequence

skedastic exports three functions relating to the variable P. `countpeaks` simply returns the observed number of peaks in a `double` vector passed as its only argument, `x`.[98] Figure 4.1 illustrates a sequence of twelve values containing four peaks.

`dpeak` and `ppeak` compute the PMF and CDF, respectively, of P. Since `ppeak` merely computes cumulative sums of probabilities computed in `dpeak`, this discussion focuses on `dpeak`. Following the notation used by Goldfeld and Quandt (1965), define $N(n,k)$ as the number of permutations of a sequence of n values containing k peaks. Defining for convenience $N(1,0)=1$, the authors make use of recursive relations to derive a general expression for the probability $P(n,k)$ that a sequence of n iid continuous random variables has exactly k peaks, namely,

$$P(n,k)=\frac{1}{n!}N(n,k). \tag{4.6}$$

`dpeak` takes arguments `k`, representing k, an integer denoting the number of peaks of which the probability should be computed,[99] `n`, representing n, an integer denoting the length of the series,[100] and `usedata`, a logical indicating whether the probability should be taken from the `dpeakdat` dataset rather than computed within the function. The `factorial` function in base R can only compute $n!$ for $n\leq 170$; for $n>170$ it returns `Inf`. Accordingly, where $n>170$, `dpeak` makes use of the `factorialZ` function from the **gmp** package (Lucas et al. 2020) to calculate $n!$, and, for similar reasons, uses the function `mpfrArray` from the **Rmpfr** package (Maechler 2020) to calculate $N(n,k)$. However, computation time is an issue for large n, and for this reason **skedastic** includes the `dpeakdat` dataset containing pre-calculated ‘peaks’ probability distributions for n up to 1000.

The function value is a double vector of the same length as `k` representing the probabilities, with the values of `k` stored in a `names` attribute. Figure 4.2 shows the expected number of peaks E(P) as a function of the sequence length n.

[98] Note that any `NA` values in `x` are ignored.

[99] `dpeak` is vectorised with respect to `k`; setting `k = 0:(n - 1)` corresponds to computing probabilities over the full support from 0 to $n-1$. Note that computation time for `k = 100` and `k = 0:100` will be similar, since the procedure is recursive.

[100] `dpeak` is *not* vectorised with respect to `n`.

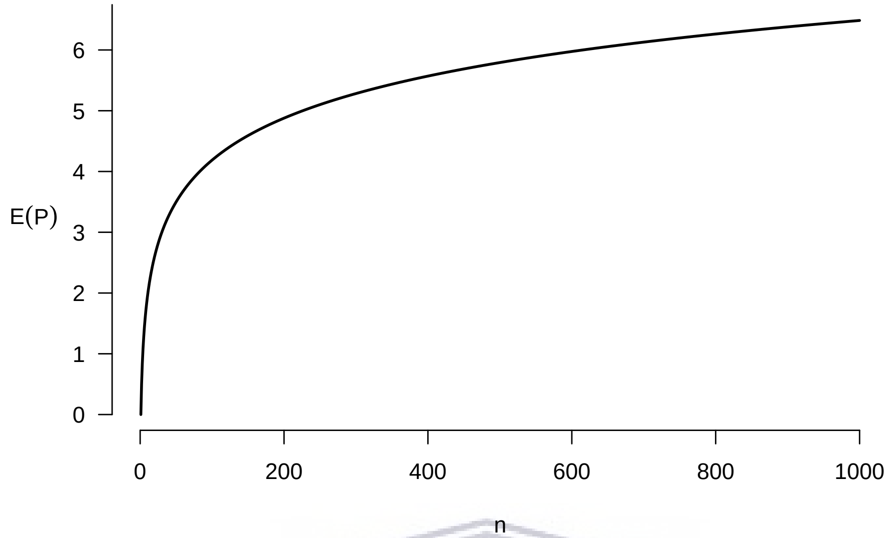


Figure 4.2: Expectation of Number of Peaks in a Sequence of n iid Random Variables

`ppeak` has the same arguments as `dpeak` with the same meaning, as well as a logical argument `lower.tail` indicating whether the lower-tailed cumulative probability should be computed. Figure 4.3 shows the PMF and CDF of the number of peaks P for an iid sequence of $n = 10$ continuous random variables.

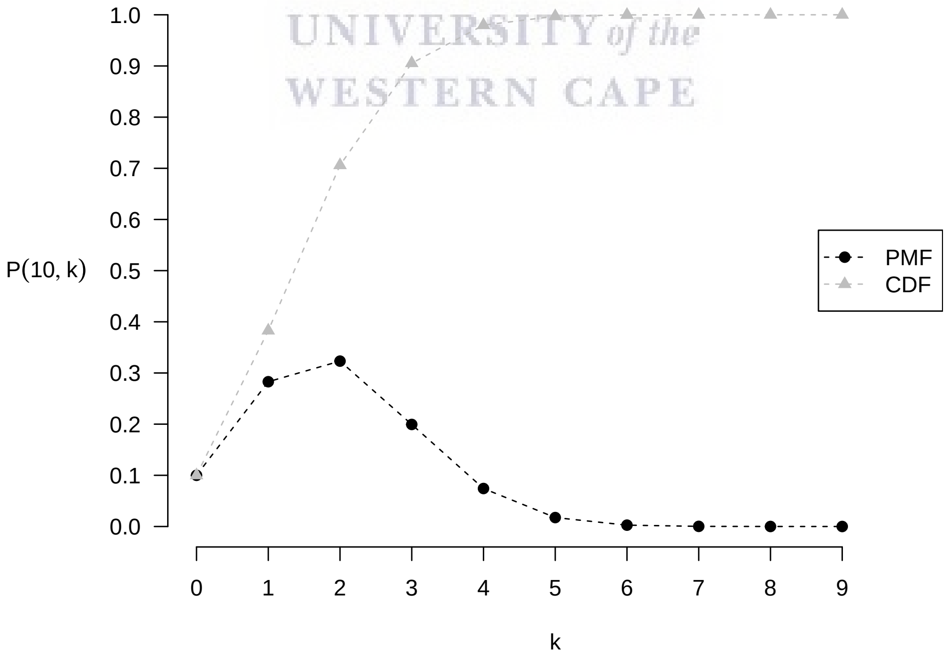


Figure 4.3: PMF and CDF of a Sequence of $n = 10$ iid Continuous Random Variables

4.2.5 Computing p-Values for a Ratio of Quadratic Forms in a Normal Random Vector (`pRQF`)

The test statistic for a number of the tests implemented in **skedastic** (Szroeter 1978, Harrison and McCabe 1979, Evans and King 1988, Honda 1989, Carapeto and Holt 2003) is, under A2 (the null hypothesis of homoskedasticity), together with A3 and A5, a Ratio of Quadratic Forms (RQF) in a normally distributed random vector with a scalar covariance matrix, namely the error vector $\boldsymbol{\epsilon}$. The R package **CompQuadForm** (Duchesne and de Micheaux 2010) contains functions that compute cumulative probabilities for a quadratic form in normally distributed random variables. The `pRQF` function computes cumulative probabilities on a Ratio of Quadratic Forms in a normal random vector. To do so, it makes use of the fact that a probability expression in a RQF can be rewritten as a probability expression in a quadratic form. Let the ratio statistic be written as

$$T = \frac{\boldsymbol{\epsilon}'\boldsymbol{A}\boldsymbol{\epsilon}}{\boldsymbol{\epsilon}'\boldsymbol{B}\boldsymbol{\epsilon}}, \tag{4.7}$$

where $\boldsymbol{A}$ and $\boldsymbol{B}$ are nonstochastic, symmetric matrices, and $\boldsymbol{\epsilon} \sim N(0, \omega \boldsymbol{I}_n)$. If the observed value of the test statistic is denoted t_0, then the p-value, in the case of an upper-tailed test, can be written as

$$\begin{aligned}\Pr(T > t_0) &= \Pr\left(\frac{\boldsymbol{\epsilon}'\boldsymbol{A}\boldsymbol{\epsilon}}{\boldsymbol{\epsilon}'\boldsymbol{B}\boldsymbol{\epsilon}} > t_0\right) \\ &= \Pr\left(\boldsymbol{\epsilon}'\boldsymbol{A}\boldsymbol{\epsilon} > t_0\boldsymbol{\epsilon}'\boldsymbol{B}\boldsymbol{\epsilon}\right) \\ &= \Pr\left(\boldsymbol{\epsilon}'(\boldsymbol{A} - t_0\boldsymbol{B})\boldsymbol{\epsilon} > 0\right). \end{aligned} \tag{4.8}$$

In light of (4.8), the functions of **CompQuadForm** can be used to compute cumulative probabilities on a RQF in normal random variables. **CompQuadForm** implements four methods for calculating probabilities in quadratic forms, of which two can be called from `pRQF`. These are the Imhof algorithm (Imhof 1961) and the Davies algorithm (Davies 1980), implemented in `imhof` and `davies`, respectively.[101]

The arguments to be passed to `pRQF` are `r`, corresponding to t_0 (the observed value of the ratio statistic), `A` (corresponding to matrix $\boldsymbol{A}$), `B` (corresponding to matrix $\boldsymbol{B}$), `lower.tail` (a logical indicating whether a lower-tailed probability is required), and `algorithm` (a character specifying the method to use). The three possible values for `algorithm` are `"imhof"` (the default), `"davies"`, and `"integrate"`. The first two correspond to calls of the eponymous functions in **CompQuadForm**. To make the function less dependent on this package, `"integrate"` results in the Imhof algorithm integral being evaluated using the `integrate` function in the **stats** package.[102]

[101] The other two functions in **CompQuadForm**, `farebrother` and `liu`, are not supported in `pRQF`. The first requires that the matrix in the quadratic form be positive semi-definite (which $\boldsymbol{A} - t_0\boldsymbol{B}$ in general is not), and the second method is shown in Duchesne and de Micheaux (2010) to be inaccurate.

[102] This is computationally slower than `"imhof"`, since the **CompQuadForm** functions use compiled code.

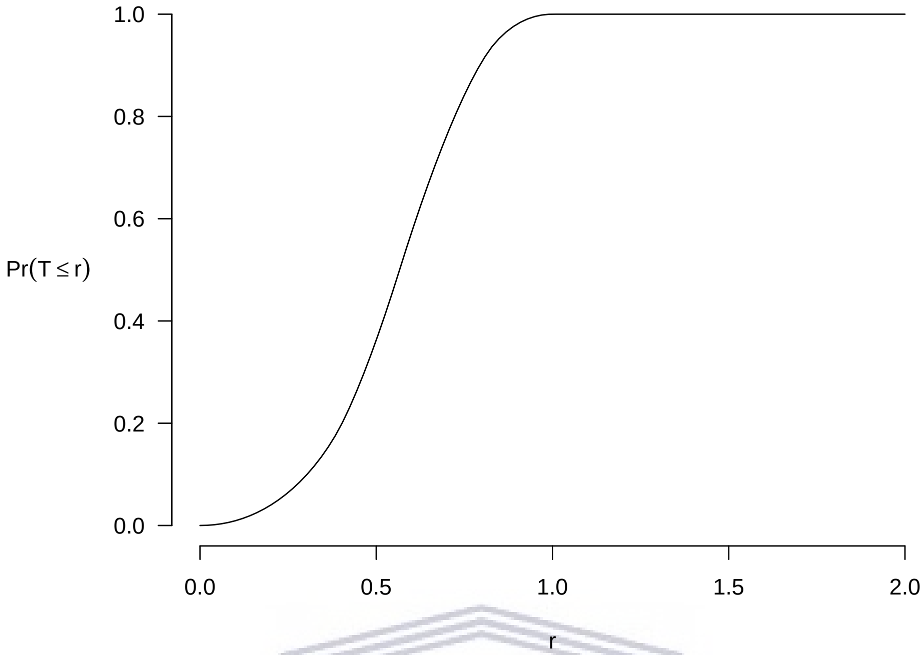


Figure 4.4: CDF of an Instance of Harrison and McCabe's (1979) Test Statistic under Homoskedasticity

Figure 4.4 shows the CDF of Harrison and McCabe's (1979) test statistic T, which is a RQF, for values of the ratio between 0 and 2, for a particular DGP with $n = 20$, $p = 3$, the two covariates generated independently from $U(0, 1)$, and $m = 0.5$ (see §2.1.9 for details of the test). Note that this is a left-tailed test.

The ability of **skedastic** to compute exact p-values for this statistic (subject to the accuracy of the Imhof algorithm's numerical approximation) contrasts with that of the `hmctest` function of the **lmtest** R package (Zeileis and Hothorn 2002). The latter gives the user the option of estimating p-values using a simulation,[103] or otherwise returning `NA` as the p-value.

4.2.6 Computing Probabilities for Lehmann's Nonparametric Trend Statistic (`dDtrend`, `pDtrend`)

Let R_i, $i = 1, 2, \ldots, n$, be the ranks of n independent and identically distributed random variables. Lehmann (1975) proposed the following statistic D as a nonparametric measure of trend in such a scenario:

$$D = \sum_{i=1}^{n} (R_i - i)^2. \tag{4.9}$$

This statistic is applied to the absolute residuals in Horn's (1981) test for heteroskedasticity. Accordingly, **skedastic** contains functions to calculate probabilities, either exact or approximate, on D under the null hypothesis. `dDtrend` computes the exact distribution of D for a sample of size n (passed as `n`) in the event that there are no ties in the sample. The support $\mathcal{S}$ of D in this case is as follows:

$$\mathcal{S} = \begin{cases} \{0\}, & n = 1 \\ \{0, 2\}, & n = 2 \\ \{0, 2, 6, 8\}, & n = 3 \\ \left\{0, 2, \ldots, \dfrac{n(n-1)(n+1)}{3}\right\}, & n \geq 4 \end{cases}.$$

[103] The `lmtest` documentation does not explain what kind of simulation is used, but presumably it is a MC simulation.

Moreover, the distribution of D is symmetric about $\frac{1}{6}(n^3 - n)$. The exact distribution is computed by counting and tabulating permutations exhaustively. The algorithm is prohibitively slow for $n > 11$. Thus, passing an `n` value greater than 11 results in an error unless the `override` logical argument is set to `TRUE` (in which case the function attempts to make the computation). The value(s) of D for which the probability should be computed is passed to `dDtrend` using the argument `k`. This argument can either be an integer vector or a character `"all"` (the default), in which case probabilities are computed for the entire support of D for the given n. `dDtrend` returns a double vector of probabilities with the corresponding values of D stored in a `names` attribute.

The exact probability distribution of D for $n = 9$ is displayed graphically in Figure 4.5. The support in this case consists of even integers between 0 and 240.

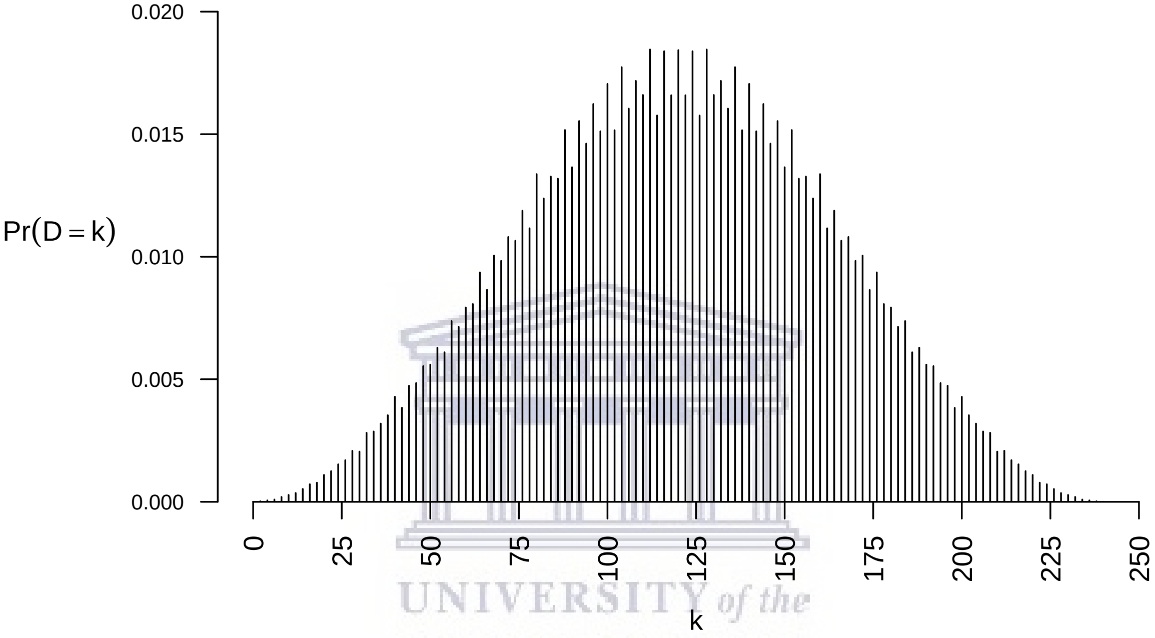


Figure 4.5: PMF of Lehmann's (1975) D Statistic for $n = 9$

`pDtrend` computes cumulative probabilities on the nonparametric statistic D, either from the exact distribution of D, via `dDtrend` (only feasible for $n \leq 10$, and where there are no ties) or using a normal approximation. The value of D for which the cumulative distribution function should be computed is passed using the argument `k`, just as in `dDtrend`. If there are no ties,[104] the expectation and variance of D are, respectively,

$$\mathrm{E}(D) = \frac{1}{6}(n^3 - n),$$

and

$$\mathrm{Var}(D) = \frac{1}{36}n^2(n+1)^2(n-1).$$

Lehmann (1975) provides a proof of the asymptotic normality of D; the rough bell shape of the exact distribution is apparent already for $n = 9$ in Figure 4.5. The normal approximation for the lower-tailed probability is

$$\Pr(D \leq k) \approx \Phi\left(\frac{k - \mathrm{E}(D) - 1}{\sqrt{\mathrm{Var}(D)}}\right),$$

where $\Phi(\cdot)$ is the standard normal cumulative distribution function.[105]

[104] `pDtrend` does implement a normal approximation for the case where ties are present, following the approach of Lehmann (1975, pp. 293-94).

[105] The -1 in the numerator is a continuity correction (-1 rather than -0.5 because the support of D consists of even numbers incrementing by 2).

4.3 A Function That Computes Heteroskedasticity-Consistent Covariance Matrix Estimators

The `hccme` function in **skedastic** is similar in purpose to the `vcovHC` function in the **sandwich** package discussed previously in §2.5.3. However, whereas `vcovHC` implements only HC0-HC5 and HC4m (whilst also allowing the user to pass a customised HCCME function via the `omega` argument), the new `hccme` function directly implements HC0-HC7, HC4m, and HC5m.[106]

The first argument of `hccme`, `object`, can either be an object of class `"lm"` (a linear model object) or a list containing named objects `X` (design matrix $\boldsymbol{X}$) and `e` or `esq` (OLS residual or squared residual vector). The `hcnum` argument controls which HCCME to compute (of those discussed in §2.3). Like the `type` argument of `vcovHC`, it defaults to HC3 due to the popularity of this HCCME. There are two logical arguments, `sandwich` and `as_matrix`. The first, like the `sandwich` argument of `vcovHC`, controls whether to compute the sandwich estimator of the form (1.6) (`TRUE`) or just the error covariance matrix $\hat{\boldsymbol{\Omega}}$ without the 'bread'. Unlike in `vcovHC`, `sandwich` defaults to `FALSE` in `hccme`. The `as_matrix` argument controls whether or not to return a matrix as the result. If `FALSE` (the default), the function returns a vector representing the diagonal of the estimated matrix.

4.4 Functions for Estimating Error Variances

4.4.1 Fitting an Auxiliary Linear Variance Model (`alvm.fit`)

The `alvm.fit` function in **skedastic** fits an ALVM to a linear regression model, applying all of the estimation, tuning, and feature selection methods described in §3.3. This function is fairly complicated, but since ALVMs are the main methodological contribution of this research, a detailed explanation is necessary. Yet, for sake of conciseness, not every argument and feature of the function will be described, but only the essential points.

Firstly, the `mainlm` argument contains the information from the underlying linear regression model. It can either be an object of class `"lm"` or a list containing named objects `y` (response vector $\boldsymbol{y}$), `X` (design matrix $\boldsymbol{X}$), and `e` (OLS residual vector $\boldsymbol{e}$).

The `model` argument controls which particular ALVM to fit. Possible options are `"cluster"` (for the clustering ALVM discussed in §3.2.4), `"spline"` (for the thin-plate spline ALVM discussed in §3.2.3.2), `"linear"` (for the linear ALVM described in (3.40)), `"polynomial"` (for the penalised polynomial ALVM discussed in §3.2.3.1), `"basic"` (for the basic ALVM described in (3.34)), and `"homoskedastic"` (to estimate all error variances using $\hat{\omega}_i = \hat{\omega}_{\text{ub}}, i = 1, 2, \ldots, n$).

The `varselect` argument controls the method to be used for feature selection within the ALVM. Possible values are `"none"` (for no feature selection, meaning that all explanatory variables in $\boldsymbol{X}$ are used in the ALVM), `"hettest"` (feature selection by heteroskedasticity testing, as discussed in §3.3.3.2), `"cv.linear"` or `"cv.cluster"` (best subset selection by K-fold CV applied to either the linear or clustering ALVM), or `"qgcv.linear"` or `"qgcv.cluster"` (best subset selection by QGCV applied to either the linear or clustering ALVM). For details on these best subset selection methods, see §3.3.3.3. Note that, in the case of the penalised polynomial and thin-plate spline models, the penalty indirectly performs feature selection (see §3.3.3.1). Therefore, `varselect` would normally be set to `"none"` when using one of these models. There are supporting functions for carrying out each of the above feature selection methods: `hetvarsel` for feature selection by heteroskedasticity testing, and `varsel.cv.linear`, `varsel.cv.cluster`, `varsel.qgcv.linear`, and `varsel.qgcv.cluster` for the other four techniques. For details on these functions, see §4.4.4.

The `lambda` argument controls the setting of the hyperparameter λ used in the penalised polynomial ALVMs and the thin-plate spline ALVM. The default value, `"foldcv"`, results in λ being chosen by K-fold CV. `"qgcv"` results in λ being chosen by QGCV. Alternatively, the user can pass `lambda` as a double of length 1, representing the value of λ.

The `nclust` argument controls the setting of the hyperparameter n_c in the clustering ALVM, denoting the number of clusters. The default value, `"elbow.swd"`, results in n_c being chosen by the elbow method using the SWD criterion (see §3.3.2.2). The next two values correspond to the elbow method with other criteria (MWD and the average of the elbow method results using SWD and MWD). `"foldcv"` corresponds to selecting n_c by K-fold CV (also discussed in §3.3.2.2). Alternatively, `nclust` can be passed as an integer of length 1, representing the value of n_c.

[106] Both functions also allow computation of the homoskedastic estimator of $\boldsymbol{\Omega}$, namely $\hat{\omega}_{\text{ub}}\boldsymbol{I}_n$, by setting the relevant argument to `"const"`.

The `clustering` argument allows the user to pass a clustering object generated by `doclust` (see §4.4.4), thus circumventing the need to perform clustering from within the `alvm.fit` routine. `clustering` defaults to `NULL`; if it is not `NULL`, the clustering object will already have a fixed n_c value assigned to it, and the `nclust` argument is therefore ignored. Both `nclust` and `clustering` are ignored if `model` is other than `"cluster"`.

The `polypen` argument specifies the type of penalty to be used in the penalised polynomial ALVM. `"L2"`, the default, corresponds to the L_2-norm penalty (like RR), while `"L1"` corresponds to the L_1-norm penalty (like LASSO regression).

The `solver` argument controls which QP solver to use to estimate the ALVM parameters. The default value is `"auto"`, which causes the solver to be selected automatically, as experience has shown that some solvers work better than others for a particular `model`.[107] The rest of the allowed values of `solver` correspond to different QP solvers available within R packages; these have all been mentioned previously in §3.3.1.3.

The `constol` argument sets the value of 0^+, the positive boundary of the inequality constraint that ensures no variance estimates are numerically zero. The default value is 10^{-10}, which is large enough not to result in infinite or `NaN` weights if the estimated covariance matrix $\hat{\boldsymbol{\Omega}}$ is inverted and used with `lm` or `lm.wfit` to compute FWLS estimates of $\boldsymbol{\beta}$ (as discussed in §2.5.2).

The `cvoption` controls how K-fold CV is performed, if necessary. The two possible values are `"testsetols"` and `"partitionres"`, corresponding to the two CV techniques depicted in Figure 3.6. `"testsetols"` is the default value, and due to its superior theoretical grounding (as discussed in §3.3.2.1), it is the only CV technique that has been thoroughly tested and used in §5.

`nfolds` denotes the number of folds K to use for CV, and defaults to 5. `d` denotes the degree d of the polynomial, if the penalised polynomial ALVM is used. `reduce2homosked` is a logical that defaults to `TRUE`. If `TRUE`, then if the feature selection procedure selects none of the features in $\boldsymbol{X}$, the homoskedastic estimator $\hat{\omega}_{\text{ub}}$ will be used instead of fitting a 'null' ALVM (e.g., a clustering ALVM with only one cluster, or a linear ALVM with only an intercept).

The value returned by `alvm.fit` is a list object of class `"alvm.fit"` containing several other objects. `coef.est` contains the estimate $\hat{\gamma}$ of the ALVM parameter vector $\boldsymbol{\gamma}$. `var.est` contains the estimate $\hat{\boldsymbol{\omega}}$ of the error variance vector $\boldsymbol{\omega}$. Other arguments, such as `method`, `fitinfo`, `hyperpar`, and `selectinfo`, contain information on the ALVM used, relevant matrices such as $\boldsymbol{M} \circ \boldsymbol{M}$ and $\boldsymbol{L}$, hyperparameter values such as λ and n_c, and feature selection results. Other relevant information such as the `lm` object for the original linear model, the `constol` value, and the QP `solver` used, is also returned.

4.4.2 Fitting an Auxiliary Nonlinear Variance Model (`anlvm.fit`)

The `anlvm.fit` function in **skedastic** fits an ANLVM using the MQL estimation method described in §3.3.1.4. Some arguments (`mainlm`, `M`, `varselect`, `nclust`, `clustering`, and `reduce2homosked`) have the same meaning as in `alvm.fit`.[108] The `g` argument is a function of one variable specifying the form of $g(\cdot)$, or a character naming such a function. This would normally be either `function(x) x ^ 2` or `function(x) exp(x)` (the character `"exp"` would be treated the same as the latter). `cluster` is a logical argument that defaults to `FALSE`; if true, the clustering ANLVM is used. Experience suggests that it is best to set `g` to `function(x) x ^ 2` with the clustering ANLVM.

The rest of the arguments pertain to the Gauss-Newton numerical scheme for solving the system (3.69). `maxgridrows` specifies the maximum number of initial values $\boldsymbol{\gamma}^{(0)}$ of the parameter vector to try, and defaults to 20. `param.init` specifies the initial value(s) of the parameter vector, $\boldsymbol{\gamma}^{(0)}$. This defaults to a function that will generate the elements of $\boldsymbol{\gamma}^{(0)}$ independently from a $U(-5, 5)$ distribution, `maxgridrows` times. Alternatively, `param.init` can be a numerical vector of length q, in which case this is the only initial value $\boldsymbol{\gamma}^{(0)}$ that is attempted (regardless of `maxgridrows`). Or, `param.init` can be a list containing the named objects `from`, `to`, and either `by` or `length.out`, specifying arguments to pass to `seq` to create a sequence. This sequence is then passed to `expand.grid` to generate a search grid. For instance, if `param.init = list("from" = 1, "to" = 3, "by" = 1)`, the grid will contain 3^q different initial values $\boldsymbol{\gamma}^{(0)}$, namely every possible q-vector consisting of some permutation of ones, twos and threes. However, only a random sample of `maxgridrows` of these 3^q initial values will actually be attempted, unless 3^q is less than

[107] Specifically, `"auto"` results in the `quadprogpp` solver being used for the clustering, linear, and basic ALVMs, the `osqp` solver being used for the L_2-norm penalised polynomial and thin-plate spline ALVMs, and the `roi` solver being used for the L_1-norm penalised polynomial ALVM. The `roi` solver has a stronger tendency to shrink coefficients to zero for the L_1-norm penalised polynomial ALVM, thus better exploiting its sparsity properties.

[108] Note however that `anlvm.fit` does not support the CV technique for choosing n_c in the clustering ANLVM. Only elbow methods can be used.

maxgridrows. nconvstop specifies a stopping rule: once the Gauss-Newton routine has achieved convergence for nconvstop different initial initial values $\gamma^{(0)}$, the search stops and the converged solution that optimises the objective function is returned. maxitql specifies the maximum number of iterations to use in the Gauss-Newton routine, and defaults to 100. tolql specifies the tolerance to use as a convergence criterion, and defaults to 10^{-8}. nestedql is a logical specifying whether to use the nested updating procedure in (3.72). It defaults to FALSE due to the computational cost of the nested procedure.

anlvm.fit returns a list object of class "anlvm.fit" containing objects such as, inter alia, coef.est (a numeric vector with the $\hat{\gamma}$ coefficient estimate), var.est (a numeric vector with the $\hat{\boldsymbol{\omega}}$ variance estimate), and qlinfo, a list containing information on the Gauss-Newton routine, such as the number of iterations used, whether convergence was achieved, and the optimal value of the objective function.

4.4.3 Obtaining Bootstrap Confidence Intervals for Error Variances from an Auxiliary Linear Variance Model (avm.ci)

skedastic contains a function called avm.ci that computes bootstrap CIs for the individual error variances ω_i, $i = 1, 2, \ldots, n$, using the methods described in §3.4. avm.ci takes as its object argument either an object of class "alvm.fit" of the kind produced by alvm.fit, the function discussed above in §4.4.1, or an object of class "anlvm.fit" of the kind produced by anlvm.fit, the function discussed above in §4.4.2.

Three optional arguments—set to NULL by default—are bootobject, bootavmobject, and jackobject. bootobject is an object of class "bootlm" generated by the bootlm function, representing a sample of B bootstrapped regression models. bootavmobject is an object of class "bootavm" generated by the bootavm function, representing a set of B ALVMs or ANLVMs fitted to each of B bootstrapped regression models ("bootlm" class objects). jackobject is an object generated by the jackavm function, representing a set of n jackknife (leave-one-out) ALVMs or ANLVMs, or at least the coefficients thereof. If any of bootobject, bootavmobject, or jackobject is NULL, it is computed from within avm.ci, but passing these objects to avm.ci can save on computation time where CIs are being computed repeatedly from the same model (e.g., several different bootstrap CI methods are being used).

The bootCImethod argument is a character that indicates the method to use for calculating the bootstrap CI. It takes on one of four values: "pct" for a percentile interval (the default), "bca" for a BCa interval, or "stdnorm" for a naïve standard normal bootstrap interval (all three of which are described in §3.4.2).

The bootsampmethod is a character that indicates the nonparametric bootstrap method to be used to generate bootstrap replications of the underlying linear regression model. It can take on two values, either "pairs" (the default) for the pairs bootstrap, or "wild" for the wild bootstrap. Both have been described in §3.4.1. This argument will be ignored if a set of bootstrap linear regression models are passed via the bootobject argument. If the wild bootstrap is used, the resfunc argument sets the transformation $f_i(\cdot)$ to be applied to each OLS residual in the bootstrap DGP. The argument is a character denoting the name of a function, and defaults to "identity", for $f_i(e_i) = e_i$.

The Brequired and Bextra arguments both refer to the number of bootstrap regression models, B. Brequired refers to the desired number of bootstrap models, whereas Bextra allows a larger number of bootstrap models to be generated. The reason is this: experience has shown that where some of the ALVMs fitted to bootstrap samples fall on the QP constraint boundary, the coverage probability of the resulting CI suffers. Consequently, by setting Bextra to a value larger than 0, the total number of bootstrap models generated will be Brequired+Bextra, and the first Brequired bootstrap ALVMs where the QP solution does *not* fall on the constraint boundary are retained. (Thus, the nonparametric bootstrap resampling procedure is modified to include a rejection sampling component). If the number of such ALVMs is less than Brequired, the set of Brequired ALVMs will include some models with QP solutions on the constraint boundary.

conf.level is a double representing the desired confidence level, $1 - \alpha$, for the interval. It defaults to 0.95. expand is a logical, defaulting to TRUE, controlling whether to apply Hesterberg's (1999) expansion technique to the quantiles. retune is a logical controlling whether to retune hyperparameters (e.g., λ or n_c) and select features anew when fitting the ALVM to each bootstrap linear regression. If FALSE, the hyperparameter value and selected features from the original ALVM are reused in each bootstrap ALVM. This is the default, due to the high computation time required to retune hyperparameters and perform feature selection many times.

avm.ci returns a list object of class "avm.ci", of which the most important element is climits, a two-column numeric matrix containing the lower and upper confidence limits, respectively, for the ω_i.

4.4.4 Supporting Functions for Auxiliary Variance Model Implementation

To describe in detail every function that was written for **skedastic** in support of ALVM and ANLVM implementation would require a lot of space and would make for very dull reading. In any case, most of these functions are not exported with the package, meaning that they are not included in the package documentation and are not intended to be called directly by users of the package.[109] However, to give the reader a sense of the amount of programming required to implement the ALVMs, ANLVMs, and associated bootstrap CIs, a summary of these supporting functions is given in Table 4.3.

Table 4.3: Supporting Functions Created for Implementation of ALVMs, ANLVMs, and Bootstrap CIs in **skedastic**

Function Name	What Function Does
`add2clust`	Adds new observations to existing clusters (necessary for computing $\boldsymbol{L}_{\text{test}}$ during cross-validation of a clustering ALVM)
`bootavm`	Fits an ALVM or ANLVM to each of B bootstrapped linear models generated by `bootlm`
`bootlm`	Generates B bootstrap replications of a linear regression model using a nonparametric method suitable for heteroskedastic linear models (bootstrapping pairs or wild bootstrap)
`bracket`	Applies a bracketing method to narrow down the search interval for optimising a continuous function
`CVObjFun.lambda`	Computes a value of the CV loss for a penalised polynomial or spline ALVM, as a function of λ
`CVObjFun.nclust`	Computes a value of the CV loss function for a clustering ALVM, as a function of n_c
`doclust`	Performs agglomerative hierarchical clustering on a data matrix, cutting at a number of clusters n_c chosen by a specified method
`GSS`	Implements the GSS algorithm to minimise a continuous univariate function
`hetvarsel`	Applies a deflator-based heteroskedasticity test to each covariate of a linear regression model (useful for feature selection in an ALVM or ANLVM)
`jackavm`	Obtains jackknife estimates of error variances based on an ALVM or ANLVM (useful for BCa modification of percentile bootstrap interval)
`makepolydesign`	Extends a design matrix $\boldsymbol{X}$ to include all main and cross terms of a polynomial up to a specified degree
`MWDelbow`	Finds the elbow point on the MWD curve using the Unit Invariant Knee (UIK) technique in order to tune n_c for a clustering ALVM
`qpest`	Applies a QP solver to solve the QP necessary to fit an ALVM
`quasiopt`	Implements the Gauss-Newton algorithm necessary to fit an ANLVM by MQL estimation
`SWDelbow`	Finds the elbow point on the SWD curve using the UIK technique in order to tune n_c for a clustering ALVM
`testcalc`	Computes the necessary matrices and vectors to prepare for fitting an ALVM to K training folds

[109] Note, however, that while functions exported with an R package can be called using the `::` syntax (i.e., `packagename::functionname`), non-exported functions defined within the source code of an R package can also be called by users using the `:::` syntax.

Table 4.3: Supporting Functions Created for Implementation of ALVMs, ANLVMs, and Bootstrap CIs in **skedastic** *(continued)*

Function Name	What Function Does
`traincalc`	Computes the necessary matrices and vectors to prepare for predicting ALVM responses in K test folds
`tune.lambda.cv`	Tunes the λ hyperparameter for a penalised polynomial or spline ALVM using K-fold CV
`tune.lambda.qgcv`	Tunes the λ hyperparameter for a penalised polynomial or spline ALVM using QGCV
`tune.nclust`	Tunes the n_c hyperparameter for a clustering ALVM using K-fold CV
`varsel.cv.linear`	Performs feature selection for an ALVM by applying best subset selection to a linear ALVM using K-fold CV loss
`varsel.cv.cluster`	Performs feature selection for an ALVM by applying best subset selection to a clustering ALVM using K-fold CV loss
`varsel.qgcv.linear`	Performs feature selection for an ALVM by applying best subset selection to a linear ALVM using QGCV loss
`varsel.qgcv.cluster`	Performs feature selection for an ALVM by applying best subset selection to a clustering ALVM using QGCV loss

4.5 Chapter Summary

This chapter provided an overview of the functions written in the R package **skedastic**, which was created specifically for this research, to make the existing and new methods discussed in earlier chapters accessible to practitioners.

The first category of functions in **skedastic** is the set of functions that implement heteroskedasticity tests. Twenty-five functions that implement existing heteroskedasticity tests from the literature were described. Effective implementation of these tests required programming of a number of supporting functions that are also exported with the **skedastic** package since they may have other applications. These include a function `blus` for computing BLUS residuals, a function `twosidedpval` for computing two-sided p-values from asymmetric distributions, a function `hetplot` for producing heteroskedasticity diagnostic plots, functions `countpeaks`, `dpeak`, and `ppeak`, for computing the number of peaks in an iid random sequence and probabilities thereof, a function `pRQF` for computing the CDF of a RQF in normal random vectors, and functions `dDtrend` and `pDtrend` for computing the PMF and CDF, respectively, of Lehmann's (1975) nonparametric trend statistic.

Another important function in **skedastic** is `hccme`, which computes an HCCME for a linear regression model based on any of the methods discussed in §2.3.

The functions most central to the objectives of this research project are those that produce point estimates of error variances by implementing ALVMs (`alvm.fit`) and ANLVMs (`anlvm.fit`) and the function that computes bootstrap confidence intervals for error variances in conjunction with an ALVM or ANLVM (`avm.ci`). There are, naturally, various supporting functions that had to be created to implement ALVMs and ANLVMs and compute bootstrap CIs based on them. These include functions pertaining to tuning of hyperparameters (including implementation of CV and QGCV routines), functions pertaining to feature selection, functions pertaining to clustering and elbow methods, and functions pertaining to bootstrapping of linear regression models and auxiliary variance models.

Now that the methods developed for this research *and* the R package developed to implement them have been discussed, the stage is set for the Results and Discussion chapter, where the methods in the Methodology chapter are applied in Monte Carlo simulations to evaluate their performance empirically.

5 Results and Discussion

In this chapter, results are reported and discussed for a variety of Monte Carlo simulations conducted to empirically evaluate the performance of methods discussed in the previous chapters. An overview of the chapter is as follows. In §5.1, the performance of some of the heteroskedasticity tests discussed in Chapter 2 and the new heteroskedasticity test introduced in §3.5 is evaluated empirically using a metric called Average Excess Power over Size (AEPS).

In §5.2, the design of the main MC experiment—looking at the performance of the ALVMs—is described. A number of metrics that are used to measure model performance are also introduced; in particular, four Mean Squared Error metrics. Methods for estimating the standard errors of MC estimates of metrics are also described in this subsection.

Section 5.3 presents the results of the main MC simulation evaluating the performance of the newly developed ALVMs and ANLVMs. The results presented are for simulations with $n = 100$ observations, with all covariates generated independently from uniform distributions, and with the number of covariates $p - 1$ varied between 1, 2, 8, and 16. To keep the volume of tables of results manageable, results under some other simulation settings (e.g., smaller and larger sample sizes; correlated covariates; non-normal errors; etc.) have been relegated to Appendix E.

The results from several supplementary MC simulations, designed to check the performance of other aspects of the ALVMs are reported in §5.4. These aspects are, specifically, the effectiveness of certain feature selection techniques proposed in §3.3.3 and the stability of the ALVMs when the design matrix $\boldsymbol{X}$ is allowed to vary.

Section 5.5 presents results on a MC simulation looking at coverage probabilities of the bootstrap methods described in §3.4 for obtaining approximate CIs for the error variances.

Finally, §5.6 explores the application of ALVMs to three real data sets, for illustrative purposes.

It should to be noted here that the simulation results presented in this chapter for ANLVMs are far less extensive than those for ALVM. There are two reasons for this. One is that the ANLVMs (apart from the clustering ANLVM) require stronger assumptions than the ALVMs, as one must specify the heteroskedastic function $g(\cdot)$. Preliminary simulations show that the performance of the ANLVMs can suffer massively when the form of $g(\cdot)$ is mis-specified, and given that the heteroskedastic function would seldom be known in practice, this is a significant limitation. The second reason is that the ANLVMs are slower to fit than the ALVMs, due to the need to run the Gauss-Newton algorithm for MQL estimation over a grid of initial parameter values to increase the chances of convergence.

5.1 Comparing the Performance of Heteroskedasticity Tests

A significant number of simulation studies have been published over the years on the relative performance of different heteroskedasticity tests in terms of size, power, and robustness (e.g., Griffiths and Surekha 1986, Evans 1992, Lyon and Tsai 1996, Godfrey and Orme 1999, Adamec 2017, Uyanto 2019). Dufour et al. (2004) provide a systematic review of empirical studies up to that time.

Table 5.1 summarises the design of some of the past MC simulation experiments studying heteroskedasticity. The 'additive,' 'multiplicative,' and 'log-multiplicative' heteroskedastic functions are as indicated in (2.17), (2.18), and (2.19). The sinusoidal heteroskedastic function used by Li and Yao (2019) is of the form $g(\boldsymbol{Z};\boldsymbol{\gamma}) = \left(1+\boldsymbol{\gamma}'\left[\sin\left(10X_{i1}\right), \sin\left(10X_{i2}\right), \ldots, \sin\left(10X_{ip}\right)\right]'\right)^2$. Unlike the three other heteroskedastic functions, this one is nonmonotonic in the covariates.

For all studies that mentioned the point, the design matrix was held fixed across MC replications, with the exception of Li and Yao (2019), whose heteroskedasticity test is derived on the basis of a random, multivariate normally distributed design matrix. The number of replications used in the MC simulation experiments in these studies varied from 1000 to 20000.

5.1.1 A New Monte Carlo Simulation of Heteroskedasticity Test Performance

A shortcoming of all of the past MC simulation experiments described in Table 5.1 concerns the performance metric used to evaluate the heteroskedasticity tests. These studies considered empirical power (the proportion of replications under a heteroskedastic DGP for which the null hypothesis was rejected) and, in some cases, empirical size (the proportion of replications under a homoskedastic DGP for which the null hypothesis was rejected), both at one particular nominal size level. This is problematic for two reasons. Firstly, evidence that a test achieves higher power than another test at a particular nominal size level is not necessarily evidence

Table 5.1: Settings of Past Monte Carlo Simulation Experiments on Heteroskedasticity in Linear Regression

Study	n	p	Het. Function(s)	Design Dist.	Other
Griffiths and Surekha (1986)	20; 50	2	additive; log-multiplicative	uniform; lognormal	
Evans (1992)	24; 64	3	additive	uniform; lognormal; normal	Non-normal error distributions used (t; lognormal; chi-square; uniform)
Lyon and Tsai (1996)	20; 30; 50; 100	2	multiplicative	uniform; contaminated uniform; normal	Non-normal error distributions used (t; contaminated normal)
Godfrey and Orme (1999)	40; 80	4	additive; multiplicative; exponential	uniform; lognormal	autocorrelated design points used; Non-normal error distributions used (t; lognormal; chi-square; mixture normal)
Dufour et al. (2004)	50; 100	6	additive; grouped	uniform	considered one vs. all covariates involved in heteroskedasticity
Adamec (2017)	10; 30; 50; 70	2	additive	uniform	
Li and Yao (2019)	100; 500; 1000	$p/n =$ 0.05; 0.1; 0.3; 0.5; 0.7; 0.9	additive; multiplicative; sinusoidal	normal	design matrix varied in each MC replication; considered 1 and $0.1p$ as number of covariates involved in heteroskedasticity
Uyanto (2019)	10; 30; 60; 90; 120; 150	2	various (mostly monotonic)	normal	

that the same holds true at other nominal size levels. (For an illustration of this, see Figure 1 in Lloyd (2005)). Secondly, suppose Test A achieves higher power than Test B at a given nominal size level but Test A is empirically oversized while Test B adheres to the nominal size well. The results are then ambiguous: it is impossible to say whether the superior power of A is outweighed by its inferior fidelity to nominal size.

Lloyd (2005) proposes an Average Excess Power over Size (AEPS) metric that addresses both of these problems. This author defines, for a continuous test statistic T, a survivor function $\mathcal{G}(t) = \Pr(T \geq t)$. The critical value for the test is defined as $c^{\star} = \inf\{t : \mathcal{G}(t) \leq \alpha^{\star}\}$, where $\alpha^{\star}$ is the significance level. Denote the true null distribution of T by $\mathcal{G}_0(t)$ and, for a given alternative hypothesis, the true alternative distribution of T by $\mathcal{G}_1(t)$. A Receiver Operating Characteristic (ROC) curve is a plot of the size $\alpha = \mathcal{G}_0(c^{\star})$ (horizontal axis) against the power $1-\beta = \mathcal{G}_1(c^\star)$ (vertical axis) for different values of $c^\star$ (or, equivalently, nominal size $\alpha^\star$). Specifically, $c^\star$ is allowed to vary from a value sufficiently large so that both $\alpha = 0$ and $1-\beta = 0$ (a certain Type II error) to a value sufficiently small so that $\alpha = 1$ and $1-\beta = 1$ (a certain Type I error). Thus, the points $(0, 0)$ and $(1, 1)$ always fall on the ROC curve. It is customary also to draw a 45 degree line on the plot connecting these two points, as this represents a completely non-informative test statistic for which $\mathcal{G}_0(c^\star) = \mathcal{G}_1(c^\star)$ (the distribution of T is the same under both hypotheses). The extent to which the curve rises above this line and approaches the upper left corner of the plot is thus a graphical representation of the test's performance over different nominal sizes. For a simple example of an ROC curve, see Figure 5.1, which shows the performance of a one-sample t-test for a particular effect size with sample size $n = 10$ vs. $n = 20$. The $n = 20$ curve is closer to the top left and is everywhere above the $n = 10$ curve and therefore dominates it.

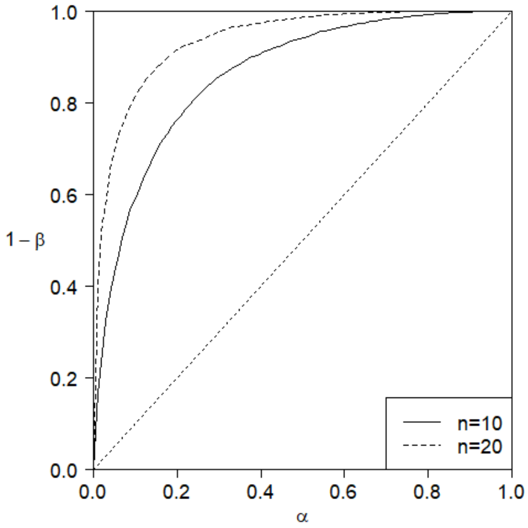


Figure 5.1: Example of a Receiver Operating Characteristic Curve for a One-Sample t-Test

Lloyd (2005) shows that the ROC curve function can be written explicitly as

$$R(a) = \mathcal{G}_1\left\{\mathcal{G}_0^{-1}(a)\right\}. \tag{5.1}$$

He further observes that a generalised metric for the performance of a hypothesis test is obtained by computing the average height $W(l, u)$ of $R(a)$ over an interval of relevant sizes $[l, u]$ and then subtracting the average size over this interval, $(l + u)/2$. The resulting quantity is the AEPS,

$$Q(l, u) = W(l, u) - (l + u)/2. \tag{5.2}$$

A sensible choice of $[l, u]$ might be $[0.01, 0.1]$, since practitioners are seldom interested in sizes outside this interval.

The advantages of Lloyd's (2005) metric are twofold. First, the metric is averaged over a range of relative sizes rather than being valid for only one arbitrarily chosen size value. Second, and more importantly, the metric takes into account *both* size performance and power performance simultaneously, in contrast to many power simulation studies (such as those cited in §5.1) that compare several hypothesis testing methods in terms of power even though not all are equally capable of meeting the nominal size.

Lloyd (2005) proposes a simple way to estimate (5.1), and thus (5.2), using a MC simulation. First, one generates values of the test statistic T under both the null and alternative hypotheses; call these $\boldsymbol{t}_0$ and $\boldsymbol{t}_1$, respectively. One then estimates $W(l, u)$ using a transformation of the Mann-Whitney test statistic (Mann and Whitney 1947), which is typically used for a well-known nonparametric two-sample test of location. The first 'sample' consists of $\boldsymbol{t}_0^{(l,u)}$, the subset of $\boldsymbol{t}_0$ that falls between its l and u empirical quantiles. The second 'sample' is the full vector $\boldsymbol{t}_1$ of values generated under the alternative hypothesis. Subtracting the average size $(l + u)/2$ yields the AEPS estimate,

$$\hat{Q}(l,u) = (n_1 n_2)^{-1}\left(R_1 - n_2(n_2+1)/2\right) - (l+u)/2, \tag{5.3}$$

where R_1 denotes the sum of the ranks of $\boldsymbol{t}_1$ computed from the combined sample, and n_1 and n_2 are the length of $\boldsymbol{t}_0^{(l,u)}$ and $\boldsymbol{t}$, respectively.[110] This approach assumes there are no ties among the $\boldsymbol{t}_0^{(l,u)}$ and $\boldsymbol{t}_1$ values. Since $0 \leq R_1 - n_2(n_2+1)/2 \leq n_1 n_2$, the factor $(n_1 n_2)^{-1}$ transforms the Mann-Whitney statistic onto the interval $[0,1]$.

Lloyd (2005) notes that in practice one would often wish to compare two methods (say, A and B) and would thus need to estimate the standard error of the difference between the two AEPS estimates, $\hat{Q}_{\mathrm{A}}(l,u) - \hat{Q}_{\mathrm{B}}(l,u)$. Since $\hat{Q}_{\mathrm{A}}(l,u)$ and $\hat{Q}_{\mathrm{B}}(l,u)$ will generally be dependent, he suggests using nonparametric bootstrap for this purpose. However, an alternative approach, as will be discussed below in §5.2.3, is simply to compute $\hat{Q}_{\mathrm{A}}(l,u)$ and $\hat{Q}_{\mathrm{B}}(l,u)$ from two separate, independently generated sets of MC replications.

5.1.1.1 Empirical Performance of Deflator-Based Tests

Two MC simulations of heteroskedasticity test performance are undertaken herein. The first focuses on those tests that rely on prior knowledge of a putative 'deflator'. These tests are those of Goldfeld and Quandt (1965) (both the parametric F test and the nonparametric peaks test), Ramsey (1969), Szroeter (1978), Breusch and Pagan (1979),[111] Harrison and McCabe (1979), Horn (1981), Evans and King (1988) (both the LM test and the GLS test), Honda (1989), and Carapeto and Holt (2003), as well as the new ALVM-based test introduced in §3.5, both using the clustering model with $n_c = 2$, and using the linear model.

The simulation used a Data Generating Process (DGP) with two covariates, both generated independently from $U(0,5)$. Sample size was varied from $n = 20$ to $n = 100$ in increments of 10. Errors were generated independently from $N(0,1)$ for the null case and from $N(0,\omega_i)$ with $\omega_i = g(x_{2i}) = (1 + x_{2i}/2)^2$ for the alternative case. Importantly, all of the deflator-based tests were implemented with the correct choice of the deflator and the direction of its relationship to the error variances. The test statistic for each test was computed for $R = 10^4$ MC replications under both the null and alternative cases, and the Mann-Whitney statistic was then computed and used to estimate the AEPS, $\hat{Q}(0.01, 0.1)$. Note that, since $(l+u)/2 = 0.055$ in this case, the possible range of $\hat{Q}(0.01, 0.1)$ is the interval $[-0.055, 0.945]$.

Standard errors for each AEPS metric were computed using $B = 500$ bootstrap samples. The bootstrap-estimated Standard Errors (SEs) are not all reported, but ranged between approximately 2×10^{-4} and 1×10^{-2}, with a median of 6.7×10^{-3}. The AEPS estimates can be regarded as accurate to within roughly one percentage point.

[110] Thus n_2 is the number of MC replications, and n_1 is approximately $(u-l)/n_2$.

[111] Breusch and Pagan (1979) is technically not a deflator-based test, but due to its popularity a deflator version is included whereby the auxiliary design matrix consists of an intercept and two covariates, the deflator and its square.

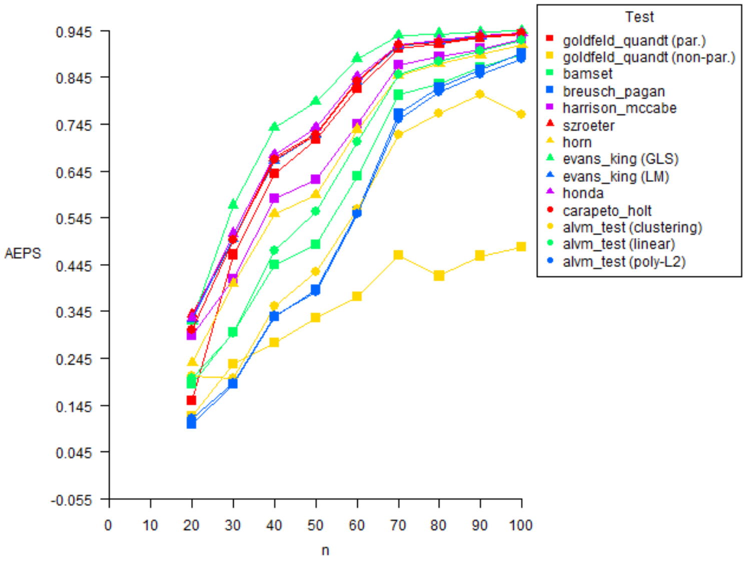


Figure 5.2: Monte Carlo Estimates of Average Excess Power over Size for Deflator-Based Heteroskedasticity Tests

Empirical AEPS estimates are shown, in Figure 5.2, for the 13 different tests. It is evident that the GLS test of Evans and King (1988) performs best, especially when $30 \leq n \leq 60$. A host of other contenders appear to be second-best with little difference between them, including Evans and King's (1988) LM test, Honda's (1989) test, Carapeto and Holt's (2003) test, and Szroeter's (1978) test. Goldfeld and Quandt's (1965) parametric F test performs poorly at $n = 20$ but joins the leaders for larger sample sizes. Harrison and McCabe's (1979) and Horn's (1981) tests and the ALVM-based test (linear version) show mediocre performance, while the lower-performing tests in this simulation include Ramsey's (1969) BAMSET test, Breusch and Pagan's (1979) test (deflator version), the polynomial and clustering versions of the ALVM-based test, and Goldfeld and Quandt's (1965) nonparametric peaks test.

The clustering version of the ALVM-based test, using only $n_c = 2$ clusters, is not as well-designed as some of the other methods to capture a smooth, monotonic heteroskedastic function of one of the covariates. It may perform better at detecting nonmonotonic heteroskedasticity, or as an omnibus test in higher dimensions.

5.1.1.2 Empirical Performance of Omnibus Tests

A similar simulation, with the same DGP, was used to estimate the AEPS of omnibus heteroskedasticity tests, namely, those that seek to make a general judgment about the presence or absence of heteroskedasticity, without positing an association between the error variances and a particular covariate. (Some of these tests make use of an auxiliary design matrix and thus do posit an association between the error variances and at least one of the covariates).

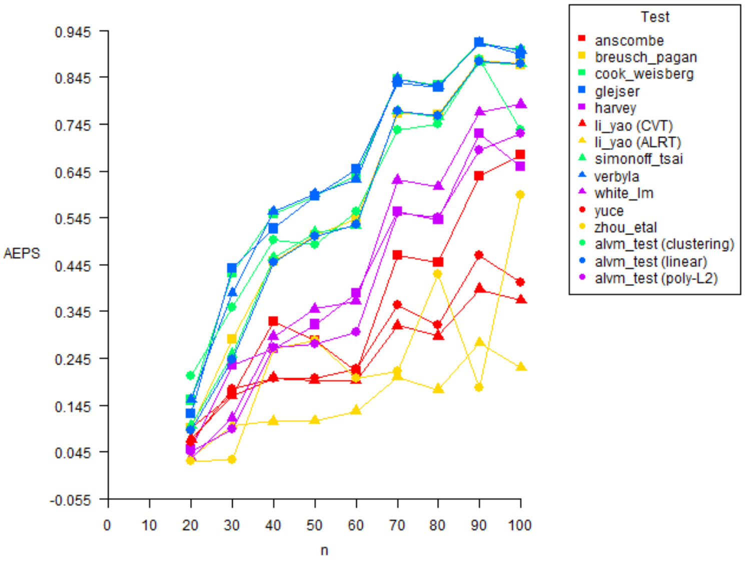


Figure 5.3: Monte Carlo Estimates of Average Excess Power over Size for Omnibus Heteroskedasticity Tests

In Figure 5.3, the top tier of tests include Verbyla's (1993) test, Glejser's (1969) test, and Cook and Weisberg's (1983) test. The ALVM-based tests are in the next tier, along with Breusch and Pagan's (1979) test and Simonoff and Tsai's (1994) score test. Notably, the clustering version of the ALVM-based test has the best AEPS among all tests when $n = 20$. Among the mediocre performers are White's (1980) test, Harvey's (1976) test, Anscombe's (1961) test, and the polynomial version of the ALVM-based test.[112] The weaker performing tests are Yüce's (2008) test, Zhou et al.'s (2015) test, and both tests of Li and Yao's (2019). In fairness to the latter, Li and Yao (2019) designed their test to specialise in high-dimensional regressions, so it is not surprising that it performs relatively poorly in the $p = 3$ case. Bootstrap estimates of standard errors of the AEPS estimates were of a similar magnitude to those of the deflator-based tests.

It is interesting to observe that the top-performing deflator-based and omnibus tests for heteroskedasticity, according to this admittedly limited experiment, are not among the most popular or widely cited tests. Indeed, to this author's knowledge, the tests of Evans and King (1988) and Verbyla (1993) were not available in any statistical software until the author deployed them in the **skedastic** R package.

5.2 Design of Monte Carlo Experiments for Evaluating the Performance of the Auxiliary Variance Models

This subsection describes the design of a Monte Carlo (MC) simulation experiment that has been undertaken to investigate the performance of the ALVMs and ANLVMs as estimators of error variances under different circumstances, in comparison to existing methods. Specifically, the methods to which comparisons are made are the classical methods of estimation and inference under assumptions A1-A5, the HCCMEs discussed in §2.3, and Miller and Startz's (2019) SVR auxiliary modelling procedure (discussed in §2.2.1.3). All of the ALVMs and ANLVMs introduced in §3.2.2-§3.2.4 are used in the simulations except for the B-spline and smoothing spline ALVMs. These are omitted because they are only applicable in the univariate case, and because the thin-plate spline is equivalent to the smoothing spline in the univariate case.

Unless otherwise indicated, the penalty parameter λ for the penalised polynomial ALVMs was tuned using five-fold CV, while the number of clusters n_c for the clustering ALVM and the clustering ANLVM was tuned using the elbow method with the SWD criterion. Feature selection was performed on the linear and clustering

[112] For this simulation, the QGCV method was used rather than K-fold CV to choose the penalty hyperparameter λ. This was to save computation time, although it is known that the CV approach works better. Thus, higher AEPS may be achieved by the polynomial ALVM if CV were used instead of QGCV.

ALVMs, and on all of the ANLVMs, using best subset selection in terms of QGCV loss computed on the linear ALVM.

The number of MC replications used for each factor combination in this experiment was $R = 10^4$, unless otherwise indicated. Within each factor combination, the design matrix $\boldsymbol{X}$ was held fixed across all MC replications. This is in line with the statement in §1.1.3 that all statistical results in this study are conditioned on $\boldsymbol{X}$. However, the further simulations discussed in §5.4.2 serve as a robustness check to ensure that the models' performance is not too sensitive to the particular form of $\boldsymbol{X}$.

5.2.1 Experimental Factors

The factors of interest and associated factor levels are summarised in Table 5.2.

Table 5.2: Factors and Factor Levels Used in Monte Carlo Experiment

Factor	Factor Levels
Sample Size n	20; 100; 1000
Number of Design Variables $p-1$	1; 2; 8; 16
Design Distribution	Independent Uniform; Correlated Normal

Heteroskedastic Function	Constant (homoskedastic);
	Additive (quadratic, as in (2.17));
	Multiplicative (exponential, as in (2.18))

Number of Design Variables Involved in Heteroskedasticity	1; $(p-1)/2$
Error Distribution	Normal; Laplace; Uniform

The experimental design is not even close to full factorial. Some factor combinations do not apply (e.g., 'number of design variables involved' does not apply under homoskedasticity). Moreover, implementing too many factor combinations would not only require a massive amount of computation time, but would lead to too many results. The factor combinations that were used are summarised in Table 5.3, which also indicates whether a particular factor combination was used for both ALVMs and ANLVMs or for ALVMs only.

Table 5.3: Factor Combinations Used in the Monte Carlo Simulation Experiment

Factor Combination	Location of Results	Run for ALVMs	Run for ANLVMs
$n = 20$, $p - 1 = 1$ Covariate	Appendix E.1	Yes	Yes
$n = 100$, $p - 1 = 1$ Covariate	§5.3.1	Yes	Yes
$n = 1000$, $p - 1 = 1$ Covariate	Appendix E.2	Yes	Yes ($R = 10^3$)
$n = 100$, $p - 1 = 1$ Covariate, Nonmonotonic Heteroskedasticity	Appendix E.3	Yes	Yes
$n = 100$, $p - 1 = 1$ Covariate, Non-Normal Errors	Appendix E.4	Yes	Yes
$n = 100$, $p - 1 = 2$ Independent Covariates	§5.3.2	Yes	Yes
$n = 100$, $p - 1 = 2$ Correlated Covariates	Appendix E.5	Yes	No
$n = 100$, $p - 1 = 8$ Independent Covariates	§5.3.3	Yes	Yes
$n = 100$, $p - 1 = 8$ Correlated Covariates	Appendix E.6	Yes	No
$n = 100$, $p - 1 = 16$ Independent Covariates	§5.3.4	Yes	No

The error distribution was always normal, with the exception of the simple linear regression simulation based on Laplace and uniform errors discussed in §5.3.1.7, with results tables in Appendix E.4. The three heteroskedastic functions mentioned in Table 5.2 were all considered in every case; a nonmonotonic (sinusoidal) heteroskedastic function was used in one simple linear regression simulation (see §5.3.1.6 and Appendix E.3). The two levels for number of design variables involved in heteroskedasticity were used in all cases where the number of predictors exceeded two.

5.2.2 Performance Metrics

The metrics used to evaluate the performance of the heteroskedastic variance estimators are as follows. Throughout this subsection, R denotes the number of MC replications used in the experiment.

5.2.2.1 Unstandardised Mean Squared Error for Individual Variance Estimates

If $\hat{\omega}_i^{(r)}$ is an estimate of the ith error variance ω_i from the rth MC replication, the unstandardised empirical MSE for an individual variance estimate is computed as,[113]

$$\mathrm{MSE}_{\mathrm{ust}}(\hat{\omega}_i) = \frac{1}{R}\sum_{r=1}^{R}\left(\hat{\omega}_i^{(r)} - \omega_i\right)^2, i = 1, 2, \ldots, n. \tag{5.4}$$

By taking the mean of (5.4) across all n observations, one obtains an overall (unstandardised) MSE metric:

$$\overline{\mathrm{MSE}}_{\mathrm{ust}}(\hat{\boldsymbol{\omega}}) = \frac{1}{n}\sum_{i=1}^{n}\mathrm{MSE}_{\mathrm{ust}}(\hat{\omega}_i). \tag{5.5}$$

It is also necessary to define the mean MSE estimate for a particular replication, which will be used in computing a standard error estimate for (5.5):

[113] Note that (5.4) represents a MC *estimate* of the true unknown MSE. For notational convenience no $\widehat{}$ is displayed. The same goes for the rest of the metrics in this section.

$$\mathrm{MSE}_{\mathrm{ust}}(\hat{\boldsymbol{\omega}})^{(r)} = \frac{1}{n}\sum_{i=1}^{n}\left(\hat{\omega}_i^{(r)} - \omega_i\right)^2. \tag{5.6}$$

(5.4) can also be decomposed into squared bias and variance components, where the bias and variance are given by

$$\mathrm{Bias}_{\mathrm{ust}}(\hat{\omega}_i) = \bar{\hat{\omega}}_i - \omega_i \tag{5.7}$$

and

$$\mathrm{Var}_{\mathrm{ust}}(\hat{\omega}_i) = \frac{1}{R-1}\sum_{r=1}^{R}\left(\hat{\omega}_i^{(r)} - \bar{\hat{\omega}}_i\right)^2, \tag{5.8}$$

where $\bar{\hat{\omega}}_i = \frac{1}{R}\sum_{r=1}^{R}\hat{\omega}_i^{(r)}$. Averaged-out versions of the bias and variance analogous to (5.5), i.e., $\overline{\mathrm{Bias}}_{\mathrm{ust}}(\hat{\boldsymbol{\omega}})$ and $\overline{\mathrm{Var}}_{\mathrm{ust}}(\hat{\boldsymbol{\omega}})$, can also be computed.

5.2.2.2 Standardised Mean Squared Error for Individual Variance Estimates

Standardised versions of the above metrics are obtained by considering $\hat{\omega}_i/\omega_i$ and comparing it to $\omega_i/\omega_i = 1$, so that each error variance carries equal weight toward the metric, regardless of its magnitude. The standardised MSE is thus

$$\mathrm{MSE}_{\mathrm{st}}(\hat{\omega}_i) = \frac{1}{R}\sum_{r=1}^{R}\left(\frac{\hat{\omega}_i^{(r)}}{\omega_i} - 1\right)^2 = \frac{1}{\omega_i^2}\mathrm{MSE}_{\mathrm{ust}}, i = 1, 2, \ldots, n. \tag{5.9}$$

(5.9) can likewise be decomposed into squared bias and variance components, where the bias and variance are given by

$$\mathrm{Bias}_{\mathrm{st}}(\hat{\omega}_i) = \frac{\bar{\hat{\omega}}_i}{\omega_i} - 1 = \frac{1}{\omega_i}\mathrm{Bias}_{\mathrm{ust}}(\hat{\omega}_i), \text{ and} \tag{5.10}$$

$$\mathrm{Var}_{\mathrm{st}}(\hat{\omega}_i) = \frac{1}{R-1}\sum_{r=1}^{R}\left(\frac{\hat{\omega}_i^{(r)}}{\omega_i} - \frac{\bar{\hat{\omega}}_i}{\omega_i}\right)^2 = \frac{1}{\omega_i^2}\mathrm{Var}_{\mathrm{ust}}(\hat{\omega}_i). \tag{5.11}$$

One can again take the mean of (5.9) across all n observations to obtain an overall standardised MSE metric, $\overline{\mathrm{MSE}}_{\mathrm{st}}(\hat{\boldsymbol{\omega}})$:

$$\overline{\mathrm{MSE}}_{\mathrm{st}}(\hat{\boldsymbol{\omega}}) = \frac{1}{n}\sum_{i=1}^{n}\mathrm{MSE}_{\mathrm{st}}(\hat{\omega}_i). \tag{5.12}$$

As before, one can likewise compute $\overline{\mathrm{Bias}}_{\mathrm{st}}(\hat{\boldsymbol{\omega}})$ and $\overline{\mathrm{Var}}_{\mathrm{st}}(\hat{\boldsymbol{\omega}})$, by averaging (5.10) and (5.11), respectively, across all n observations.

The standardised versions of these metrics are superfluous in the homoskedastic case, since they only differ by a constant scaling factor from the unstandardised versions. In other cases, results on unstandardised and standardised versions of the metrics will be reported separately.

5.2.2.3 Mean Squared Error Metric for FWLS Estimation of β

Let $\hat{\boldsymbol{\beta}}_{\mathrm{FWLS}}^{(r)} = (\boldsymbol{X}'\left[\hat{\boldsymbol{\Omega}}^{(r)}\right]^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}'\left[\hat{\boldsymbol{\Omega}}^{(r)}\right]^{-1}\boldsymbol{y}^{(r)}$ be a feasible weighted least squares estimator of $\boldsymbol{\beta}$ based on the error variance estimate vector $\hat{\boldsymbol{\omega}}^{(r)}$ from the rth MC replication (where $\hat{\boldsymbol{\Omega}}^{(r)} = \operatorname{diag}\left\{\hat{\boldsymbol{\omega}}^{(r)}\right\}$). Then, a performance metric for this FWLS estimate is,

$$\mathrm{MSE}(\hat{\boldsymbol{\beta}}_{\mathrm{FWLS}}) = \frac{1}{R}\sum_{r=1}^{R}\frac{1}{p}\left|\left|\hat{\boldsymbol{\beta}}_{\mathrm{FWLS}}^{(r)} - \boldsymbol{\beta}\right|\right|_2^2. \tag{5.13}$$

5.2.2.4 Mean Squared Error Metric for HCCME Estimation of $\mathrm{SE}(\hat{\beta}_j)$

The size performance of a quasi-t-test of hypothesis on an element of $\boldsymbol{\beta}$ (e.g., $H_0 : \beta_j = 0$) depends on obtaining a good estimate of $\mathrm{SE}(\hat{\beta}_j) = \sqrt{\mathrm{Var}(\hat{\beta}_j)}$, a diagonal element of (1.6), by replacing $\boldsymbol{\Omega}$ in (1.6) with a suitable estimator $\hat{\boldsymbol{\Omega}}$. A suitable metric for evaluating performance in this case is

$$\mathrm{MSE}(\widehat{\mathrm{SE}}(\hat{\boldsymbol{\beta}})) = \frac{1}{R}\sum_{r=1}^{R}\frac{1}{p}\left|\left|\widehat{\mathrm{SE}}(\hat{\boldsymbol{\beta}})^{(r)} - \mathrm{SE}(\hat{\boldsymbol{\beta}})\right|\right|_2^2, \tag{5.14}$$

where $\mathrm{SE}(\hat{\boldsymbol{\beta}}) = \sqrt{\mathrm{diag}\left((\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{\Omega}\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\right)}$ and $\widehat{\mathrm{SE}}(\hat{\boldsymbol{\beta}})^{(r)} = \sqrt{\mathrm{diag}\left((\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\hat{\boldsymbol{\Omega}}^{(r)}\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\right)}$, the square root being applied elementwise.

It was previously discussed in connection with (3.75) that accurate estimation of this standard error is not just about accurate estimation of the elements of $\boldsymbol{\omega}$ 'on average'; it is about accurate estimation of those elements that figure most prominently in the sandwich estimator.

5.2.3 Estimating Standard Errors of Monte Carlo Estimates

To reiterate, $R = 10^4$ MC replications were generated for each factor combination in this experiment, with one exception. In order to assess statistical significance of differences between MC mean estimates of quantities of interest such as those described above, standard error estimates are needed—not only of the MC mean estimates but also of the differences between them. Some discussion follows of the method used to compute these standard error estimates.

Let $\hat{\theta}_1^{(r)}$ and $\hat{\theta}_2^{(r)}$ be two estimators of some unknown quantity θ based on R randomly generated data sets indexed by r, and suppose that $\mathrm{Var}(\hat{\theta}_1^{(r)}) = \sigma_1^2$ and $\mathrm{Var}(\hat{\theta}_2^{(r)}) = \sigma_2^2$. Let $\bar{\hat{\theta}}_1 = R^{-1}\sum_{r=1}^{R}\hat{\theta}_1^{(r)}$, the MC mean, be an estimate of $\mathrm{E}\left(\hat{\theta}_1\right)$ and define $\bar{\hat{\theta}}_2$ analogously.

Using basic properties of the variance operator under independence, it follows that $\mathrm{Var}(\bar{\hat{\theta}}_1) = \sigma_1^2/R$ and $\mathrm{Var}(\bar{\hat{\theta}}_2) = \sigma_2^2/R$. A MC estimate of $\mathrm{SE}(\bar{\hat{\theta}}_1)$ is given by

$$\widehat{\mathrm{SE}}\left(\bar{\hat{\theta}}_1\right) = \left[\frac{\hat{\sigma}_1^2}{R}\right]^{1/2} = \left[\frac{1}{R(R-1)}\sum_{r=1}^{R}\left(\hat{\theta}_1^{(r)} - \bar{\hat{\theta}}_1\right)^2\right]^{1/2}, \tag{5.15}$$

and similarly for $\mathrm{SE}(\bar{\hat{\theta}}_2)$. However, if the goal is to demonstrate a statistically significant difference between $\mathrm{E}\left(\hat{\theta}_1\right)$ and $\mathrm{E}\left(\hat{\theta}_2\right)$, that is, that $\mathrm{E}\left(\hat{\theta}_1 - \hat{\theta}_2\right) \neq 0$, the real quantity of interest is $\mathrm{SE}(\bar{\hat{\theta}}_1 - \bar{\hat{\theta}}_2)$. Here arises the issue of whether or not $\hat{\theta}_1^{(r)}$ and $\hat{\theta}_2^{(r)}$ are independent. One can ensure independence simply by computing the two estimates from separate and independently drawn data sets for each $r = 1, 2, \ldots, R$.[114] In this case, $\mathrm{Var}\left(\bar{\hat{\theta}}_1 - \bar{\hat{\theta}}_2\right) = \dfrac{\sigma_1^2 + \sigma_2^2}{R}$, and a MC estimate of $\mathrm{SE}(\bar{\hat{\theta}}_1 - \bar{\hat{\theta}}_2)$ is given by

$$\begin{aligned}\widehat{\mathrm{SE}}\left(\bar{\hat{\theta}}_1 - \bar{\hat{\theta}}_2\right) &= \left[\frac{\hat{\sigma}_1^2 + \hat{\sigma}_2^2}{R}\right]^{1/2} = \left[\left(\widehat{\mathrm{SE}}\left(\bar{\hat{\theta}}_1\right)\right)^2 + \left(\widehat{\mathrm{SE}}\left(\bar{\hat{\theta}}_2\right)\right)^2\right]^{1/2} \\ &= \left[\frac{1}{R(R-1)}\sum_{r=1}^{R}\left\{\left(\hat{\theta}_1^{(r)} - \bar{\hat{\theta}}_1\right)^2 + \left(\hat{\theta}_2^{(r)} - \bar{\hat{\theta}}_2\right)^2\right\}\right]^{1/2}.\end{aligned} \tag{5.16}$$

If, on the other hand, $\hat{\theta}_1^{(r)}$ and $\hat{\theta}_2^{(r)}$ are computed using the same random data set, they cannot be treated as independent. In this case, $\mathrm{Var}\left(\bar{\hat{\theta}}_1 - \bar{\hat{\theta}}_2\right) = \dfrac{\sigma_1^2 + \sigma_2^2}{R} - 2\,\mathrm{Cov}\left(\bar{\hat{\theta}}_1, \bar{\hat{\theta}}_2\right)$. Now, making use of basic properties of the covariance of linear combinations of random variables, it can be shown that

[114] In the application at hand, this means drawing the random errors $\boldsymbol{\epsilon}$ independently for each MC replication. The design matrix $\boldsymbol{X}$ remains fixed across all replications and estimation methods.

$\mathrm{Cov}(\bar{\hat{\theta}}_1, \bar{\hat{\theta}}_2) = R^{-2} \sum_{r=1}^{R} \sum_{s=1}^{R} \mathrm{Cov}(\hat{\theta}_1^{(r)}, \hat{\theta}_2^{(s)})$. However, by the mutual independence of the MC replications, $\mathrm{Cov}(\hat{\theta}_1^{(r)}, \hat{\theta}_2^{(s)}) = 0$ for $r \neq s$. Thus,

$$\begin{aligned} \mathrm{Cov}(\bar{\hat{\theta}}_1, \bar{\hat{\theta}}_2) &= \frac{1}{R^2} \sum_{r=1}^{R} \mathrm{Cov}(\hat{\theta}_1^{(r)}, \hat{\theta}_2^{(r)}) \\ &= \frac{1}{R} \mathrm{Cov}(\hat{\theta}_1^{(r)}, \hat{\theta}_2^{(r)}). \end{aligned} \tag{5.17}$$

A MC estimate of (5.17) is obtained by substituting the empirical MC covariance estimate:

$$\widehat{\mathrm{Cov}}(\hat{\theta}_1^{(r)}, \hat{\theta}_2^{(r)}) = (R-1)^{-1} \sum_{r=1}^{R} \left(\hat{\theta}_1^{(r)} - \bar{\hat{\theta}}_1\right) \left(\hat{\theta}_2^{(r)} - \bar{\hat{\theta}}_2\right).$$

It follows that, where $\hat{\theta}_1^{(r)}$ and $\hat{\theta}_2^{(r)}$ are both computed from the same random sample,

$$\begin{aligned} \widehat{\mathrm{SE}}\left(\bar{\hat{\theta}}_1 - \bar{\hat{\theta}}_2\right) &= \left[\frac{1}{R(R-1)} \sum_{r=1}^{R} \left\{ \left(\hat{\theta}_1^{(r)} - \bar{\hat{\theta}}_1\right)^2 + \left(\hat{\theta}_2^{(r)} - \bar{\hat{\theta}}_2\right)^2 - 2\left(\hat{\theta}_1^{(r)} - \bar{\hat{\theta}}_1\right)\left(\hat{\theta}_2^{(r)} - \bar{\hat{\theta}}_2\right) \right\}\right]^{1/2} \\ &= \left[\frac{1}{R(R-1)} \sum_{r=1}^{R} \left\{ \left((\hat{\theta}_1^{(r)} - \bar{\hat{\theta}}_1) - (\hat{\theta}_2^{(r)} - \bar{\hat{\theta}}_2)\right)^2 \right\}\right]^{1/2}. \end{aligned} \tag{5.18}$$

Since two similar estimation methods are likely to be positively correlated, it is likely that for the present purposes, use of (5.18) (with multiple estimators computed from the same random sample in each MC replication) will result in smaller standard errors than (5.16) (with each estimator computed from a separately drawn random sample in each MC replication).

On the other hand, an advantage of (5.16) is that it can be easily computed after the fact from stored results of (5.15), using the relation,

$$\mathrm{SE}\left(\bar{\hat{\theta}}_1 - \bar{\hat{\theta}}_2\right) = \left[\mathrm{SE}\left(\bar{\hat{\theta}}_1\right)^2 + \mathrm{SE}\left(\bar{\hat{\theta}}_2\right)^2\right]^{1/2}, \tag{5.19}$$

which does not hold for (5.18). The approach used to assess whether a particular method has a performance metric that is better (smaller) than all others by a statistically significant margin is then as follows (assuming that c different methods are being compared):

1. Compute $\bar{\hat{\theta}}_j$, $j = 1, 2, \ldots, c$, and the corresponding standard error estimates and set $k = \arg\min_j \bar{\hat{\theta}}_j$.
2. Compute $Z_j = \dfrac{\bar{\hat{\theta}}_k - \bar{\hat{\theta}}_j}{\mathrm{SE}\left(\bar{\hat{\theta}}_k - \bar{\hat{\theta}}_j\right)}$, $j = 1, 2, \ldots, c$, $j \neq k$, where the standard error is computed from (5.19).
3. Compare each Z_j to $z_{1-\alpha'}$, the upper $\alpha' = \alpha/(c-1)$ standard normal quantile, where α is the maximum permissible family-wise Type I error probability and $c - 1$ is the number of comparisons being made.[115] (This is a one-tailed test with Bonferroni correction).

$\alpha = 0.05$ will be used throughout the results unless otherwise stated.

[115] If all c methods were to be compared pairwise, the number of pairwise comparisons would be $\binom{c}{2}$. However, in this case the question of interest is whether one particular method's metric—the one with the best (lowest) point estimate—is significantly lower than the other $c - 1$ methods' metrics.

5.2.4 Relative Performance Metrics

It will prove useful, in subsequent results, to report the MC mean estimates of the performance metrics in relative rather than absolute terms. In terms of notation introduced in §5.2.3, if $\bar{\hat{\theta}}_j$, $j = 1, 2, \ldots, c$, are the MC mean estimates of a particular metric for c different methods, the relative MC mean estimate is,

$$\bar{\hat{\theta}}_j^{\mathrm{rel}} = \frac{\bar{\hat{\theta}}_j}{\min\left\{\bar{\hat{\theta}}_1, \bar{\hat{\theta}}_2, \ldots, \bar{\hat{\theta}}_c\right\}}. \tag{5.20}$$

It is clear from (5.20) that the best-performing method will have a $\bar{\hat{\theta}}_j^{\mathrm{rel}}$ value of 1, while all other methods will yield values greater than 1. This allows quick identification of the best-performing method and the magnitude by which other methods underperform relative to this one.

5.2.5 Format of Results Tables and Graphs

For the simple linear regression case ($p = 2$, i.e., an intercept and one predictor), results on the $\overline{\mathrm{MSE}}_{\mathrm{ust}}(\hat{\boldsymbol{\omega}})$ metric (5.5) and the $\overline{\mathrm{MSE}}_{\mathrm{st}}(\hat{\boldsymbol{\omega}})$ metric (5.12) are presented in graphical form. Each of these metrics, along with their squared bias and variance components, are plotted against the predictor variable x_i for all three error variance settings (homoskedasticity; additive and multiplicative heteroskedasticity). These visualisations help to illustrate how the models' performance varies with x_i, and thus also with the magnitude of the error variance.

These MSE plots are not used for factor combinations with multiple predictors ($p > 2$). This is mainly for reasons of conciseness, but also because the relationship between performance metric and predictor will not appear as clearly (in two dimensions, at least) when there are multiple predictors.

The most important results, for both simple and multiple linear regression settings, are displayed in tabular form. For each simulation setting reported on in §5.3 and in Appendix E, four tables appear, each containing results in terms of one of the four main metrics of interest: $\overline{\mathrm{MSE}}_{\mathrm{ust}}(\hat{\boldsymbol{\omega}})$ (Equation 5.5), $\overline{\mathrm{MSE}}_{\mathrm{st}}(\hat{\boldsymbol{\omega}})$ (Equation 5.12), $\mathrm{MSE}(\hat{\boldsymbol{\beta}}_{\mathrm{FWLS}})$ (Equation 5.13), and $\mathrm{MSE}(\widehat{\mathrm{SE}}(\hat{\boldsymbol{\beta}}))$ (Equation 5.14).

The rows of these tables correspond to the different methods: selected HCCMEs, selected ALVMs, and Miller and Startz's (2019) SVR model. The only HCCMEs reported on in these tables are HC3, HC4, and HC6. This is for conciseness, because the results tended to be similar across most HCCMEs (HC6 being the exception), while HC3 and HC4 are probably the most widely used HCCMEs. The ALVMs used in the simulations normally included the basic ALVM (3.34), the linear ALVM (3.40), the clustering ALVM (3.59), the L_2-norm and L_1-norm penalised polynomial ALVMs (3.45) and (3.67), and the thin-plate spline ALVM (C.19). In some instances, particularly in higher dimensions, the L_1-norm (LASSO) penalised polynomial ALVM and the thin-plate spline ALVM were omitted due to their high computation time.

The columns of these results tables correspond to different DGPs—different heteroskedastic functions, for the most part. The column headers indicate the type of heteroskedasticity (e.g., homoskedastic, additive heteroskedasticity, multiplicative heteroskedasticity) as well as the predictors that are related to the error variances through the heteroskedastic function. The latter are indicated using set notation, with $\mathcal{H}$ denoting the indices of predictors involved in heteroskedasticity. $\mathcal{H} = \emptyset$ thus represents homoskedasticity, while $\mathcal{H} = \{2, 3\}$ would indicate that two predictors, with indices 2 and 3, are involved in heteroskedasticity. In keeping with the notation used throughout this document, the index 1 corresponds to the first column of $\boldsymbol{X}$, a column of ones, and not a predictor; hence it does not appear in the $\mathcal{H}$ sets in the results tables.

Each cell in the results tables contains two values. The upper value in the cell is the *relative* MC mean estimate of the metric of interest, as per (5.20). Beneath this, in brackets, is the estimated standard error of the *absolute* MC mean estimate, as per (5.15), expressed in scientific notation.

The cells are also colour-coded for ease of interpretation. The cell for the best-performing method in each column (DGP)—thus having a relative MC mean estimate of 1—has a green background. Cells for methods that are *not* inferior to the best-performing method by a statistically significant margin at (family-wise) level $\alpha = 0.05$, as explained in §5.2.3, have yellow backgrounds. Cells for methods that *are* inferior to the best-performing method by a statistically significant margin have white backgrounds.

The ANLVM results on the four above-mentioned metrics are presented in separate tables from the ALVM results. One reason for this is that, since ANLVMs (other than the clustering ANLVM) entail an assumed specification of the heteroskedastic function $g(\cdot)$, it is not 'fair' to compare the ANLVM results to those of ALVMs in cases where an ANLVM is based on the exact heteroskedastic functional form of the DGP. The tables of ANLVM results are also structured differently, since the small number of models makes it possible to include all four metrics in one table. The rows of these tables are organised firstly by metric, and nested within each metric, by ANLVM. The columns again represent the heteroskedastic functions of the respective DGPs, with $\mathcal{H}$ representing the indices of predictors related to the error variances.

The relative metrics in the ANLVM results tables are computed relative to the lowest metric among the ALVMs used under the same DGPs. Thus, the relative metric for an ANLVM could be less than 1, which would indicate that the this ANLVM outperforms the best-performing ALVM for this experimental setting (in which case the cell is highlighted in green). Statistical significance is also assessed relative to the metric of the best-performing ALVM for purposes of yellow colour-coding.

Table 5.4: Illustration of Table Format for Displaying ALVM Results for Metric 1

Model	DGP 1	DGP 2
Model 1	1 (3.21×10^{-3})	1.17 (1.98×10^{-4})
Model 2	1.28 (2.89×10^{-3})	2.19 (5.58×10^{-4})
Model 3	1.02 (3.07×10^{-3})	1 (4.46×10^{-4})

Examples of the two types of results tables described above are given in Tables 5.4 and 5.5. Table 5.4 shows results for three models (e.g., HCCMEs and/or ALVMs) for two experimental settings (DGPs) in terms of an arbitrary performance metric called Metric 1. The top value in each cell is the MC mean estimate of the metric relative to the best-performing metric (see (5.20)). Thus, the smallest value in each column is always a 1; the cell containing this value is highlighted in green. Cells whose metric value is not greater than (inferior to) that of the best model by a statistically significant margin are highlighted in yellow, while other cells are white. The estimated MC standard error of the absolute metric estimate is displayed in brackets below the corresponding relative MC mean estimate. In the case of DGP 1, Model 1 performs best (hence green), but Model 3 is not inferior by a statistically significant margin (hence yellow). In the case of DGP 2, Model 3 performs best (hence green) and is better than Model 1 and Model 2 by a statistically significant margin (hence they are both white).

Table 5.5: Illustration of Table Format for Displaying ANLVM Results for Metrics 1 and 2

Metric	ANLVM	DGP 1	DGP 2
Metric 1	ANLVM 1	1.42 (1.47×10^{-4})	1.18 (6.14×10^{-4})
	ANLVM 2	1.01 (6.70×10^{-4})	0.939 (4.65×10^{-4})
Metric 2	ANLVM 1	1.36 (3.59×10^{-4})	1.29 (8.59×10^{-4})
	ANLVM 2	1.07 (7.34×10^{-4})	0.965 (3.14×10^{-4})

Table 5.5 shows the results for two ANLVMs under the same DGPs shown in Table 5.4, in terms of two metrics, Metric 1 and Metric 2. Importantly, the relative metric values and background colours in Table 5.5 are not computed relative to these ANLVMs only, but relative to the models in Table 5.4 (for Metric 1) and to another set of results (not shown) for Metric 2. The relative metric values in the DGP 1 column of Table 5.5 are all greater than 1; the absence of any green cell indicates that none of the ANLVMs outperforms all of the other ANLVMs *and the models shown in the DGP 1 column of Table 5.4*. However, the yellow cell for ANLVM 2 in terms of Metric 1 indicates that this model is not inferior to that of Model 1 in Table 5.4 by a statistically significant margin (in terms of Metric 1, under DGP 1). Of the relative metric values in the DGP 2 column of Table 5.5, two are less than 1, and thus highlighted in green. This indicates that ANLVM 2 performs better than ANLVM 1 *and Models 1-3 of Table 5.4* in terms of both Metric 1 and Metric 2 under DGP 2.

Again, the reason for having a self-contained comparison of ALVMs in one table, and adding a second table with ANLVM results that are compared to the first table, is that it is not 'fair' in some sense to declare an

ANLVM the best-performing model under a particular DGP if this ANLVM used information not available to the ALVMs, namely the correct specification of the heteroskedastic function $g(\cdot)$.

5.3 Results of Auxiliary Variance Model Performance Simulations

5.3.1 Linear Regression with One Covariate

Here, a detailed set of results is given for a simulation where the DGP is a simple linear regression model ($p = 2$; an intercept is included in all models). The predictor $\boldsymbol{x}$ was drawn from a $U(0, 3)$ distribution with $n = 100$ observations. In each MC replication, the random error vector $\boldsymbol{\epsilon}$ was drawn independently from a $N(\mathbf{0}, \boldsymbol{\Omega})$ distribution, where $\boldsymbol{\Omega} = \text{diag}\{\boldsymbol{\omega}\}$. The responses were then generated as $\boldsymbol{y} = [\mathbf{1} \ \ \boldsymbol{x}]\,\boldsymbol{\beta} + \boldsymbol{\epsilon}$, where $\boldsymbol{\beta} = [1, 1]'$.[116] For each replication, an OLS fit was computed and $\hat{\boldsymbol{\beta}}_{\text{OLS}}$ and $\boldsymbol{e}$ were obtained.

In the homoskedastic case, the error variance was $\omega_i = \omega = 1$ for all $i \in \{1, 2, \ldots, n\}$. In the additive case, the heteroskedastic function was $\omega_i = g(x_i) = (1 + x_i)^2$, while in the multiplicative case, the heteroskedastic function was $\omega_i = g(x_i) = \exp\{x_i\}$.

Figures 5.4 to 5.8 show the MC estimates of the two MSE metrics for estimating individual error variances, $\text{MSE}_{\text{ust}}(\hat{\omega}_i)$ (Equation 5.4) and $\text{MSE}_{\text{st}}(\hat{\omega}_i)$ (Equation 5.9), along with their squared-bias and variance components, as functions of the single explanatory variable x_i. This allows visualisation of how the model performance varies according to the magnitude of the explanatory variable—and thus the magnitude of the error variance, in the heteroskedastic DGPs. Each set of plots are split into two columns. Results for the HCCMEs and the basic ALVM appear in the left plot, while results for the other ALVMs and the Miller-Startz SVR model appear in the right plot. The reason for this split is that the first set of methods tends to have much poorer results than the second set according to these metrics; so much so that it is not visually appropriate to display all the results on the same plot.

The methods shown in the plots are denoted in the legends as follows. The homoskedastic estimator (denoted `homo.` on the plots) is $\hat{\omega}_i = \hat{\omega}_{\text{ub}}$ for all $i = 1, 2, \ldots, n$. The HCCMEs from §2.3 are denoted using the HC# nomenclature introduced there. `miller` in the plot legends denotes the auxiliary SVR modelling method of Miller and Startz (2019). Coming to the ALVMs, `basic` denotes the 'basic' or 'naïve' ALVM (3.34). In the panels on the right of the figures, `cluster-qgcv.linear` denotes the clustering ALVM (3.59) with number of clusters chosen using the elbow method with the SWD criterion (3.88), and feature selection performed by applying the QGCV metric to the linear model.[117] `linear-qgcv.linear` denotes the linear ALVM (3.40), again with QGCV applied to the linear model for variable selection. `poly-L2-foldCV` denotes the RR (L_2-norm-penalised) polynomial ALVM estimated by (3.46) while `poly-L1-foldCV` denotes the LASSO (L_1-norm-penalised) polynomial ALVM described in (3.67), with the hyperparameter λ tuned using five-fold CV in both cases. `spline-foldCV` denotes the thin-plate spline ALVM estimated by (C.19), likewise with five-fold CV used to tune λ. The SVR hyperparameters for Miller and Startz's (2019) model were tuned exactly according to their own R code.

A set of three plots is given for each of the three DGP scenarios (homoskedasticity; additive heteroskedasticity; multiplicative heteroskedasticity). Following the plots, Tables 5.6 to 5.9 summarise the results of this part of the simulation in terms of the four main metrics of interest. This system of four tables is used throughout the ALVM performance results across different factor combinations. Each table covers all of the 'skedasticities' (DGPs) in separate columns.

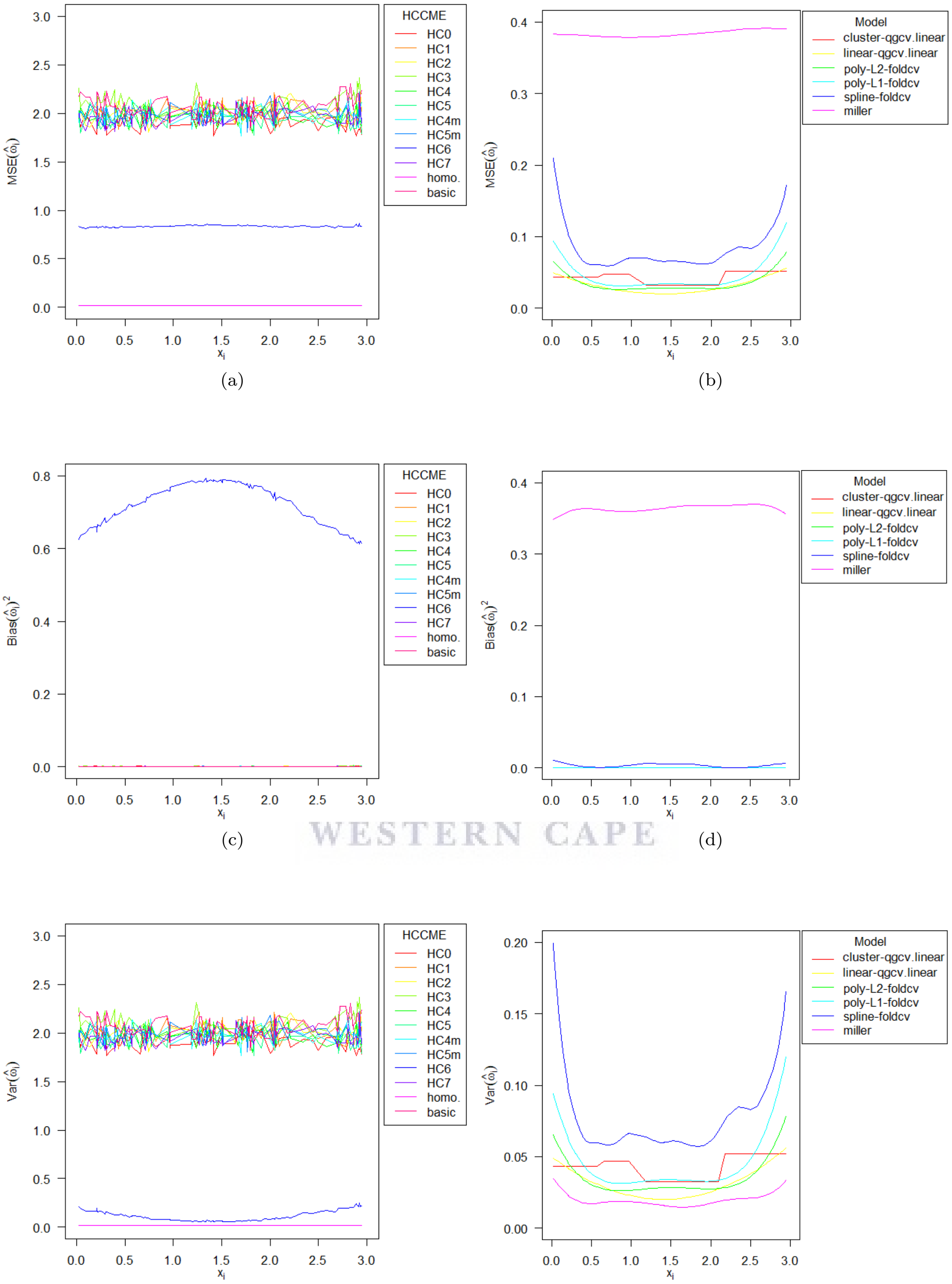
5.3.1.1 Homoskedastic Case

Figure 5.4 shows the MC MSE, squared bias, and variance of the individual $\hat{\omega}_i$ estimators for the homoskedastic DGP (in which the unstandardised and standardised metrics are identical).[118]

[116] The magnitudes of the elements of $\boldsymbol{\beta}$ have no bearing on the performance of the auxiliary variance models.

[117] Feature selection in the one-covariate model entails that, if the single feature is not selected, the homoskedastic variance estimator $\hat{\omega}_{\text{ub}}$ is used.

[118] The horizontal scale of the left and right panels differs due to the greater space required for the legend in the right panel.



(a)

(b)

(c)

(d)

(e)

(f)

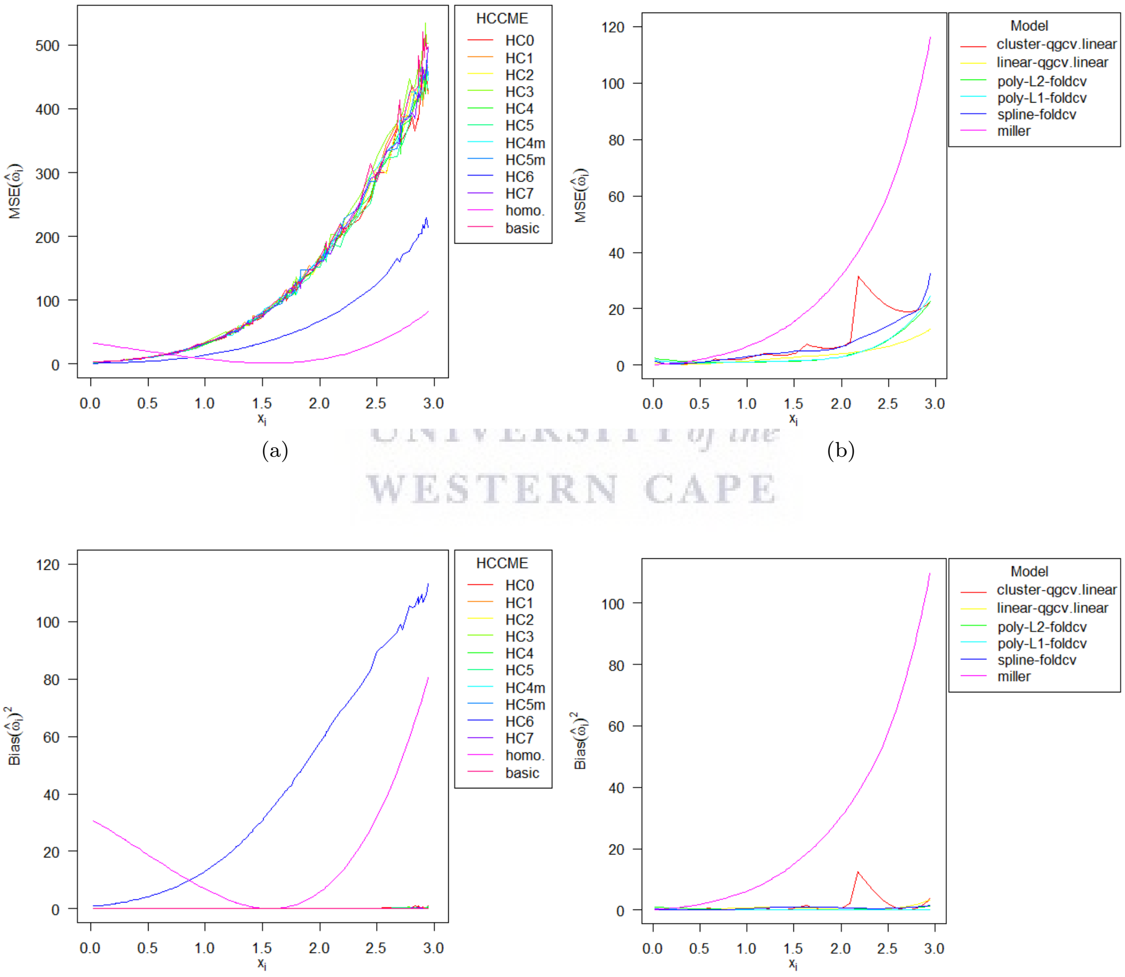
Figure 5.4: Unstandardised MSE, Squared Bias, and Variance Metrics of HCCMEs (a, c, e) and ALVMs (b, d, f) for Homoskedastic Simple Linear Regression Model

By paying attention to the scale of the vertical axis in Figure 5.4, one observes by comparing panels (a) and (b) that the homoskedastic estimator has the lowest MSE, as expected. Moreover, the ALVMs in panel (b) all outperform the HCCMEs in panel (a) in terms of MSE. Within the HCCMEs, HC6 has a lower MSE than the rest, which have little difference between them. Among the variance models, the Miller-Startz SVR model has a higher MSE than the ALVMs. The thin-plate spline ALVM also performs relatively poorly, especially close to the boundary knots. The linear, polynomial, and clustering ALVMs are all close competitors.

From panels (c) and (d), it is apparent that all of the methods have negligible squared bias with the exceptions of HC6 in panel (c), and the Miller-Startz SVR model and (to a lesser extent) the thin-plate spline ALVM in panel (d). From panels (e) and (f), it is apparent that the homoskedastic estimator has the lowest variance. Miller-Startz SVR comes next, followed by the various ALVMs and HC6. The rest of the HCCMEs have much higher variances.

5.3.1.2 Additive Heteroskedasticity Case

The unstandardised MSE, squared bias, and variance metrics for the DGP with additive heteroskedasticity ($g(x_i) = (1 + x_i)^2$) are shown in Figure 5.5, with standardised versions in Figure 5.6.

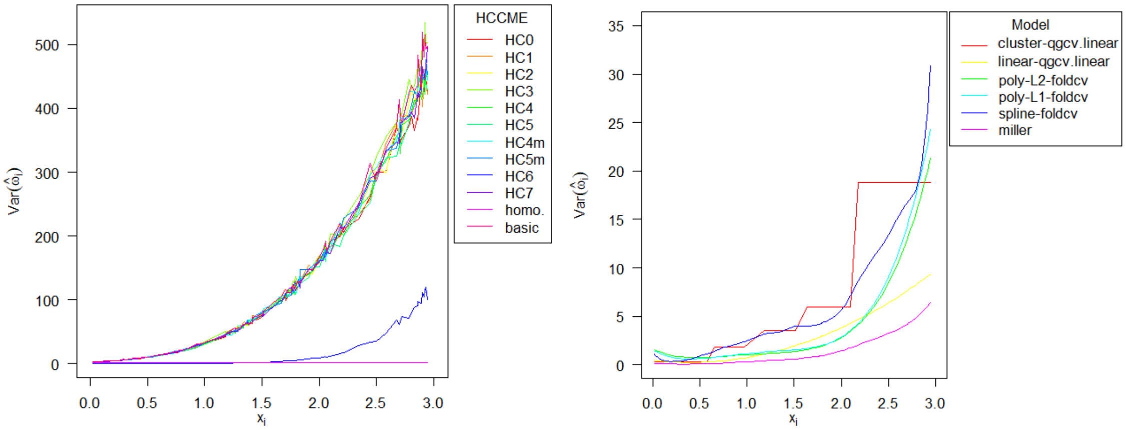


(a)

(b)

(c)

(d)

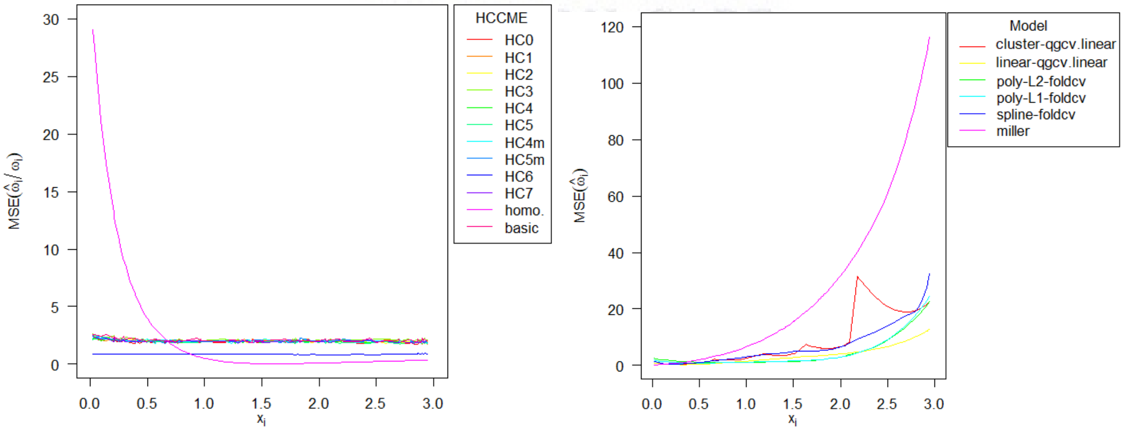


(e) (f)

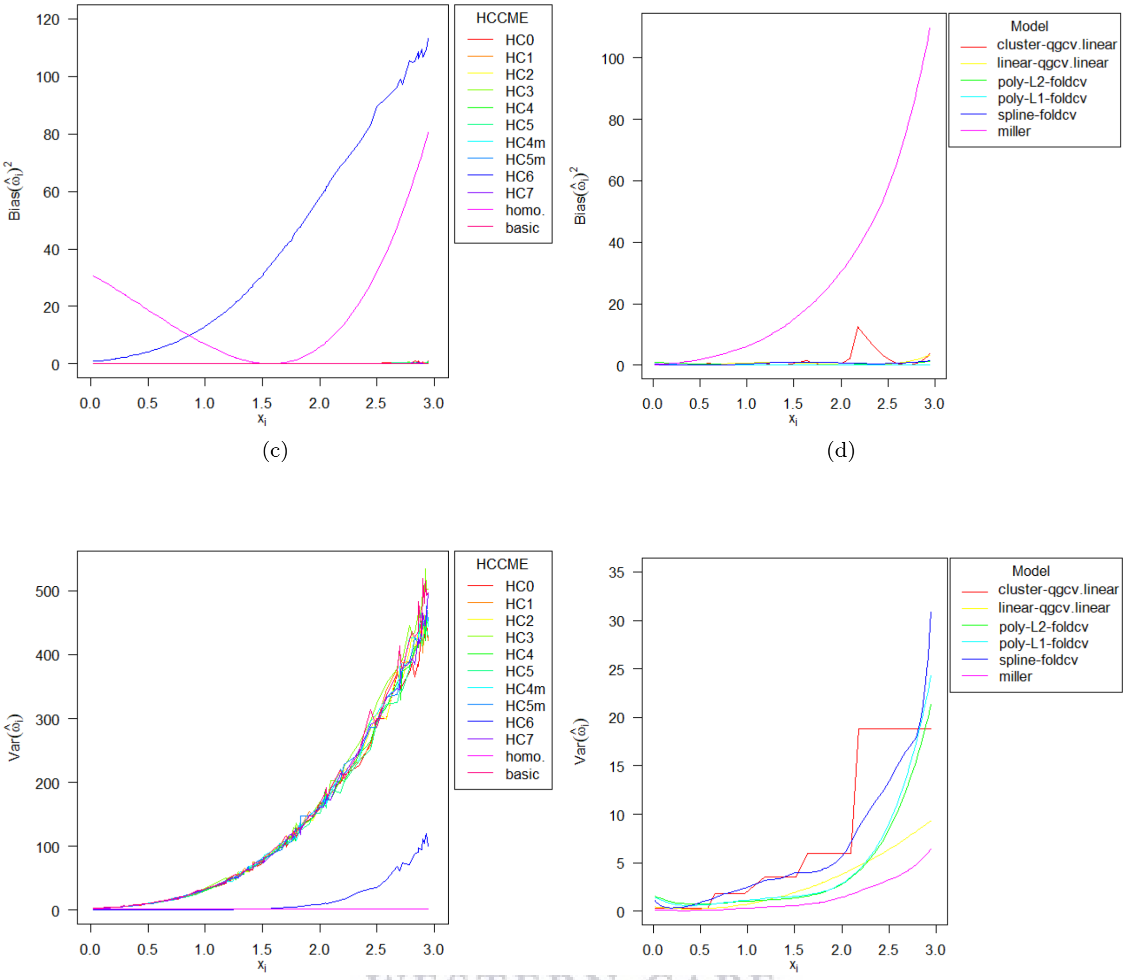
Figure 5.5: Unstandardised MSE, Squared Bias, and Variance Metrics of HCCMEs (a, c, e) and ALVMs (b, d, f) for Simple Linear Regression Model with Additive Heteroskedasticity

Panels (a) and (b) of Figure 5.5 show that the unstandardised MSE of all HCCMEs and variance models increases with x_i. This is unsurprising given that $g(x_i)$ increases with x_i quadratically in the DGP. The MSE of the homoskedastic estimator does not strictly increase with x_i. As in the homoskedastic case, HC6 has a lower MSE than the other HCCMEs, while Miller-Startz SVR has a higher MSE than the ALVMs. The MSE curve of the clustering ALVM appears discontinuous, which makes sense because of the discrete 'jumps' in estimated variance that occur according to which cluster a particular interval of x_i values is assigned to. All of the ALVMs perform better than the HCCMEs, with the linear and polynomial models performing best. Interestingly, the homoskedastic estimator has a lower MSE than the HCCMEs (but not lower than the ALVMs), except for small values of x_i.

The bias-variance trade-off story in the lower two pairs of panels is broadly similar to that in Figure 5.4, except that now the homoskedastic estimator has substantial bias.



(a) (b)



(c) (d)

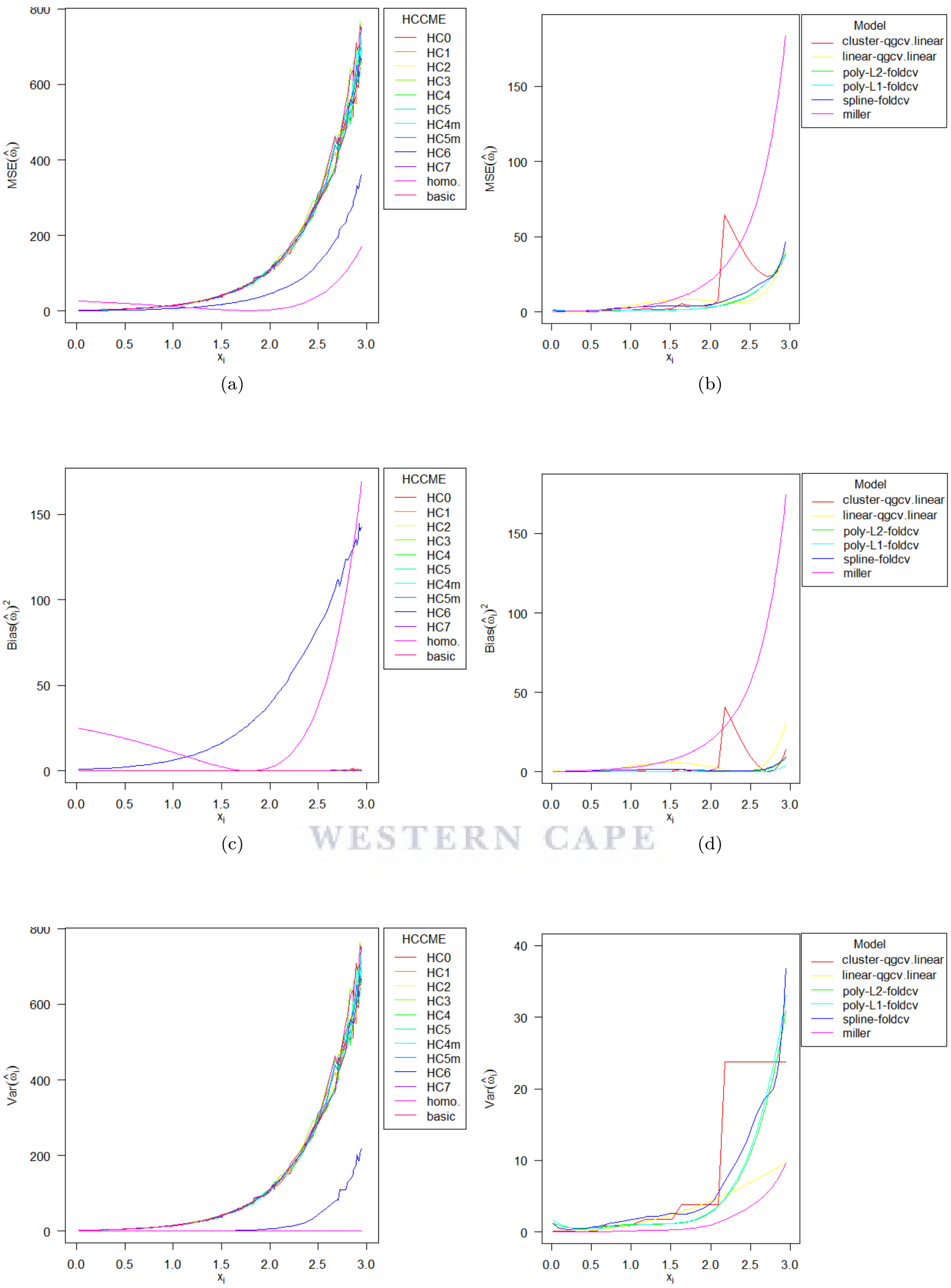
(e) (f)

Figure 5.6: Standardised MSE, Squared Bias, and Variance Metrics of HCCMEs (a, c, e) and ALVMs (b, d, f) for Simple Linear Regression Model with Additive Heteroskedasticity

Looking at the standardised versions of the metrics in Figure 5.6, a broadly similar picture emerges, except that the performance of the homoskedastic estimator is reversed: in standardised terms, it performs very poorly for small x_i but better than any other method for large x_i.

5.3.1.3 Multiplicative Heteroskedasticity Case

The unstandardised MSE, squared bias, and variance metrics for the DGP with multiplicative heteroskedasticity ($g(x_i) = e^{x_i}$) are shown in Figure 5.7, with standardised versions in Figure 5.8.



(a)

(b)

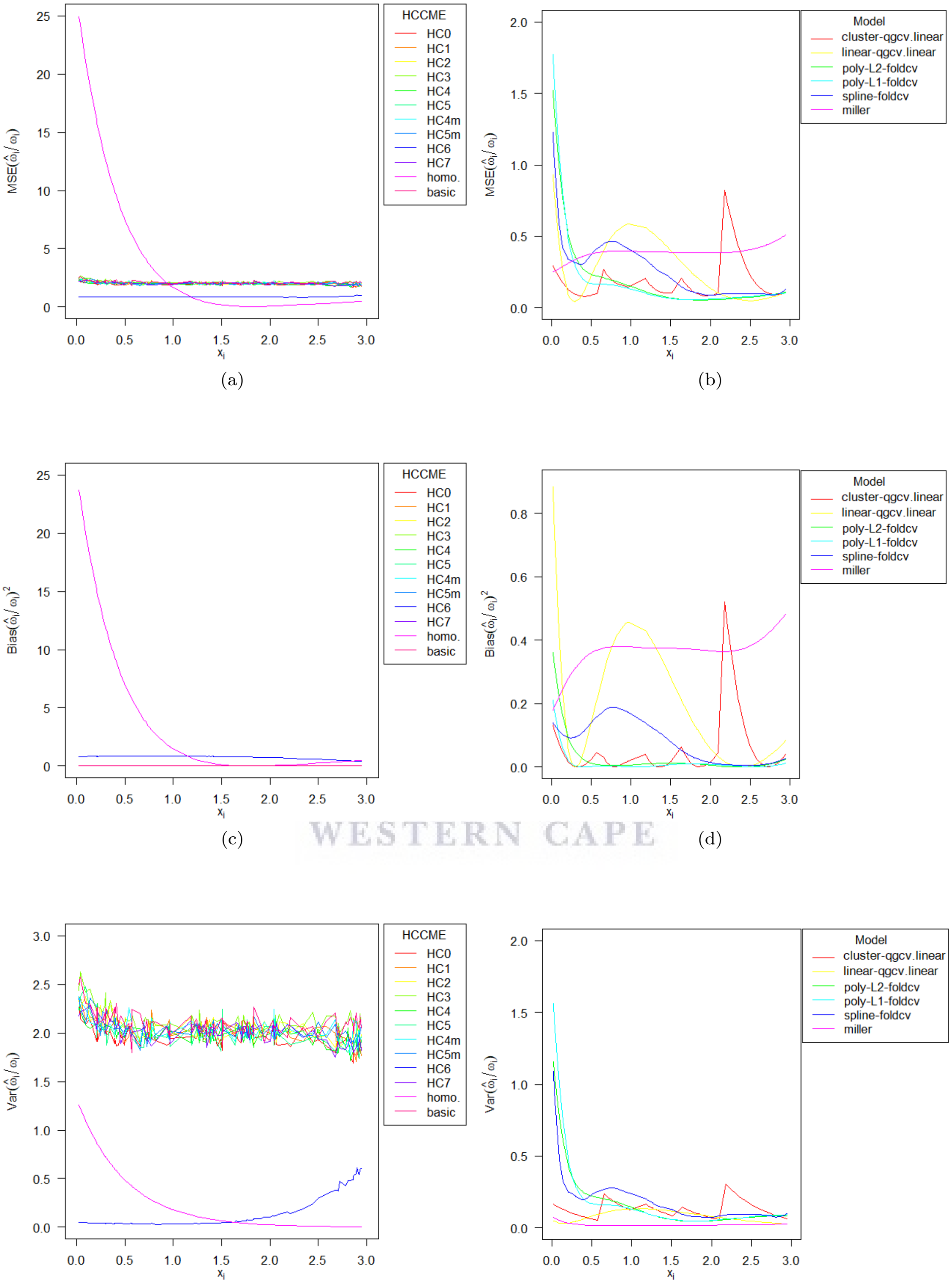
(c)

(d)

(e)

(f)

Figure 5.7: Unstandardised MSE, Squared Bias, and Variance Metrics of HCCMEs (a, c, e) and ALVMs (b, d, f) for Simple Linear Regression Model with Multiplicative Heteroskedasticity



(a)

(b)

(c)

(d)

(e)

(f)

Figure 5.8: Standardised MSE, Squared Bias, and Variance Metrics of HCCMEs (a, c, e) and ALVMs (b, d, f) for Simple Linear Regression Model with Multiplicative Heteroskedasticity

The relative performances of the different models under the multiplicative heteroskedasticity DGP, as displayed in Figures 5.7 and 5.8, are very similar to those under the additive heteroskedasticity DGP from Figures 5.5 and 5.6.

5.3.1.4 Results on Relative Performance Metrics with Statistical Significance Comparisons

Tables 5.6-5.9 show results on the four key performance metrics discussed in §5.2.2 across all three error variance settings (homoskedasticity; additive heteroskedasticity; multiplicative heteroskedasticity). In each of these four tables, the homoskedastic case is shown in the first column followed by the case of additive heteroskedasticity, $\omega_i = (1 + x_i)^2$ and the case of multiplicative heteroskedasticity, $\omega_i = e^{x_i}$. MC mean estimates are presented in relative terms, as explained in §5.2.4, so that a value of 1 indicates the best-performing model (highlighted in green). As explained in §5.2.5, if the model's performance does not differ from the best-performing model by a statistically significant margin (at 5% significance level, with Bonferroni correction), it is highlighted in yellow. MC standard error estimates of the absolute metric values appear in brackets beneath the corresponding relative metric values. The metrics are averaged across all n observations, and thus do not convey information about how performance varies by observation, as do the graphical results in Figures 5.4 to 5.8.

Table 5.6: (Relative) Unstandardised MSE-of-Variances Estimate (with Estimated Standard Error) for One-Covariate Linear Regression Model

	Homosked.	Add. Het.	Mult. Het.
Model	$\boldsymbol{\mathcal{H}} = \emptyset$	$\boldsymbol{\mathcal{H}} = \{2\}$	$\boldsymbol{\mathcal{H}} = \{2\}$
HC3	102 (8.06×10^{-3})	38.5 (7.89×10^{-1})	23.2 (9.58×10^{-1})
HC4	96.1 (7.27×10^{-3})	35.6 (7.03×10^{-1})	21.2 (8.71×10^{-1})
HC6	40.8 (6.30×10^{-4})	16.1 (1.93×10^{-1})	10.1 (2.89×10^{-1})
Homoskedastic	1 (2.99×10^{-4})	6.1 (1.90×10^{-2})	5.07 (1.98×10^{-2})
Basic ALVM	101 (7.80×10^{-3})	38 (7.40×10^{-1})	23 (9.45×10^{-1})
Clustering ALVM	2.01 (6.43×10^{-4})	2.07 (7.28×10^{-2})	1.49 (8.28×10^{-2})
Linear ALVM	1.56 (4.52×10^{-4})	1 (4.25×10^{-2})	1.35 (4.59×10^{-2})
L_2-Norm Pen. Poly. ALVM	1.8 (4.52×10^{-4})	1.29 (5.27×10^{-2})	1.03 (7.38×10^{-2})
L_1-Norm Pen. Poly. ALVM	2.37 (5.24×10^{-4})	1.31 (5.99×10^{-2})	1 (7.72×10^{-2})
Thin-Plate spline ALVM	4.07 (7.29×10^{-4})	1.96 (7.17×10^{-2})	1.28 (7.95×10^{-2})
Miller-Startz SVR	18.8 (9.41×10^{-4})	7.89 (7.95×10^{-2})	4.98 (9.39×10^{-2})

Table 5.7: (Relative) Standardised MSE-of-Variances Estimate (with Estimated Standard Error) for One-Covariate Linear Regression Model

Model	Homosked.	Add. Het.	Mult. Het.
	$\mathcal{H} = \emptyset$	$\mathcal{H} = \{2\}$	$\mathcal{H} = \{2\}$
HC3	102 (8.06×10^{-3})	23.4 (8.07×10^{-3})	14.4 (8.07×10^{-3})
HC4	96.1 (7.27×10^{-3})	21.9 (7.40×10^{-3})	13.5 (7.52×10^{-3})
HC6	40.8 (6.30×10^{-4})	9.49 (9.60×10^{-4})	5.9 (1.02×10^{-3})
Homoskedastic	1 (2.99×10^{-4})	32.2 (1.30×10^{-2})	26.7 (1.96×10^{-2})
Basic ALVM	101 (7.80×10^{-3})	23.3 (7.88×10^{-3})	14.2 (7.82×10^{-3})
Clustering ALVM	2.01 (6.43×10^{-4})	1.56 (1.00×10^{-3})	1 (9.82×10^{-4})
Linear ALVM	1.56 (4.52×10^{-4})	1 (9.92×10^{-4})	1.79 (2.46×10^{-3})
L_2-Norm Pen. Poly. ALVM	1.8 (4.52×10^{-4})	2.32 (2.59×10^{-3})	1.31 (2.54×10^{-3})
L_1-Norm Pen. Poly. ALVM	2.37 (5.24×10^{-4})	1.7 (1.71×10^{-3})	1.24 (2.38×10^{-3})
Thin-Plate spline ALVM	4.07 (7.29×10^{-4})	1.82 (1.78×10^{-3})	1.73 (2.13×10^{-3})
Miller-Startz SVR	18.8 (9.41×10^{-4})	4.3 (9.21×10^{-4})	2.64 (9.28×10^{-4})

Table 5.8 (Relative) MSE of FWLS Estimate of β (with Estimated Standard Error) for One-Covariate Linear Regression Model

Model	Homosked.	Add. Het.	Mult. Het.
	$\mathcal{H} = \emptyset$	$\mathcal{H} = \{2\}$	$\mathcal{H} = \{2\}$
OLS	1.01 (3.29×10^{-4})	1.48 (1.54×10^{-3})	1.69 (1.58×10^{-3})
HC3	1.01 (3.22×10^{-4})	1.47 (1.51×10^{-3})	1.67 (1.53×10^{-3})
HC4	1 (3.20×10^{-4})	1.49 (1.53×10^{-3})	1.66 (1.52×10^{-3})
HC6	1.06 (3.51×10^{-4})	1.48 (1.54×10^{-3})	1.68 (1.55×10^{-3})
Homoskedastic	1.01 (3.29×10^{-4})	1.48 (1.54×10^{-3})	1.69 (1.58×10^{-3})
Basic ALVM	1.2 (1.83×10^{-3})	1.59 (1.67×10^{-3})	1.81 (1.69×10^{-3})
Clustering ALVM	1.04 (3.28×10^{-4})	1 (9.92×10^{-4})	1 (8.73×10^{-4})
Linear ALVM	1.04 (3.36×10^{-4})	5.98 (9.02×10^{-3})	9.94 (1.02×10^{-2})
L_2-Norm Pen. Poly. ALVM	1.03 (3.38×10^{-4})	2.02 (5.45×10^{-3})	72.7 (3.03×10^{0})
L_1-Norm Pen. Poly. ALVM	1.01 (3.29×10^{-4})	2.55 (7.72×10^{-3})	156 (4.17×10^{0})
Thin-Plate spline ALVM	1.16 (6.18×10^{-4})	3900 (4.49×10^{1})	12900 (6.55×10^{1})
Miller-Startz SVR	1.05 (3.43×10^{-4})	1.04 (1.05×10^{-3})	1.01 (9.06×10^{-4})

Table 5.9: (Relative) MSE of Standard Errors of $\hat{\boldsymbol{\beta}}_{\text{OLS}}$ (with Estimated Standard Error) for One-Covariate Linear Regression Model

Model	Homosked.	Add. Het.	Mult. Het.
	$\boldsymbol{\mathcal{H}} = \emptyset$	$\boldsymbol{\mathcal{H}} = \{2\}$	$\boldsymbol{\mathcal{H}} = \{2\}$
HC3	2.07 (3.45×10^{-6})	1.69 (2.41×10^{-5})	1.27 (2.91×10^{-5})
HC4	1.91 (3.07×10^{-6})	1.56 (2.08×10^{-5})	1.23 (2.61×10^{-5})
HC6	67.8 (1.54×10^{-5})	35.2 (1.04×10^{-4})	19.7 (1.15×10^{-4})
Homoskedastic	1 (1.76×10^{-6})	8.95 (4.74×10^{-5})	5.75 (4.05×10^{-5})
Basic ALVM	2.01 (3.18×10^{-6})	1.6 (2.11×10^{-5})	1.26 (2.70×10^{-5})
Clustering ALVM	1.31 (2.42×10^{-6})	1.37 (1.79×10^{-5})	1.05 (2.12×10^{-5})
Linear ALVM	1.24 (2.18×10^{-6})	1 (1.28×10^{-5})	1.13 (1.96×10^{-5})
L_2-Norm Pen. Poly. ALVM	1.36 (2.40×10^{-6})	1.52 (1.71×10^{-5})	1 (2.12×10^{-5})
L_1-Norm Pen. Poly. ALVM	1.62 (2.85×10^{-6})	1.41 (1.74×10^{-5})	1.04 (2.28×10^{-5})
Thin-Plate spline ALVM	2.4 (3.84×10^{-6})	2.09 (2.59×10^{-5})	2.17 (3.63×10^{-5})
Miller-Startz SVR	28.7 (1.47×10^{-5})	19.2 (7.49×10^{-5})	13.3 (8.41×10^{-5})

Discussion of Tables 5.6-5.9

From Table 5.6, it is evident that the homoskedastic variance estimator significantly outperforms all others under homoskedasticity. Its unstandardised MSE estimate is almost 100 times smaller than those of some of the HCCMEs. Notably, however, the unstandardised MSEs of the linear ALVM and polynomial ALVM (with L_2-norm penalty) are less than double that of the homoskedastic estimator. Under additive (quadratic) heteroskedasticity, the linear ALVM is significantly better than all others in terms of unstandardised MSE. Under multiplicative (exponential) heteroskedasticity, the polynomial ALVM with L_1-norm penalty is significantly better than all others except the same model with L_2-norm penalty.

Table 5.7 tells a similar tale. The homoskedastic estimator is significantly better than all others in terms of standardised MSE under homoskedasticity, while the linear ALVM is better than all competitors under an additive heteroskedastic DGP. A difference between the unstandardised and standardised MSE metrics appears under the multiplicative heteroskedastic DGP. Here, the clustering ALVM performs significantly better than all other methods in terms of standardised MSE, whereas the polynomial models had been the winners in terms of unstandardised MSE.

Turning attention to Table 5.8, under homoskedasticity there is very little separation between the different variance estimation methods in terms of MSE for estimating $\boldsymbol{\beta}$ using FWLS. The HCCMEs and ALVMs all produce similar results to each other and, notably, to OLS. Under the two heteroskedastic DGPs, the clustering ALVM yields the best results, but not by a statistically significant margin over the Miller-Startz SVR method or (in the multiplicative heteroskedasticity case) the polynomial ALVMs. The performance of the polynomial and spline models, and to a lesser extent the linear model, is rather unstable by this metric, especially under multiplicative heteroskedasticity.[119]

Finally, Table 5.9 shows results on the MSE metric for estimation of the standard errors of the elements of $\hat{\boldsymbol{\beta}}_{\text{OLS}}$. Here, the homoskedastic estimator produces significantly better results than other methods under the homoskedastic DGP. The linear ALVM is the clear winner under additive (quadratic) heteroskedasticity,

[119] Note that the two polynomial models' results are highlighted in yellow in the last column only because their MC standard errors are too large to allow for a statistically significant comparison with the cluster model.

and the clustering ALVM has the best result under multiplicative (exponential) heteroskedasticity, but is not significantly better than the linear or polynomial ALVMs.

Results for the metrics in Tables 5.6 to 5.9 for simulations with the same specifications except for different sample sizes can can be found in Appendix E.1 ($n = 20$) and Appendix E.2 ($n = 1000$). Only $R = 10^3$ MC replications were used with the $n = 1000$ simulation due to the large computation time required.

In the $n = 20$ case (Tables E.1-E.4), the main difference in terms of $\overline{\text{MSE}}_{\text{ust}}(\hat{\boldsymbol{\omega}})$ is that the linear ALVM, and not the penalised polynomial ALVMs, performs best under multiplicative heteroskedasticity. In terms of $\overline{\text{MSE}}_{\text{st}}(\hat{\boldsymbol{\omega}})$, the Miller-Startz SVR model is the clear winner under both DGPs, in contrast to the $n = 100$ case where the clustering and linear ALVMs, respectively, performed best. The good small-sample performance of the Miller-Startz model under heteroskedasticity carries over into the $\text{MSE}(\hat{\boldsymbol{\beta}}_{\text{FWLS}})$ metric, where also the homoskedastic estimator is more distinctly better than other methods under homoskedasticity than in the $n = 100$ case. The linear ALVM is the clear winner in terms of the $\text{MSE}(\text{SE}(\hat{\boldsymbol{\beta}}))$ metric in the $n = 20$ case for both heteroskedastic DGPs, whereas in the $n = 100$ case, the L_2-norm penalised polynomial ALVM won under multiplicative heteroskedasticity.

Comparing the large-sample $n = 1000$ case (Tables E.6-E.9) to the $n = 100$ case, the main difference in terms of $\overline{\text{MSE}}_{\text{ust}}(\hat{\boldsymbol{\omega}})$ is that the clustering ALVM, and not the penalised polynomial ALVMs, performs best under multiplicative heteroskedasticity. In terms of $\overline{\text{MSE}}_{\text{st}}(\hat{\boldsymbol{\omega}})$, the clustering ALVM is the clear winner under both DGPs, in contrast to the $n = 100$ case where the linear ALVMs performed best under multiplicative heteroskedasticity. The good large-sample performance of the clustering ALVM under heteroskedasticity carries over into the $\text{MSE}(\hat{\boldsymbol{\beta}}_{\text{FWLS}})$ metric, it is the clear winner under both DGPs. This was also the case with $n = 100$, but not by a statistically significant margin. The clustering and basic ALVMs are neck-and-neck with HC3 and HC4 in terms of the $\text{MSE}(\text{SE}(\hat{\boldsymbol{\beta}}))$ metric in the $n = 1000$ case for both heteroskedastic DGPs. The other ALVMs perform poorly here.

5.3.1.5 ANLVM Results for this Simulation Configuration

Table 5.10 reports performance metrics for three ANLVMs for the same simulation reported on in Tables 5.6-5.9. 'Quadratic' refers to the ANLVM with quadratic heteroskedastic function $g(\boldsymbol{Z}_{k\cdot};\boldsymbol{\gamma}) = (\boldsymbol{Z}'_{k\cdot}\boldsymbol{\gamma})^2$ (3.38). 'Exponential' refers to the ANLVM with exponential heteroskedastic function $g(\boldsymbol{Z}_{k\cdot};\boldsymbol{\gamma}) = \exp\{\boldsymbol{Z}'_{k\cdot}\boldsymbol{\gamma}\}$ (3.39). 'Clustering' refers to the clustering ANLVM described in (3.61). In all ANLVM simulations reported on in the thesis, feature selection was performed by best subset selection using QGCV loss in the linear ALVM. MC means and standard error estimates for the metrics are computed only across replications where the Gauss-Newton algorithm used for MQL estimation achieved convergence. For convergence rates for all ANLVM simulations, see Table 5.31.

Table 5.10: Relative Performance Metrics (with Estimated Standard Errors) for ANLVMs Fit to One-Covariate Linear Regression Model

		Homosked.	Add. Het.	Mult. Het.
Metric	ANLVM	$\mathcal{H} = \emptyset$	$\mathcal{H} = \{2\}$	$\mathcal{H} = \{2\}$
$\overline{\text{MSE}}_{\text{ust}}(\hat{\boldsymbol{\omega}})$	Quadratic	1.58 (4.57×10^{-4})	0.771 (3.76×10^{-2})	0.798 (3.82×10^{-2})
	Exponential	1.63 (4.71×10^{-4})	1.94 (1.06×10^{-1})	0.706 (6.80×10^{-2})
	Clustering	1.98 (6.36×10^{-4})	2.09 (7.10×10^{-2})	1.47 (7.64×10^{-2})
$\overline{\text{MSE}}_{\text{std}}(\hat{\boldsymbol{\omega}})$	Quadratic	1.58 (4.57×10^{-4})	0.491 (6.43×10^{-4})	0.451 (1.11×10^{-3})
	Exponential	1.63 (4.71×10^{-4})	0.855 (7.40×10^{-4})	0.291 (4.67×10^{-4})
	Clustering	1.98 (6.36×10^{-4})	1.59 (1.02×10^{-3})	0.991 (9.26×10^{-4})
$\text{MSE}(\hat{\boldsymbol{\beta}}_{\text{FWLS}})$	Quadratic	1.04 (3.39×10^{-4})	0.951 (9.43×10^{-4})	0.966 (8.40×10^{-4})
	Exponential	1.02 (3.28×10^{-4})	0.945 (9.30×10^{-4})	0.971 (8.55×10^{-4})
	Clustering	1.02 (3.27×10^{-4})	1 (9.88×10^{-4})	1 (8.60×10^{-4})
$\text{MSE}(\text{SE}(\hat{\boldsymbol{\beta}}))$	Quadratic	1.27 (2.27×10^{-6})	0.808 (1.05×10^{-5})	0.976 (1.72×10^{-5})
	Exponential	1.3 (2.37×10^{-6})	1.73 (2.50×10^{-5})	0.748 (1.63×10^{-5})
	Clustering	1.28 (2.35×10^{-6})	1.4 (1.82×10^{-5})	1.05 (2.03×10^{-5})

As was explained in §5.2.5, the relative metric values in Table 5.10 are relative to the best-performing ALVM in Tables 5.6-5.9 (depending on metric). A cell in Table 5.10 is highlighted in green only if the metric for that ANLVM is lower than those of all other ANLVMs *and ALVMs*, in which case the relative metric value is less than 1. Hence, it is possible that for some settings and metrics, there is no green cell; this is the case for all metrics estimated under homoskedasticity in Table 5.10. A yellow cell in an ANLVM results table indicates that this ANLVM is inferior to the best model (whether an ALVM or an ANLVM), but not by a statistically significant margin.

From Table 5.10, it is evident that under additive (quadratic) heteroskedasticity, the quadratic ANLVM outperforms all competitors, including all of the ALVMs, in all four metrics. Of course, in this instance it has the advantage of having exactly specified the true heteroskedastic function $g(\cdot)$. Similarly, under multiplicative (exponential) heteroskedasticity, the exponential ANLVM outperforms all competitors, including all of the ALVMs, in three out of four metrics. The quadratic ANLVM actually performs slightly better than the exponential ANLVM in terms of $\text{MSE}(\hat{\boldsymbol{\beta}}_{\text{FWLS}})$ under the multiplicative (exponential) DGP, and vice versa. The performance of the clustering ANLVM is satisfactory across the board, and very similar to that of the clustering ALVM.

Table E.5 displays results for an ANLVM situation run on the same DGPs but with $n = 20$. Although the exponential ANLVM is still the best-performing model in terms of $\text{MSE}(\hat{\boldsymbol{\beta}}_{\text{FWLS}})$ under both additive and multiplicative heteroskedasticity, it is clear that the ANLVMs' performance has suffered more due to the reduction in sample size than that of the ALVMs. The Gauss-Newton algorithm's convergence rates also declined under the smaller sample size; the lowest rate was 80%, for the clustering ANLVM under multiplicative heteroskedasticity.

Table E.10 displays results for a large-sample ANLVM simulation ($n = 1000$). Here, the quadratic ANLVM is the best-performing model under the additive heteroskedastic DGP for all metrics except for $\text{MSE}(\hat{\boldsymbol{\beta}}_{\text{FWLS}})$, and similarly the exponential ANLVM is the best-performing model under the multiplicative heteroskedastic DGP for all metrics except for $\text{MSE}(\hat{\boldsymbol{\beta}}_{\text{FWLS}})$. Meanwhile, the clustering ANLVM is the best-performing model under both heteroskedastic DGPs in terms of $\text{MSE}(\hat{\boldsymbol{\beta}}_{\text{FWLS}})$. The convergence rate was also 100% for the ANLVMs throughout this simulation. Unfortunately, due to high computation time, only $R = 10^2$ MC replications could be run, and the resulting high MC standard errors mean that the ANLVMs are not always better than other methods by a statistically significant margin. Nonetheless, Table E.10 provides evidence that the ANLVMs are a very attractive option when the sample size is large.

5.3.1.6 Simulation on a Nonmonotonic Heteroskedasticity Case

As a robustness check, only for the one-covariate regression with $n = 100$, a simulation was conducted under a nonmonotonic heteroskedastic DGP—specifically, the heteroskedastic function is $g(x) = \left[\sin^2\left(\frac{2\pi x}{3}\right)\right] + \frac{1}{5}$ (see Figure E.1). Graphical and tabular results are displayed in Appendix E.3. Two clustering ALVMs were used in this simulation, with the number of clusters being $n_c = 5$ (chosen by the elbow method with SWD criterion) and $n_c = 8$, respectively. It was suspected that a larger number of clusters might be more effective in this case, due to the magnitude of rate of change in the heteroskedastic function. Two clustering ANLVMs were also fit, again with $n_c = 5$ and $n_c = 8$. An ANLVM was also fitted with $g(x)$ correctly specified; this is called `sinsq` in Table E.12.

Panels (a) and (b) of Figure E.2 show that the highest unstandardised MSEs are achieved at the maxima of the heteroskedastic function, whereas Panels (a) and (b) of Figure E.3 show that the highest *standardised* MSEs are achieved at the minima of the heteroskedastic function. From the first two columns of Table E.11, it is evident that the thin-plane spline ALVM achieves the lowest average unstandardised and standardised MSE for estimating the ω_i. However, this does not translate into optimal FWLS estimation of $\boldsymbol{\beta}$ or estimation of $\text{SE}(\hat{\boldsymbol{\beta}})$, as the Miller-Startz SVR model and the homoskedastic estimator perform better in terms of these two metrics, respectively.

From Table E.12, it appears that the ANLVMs have not been particularly successful in modelling the sinusoidal heteroskedastic function. Even the correctly specified 'squared sinusoidal' ANLVM in the last row is inferior to the thin-plate spline ALVM, in terms of the first two metrics. This method does perform best in case of the fourth metric, however, while the clustering ANLVM—like the clustering ALVM—performs fairly well in terms of FWLS estimation of $\boldsymbol{\beta}$, especially when $n_c = 8$.

5.3.1.7 Simulation under Non-Normal Errors

As a further robustness check, only for the one-covariate regression with $n = 100$, a performance evaluation of the ALVMs and ANLVMs was conducted under two DGPs with non-normal errors. These are, specifically, a Laplace or double exponential distribution (which is leptokurtic) and a uniform distribution (which is platykurtic). In both instances, the distributions were parametrised so that the errors have zero mean and variance ω_i. In the Laplace case, by generating $\epsilon_i \sim \text{Laplace}(0, \sqrt{\omega_i/2})$ (where the two parameters of the Laplace distribution are a location parameter and scale parameter, respectively). In the uniform case, this was achieved by generating $\epsilon_i \sim U(-\sqrt{3\omega_i}, \sqrt{3\omega_i})$. Thus, the respective marginal PDFs of the errors are,

$$f_{\epsilon_i}(x) = (2\omega_i)^{-1/2} \exp\left\{-(\omega_i/2)^{-1/2}|x|\right\} \text{ for } -\infty < x < \infty \text{ (Laplace case), and} \tag{5.21}$$

$$f_{\epsilon_i}(x) = \left[2(3\omega)^{1/2}\right]^{-1} \text{ for } -(3\omega_i)^{1/2} \leq x \leq (3\omega_i)^{1/2} \text{ (Uniform case).} \tag{5.22}$$

Performance results for the ALVMs, using the usual four metrics, are shown in Tables E.13-E.16. The results are generally similar to those obtained under normal errors. The thin-plate spline ALVM fares better under Laplace-distributed errors than under normal errors in terms of $\overline{\text{MSE}}_{\text{ust}}(\hat{\boldsymbol{\omega}})$ and $\overline{\text{MSE}}_{\text{st}}(\hat{\boldsymbol{\omega}})$, but for the most part, the models that performed best under normal errors also perform best under non-normal errors.

Performance results for the ANLVMs are shown in Table E.17. Again, the results are broadly similar to those under normal errors. The quadratic ANLVM performs very well under additive heteroskedasticity, and the exponential ANLVM performs very well under multiplicative heteroskedasticity. One notable change is that, for both additive and multiplicative heteroskedasticity, when the errors were generated from a Laplace distribution, the clustering ANLVM performed best in terms of the $\text{MSE}(\hat{\boldsymbol{\beta}}_{\text{FWLS}})$ metric, which is not the case under normal or uniform errors.

It appears that the performance of the ALVMs and ANLVMs are not seriously affected under the kinds of deviation from normality considered here. This is unsurprising in the case of ALVMs, which do not make use of the normality assumption A5. The ANLVMs are, in theory, more prone to being affected by non-normality, because they make use of the variance-covariance matrix (3.11) (see $\boldsymbol{V}(\boldsymbol{\gamma})$ in §3.3.1.4), whereas under non-normality the true variance-covariance matrix of the squared OLS residuals is given by (3.17).

5.3.2 Linear Regression with Two Covariates

Tables 5.11-5.14 show results on the same four performance metrics considered previously, for a simulation with two covariates generated independently from $U(0,3)$.

The homoskedastic case is shown in the first column followed by two cases of quadratic heteroskedasticity, $\omega_i = (1+x_{i2})^2$ and $\omega_i = (1+x_{i2}+x_{i3})^2$, followed by two cases of exponential heteroskedasticity, $\omega_i = e^{x_{i2}}$ and $\omega_i = e^{x_{i2}+x_{i3}}$.

Again, the penalty parameter λ was tuned using five-fold CV for the L_2-norm penalised polynomial ALVM and the thin-plate spline ALVM, while the elbow method with SWD criterion was used to choose the number of clusters n_c for the clustering ALVM. Feature selection for the linear and clustering ALVMs was performed using best subset selection on the linear model by QGCV loss. The LASSO penalised polynomial ALVM was not included in any of the multiple linear regression simulations due to its high computation time.

Table 5.11 (Relative) Unstandardised MSE-of-Variances Estimate (with Estimated Standard Error) for Two-Covariate Linear Regression Model

Model	Homosked.	Add. Het.		Mult. Het.	
	$\boldsymbol{\mathcal{H}} = \emptyset$	$\boldsymbol{\mathcal{H}} = \{2\}$	$\boldsymbol{\mathcal{H}} = \{2,3\}$	$\boldsymbol{\mathcal{H}} = \{2\}$	$\boldsymbol{\mathcal{H}} = \{2,3\}$
HC3	107 (8.19×10^{-3})	36.9 (7.68×10^{-1})	23.2 (4.77×10^{0})	16.8 (9.37×10^{-1})	9.79 (8.91×10^{1})
HC4	96.2 (7.13×10^{-3})	33.2 (6.74×10^{-1})	20.3 (3.93×10^{0})	15 (8.09×10^{-1})	8.63 (7.71×10^{1})
HC6	41.8 (5.58×10^{-4})	14.9 (1.39×10^{-1})	9.13 (8.06×10^{-1})	7.02 (2.26×10^{-1})	4.72 (5.16×10^{1})
Homoskedastic	1 (2.88×10^{-4})	5.81 (1.94×10^{-2})	2.89 (1.20×10^{-1})	3.69 (2.05×10^{-2})	3.01 (1.34×10^{0})
Basic ALVM	106 (7.96×10^{-3})	37.4 (7.97×10^{-1})	23 (4.69×10^{0})	16.8 (9.61×10^{-1})	9.95 (9.10×10^{1})
Clustering ALVM	3.14 (8.86×10^{-4})	2.17 (7.71×10^{-2})	2.16 (5.57×10^{-1})	1.13 (8.15×10^{-2})	1.46 (7.76×10^{0})
Linear ALVM	2.2 (5.65×10^{-4})	1 (4.20×10^{-2})	1 (3.16×10^{-1})	1.01 (5.10×10^{-2})	1.73 (3.82×10^{0})
L_2-Norm Pen. Poly. ALVM	2.58 (6.74×10^{-4})	1.73 (6.19×10^{-2})	1.04 (3.68×10^{-1})	1 (8.55×10^{-2})	1 (7.16×10^{0})
Thin-Plate spline ALVM	5.59 (8.85×10^{-4})	2.06 (9.73×10^{-2})	1.29 (5.64×10^{-1})	1.24 (1.20×10^{-1})	1.3 (2.09×10^{1})
Miller-Startz SVR	19.6 (9.15×10^{-4})	7.86 (7.63×10^{-2})	4.72 (4.41×10^{-1})	3.86 (8.70×10^{-2})	2.53 (6.17×10^{0})

Table 5.12: (Relative) Standardised MSE-of-Variances Estimate (with Estimated Standard Error) for Two-Covariate Linear Regression Model

	Homosked.	Add. Het.		Mult. Het.	
Model	$\mathcal{H} = \emptyset$	$\mathcal{H} = \{2\}$	$\mathcal{H} = \{2,3\}$	$\mathcal{H} = \{2\}$	$\mathcal{H} = \{2,3\}$
HC3	107 (8.19×10^{-3})	19 (8.47×10^{-3})	11.9 (8.85×10^{-3})	14 (8.58×10^{-3})	6.32 (1.49×10^{-2})
HC4	96.2 (7.13×10^{-3})	17.1 (7.47×10^{-3})	10.6 (7.44×10^{-3})	12.7 (7.63×10^{-3})	5.66 (1.39×10^{-2})
HC6	41.8 (5.58×10^{-4})	7.37 (7.40×10^{-4})	4.54 (7.56×10^{-4})	5.45 (8.59×10^{-4})	2.08 (1.07×10^{-3})
Homoskedastic	1 (2.88×10^{-4})	25.4 (1.31×10^{-2})	20.6 (1.53×10^{-2})	24.7 (1.98×10^{-2})	109 (2.39×10^{-1})
Basic ALVM	106 (7.96×10^{-3})	19.1 (8.54×10^{-3})	11.8 (8.60×10^{-3})	13.8 (8.26×10^{-3})	5.96 (1.29×10^{-2})
Clustering ALVM	3.14 (8.86×10^{-4})	1.43 (1.38×10^{-3})	2.34 (4.30×10^{-3})	1 (1.09×10^{-3})	2.7 (1.33×10^{-2})
Linear ALVM	2.2 (5.65×10^{-4})	1 (1.42×10^{-3})	1.44 (3.63×10^{-3})	1.78 (2.67×10^{-3})	11.8 (4.36×10^{-2})
L_2-Norm Pen. Poly. ALVM	2.58 (6.74×10^{-4})	3.29 (3.82×10^{-3})	3.08 (6.93×10^{-3})	2.21 (4.45×10^{-3})	26.9 (3.34×10^{-1})
Thin-Plate spline ALVM	5.59 (8.85×10^{-4})	1.86 (1.68×10^{-3})	1 (1.66×10^{-3})	2.07 (2.62×10^{-3})	3.44 (3.91×10^{-2})
Miller-Startz SVR	19.6 (9.15×10^{-4})	3.43 (8.74×10^{-4})	2.13 (8.63×10^{-4})	2.49 (8.90×10^{-4})	1 (1.95×10^{-3})

Table 5.13 (Relative) MSE of FWLS Estimate of β (with Estimated Standard Error) for Two-Covariate Linear Regression Model

	Homosked.	Add. Het.		Mult. Het.	
Model	$\mathcal{H} = \emptyset$	$\mathcal{H} = \{2\}$	$\mathcal{H} = \{2,3\}$	$\mathcal{H} = \{2\}$	$\mathcal{H} = \{2,3\}$
OLS	1.01 (4.01×10^{-4})	1.52 (2.01×10^{-3})	1.59 (5.04×10^{-3})	1.76 (1.86×10^{-3})	3.43 (1.48×10^{-2})
HC3	1.04 (4.03×10^{-4})	1.52 (2.05×10^{-3})	1.57 (4.86×10^{-3})	1.76 (1.90×10^{-3})	3.16 (1.36×10^{-2})
HC4	1.04 (4.08×10^{-4})	1.53 (2.02×10^{-3})	1.56 (4.88×10^{-3})	1.74 (1.85×10^{-3})	3.14 (1.37×10^{-2})
HC6	1.04 (4.10×10^{-4})	1.59 (2.09×10^{-3})	1.59 (4.93×10^{-3})	1.8 (1.87×10^{-3})	3.41 (1.49×10^{-2})
Homoskedastic	1.01 (4.01×10^{-4})	1.52 (2.01×10^{-3})	1.59 (5.04×10^{-3})	1.76 (1.86×10^{-3})	3.43 (1.48×10^{-2})
Basic ALVM	1.08 (4.17×10^{-4})	1.62 (2.12×10^{-3})	1.63 (5.11×10^{-3})	1.9 (2.01×10^{-3})	3.08 (1.41×10^{-2})
Clustering ALVM	1 (3.86×10^{-4})	1 (1.29×10^{-3})	1.03 (3.35×10^{-3})	1 (1.03×10^{-3})	1.41 (1.21×10^{-2})
Linear ALVM	1.01 (4.15×10^{-4})	5.65 (2.10×10^{-2})	3.08 (1.90×10^{-2})	10.1 (1.94×10^{-2})	5.33 (7.78×10^{-2})
L_2-Norm Pen. Poly. ALVM	1.12 (6.84×10^{-4})	10.3 (2.64×10^{-1})	6.24 (6.08×10^{-1})	8390 (4.59×10^{2})	12300 (9.03×10^{2})
Thin-Plate spline ALVM	2.59 (1.65×10^{-2})	190 (2.00×10^{0})	36.9 (1.41×10^{0})	254 (1.69×10^{0})	314 (1.24×10^{1})
Miller-Startz SVR	1.05 (4.08×10^{-4})	1.07 (1.40×10^{-3})	1 (3.01×10^{-3})	1.1 (1.14×10^{-3})	1 (4.70×10^{-3})

Table 5.14: (Relative) MSE of Standard Errors of $\hat{\beta}_{\text{OLS}}$ (with Estimated Standard Error) for Two-Covariate Linear Regression Model

Model	Homosked.	Add. Het.		Mult. Het.	
	$\mathcal{H} = \emptyset$	$\mathcal{H} = \{2\}$	$\mathcal{H} = \{2,3\}$	$\mathcal{H} = \{2\}$	$\mathcal{H} = \{2,3\}$
HC3	2.66 (5.56×10^{-6})	1.71 (3.46×10^{-5})	1.91 (9.65×10^{-5})	1.65 (4.00×10^{-5})	1.94 (7.03×10^{-4})
HC4	2.41 (4.61×10^{-6})	1.59 (2.90×10^{-5})	1.73 (7.54×10^{-5})	1.52 (3.32×10^{-5})	1.79 (5.84×10^{-4})
HC6	71.2 (2.15×10^{-5})	37.2 (1.37×10^{-4})	39.2 (3.66×10^{-4})	27.2 (1.49×10^{-4})	12.3 (1.49×10^{-3})
Homoskedastic	1 (2.18×10^{-6})	2.88 (3.92×10^{-5})	4.44 (1.21×10^{-4})	2.75 (3.81×10^{-5})	1.19 (2.43×10^{-4})
Basic ALVM	2.52 (5.01×10^{-6})	1.66 (3.15×10^{-5})	1.81 (8.49×10^{-5})	1.62 (3.76×10^{-5})	1.92 (6.76×10^{-4})
Clustering ALVM	1.34 (2.99×10^{-6})	1 (1.97×10^{-5})	1.29 (5.66×10^{-5})	1 (2.30×10^{-5})	1.18 (3.08×10^{-4})
Linear ALVM	1.3 (2.83×10^{-6})	1 (2.09×10^{-5})	1 (4.91×10^{-5})	1.35 (2.88×10^{-5})	1 (2.87×10^{-4})
L_2-Norm Pen. Poly. ALVM	1.38 (3.15×10^{-6})	1.14 (2.22×10^{-5})	1.32 (6.03×10^{-5})	1.07 (2.99×10^{-5})	1.67 (6.09×10^{-4})
Thin-Plate spline ALVM	2.83 (5.42×10^{-6})	1.86 (3.22×10^{-5})	1.79 (7.34×10^{-5})	1.82 (3.71×10^{-5})	2.17 (5.84×10^{-4})
Miller-Startz SVR	29.3 (1.80×10^{-5})	18.5 (1.04×10^{-4})	18.7 (2.42×10^{-4})	15.9 (1.06×10^{-4})	11.4 (1.03×10^{-3})

Discussion of Tables 5.11-5.14

It is evident from Table 5.11 that the homoskedastic estimator performs best in terms of unstandardised MSE in the homoskedastic case, while the linear ALVM and L_2-penalised polynomial ALVM perform best in the additive heteroskedasticity cases and multiplicative heteroskedasticity cases, respectively. The results are more varied for standardised MSE (Table 5.12): the homoskedastic estimator is still the best under homoskedasticity, but the linear, spline, and clustering ALVMs and the Miller-Startz model each perform best in one heteroskedastic case.

From Table 5.13, one observes that OLS, all of the HCCMEs, the clustering and linear ALVM perform best under homoskedasticity. The clustering ALVM also performs best under two of the heteroskedastic scenarios (where the error variance is a function of one covariate), while the Miller-Startz model performs best under the other two scenarios (the error variance is a function of both covariates). The spline and polynomial models show some instability here, with erratic performance and very high standard errors, especially under multiplicative heteroskedasticity. Looking at MSE for estimating the SE($\hat{\beta}_j$), $j = 1, 2, \ldots, p$ (Table 5.14), the homoskedastic estimator is again the best under homoskedasticity, while the linear and clustering ALVMs perform best under the heteroskedastic DGPs.

Appendix E.5 provides results for a simulation like the one discussed in this section but with two covariates generated from a bivariate normal distribution with mean vector $[3, 3]$ and covariance matrix $\begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$ (thus correlation coefficient $\rho = 0.5$). The purpose of this simulation case is to monitor the performance of the methods under multicollinearity, an issue which occurs commonly in practice with multiple linear regression models.

Comparing Tables E.18-E.21 with Tables 5.11-5.14, the following characteristics of performance between the DGP with two independent uniform covariates and with two correlated normal covariates emerge. There are no substantial differences in $\overline{\text{MSE}}_{\text{ust}}(\hat{\boldsymbol{\omega}})$ between the two cases. In terms of $\overline{\text{MSE}}_{\text{st}}(\hat{\boldsymbol{\omega}})$, the only difference emerges in the last heteroskedasticity setting, multiplicative with $\mathcal{H} = \{2, 3\}$. Whereas the HC6 HCCME performed best under an independent uniform design, Miller-Startz SVR performs best under a correlated normal design. There is also a change in performance in terms of MSE($\hat{\beta}_{\text{FWLS}}$). Under the independent uniform DGP, the clustering ALVM and Miller-Startz SVR alternated as the best-performing methods with heteroskedasticity. Under the correlated normal DGP, Miller-Startz SVR is superior in three out of four heteroskedastic scenarios, and the linear ALVM in the fourth instance. The performance of the linear ALVM also improves under the correlated normal design in terms of the MSE(SE($\hat{\boldsymbol{\beta}}$)) metric. It is the best-performing method in three out of four heteroskedastic scenarios; interestingly, the homoskedastic estimator performs best in the fourth heteroskedastic case (multiplicative heteroskedasticity with $\mathcal{H} = \{2, 3\}$).

5.3.2.1 ANLVM Results for this Simulation Configuration

Table 5.15 reports performance metrics for three ANLVMs for the same simulation reported on in Tables 5.11-5.14.

Table 5.15: Relative Performance Metrics (with Estimated Standard Errors) for ANLVMs Fitted to Two-Covariate Linear Regression Model

		Homosked.	Add. Het.		Mult. Het.	
Metric	ANLVM	$\mathcal{H} = \emptyset$	$\mathcal{H} = \{2\}$	$\mathcal{H} = \{2, 3\}$	$\mathcal{H} = \{2\}$	$\mathcal{H} = \{2, 3\}$
$\overline{\text{MSE}}_{\text{ust}}(\hat{\boldsymbol{\omega}})$	Quadratic	2.22 (5.67×10^{-4})	0.765 (3.96×10^{-2})	0.922 (4.87×10^{-1})	0.587 (4.06×10^{-2})	1.22 (8.28×10^{0})
	Exponential	2.22 (5.81×10^{-4})	1.88 (1.00×10^{-1})	1.5 (5.92×10^{-1})	0.559 (7.13×10^{-2})	0.719 (8.82×10^{0})
	Clustering	3.26 (9.29×10^{-4})	2.16 (7.64×10^{-2})	2.13 (5.47×10^{-1})	1.12 (9.16×10^{-2})	1.47 (8.62×10^{0})
$\overline{\text{MSE}}_{\text{std}}(\hat{\boldsymbol{\omega}})$	Quadratic	2.22 (5.67×10^{-4})	0.451 (7.38×10^{-4})	1.34 (1.32×10^{-2})	0.46 (1.21×10^{-3})	8.83 (5.36×10^{-1})
	Exponential	2.22 (5.81×10^{-4})	0.745 (9.02×10^{-4})	1.48 (3.94×10^{-3})	0.307 (5.99×10^{-4})	1.14 (1.16×10^{-2})
	Clustering	3.26 (9.29×10^{-4})	1.41 (1.36×10^{-3})	2.34 (4.21×10^{-3})	0.989 (1.10×10^{-3})	2.84 (1.54×10^{-2})
MSE($\hat{\boldsymbol{\beta}}_{\text{FWLS}}$)	Quadratic	1.01 (4.01×10^{-4})	0.935 (2.01×10^{-3})	0.931 (5.04×10^{-3})	0.968 (1.86×10^{-3})	1.5 (1.48×10^{-2})
	Exponential	1.02 (4.03×10^{-4})	0.955 (2.05×10^{-3})	0.923 (4.86×10^{-3})	0.957 (1.90×10^{-3})	0.748 (1.36×10^{-2})
	Clustering	1.01 (4.08×10^{-4})	0.976 (2.02×10^{-3})	1.03 (4.88×10^{-3})	1.02 (1.85×10^{-3})	0.958 (1.37×10^{-2})
MSE(SE($\hat{\boldsymbol{\beta}}$))	Quadratic	1.3 (5.56×10^{-6})	0.769 (3.46×10^{-5})	0.883 (9.65×10^{-5})	0.886 (4.00×10^{-5})	1.85 (7.03×10^{-4})
	Exponential	1.3 (4.61×10^{-6})	1.18 (2.90×10^{-5})	1.55 (7.54×10^{-5})	0.829 (3.32×10^{-5})	0.952 (5.84×10^{-4})
	Clustering	1.4 (2.15×10^{-5})	1.02 (1.37×10^{-4})	1.26 (3.66×10^{-4})	0.977 (1.49×10^{-4})	1.22 (1.49×10^{-3})

From Table 5.15, it is evident that under additive (quadratic) heteroskedasticity linked to *one* covariate, the quadratic ANLVM outperforms all competitors, including all of the ALVMs, in all four metrics. Under additive heteroskedasticity linked to *both* covariates, the quadratic ANLVM outperforms all competitors in terms of two metrics ($\overline{\text{MSE}}_{\text{ust}}(\hat{\boldsymbol{\omega}})$ and MSE(SE($\hat{\boldsymbol{\beta}}$))).

Under multiplicative (exponential) heteroskedasticity linked to *one* covariate, the exponential ANLVM outperforms all competitors, including all of the ANLVMs, in all four metrics. Under multiplicative heteroskedasticity linked to *both* covariates, the exponential ANLVM outperforms all competitors in terms of three metrics, the exception being $\overline{\text{MSE}}_{\text{st}}(\hat{\boldsymbol{\omega}})$. The exponential ANLVM also achieves the best result under the DGP with *additive* heteroskedasticity linked to two covariates.

The performance of the clustering ANLVM is satisfactory across the board, and similar to that of the clustering ALVM, but actually markedly better in certain instances, such as the MSE($\hat{\boldsymbol{\beta}}_{\text{FWLS}}$) metric under multiplicative heteroskedasticity linked to two covariates, where the clustering ALVM has a relative MC mean value of 1.41 and the clustering ANLVM relative MC mean value is 0.958.

5.3.3 Linear Regression with Eight Covariates

Tables 5.16-5.19 show results for a simulation with eight covariates generated independently from $U(0, 3)$. The homoskedastic case is shown in the first column followed by two cases of quadratic heteroskedasticity,

$\omega_i = (1 + x_{i2})^2$ and $\omega_i = (1 + x_{i2} + x_{i3} + x_{i4} + x_{i5})^2$, followed by two cases of exponential heteroskedasticity, $\omega_i = e^{x_{i2}}$ and $\omega_i = e^{x_{i2}+x_{i3}+x_{i4}+x_{i5}}$.

The models run are the same as those in §5.3.2 except that the thin-plate spline model has now been omitted for reasons of computation time.

Table 5.16: (Relative) Unstandardised MSE-of-Variances Estimate (with Estimated Standard Error) for Eight-Covariate Linear Regression Model

Model	Homosked.	Add. Het.		Mult. Het.	
	$\mathcal{H} = \emptyset$	$\mathcal{H} = \{2\}$	$\mathcal{H} = \{2,3,4,5\}$	$\mathcal{H} = \{2\}$	$\mathcal{H} = \{2,3,4,5\}$
HC3	115 (9.72×10^{-3})	25.6 (8.48×10^{-1})	14 (3.64×10^{1})	15.7 (1.00×10^{0})	2.86 (2.92×10^{5})
HC4	84.1 (6.64×10^{-3})	18.8 (5.90×10^{-1})	9.9 (2.25×10^{1})	11.5 (6.96×10^{-1})	1.88 (1.33×10^{5})
HC6	39.2 (4.84×10^{-4})	9.13 (8.43×10^{-2})	4.8 (2.96×10^{0})	5.78 (1.32×10^{-1})	1.55 (1.78×10^{5})
Homoskedastic	1 (3.04×10^{-4})	3.67 (2.07×10^{-2})	1 (9.68×10^{-1})	3.19 (2.14×10^{-2})	1.13 (2.79×10^{3})
Basic ALVM	113 (9.25×10^{-3})	25.6 (8.55×10^{-1})	13.6 (3.38×10^{1})	15.4 (1.01×10^{0})	2.92 (2.74×10^{5})
Clustering ALVM	9.19 (1.70×10^{-3})	2.57 (1.34×10^{-1})	1.91 (6.12×10^{0})	1.57 (1.45×10^{-1})	1.13 (1.69×10^{4})
Linear ALVM	5.79 (9.71×10^{-4})	1 (5.38×10^{-2})	1.05 (2.59×10^{0})	1 (5.62×10^{-2})	1 (7.10×10^{3})
L_2-Norm Pen. Poly. ALVM	24.2 (5.18×10^{-3})	6.62 (3.26×10^{-1})	3.44 (1.50×10^{1})	4.35 (3.72×10^{-1})	1.27 (1.02×10^{5})
Miller-Startz SVR	23 (9.13×10^{-4})	5.84 (6.67×10^{-2})	2.91 (2.76×10^{0})	4.02 (6.99×10^{-2})	1.03 (8.14×10^{3})

Table 5.17: (Relative) Standardised MSE-of-Variances Estimate (with Estimated Standard Error) for Eight-Covariate Linear Regression Model

Model	Homosked.	Add. Het.		Mult. Het.	
	$\mathcal{H} = \emptyset$	$\mathcal{H} = \{2\}$	$\mathcal{H} = \{2,3,4,5\}$	$\mathcal{H} = \{2\}$	$\mathcal{H} = \{2,3,4,5\}$
HC3	115 (9.72×10^{-3})	8.34 (1.19×10^{-2})	5.64 (1.08×10^{-2})	8.5 (1.25×10^{-2})	61.7 (1.50×10^{0})
HC4	84.1 (6.64×10^{-3})	6.02 (8.14×10^{-3})	4.06 (7.26×10^{-3})	6.15 (8.69×10^{-3})	42.8 (1.14×10^{0})
HC6	39.2 (4.84×10^{-4})	2.48 (5.54×10^{-4})	1.8 (5.66×10^{-4})	2.45 (6.32×10^{-4})	1 (1.13×10^{-2})
Homoskedastic	1 (3.04×10^{-4})	8.7 (1.35×10^{-2})	2.79 (5.36×10^{-3})	11.3 (2.02×10^{-2})	1010 (8.62×10^{0})
Basic ALVM	113 (9.25×10^{-3})	7.94 (1.11×10^{-2})	5.47 (1.03×10^{-2})	7.96 (1.12×10^{-2})	26.7 (8.29×10^{-1})
Clustering ALVM	9.19 (1.70×10^{-3})	1.15 (4.49×10^{-3})	1.69 (5.13×10^{-3})	1 (4.23×10^{-3})	101 (2.25×10^{0})
Linear ALVM	5.79 (9.71×10^{-4})	1 (3.99×10^{-3})	1.13 (4.01×10^{-3})	1.32 (4.75×10^{-3})	180 (3.43×10^{0})
L_2-Norm Pen. Poly. ALVM	24.2 (5.18×10^{-3})	6.18 (2.11×10^{-2})	2.72 (1.24×10^{-2})	8.59 (3.46×10^{-2})	6430 (1.74×10^{2})
Miller-Startz SVR	23 (9.13×10^{-4})	1.29 (1.03×10^{-3})	1 (9.91×10^{-4})	1.23 (1.05×10^{-3})	5.28 (1.66×10^{-1})

Table 5.18 (Relative) MSE of FWLS Estimate of $\boldsymbol{\beta}$ (with Estimated Standard Error) for Eight-Covariate Linear Regression Model

Model	Homosked.	Add. Het.		Mult. Het.	
	$\mathcal{H} = \emptyset$	$\mathcal{H} = \{2\}$	$\mathcal{H} = \{2,3,4,5\}$	$\mathcal{H} = \{2\}$	$\mathcal{H} = \{2,3,4,5\}$
OLS	1 (6.72×10^{-4})	1.21 (4.03×10^{-3})	1.03 (2.87×10^{-2})	1.41 (3.47×10^{-3})	2.7 (7.47×10^{-1})
HC3	1.07 (7.27×10^{-4})	1.21 (3.97×10^{-3})	1.07 (3.07×10^{-2})	1.4 (3.41×10^{-3})	2.24 (6.34×10^{-1})
HC4	1.04 (7.16×10^{-4})	1.2 (4.06×10^{-3})	1.07 (3.05×10^{-2})	1.4 (3.37×10^{-3})	2.33 (6.43×10^{-1})
HC6	1.08 (7.17×10^{-4})	1.25 (4.21×10^{-3})	1.12 (3.16×10^{-2})	1.46 (3.58×10^{-3})	2.51 (6.94×10^{-1})
Homoskedastic	1 (6.72×10^{-4})	1.21 (4.03×10^{-3})	1.03 (2.87×10^{-2})	1.41 (3.47×10^{-3})	2.7 (7.47×10^{-1})
Basic ALVM	1.2 (8.29×10^{-4})	1.32 (4.44×10^{-3})	1.2 (3.42×10^{-2})	1.58 (3.97×10^{-3})	2.07 (6.03×10^{-1})
Clustering ALVM	1.14 (7.90×10^{-4})	1.08 (4.33×10^{-2})	1.26 (2.12×10^{-1})	1 (4.43×10^{-3})	2.39 (5.29×10^{0})
Linear ALVM	1.29 (9.99×10^{-4})	1.28 (5.11×10^{-3})	1.52 (5.35×10^{-2})	1.5 (5.73×10^{-3})	1.85 (7.08×10^{-1})
L_2-Norm Pen. Poly. ALVM	1240 (4.98×10^{1})	76 (9.31×10^{0})	89 (1.07×10^{2})	470 (7.09×10^{1})	5.65 (3.88×10^{1})
Miller-Startz SVR	1.04 (6.91×10^{-4})	1 (3.36×10^{-3})	1 (2.89×10^{-2})	1.04 (2.62×10^{-3})	1 (2.96×10^{-1})

Table 5.19: (Relative) MSE of Standard Errors of $\hat{\beta}_{\text{OLS}}$ (with Estimated Standard Error) for Eight-Covariate Linear Regression Model

Model	Homosked.	Add. Het.		Mult. Het.	
	$\mathcal{H} = \emptyset$	$\mathcal{H} = \{2\}$	$\mathcal{H} = \{2,3,4,5\}$	$\mathcal{H} = \{2\}$	$\mathcal{H} = \{2,3,4,5\}$
HC3	3.47 (1.51×10^{-5})	2.63 (1.14×10^{-4})	2.31 (7.71×10^{-4})	2.46 (1.05×10^{-4})	2.51 (8.64×10^{-2})
HC4	2.73 (9.86×10^{-6})	2.05 (7.09×10^{-5})	1.74 (4.70×10^{-4})	1.95 (6.90×10^{-5})	2.07 (4.91×10^{-2})
HC6	71 (3.97×10^{-5})	42.8 (2.74×10^{-4})	43 (1.97×10^{-3})	36.4 (2.63×10^{-4})	10.7 (9.76×10^{-2})
Homoskedastic	1 (4.29×10^{-6})	1.31 (5.15×10^{-5})	1.13 (3.43×10^{-4})	1.66 (6.18×10^{-5})	1 (2.38×10^{-2})
Basic ALVM	2.9 (1.17×10^{-5})	2.26 (8.73×10^{-5})	1.9 (5.79×10^{-4})	2.19 (8.72×10^{-5})	2.52 (7.61×10^{-2})
Clustering ALVM	1.23 (5.11×10^{-6})	1 (3.96×10^{-5})	1 (3.09×10^{-4})	1 (4.00×10^{-5})	1.27 (2.19×10^{-2})
Linear ALVM	1.17 (4.99×10^{-6})	1.28 (5.71×10^{-5})	1.02 (3.28×10^{-4})	1.9 (8.25×10^{-5})	1.14 (3.09×10^{-2})
L_2-Norm Pen. Poly. ALVM	2.14 (1.39×10^{-5})	2.13 (1.18×10^{-4})	1.75 (8.30×10^{-4})	2.62 (1.43×10^{-4})	3.8 (1.47×10^{-1})
Miller-Startz SVR	28.5 (3.46×10^{-5})	20.8 (2.16×10^{-4})	18 (1.53×10^{-3})	19.4 (2.01×10^{-4})	10.5 (5.75×10^{-2})

Discussion of Tables 5.16-5.19

In terms of unstandardised MSE for estimating $\boldsymbol{\omega}$, the homoskedastic estimator performs best in the homoskedastic DGP but also, surprisingly, in the additive heteroskedastic DGP where the error variance was a quadratic function of four covariates (third column of Table 5.16). In the other three heteroskedastic DGPs, the linear ALVM performed best by this metric.

The results were more varied in terms of the standardised MSE for estimating $\boldsymbol{\omega}$ (Table 5.17). The homoskedastic estimator performed best under homoskedasticity, while the linear ALVM, clustering ALVM, Miller-Startz SVR model, and HC6 HCCME each performed best in one of the heteroskedastic DGPs.

Considering the MSE of the FWLS estimator (Table 5.18), the homoskedastic approach was best under homoskedasticity (although not statistically significantly better than some of the HCCMEs or Miller-Startz SVR model). The Miller-Startz SVR model performed best in three of the four heteroskedastic scenarios, though only by a statistically significant margin in one of these. In the other heteroskedastic DGP—multiplicative heteroskedasticity involving only one covariate—the clustering ALVM performed best, but not by a significant margin over Miller-Startz SVR.

Finally, referring to Table 5.19, in terms of the MSE of standard error estimates of $\hat{\boldsymbol{\beta}}_{\text{OLS}}$, the homoskedastic estimator performed best in the homoskedastic DGP and one heteroskedastic DGP, while the clustering ALVM performed best in three heteroskedastic DGPs—in one case, virtually neck-and-neck with the linear ALVM.

As with the two-covariate case, an eight-covariate simulation was conducted with correlated normal covariates. These covariates were generated from a multivariate normal distribution with a mean vector of 3s and covariance matrix

$$\boldsymbol{\Sigma} = \begin{bmatrix} 1 & 0.5 & 0.5 & 0.5 & -0.5 & -0.5 & -0.5 & -0.5 \\ 0.5 & 1 & 0.5 & 0.5 & -0.5 & -0.5 & -0.5 & -0.5 \\ 0.5 & 0.5 & 1 & 0.5 & -0.5 & -0.5 & -0.5 & -0.5 \\ 0.5 & 0.5 & 0.5 & 1 & -0.5 & -0.5 & -0.5 & -0.5 \\ -0.5 & -0.5 & -0.5 & -0.5 & 1 & 0.5 & 0.5 & 0.5 \\ -0.5 & -0.5 & -0.5 & -0.5 & 0.5 & 1 & 0.5 & 0.5 \\ -0.5 & -0.5 & -0.5 & -0.5 & 0.5 & 0.5 & 1 & 0.5 \\ -0.5 & -0.5 & -0.5 & -0.5 & 0.5 & 0.5 & 0.5 & 1 \end{bmatrix}.$$

Thus, the first four covariates are all positively correlated with one another, the last four covariates are all positively correlated with one another, and the first four covariates are all negatively correlated with the last four, with all correlation coefficients having a magnitude of 0.5. The purpose of this simulation case is to monitor the performance of the methods under higher-dimensional multicollinearity, an issue which occurs commonly in practice with multiple linear regression models.

Tables E.22-E.25 in §E.6 show results for this eight-covariate simulation with multicollinearity. Comparing these results with those in Tables 5.16-5.19, the following similarities and differences are observed. In Table 5.16, the lowest unstandardised MSE for the variance estimates was achieved by the homoskedastic estimator in the first and third DGPs and by the linear ALVM in the other three DGPs. In Table E.22, the 'winner' was the same in the first, second, and fourth DGPs. However, in the third DGP, the linear ALVM was now better than the homoskedastic DGP, while in the fifth DGP, the Miller-Startz SVR model performed best.

In Table 5.17, the lowest standardised MSE for the variance estimates was obtained by a different method for each DGP: going from left to right, the 'winners' were the homoskedastic estimator, the linear ALVM, the Miller-Startz SVR model, the clustering ALVM, and the HC6 HCCME. In Table E.23, it is apparent that the 'winners' in the multicollinear simulation were nearly the same; the only change is that the linear ALVM performs best in the third DGP, rather than the clustering ALVM.

In Table 5.18, the lowest MSE for FWLS estimation of $\boldsymbol{\beta}$ was achieved by the homoskedastic estimator (OLS) for the first (homoskedastic) DGP, and by the Miller-Startz SVR model for the other four DGPs with the exception of the fourth, for which the clustering ALVM was superior, albeit not by a statistically significant margin. In Table E.24, the results were similar except that in the fourth DGP, the Miller-Startz SVR model now outperformed the clustering ALVM (by a statistically significant margin).

In Table 5.19, the lowest MSE for estimating $\text{SE}(\hat{\boldsymbol{\beta}}_{\text{OLS}})$ was achieved by the homoskedastic estimator in the first and fifth DGPs and by the clustering ALVM in the other three DGPs. However, Table E.25 shows that in the multicollinear simulation, the homoskedastic estimator produces the lowest MSE for estimating $\text{SE}(\hat{\boldsymbol{\beta}}_{\text{OLS}})$ in all five DGPs.

To summarise, the performance of the ALVMs seems to have deteriorated slightly in the presence of multicollinearity in the eight-covariate linear regression model, relative to the homoskedastic estimator and the Miller-Startz SVR model. This is particularly true for the fourth metric, MSE of $\text{SE}(\hat{\boldsymbol{\beta}}_{\text{OLS}})$, relative to the homoskedastic estimator. However, by this metric the ALVMs are still outperforming the Miller-Startz SVR model by a wide margin under multicollinearity.

5.3.3.1 ANLVM Results for this Simulation Configuration

Table 5.20 reports performance metrics for three ANLVMs for the same simulation reported on in Tables 5.16-5.19.

Table 5.20: Relative Performance Metrics (with Estimated Standard Errors) for ANLVMs Fitted to Eight-Covariate Linear Regression Model

		Homosked.	Add. Het.		Mult. Het.	
Metric	ANLVM	$\mathcal{H} = \emptyset$	$\mathcal{H} = \{2\}$	$\mathcal{H} = \{2,3,4,5\}$	$\mathcal{H} = \{2\}$	$\mathcal{H} = \{2,3,4,5\}$
$\overline{\text{MSE}}_{\text{ust}}(\hat{\boldsymbol{\omega}})$	Quadratic	8.99 (7.69×10^{-3})	1.08 (1.53×10^{-1})	1.54 (1.72×10^{1})	0.741 (1.38×10^{-1})	0.954 (1.11×10^{4})
	Exponential	6.44 (1.31×10^{-3})	1.82 (1.50×10^{-1})	1.5 (7.23×10^{0})	0.828 (2.13×10^{-1})	1.08 (1.56×10^{5})
	Clustering	9.48 (1.83×10^{-3})	2.57 (1.33×10^{-1})	1.9 (6.09×10^{0})	1.55 (1.46×10^{-1})	1.08 (1.28×10^{4})
$\overline{\text{MSE}}_{\text{std}}(\hat{\boldsymbol{\omega}})$	Quadratic	8.99 (7.69×10^{-3})	0.639 (1.59×10^{-2})	1.51 (2.83×10^{-2})	0.848 (5.67×10^{-2})	227 (1.64×10^{1})
	Exponential	6.44 (1.31×10^{-3})	0.498 (2.05×10^{-3})	1.11 (3.94×10^{-3})	0.299 (1.53×10^{-3})	45 (1.57×10^{0})
	Clustering	9.48 (1.83×10^{-3})	1.16 (4.82×10^{-3})	1.68 (4.95×10^{-3})	0.983 (4.36×10^{-3})	85.1 (1.91×10^{0})
$\text{MSE}(\hat{\boldsymbol{\beta}}_{\text{FWLS}})$	Quadratic	1.13 (6.72×10^{-4})	0.854 (4.03×10^{-3})	1.07 (2.87×10^{-2})	0.949 (3.47×10^{-3})	1.84 (7.47×10^{-1})
	Exponential	1.08 (7.27×10^{-4})	0.776 (3.97×10^{-3})	1.01 (3.07×10^{-2})	0.787 (3.41×10^{-3})	0.959 (6.34×10^{-1})
	Clustering	1.12 (7.16×10^{-4})	0.878 (4.06×10^{-3})	1.1 (3.05×10^{-2})	0.885 (3.37×10^{-3})	1.22 (6.43×10^{-1})
$\text{MSE}(\text{SE}(\hat{\boldsymbol{\beta}}))$	Quadratic	1.58 (1.51×10^{-5})	0.992 (1.14×10^{-4})	1.2 (7.71×10^{-4})	0.968 (1.05×10^{-4})	1.37 (8.64×10^{-2})
	Exponential	1.2 (9.86×10^{-6})	0.94 (7.09×10^{-5})	0.96 (4.70×10^{-4})	0.896 (6.90×10^{-5})	1.44 (4.91×10^{-2})
	Clustering	1.27 (3.97×10^{-5})	1.01 (2.74×10^{-4})	0.998 (1.97×10^{-3})	0.984 (2.63×10^{-4})	1.62 (9.76×10^{-2})

From Table 5.20, it is evident that the exponential ANLVM is more successful than the other two under most DGPs in this eight-covariate simulation. Under additive (quadratic) heteroskedasticity linked to one covariate, the exponential ANLVM, surprisingly, is the winner over the quadratic ANLVM and all other models in terms of three out of four metrics. It is also superior to the quadratic ANLVM under additive heteroskedasticity linked to four covariates. The exponential ANLVM keeps up its good performance under the multiplicative heteroskedastic DGPs, except that in terms of $\overline{\text{MSE}}_{\text{ust}}(\hat{\boldsymbol{\omega}})$, the quadratic ALVM actually performs better in both DGPs (heteroskedasticity linked to one covariate and linked to four covariates). As in the lower-dimensional simulations, the clustering ANLVM yields results comparable to those of the clustering ALVM, but better in certain respects.

5.3.4 Linear Regression with Sixteen Covariates

Tables 5.21-5.24 show results for a simulation with sixteen covariates generated independently from $U(0,3)$. In each instance, the MC mean estimate of the metric is shown first with the estimated standard error beneath it in brackets. The homoskedastic case is shown in the first column followed by two cases of quadratic heteroskedasticity, $\omega_i = (1 + x_{i2})^2$ and $\omega_i = (1 + x_{i2} + \cdots + x_{i9})^2$, followed by two cases of exponential heteroskedasticity, $\omega_i = e^{x_{i2}}$ and $\omega_i = e^{x_{i2}+\cdots+x_{i9}}$. The metrics used and the format of the results are the same as for the lower-dimensional simulations. The models run are the same as in the eight-covariate simulations, except that the L_2-norm penalised polynomial has now also been omitted due to high computation time.

Table 5.21: (Relative) Unstandardised MSE-of-Variances Estimate (with Estimated Standard Error) for Sixteen-Covariate Linear Regression Model

	Homosked.	Add. Het.		Mult. Het.	
Model	$\mathcal{H} = \emptyset$	$\mathcal{H} = \{2\}$	$\mathcal{H} = \{2, \ldots, 9\}$	$\mathcal{H} = \{2\}$	$\mathcal{H} = \{2, \ldots, 9\}$
HC3	125 (1.22×10^{-2})	20.1 (1.15×10^{0})	21.8 (4.75×10^{2})	17.1 (1.36×10^{0})	2.26 $(9.14 \times 10^{+12})$
HC4	70.1 (6.29×10^{-3})	11.2 (5.81×10^{-1})	12.2 (2.41×10^{2})	9.42 (6.31×10^{-1})	1.23 $(4.23 \times 10^{+12})$
HC6	35 (4.73×10^{-4})	6.02 (6.68×10^{-2})	6.3 (2.41×10^{1})	5.26 (9.90×10^{-2})	1.09 $(3.17 \times 10^{+12})$
Homoskedastic	1 (3.57×10^{-4})	2.14 (2.57×10^{-2})	1 (1.21×10^{1})	2.59 (2.65×10^{-2})	1.06 $(8.83 \times 10^{+10})$
Basic ALVM	117 (1.12×10^{-2})	19.2 (1.12×10^{0})	20.8 (4.51×10^{2})	16.5 (1.25×10^{0})	1.81 $(8.37 \times 10^{+12})$
Clustering ALVM	15 (2.44×10^{-3})	2.62 (2.14×10^{-1})	3.28 (9.38×10^{1})	2.12 (2.23×10^{-1})	1.05 $(2.97 \times 10^{+11})$
Linear ALVM	8.97 (1.22×10^{-3})	1 (8.44×10^{-2})	1.93 (3.85×10^{1})	1 (7.59×10^{-2})	1.04 $(1.30 \times 10^{+11})$
Miller-Startz SVR	21.8 (7.47×10^{-4})	4.23 (6.38×10^{-2})	4.12 (2.61×10^{1})	3.97 (7.03×10^{-2})	1 $(1.46 \times 10^{+11})$

Table 5.22: (Relative) Standardised MSE-of-Variances Estimate (with Estimated Standard Error) for Sixteen-Covariate Linear Regression Model

	Homosked.	Add. Het.		Mult. Het.	
Model	$\mathcal{H} = \emptyset$	$\mathcal{H} = \{2\}$	$\mathcal{H} = \{2, \ldots, 9\}$	$\mathcal{H} = \{2\}$	$\mathcal{H} = \{2, \ldots, 9\}$
HC3	125 (1.22×10^{-2})	8.31 (1.92×10^{-2})	11.8 (1.37×10^{-2})	9.61 (2.26×10^{-2})	162 (4.82×10^{3})
HC4	70.1 (6.29×10^{-3})	4.58 (1.01×10^{-2})	6.55 (7.02×10^{-3})	5.16 (1.14×10^{-2})	71.9 (1.92×10^{3})
HC6	35 (4.73×10^{-4})	1.71 (5.06×10^{-4})	3.08 (4.48×10^{-4})	1.75 (5.93×10^{-4})	1 (4.41×10^{1})
Homoskedastic	1 (3.57×10^{-4})	5.56 (1.26×10^{-2})	1 (1.45×10^{-3})	8.72 (2.05×10^{-2})	671 (1.07×10^{4})
Basic ALVM	117 (1.12×10^{-2})	7.19 (1.65×10^{-2})	11 (1.27×10^{-2})	8.09 (1.88×10^{-2})	78.3 (1.83×10^{3})
Clustering ALVM	15 (2.44×10^{-3})	1.58 (9.03×10^{-3})	2.29 (4.02×10^{-3})	1.46 (9.25×10^{-3})	931 (6.05×10^{4})
Linear ALVM	8.97 (1.22×10^{-3})	1.35 (6.80×10^{-3})	1.57 (2.45×10^{-3})	1.64 (8.04×10^{-3})	427 (1.27×10^{4})
Miller-Startz SVR	21.8 (7.47×10^{-4})	1 (9.89×10^{-4})	1.87 (8.16×10^{-4})	1 (1.09×10^{-3})	10.8 (2.99×10^{2})

Table 5.23: (Relative) MSE of FWLS Estimate of $\boldsymbol{\beta}$ (with Estimated Standard Error) for Sixteen-Covariate Linear Regression Model

Model	Homosked.	Add. Het.		Mult. Het.	
	$\boldsymbol{\mathcal{H}} = \emptyset$	$\boldsymbol{\mathcal{H}} = \{2\}$	$\boldsymbol{\mathcal{H}} = \{2, \dots, 9\}$	$\boldsymbol{\mathcal{H}} = \{2\}$	$\boldsymbol{\mathcal{H}} = \{2, \dots, 9\}$
OLS	1 (4.89×10^{-4})	1.13 (3.74×10^{-3})	1 (9.04×10^{-2})	1.41 (3.61×10^{-3})	3.2 (4.79×10^{3})
HC3	1.07 (4.96×10^{-4})	1.17 (3.84×10^{-3})	1.04 (9.29×10^{-2})	1.39 (3.64×10^{-3})	1.72 (2.52×10^{3})
HC4	1.11 (5.26×10^{-4})	1.14 (3.82×10^{-3})	1.04 (9.20×10^{-2})	1.4 (3.70×10^{-3})	1.72 (2.60×10^{3})
HC6	1.15 (5.61×10^{-4})	1.23 (4.13×10^{-3})	1.07 (9.79×10^{-2})	1.5 (3.94×10^{-3})	2.26 (3.38×10^{3})
Homoskedastic	1 (4.89×10^{-4})	1.13 (3.74×10^{-3})	1 (9.04×10^{-2})	1.41 (3.61×10^{-3})	3.2 (4.79×10^{3})
Basic ALVM	1.33 (7.13×10^{-4})	1.33 (4.45×10^{-3})	1.22 (1.10×10^{-1})	1.56 (4.13×10^{-3})	2.1 (3.31×10^{3})
Clustering ALVM	1.22 (6.20×10^{-4})	1.14 (1.19×10^{-2})	1.15 (1.05×10^{-1})	1.49 (4.58×10^{-2})	2.69 (2.48×10^{4})
Linear ALVM	1.37 (6.83×10^{-4})	1 (3.52×10^{-3})	1.28 (1.18×10^{-1})	1 (2.64×10^{-3})	1.91 (3.75×10^{3})
Miller-Startz SVR	1.08 (5.14×10^{-4})	1.03 (3.49×10^{-3})	1.02 (9.28×10^{-2})	1.21 (3.15×10^{-3})	1 (1.85×10^{3})

Table 5.24: (Relative) MSE of Standard Errors of $\hat{\boldsymbol{\beta}}_{\text{OLS}}$ (with Estimated Standard Error) for Sixteen-Covariate Linear Regression Model

Model	Homosked.	Add. Het.		Mult. Het.	
	$\boldsymbol{\mathcal{H}} = \emptyset$	$\boldsymbol{\mathcal{H}} = \{2\}$	$\boldsymbol{\mathcal{H}} = \{2, \dots, 9\}$	$\boldsymbol{\mathcal{H}} = \{2\}$	$\boldsymbol{\mathcal{H}} = \{2, \dots, 9\}$
HC3	4.85 (1.86×10^{-5})	3.73 (1.88×10^{-4})	4.64 (3.89×10^{-3})	3.29 (2.05×10^{-4})	1.13 (9.06×10^{2})
HC4	2.8 (8.53×10^{-6})	2.45 (8.96×10^{-5})	2.88 (1.85×10^{-3})	2.24 (1.02×10^{-4})	1 (5.75×10^{2})
HC6	65.5 (3.09×10^{-5})	40.5 (3.07×10^{-4})	55.7 (6.87×10^{-3})	31.4 (3.17×10^{-4})	1.85 (7.55×10^{2})
Homoskedastic	1 (4.09×10^{-6})	1 (4.43×10^{-5})	1 (8.05×10^{-4})	1 (5.05×10^{-5})	1.85 (4.74×10^{2})
Basic ALVM	3.2 (1.22×10^{-5})	2.83 (1.32×10^{-4})	3.38 (2.72×10^{-3})	2.6 (1.47×10^{-4})	3.29 (6.67×10^{2})
Clustering ALVM	1.3 (4.79×10^{-6})	1.11 (5.13×10^{-5})	1.28 (9.79×10^{-4})	1.03 (5.67×10^{-5})	1.84 (4.56×10^{2})
Linear ALVM	1.15 (4.52×10^{-6})	1.07 (5.35×10^{-5})	1.16 (8.99×10^{-4})	1.03 (6.04×10^{-5})	1.65 (4.52×10^{2})
Miller-Startz SVR	28.7 (2.92×10^{-5})	22.5 (2.56×10^{-4})	26.6 (5.45×10^{-3})	19.4 (2.61×10^{-4})	3.97 (5.08×10^{2})

Discussion of Tables 5.21-5.24

Referring to Table 5.21, it is evident that in terms of unstandardised MSE for estimating the error variances $\boldsymbol{\omega}$, the homoskedastic estimator performs best in the homoskedastic DGP but also in one additive heteroskedasticity case. The linear ALVM performs best in both the additive and multiplicative heteroskedastic DGPs where only one covariate was involved in heteroskedasticity, while in the other multiplicative heteroskedasticity case, the Miller-Startz SVR model prevails.[120]

[120] The standard errors are enormous in this case due to the huge magnitude of the error variances, which have been computed by an exponential function with the exponent being the sum of eight $U(0, 3)$ random variables.

According to Table 5.22, the ALVMs are not at their best in terms of standardised MSE for estimating the error variances $\boldsymbol{\omega}$ in this high-dimensional setting. The homoskedastic estimator again performs best in the homoskedastic DGP and one additive heteroskedastic DGP. Miller-Startz SVR performs best in both heteroskedastic DGPs where only one covariate was implicated in heteroskedasticity. In the other multiplicative heteroskedastic DGP, the HCCME HC6 performs best.

Turning to Table 5.23, the homoskedastic estimator (equivalently, OLS) results in the best MSE of the FWLS estimator in the homoskedastic DGP and one additive heteroskedastic DGP case (though not by a statistically significant margin, in the latter). The linear ALVM performs best in both heteroskedastic DGPs where only one covariate is involved in heteroskedasticity; again not by a statistically significant margin. Miller-Startz SVR is the clear winner in the other multiplicative heteroskedastic DGP case.

Finally, Table 5.24 shows that the homoskedastic estimator performs best in four out of five cases in MSE of $\mathrm{SE}(\hat{\boldsymbol{\beta}}_{\mathrm{OLS}})$, although the linear and clustering ALVMs are close behind. In the last case (multiplicative heteroskedasticity with eight covariates involved), the HCCMEs tend to outperform the modelling approaches.

No parallel simulation with correlated normal covariates was conducted in the sixteen-covariate case.

5.4 Simulation Results on Other Aspects of Auxiliary Variance Models

5.4.1 Feature Selection Performance

A limited experiment was performed to specifically explore the performance of the feature selection techniques discussed in §3.3.3. The settings of the experiment are summarised in Table 5.25. In each setting, both covariates were generated independently from $U(0,3)$ with $n = 100$.

The additive heteroskedastic function used was

$$g(\boldsymbol{X}'_{i\cdot}) = \begin{cases} \left(1 + \sum_{j\in\mathcal{H}} X_{ij}\right)^2 & \text{if } \mathcal{H} \neq \emptyset \\ 1 & \text{if } \mathcal{H} = \emptyset \end{cases}, \tag{5.23}$$

where $\mathcal{H}$ is the set of columns of $\boldsymbol{X}$ involved in heteroskedasticity (see second column of Table 5.25), and $\boldsymbol{X}_{\cdot 1}$ is a column of ones. The multiplicative heteroskedastic function used was

$$g(\boldsymbol{X}'_{i\cdot}) = \begin{cases} \exp\left\{\sum_{j\in\mathcal{H}} X_{ij}\right\} & \text{if } \mathcal{H} \neq \emptyset \\ 1 & \text{if } \mathcal{H} = \emptyset \end{cases}. \tag{5.24}$$

Table 5.25: Settings for Feature Selection Experiment

No. of Covariates $p-1$	$\mathcal{H}$	Het. Function
2	$\emptyset$	-
2	$\{2\}$	Additive
2	$\{2,3\}$	Additive
2	$\emptyset$	-
2	$\{2\}$	Multiplicative
2	$\{2,3\}$	Multiplicative

The additive heteroskedasticity configuration was repeated with a smaller sample size of $n = 20$ to investigate the small-sample performance of the feature selection techniques. $R = 10^4$ MC replications were generated for each configuration. Thus each MC proportion estimate (the proportion of times that a particular set of variables was selected) $\hat{\pi}$ in Table 5.27 has MC standard error $\sqrt{\dfrac{\pi(1-\pi)}{R}}$, which for $R = 10^4$ is maximised when $\pi = 0.5$ at a value of 0.005.

The feature selection methods used were as follows. Heteroskedasticity testing selection methods (§3.3.3.2) were used with both Breusch and Pagan's (1979) test and Evans and King's (1988) GLS test. In each instance, two different significance levels were tried ($\alpha = 0.05; 0.1$). Best subset selection methods (§3.3.3.3) were used with both the QGCV criterion (3.86) and five-fold CV (3.77). In each best subset selection case, the linear ALVM was used as well as the clustering ALVM with n_c determined by the elbow method with SWD criterion. The shrinkage method (which forms part of the LASSO ALVM fitting mechanism) was not included in this simulation due to the high computation time required.

In Tables 5.26, 5.27, and 5.28 below, the columns highlighted in green are those corresponding to the correct choice of features for each DGP. Except for the last two columns, the number shown is the proportion of replications for which a particular feature selection choice was made, within that DGP. The second-to-last column, labelled ‘Accuracy’, indicates the proportion of times that the exactly correct feature selection choice was made, across all three DGPs. The last column, labelled ‘SD’, indicates the standard deviation of the proportion of correct feature selection choices (green columns) across all three DGPs. A good feature selection method should have a high accuracy but also a low standard deviation, indicating that its performance is consistent across different DGPs.

Table 5.26: Feature Selection Relative Frequencies for Two-Covariate Model with Additive Heteroskedasticity, $n = 100$

	DGP $\mathcal{H} = \emptyset$				DGP $\mathcal{H} = \{2\}$				DGP $\mathcal{H} = \{2,3\}$				Overall Metrics	
Method	$\emptyset$	$\{2\}$	$\{3\}$	$\{2,3\}$	$\emptyset$	$\{2\}$	$\{3\}$	$\{2,3\}$	$\emptyset$	$\{2\}$	$\{3\}$	$\{2,3\}$	Accuracy	SD
QGCV (linear)	0.699	0.139	0.137	0.025	0.001	0.908	0.000	0.091	0.008	0.148	0.093	0.750	0.804	0.096
QGCV (clustering)	0.824	0.077	0.068	0.032	0.029	0.758	0.004	0.209	0.188	0.284	0.181	0.346	0.668	0.217
CV (linear)	0.605	0.174	0.170	0.051	0.011	0.737	0.002	0.250	0.039	0.149	0.103	0.709	0.693	0.060
CV (cluster)	0.728	0.103	0.103	0.066	0.119	0.638	0.018	0.225	0.258	0.247	0.192	0.302	0.554	0.183
B-P Test ($\alpha = 0.05$)	0.906	0.046	0.046	0.001	0.016	0.958	0.000	0.025	0.166	0.314	0.224	0.295	0.772	0.319
B-P Test ($\alpha = 0.1$)	0.812	0.091	0.086	0.010	0.004	0.933	0.000	0.063	0.071	0.255	0.184	0.490	0.783	0.202
E-K Test ($\alpha = 0.05$)	0.900	0.049	0.048	0.002	0.000	0.903	0.000	0.097	0.014	0.123	0.093	0.769	0.875	0.071
E-K Test ($\alpha = 0.1$)	0.807	0.089	0.094	0.011	0.000	0.840	0.000	0.160	0.004	0.070	0.062	0.864	0.846	0.030

Table 5.26 shows the proportion of replications where each variable selection outcome was achieved for each DGP setting of $\mathcal{H}$, in the first experiment with additive heteroskedasticity. In terms of overall accuracy, the Evans-King heteroskedasticity testing method with $\alpha = 0.05$ performs best, with an accuracy of over 87%, and a relatively low standard deviation of 0.071. The same method with significance level $\alpha = 0.1$ has almost as good accuracy and a much smaller standard deviation. The third-best combination of accuracy and standard deviation belongs to the QGCV method using the linear ALVM. The techniques based on the clustering ALVM and the Breusch-Pagan heteroskedasticity test both showed poor performance under the DGP where both covariates were involved in heteroskedasticity.

Table 5.27: Feature Selection Relative Frequencies for Two-Covariate Model with Multiplicative Heteroskedasticity

	DGP $\mathcal{H} = \emptyset$				DGP $\mathcal{H} = \{2\}$				DGP $\mathcal{H} = \{2,3\}$				Overall Metrics	
Method	$\emptyset$	$\{2\}$	$\{3\}$	$\{2,3\}$	$\emptyset$	$\{2\}$	$\{3\}$	$\{2,3\}$	$\emptyset$	$\{2\}$	$\{3\}$	$\{2,3\}$	Accuracy	SD
QGCV (linear)	0.709	0.137	0.130	0.024	0.000	0.961	0.000	0.039	0.000	0.044	0.077	0.879	0.878	0.119
QGCV (clustering)	0.818	0.065	0.078	0.038	0.019	0.725	0.002	0.254	0.022	0.065	0.110	0.803	0.800	0.054
CV (linear)	0.604	0.172	0.171	0.052	0.004	0.780	0.001	0.215	0.007	0.076	0.075	0.842	0.745	0.101
CV (cluster)	0.744	0.090	0.101	0.065	0.080	0.666	0.011	0.243	0.068	0.172	0.225	0.535	0.658	0.089
B-P Test ($\alpha = 0.05$)	0.909	0.042	0.046	0.002	0.003	0.969	0.000	0.028	0.008	0.052	0.085	0.856	0.928	0.057
B-P Test ($\alpha = 0.1$)	0.812	0.086	0.092	0.009	0.000	0.932	0.000	0.067	0.001	0.016	0.030	0.952	0.910	0.066
E-K Test ($\alpha = 0.05$)	0.903	0.047	0.046	0.004	0.000	0.886	0.000	0.114	0.000	0.000	0.000	0.999	0.910	0.063
E-K Test ($\alpha = 0.1$)	0.810	0.086	0.091	0.013	0.000	0.824	0.000	0.176	0.000	0.000	0.000	1.000	0.854	0.098

Table 5.27 shows the feature selection performance for the second experiment, with multiplicative heteroskedasticity. It is evident that the Breusch-Pagan heteroskedasticity testing approach performs best in terms of accuracy (and very well in terms of standard deviation), followed by the Evans-King heteroskedasticity testing approach. The QGCV approach based on the linear model is also highly competitive (despite slightly inferior performance in the homoskedastic DGP). The QGCV best subset selection approaches outperform the CV best subset selection approaches and are also much faster. It is surprising that QGCV outperforms five-fold CV, but this may be because QGCV approximates leave-one-out CV, and the lower bias of leave-one-out CV (despite higher variance) may be favourable in this instance.

Table 5.28: Feature Selection Relative Frequencies for Two-Covariate Model with Additive Heteroskedasticity, $n = 20$

	DGP $\mathcal{H} = \emptyset$				DGP $\mathcal{H} = \{2\}$				DGP $\mathcal{H} = \{2,3\}$				Overall Metrics	
Method	$\emptyset$	$\{2\}$	$\{3\}$	$\{2,3\}$	$\emptyset$	$\{2\}$	$\{3\}$	$\{2,3\}$	$\emptyset$	$\{2\}$	$\{3\}$	$\{2,3\}$	Accuracy	SD
QGCV (linear)	0.645	0.151	0.181	0.023	0.343	0.505	0.119	0.032	0.203	0.085	0.662	0.050	0.461	0.294
QGCV (clustering)	0.662	0.117	0.129	0.092	0.566	0.268	0.100	0.067	0.380	0.054	0.354	0.213	0.350	0.230
CV (linear)	0.424	0.228	0.222	0.126	0.363	0.309	0.183	0.145	0.315	0.172	0.335	0.179	0.308	0.109
CV (cluster)	0.400	0.210	0.214	0.176	0.347	0.286	0.174	0.193	0.293	0.176	0.285	0.246	0.296	0.078
B-P Test ($\alpha = 0.05$)	0.924	0.038	0.038	0.000	0.884	0.100	0.015	0.000	0.735	0.015	0.250	0.000	0.319	0.416
B-P Test ($\alpha = 0.1$)	0.821	0.087	0.086	0.005	0.711	0.237	0.044	0.008	0.528	0.040	0.418	0.013	0.371	0.344
E-K Test ($\alpha = 0.05$)	0.905	0.047	0.044	0.004	0.423	0.521	0.017	0.039	0.297	0.019	0.636	0.049	0.541	0.335
E-K Test ($\alpha = 0.1$)	0.811	0.089	0.088	0.012	0.255	0.641	0.016	0.087	0.180	0.020	0.700	0.101	0.588	0.267

Table 5.28 shows the results for a third experiment with additive heteroskedasticity but with a small sample size of $n = 20$. Since the heteroskedasticty testing feature selection technique is contingent on power, which increases with sample size, it is not surprising that these methods are less effective than in the $n = 100$ case. The heteroskedasticity testing approach based on Evans and King's (1988) test is still the best overall in terms of accuracy. The heteroskedasticity testing approach based on Breusch and Pagan's (1979) test has almost no power to detect heteroskedasticity, especially in the $\mathcal{H} = \{2, 3\}$ case. The increased sampling error from the smaller sample size has also affected the performance of the QGCV and CV best subset selection techniques, though not to the same degree. The QGCV best subset method based on the linear ALVM remains competitive in terms of accuracy, but the five-fold CV best subset methods have the lowest standard deviations.

Table 5.29: Feature Selection Metrics for Four-Covariate Model with Additive Heteroskedasticity, $n = 50$

Method	Sensitivity	Specificity	Accuracy	SD
QGCV (linear)	0.482	0.858	0.252	0.243
QGCV (clustering)	0.487	0.665	0.126	0.093
CV (linear)	0.452	0.681	0.111	0.070
CV (cluster)	0.504	0.579	0.090	0.030
B-P Test ($\alpha = 0.05$)	0.192	0.959	0.176	0.261
B-P Test ($\alpha = 0.1$)	0.307	0.915	0.203	0.249
E-K Test ($\alpha = 0.05$)	0.470	0.930	0.327	0.300
E-K Test ($\alpha = 0.1$)	0.594	0.873	0.342	0.236

Table 5.29 shows results for a higher-dimensional simulation with $(p - 1 = 4)$ features, based on $R = 10^3$ MC replications, with $n = 50$. In this case, there were 16 DGPs corresponding to the presence and absence of each feature in the quadratic heteroskedastic function. Instead of showing the relative frequencies for all the DGPs, two new metrics are shown. These are the feature selection 'sensitivity' (proportion of replications where a feature that *was* involved in heteroskedasticity was selected) and the feature selection 'specificity' (proportion of replications where a feature that *was not* involved in heteroskedasticity was *not* selected). These metrics are averaged across all features. The accuracy refers to the proportion of replications where the set of features selected was *exactly* that of the DGP. By sensitivity, specificity, and accuracy, the heteroskedasticity testing approach using Evans and King's (1988) GLS test performs the best. A test size of 0.1 seems to work slightly better than a test size of 0.05, as the gain in sensitivity more than compensates for the loss in specificity. The QGCV best subset method based on the linear ALVM is also competitive in terms of sensitivity and specificity. The heteroskedasticity testing method based on Breusch and Pagan's (1979) test have excellent specificity but low sensitivity.

Collectively, it appears that the heteroskedasticity testing approach based on Evans and King's (1988) GLS test performs best. However, one should bear in mind that the performance of a heteroskedasticity test may be sensitive to the type of heteroskedasticity.[121] Also, the feature selection simulations reported on here were run only after running the time-consuming ALVM performance simulations reported in §5.3. For these two reasons, the QGCV procedure was used predominantly as the feature selection method in the ALVM and ANLVM performance simulations discussed in §5.3.

[121] See the performance of Evans and King's (1988) test in the illustration in §5.6.3.

5.4.2 Stability of Auxiliary Linear Variance Models across Different Design Matrices

It was assumed from the outset of this research in §1.1.3 that the design matrix $\boldsymbol{X}$ is either nonstochastic or that otherwise all statistical results are conditional on $\boldsymbol{X}$. However, the question arises whether, in case of stochastic $\boldsymbol{X}$, the unconditional statistical results are stable with respect to different designs $\boldsymbol{X}$ drawn from a single distribution with CDF denoted $F_{\boldsymbol{X}}$. This section explores this question through a limited empirical simulation.

Let $\boldsymbol{X}$ be a predictor matrix drawn from $F_{\boldsymbol{X}}$. Let the vector of error variances associated with predictor matrix $\boldsymbol{X}$ be $\boldsymbol{\omega}(\boldsymbol{X}) = [\omega_1(\boldsymbol{X}'_{1\cdot}), \omega_2(\boldsymbol{X}'_{2\cdot}), \ldots, \omega_n(\boldsymbol{X}'_{n\cdot})]' = [g(\boldsymbol{X}'_{1\cdot}), g(\boldsymbol{X}'_{2\cdot}), \ldots, g(\boldsymbol{X}'_{n\cdot})]$, with heteroskedastic function g as introduced in §1.1.4 and $\boldsymbol{Z} = \boldsymbol{X}$ for simplicity.

If $\hat{\boldsymbol{\omega}}(\boldsymbol{X})$ is an estimator of $\boldsymbol{\omega}(\boldsymbol{X})$, an unconditional, unstandardised MSE for an element of this estimator, $\hat{\omega}_i(\boldsymbol{X})$,[122] can be written as

$$\mathrm{E}\left[\left(\hat{\omega}_i(\cdot) - \omega_i(\cdot)\right)^2\right] = \mathrm{E}_{\boldsymbol{X}}\left\{\mathrm{E}_{\boldsymbol{\epsilon}|\boldsymbol{X}}\left[\left(\hat{\omega}_i(\boldsymbol{X}) - \omega_i(\boldsymbol{X}'_{i\cdot})\right)^2\right]\right\}. \tag{5.25}$$

Note that a standardised version of this expectation, and all quantities introduced below based on it, is obtained by replacing $\left(\hat{\omega}_i(\boldsymbol{X}) - \omega_i(\boldsymbol{X}'_{i\cdot})\right)^2$ in (5.25) by $\left(\dfrac{\hat{\omega}_i(\boldsymbol{X})}{\omega_i(\boldsymbol{X}'_{i\cdot})} - 1\right)^2$. The standardised quantities weigh every observation as being of equal importance, regardless of the magnitude of its error variance $\omega_i(\boldsymbol{X}'_{i\cdot})$. The notation MSE in the rest of this section may refer either to $\overline{\mathrm{MSE}}_{\mathrm{ust}}$ or to $\overline{\mathrm{MSE}}_{\mathrm{std}}$.

Now, since expectation has been taken over $\boldsymbol{X}$, the whole expression is a function of functions $\hat{\omega}_i(\cdot)$ and $\omega_i(\cdot)$. Moreover, if the rows of $\boldsymbol{X}$ are independent then this expectation is the same for all $i \in \{1, 2, \ldots, n\}$, making the i subscripts on the left side of (5.25) essentially arbitrary.[123] Consequently, there is nothing to be gained by focusing on a particular index i across different designs $\boldsymbol{X}$ as in (5.25). It is more appropriate to focus on (5.26), which aggregates across all observations:

$$\begin{aligned}\mathrm{E}\left[\frac{1}{n}\sum_{i=1}^{n}\left(\hat{\omega}_i(\cdot) - \omega_i(\cdot)\right)^2\right] &= \mathrm{E}\left[\frac{1}{n}\left(\hat{\boldsymbol{\omega}}(\cdot) - \boldsymbol{\omega}(\cdot)\right)'\left(\hat{\boldsymbol{\omega}}(\cdot) - \boldsymbol{\omega}(\cdot)\right)\right] \\ &= \mathrm{E}_{\boldsymbol{X}}\left\{\mathrm{E}_{\boldsymbol{\epsilon}|\boldsymbol{X}}\left[\frac{1}{n}\left(\hat{\boldsymbol{\omega}}(\boldsymbol{X}) - \boldsymbol{\omega}(\boldsymbol{X})\right)'\left(\hat{\boldsymbol{\omega}}(\boldsymbol{X}) - \boldsymbol{\omega}(\boldsymbol{X})\right)\right]\right\} \\ &= \mathrm{E}_{\boldsymbol{X}}\left\{\mathrm{E}_{\boldsymbol{\epsilon}|\boldsymbol{X}}\left[\mathrm{MSE}(\hat{\boldsymbol{\omega}}(\boldsymbol{X}); \boldsymbol{\omega}(\boldsymbol{X}))\right]\right\}. \end{aligned} \tag{5.26}$$

If R_1 MC replications of $\boldsymbol{\epsilon}$, and therefore of $\boldsymbol{y}$ and $\boldsymbol{e}$, are generated, leading to variance estimates $\boldsymbol{\omega}(\boldsymbol{X})^{(r)}, r = 1, 2, \ldots, R_1$, then a MC estimator of $\mathrm{E}_{\boldsymbol{\epsilon}|\boldsymbol{X}}\left[\mathrm{MSE}(\hat{\boldsymbol{\omega}}(\boldsymbol{X}); \boldsymbol{\omega}(\boldsymbol{X}))\right]$ is given by

$$\begin{aligned}\widehat{\mathrm{MSE}}\left(\hat{\boldsymbol{\omega}}(\boldsymbol{X}); \boldsymbol{\omega}(\boldsymbol{X})\right) &= \frac{1}{R_1}\sum_{r=1}^{R_1}\frac{1}{n}\left(\hat{\boldsymbol{\omega}}(\boldsymbol{X})^{(r)} - \boldsymbol{\omega}(\boldsymbol{X})\right)'\left(\hat{\boldsymbol{\omega}}(\boldsymbol{X})^{(r)} - \boldsymbol{\omega}(\boldsymbol{X})\right) \\ &= \frac{1}{R_1}\sum_{r=1}^{R_1}\mathrm{MSE}(\hat{\boldsymbol{\omega}}(\boldsymbol{X})^{(r)}; \boldsymbol{\omega}(\boldsymbol{X})). \end{aligned} \tag{5.27}$$

Let $\left\{\boldsymbol{X}^{(j)}\right\}, j = 1, 2, \ldots, R_2$, be a random sample of predictor matrices drawn from $F_{\boldsymbol{X}}$. Then, a MC estimator of (5.26) is given by

[122] This is made a function of $\boldsymbol{X}$, not only of $\boldsymbol{X}'_{i\cdot}$, because the estimators being considered here depend on all rows of $\boldsymbol{X}$ through the annihilator matrix $\boldsymbol{M} = \boldsymbol{I}_n - \boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'$.

[123] This claim can be argued in more detail as follows. Let $\boldsymbol{X}^{(1)}$ be a random $n \times p$ matrix drawn from $F_{\boldsymbol{X}}$ whose rows are n mutually independent random p-vectors $\boldsymbol{X}^{(1)\prime}_{1\cdot}, \boldsymbol{X}^{(1)\prime}_{2\cdot}, \ldots, \boldsymbol{X}^{(1)\prime}_{n\cdot}$. These n random vectors can be thought of as random draws from a p-variate probability distribution. Now, let $\boldsymbol{X}^{(2)}$ be another random $n \times p$ matrix drawn from $F_{\boldsymbol{X}}$, independently of $\boldsymbol{X}^{(1)}$, whose rows consist of n mutually independent random p-vectors $\boldsymbol{X}^{(2)\prime}_{1\cdot}, \boldsymbol{X}^{(2)\prime}_{2\cdot}, \ldots, \boldsymbol{X}^{(2)\prime}_{n\cdot}$. It follows that $\boldsymbol{X}^{(1)\prime}_{1\cdot}, \boldsymbol{X}^{(1)\prime}_{2\cdot}, \ldots, \boldsymbol{X}^{(1)\prime}_{n\cdot}, \boldsymbol{X}^{(2)\prime}_{1\cdot}, \boldsymbol{X}^{(2)\prime}_{2\cdot}, \ldots, \boldsymbol{X}^{(2)\prime}_{n\cdot}$ are all independent and identically distributed random p-vectors. Consequently, the index i on $\boldsymbol{X}^{(1)\prime}_{i\cdot}$ has no bearing on its distribution, and $\boldsymbol{X}^{(1)\prime}_{i\cdot}$ and $\boldsymbol{X}^{(2)\prime}_{i\cdot}$ are no more related than $\boldsymbol{X}^{(1)\prime}_{i\cdot}$ and $\boldsymbol{X}^{(2)\prime}_{j\cdot}$, $i \neq j$.

$$\widehat{\text{MSE}}\left(\hat{\boldsymbol{\omega}}(\cdot);\boldsymbol{\omega}(\cdot),F_{\boldsymbol{X}}\right)=\frac{1}{R_2}\sum_{j=1}^{R_2}\widehat{\text{MSE}}\left(\hat{\boldsymbol{\omega}}(\boldsymbol{X}^{(j)});\boldsymbol{\omega}(\boldsymbol{X}^{(j)})\right). \tag{5.28}$$

Also of interest is the variability of the conditional MSE with respect to the covariate matrix $\boldsymbol{X}$. Consider

$$\begin{aligned}&\text{Var}_{\boldsymbol{X}}\left\{\text{E}_{\boldsymbol{\epsilon}|\boldsymbol{X}}\left[\text{MSE}(\hat{\boldsymbol{\omega}}(\boldsymbol{X});\boldsymbol{\omega}(\boldsymbol{X}))\right]\right\}\\&\quad=\text{E}_{\boldsymbol{X}}\left\{\left(\text{E}_{\boldsymbol{\epsilon}|\boldsymbol{X}}\left[\text{MSE}(\hat{\boldsymbol{\omega}}(\boldsymbol{X});\boldsymbol{\omega}(\boldsymbol{X}))\right]-\text{E}_{\boldsymbol{X}}\left\{\text{E}_{\boldsymbol{\epsilon}|\boldsymbol{X}}\left[\text{MSE}(\hat{\boldsymbol{\omega}}(\boldsymbol{X});\boldsymbol{\omega}(\boldsymbol{X}))\right]\right\}\right)^2\right\}.\end{aligned} \tag{5.29}$$

A MC estimator of the square root of (5.29), the 'between-designs' standard error of the MSE estimate, is given by

$$\sqrt{\frac{1}{R_2-1}\sum_{j=1}^{R_2}\left(\widehat{\text{MSE}}(\hat{\boldsymbol{\omega}}(\boldsymbol{X}^{(j)});\boldsymbol{\omega}(\boldsymbol{X}^{(j)}))-\widehat{\text{MSE}}(\hat{\boldsymbol{\omega}}(\cdot),\boldsymbol{\omega}(\cdot);F_{\boldsymbol{X}})\right)^2}. \tag{5.30}$$

If (5.30) is small, this implies that (5.28) is stable relative to the choice of predictor matrix $\boldsymbol{X}$ from $F_{\boldsymbol{X}}$. Of course, this raises the question of how small is 'small'. An alternative approach is to use an ANOVA sum-of-squares decomposition approach. The total sum of squared errors between the individual MSE estimates $\widehat{\text{MSE}}(\hat{\boldsymbol{\omega}}(\boldsymbol{X}^{(j)})^{(r)};\boldsymbol{\omega}(\boldsymbol{X}^{(j)}))$, from the rth replication and the jth design, and the overall estimator of (5.26), $\widehat{\text{MSE}}(\hat{\boldsymbol{\omega}}(\cdot);\boldsymbol{\omega}(\cdot),F_{\boldsymbol{X}})$, is

$$\begin{aligned}&\sum_{j=1}^{R_2}\sum_{r=1}^{R_1}\left[\widehat{\text{MSE}}(\hat{\boldsymbol{\omega}}(\boldsymbol{X}^{(j)})^{(r)};\boldsymbol{\omega}(\boldsymbol{X}^{(j)}))-\widehat{\text{MSE}}(\hat{\boldsymbol{\omega}}(\cdot);\boldsymbol{\omega}(\cdot),F_{\boldsymbol{X}})\right]^2\\&=R_1\sum_{j=1}^{R_2}\left[\widehat{\text{MSE}}(\hat{\boldsymbol{\omega}}(\boldsymbol{X}^{(j)});\boldsymbol{\omega}(\boldsymbol{X}^{(j)}))-\widehat{\text{MSE}}(\hat{\boldsymbol{\omega}}(\cdot);\boldsymbol{\omega}(\cdot),F_{\boldsymbol{X}})\right]^2\\&+\sum_{j=1}^{R_2}\sum_{r=1}^{R_1}\left[\widehat{\text{MSE}}(\hat{\boldsymbol{\omega}}(\boldsymbol{X}^{(j)})^{(r)};\boldsymbol{\omega}(\boldsymbol{X}^{(j)}))-\widehat{\text{MSE}}(\hat{\boldsymbol{\omega}}(\boldsymbol{X}^{(j)});\boldsymbol{\omega}(\boldsymbol{X}^{(j)}))\right]^2.\end{aligned} \tag{5.31}$$

The first term on the right side of (5.31) represents the 'between-designs' sum of squares, while the second term represents the 'within-designs' sum of squares. Dividing the 'between-designs' sum of squares by the total sum of squares, one obtains a kind of 'coefficient of determination' statistic:

$$\frac{R_1\displaystyle\sum_{j=1}^{R_2}\left[\widehat{\text{MSE}}(\hat{\boldsymbol{\omega}}(\boldsymbol{X}^{(j)});\boldsymbol{\omega}(\boldsymbol{X}^{(j)}))-\widehat{\text{MSE}}(\hat{\boldsymbol{\omega}}(\cdot);\boldsymbol{\omega}(\cdot),F_{\boldsymbol{X}})\right]^2}{\displaystyle\sum_{j=1}^{R_2}\sum_{r=1}^{R_1}\left[\widehat{\text{MSE}}(\hat{\boldsymbol{\omega}}(\boldsymbol{X}^{(j)})^{(r)};\boldsymbol{\omega}(\boldsymbol{X}^{(j)}))-\widehat{\text{MSE}}(\hat{\boldsymbol{\omega}}(\cdot);\boldsymbol{\omega}(\cdot),F_{\boldsymbol{X}})\right]^2}. \tag{5.32}$$

Clearly, the 'coefficient of determination' quantity in (5.32) falls within the interval $[0,1]$. If it is close to 0, this suggests that variation in the design matrices is a negligible source of variation in MSE estimates relative to the variation attributable to the randomness of $\boldsymbol{\epsilon}$.

To evaluate the stability of the ALVMs with respect to randomness in the design matrix, a MC simulation was designed as follows. $R_2=20$ different design matrices were generated, with $n=100$ and $p=3$. Both the multiplicative heteroskedastic function (5.24) and the additive heteroskedastic function (5.23) were used, each with DGPs $\mathcal{H}=\{2\}$ and $\mathcal{H}=\{2,3\}$, using the notation introduced in §5.4.1. The fifth DGP was homoskedastic ($\mathcal{H}=\emptyset$). Under each design scenario, $R_1=500$ MC replications of the errors $\boldsymbol{\epsilon}$ were generated, and six auxiliary variance models were fit: the homoskedastic model ($\hat{\omega}_{\text{ub}}=(n-p)^{-1}\boldsymbol{e}'\boldsymbol{e}$), and the 'basic', cluster, linear, penalised polynomial (with L_2 norm penalty), and thin-plate spline ALVMs.[124] The errors $\boldsymbol{\epsilon}$ were generated separately for each model within each MC replication to ensure independence of the model results.

[124] The penalised polynomial model with L_1 norm was omitted from this simulation due to its computational expense.

For the cluster and linear models, variable selection was performed using the QGCV-linear technique; variable selection in the polynomial and thin-plate models was left to the shrinkage penalty.

Table 5.30 shows the coefficient of determination metric (5.32), based on both the unstandardised and standardised MSEs, for all six models and all five DGPs in the simulation. The figures in this table are rounded to four significant digits.

Table 5.30: Coefficient of Determination Metrics (5.32) for Two-Covariate Model

		Homoskedasticity	Multiplicative Heteroskedasticity		Additive Heteroskedasticity	
Metric Type	Model	DGP $\mathcal{H} = \emptyset$	DGP $\mathcal{H} = \{2\}$	DGP $\mathcal{H} = \{2,3\}$	DGP $\mathcal{H} = \{2\}$	DGP $\mathcal{H} = \{2,3\}$
Uns.	Homoskedastic	0.001140	0.753800	0.955100	0.4498000	0.326400
	Basic ALVM	0.001668	0.035280	0.031310	0.0386800	0.023290
	Clustering ALVM	0.003249	0.018240	0.085400	0.0206500	0.016340
	Linear ALVM	0.002682	0.029440	0.576700	0.0032570	0.012810
	L_2-Norm Pen. Poly. ALVM	0.001804	0.005962	0.048860	0.0060300	0.004604
	Thin-Plate Spline ALVM	0.002770	0.003761	0.007064	0.0069650	0.006061
Std.	Homoskedastic	0.001140	0.104000	0.238700	0.1294000	0.369400
	Basic ALVM	0.001668	0.001873	0.009770	0.0009995	0.001250
	Clustering ALVM	0.003249	0.005099	0.075580	0.0039560	0.033110
	Linear ALVM	0.002682	0.019820	0.097260	0.0060520	0.017080
	L_2-Norm Pen. Poly. ALVM	0.001804	0.004279	0.034680	0.0101900	0.014880
	Thin-Plate Spline ALVM	0.002770	0.008971	0.015890	0.0059570	0.008320

It is apparent that in the homoskedastic DGP, the amount of variation in the MC MSEs that is explained by the variation in design matrices is negligible. The same is true for most of the heteroskedastic DGPs for the ALVMs, though not for the homoskedastic estimator.[125] The only ALVM where the coefficient of determination metric is large enough to be of concern is the linear ALVM under multiplicative heteroskedasticity linked to both covariates ($\mathcal{H} = \{2,3\}$). Here, more than half (57.7%) of the variation in MC MSEs is due to variation in the design matrix. This suggests that the performance of the linear ALVM may be sensitive to the specific form of the design matrix $\boldsymbol{X}$ if there is heteroskedasticity of extreme magnitude. The other ALVMs, however, seem to be stable with respect to the form of the design matrix $\boldsymbol{X}$, even in the presence of fairly extreme heteroskedasticity. Admittedly, this has only been shown empirically in a narrow set of circumstances.

5.4.3 Convergence Rates of Gauss-Newton Algorithm for Fitting Auxiliary Nonlinear Variance Models

Table 5.31 indicates the convergence rates for the Gauss-Newton algorithm used to numerically solve (3.69) for MQL estimation of the ANLVM parameters in the simulations reported on in §5.3.1.5, §5.3.2.1, and §5.3.3.1. The feature selection procedure entailed using the homoskedastic variance estimator $\hat{\omega}_{\text{ub}}$ in cases where no features were selected for inclusion in the auxiliary design matrix $\boldsymbol{Z}$. Consequently, no MQL estimation was required in such cases, so the convergence rates are computed only over those replications where the Gauss-Newton algorithm was actually run. Naturally, MQL estimation was required in relatively few replications under the homoskedastic DGP.

The settings used for the Gauss-Newton algorithm, per the arguments of the `anlvm.fit` function in the **skedastic** package (discussed in §4.4.2), entailed up to 20 initial values of the parameter vector $\boldsymbol{\gamma}$ and a maximum of 100 iterations of the updating equation. The nested updating equation (3.72) was not used due to computation time, but may have resulted in higher convergence rates and/or more accurate or precise parameter estimation.

'Multiple Covariates' in Table 5.31 refers to a DGP with heteroskedasticity where the error variances are related to multiple covariates. This corresponds to $\mathcal{H} = \{2,3\}$ in the case of $p-1=2$ covariates and $\mathcal{H} = \{2,3,4,5\}$ in the case of $p-1=8$ covariates. It is not applicable in the case of one covariate.

[125] The high values of the coefficient of determination metric (5.32) for the homoskedastic estimator under heteroskedasticity are surprising, since the error variances $\boldsymbol{\omega} = \mathbf{1}$ are independent of the design matrix in this case. However, the estimators are still affected by changes in $\boldsymbol{M}$ (a function of $\boldsymbol{X}$), while the variation explained by changes in $\boldsymbol{\epsilon}$ tends to be miniscule when $\boldsymbol{\epsilon}$ is iid across replications.

The convergence rates are very high—above 96%—under all DGPs except for multiplicative heteroskedasticity linked to multiple covariates. In this scenario, the convergence rates are still above 98% for the exponential ANLVM (presumably because the heteroskedastic function is correctly specified) but much lower for the quadratic and clustering ANLVMs.

Table 5.31: Convergence Rates of Gauss-Newton Algorithm for Fitting ANLVMs

		Homosked.	Additive Het.		Multiplicative Het.	
No. of Covariates $(p-1)$	ANLVM	$\mathcal{H}=\emptyset$	$\mathcal{H}=\{2\}$	Multiple Covariates	$\mathcal{H}=\{2\}$	Multiple Covariates
1	Quadratic	1.000	1.000		0.999	
	Exponential	0.998	0.991		0.995	
	Clustering	1.000	1.000		1.000	
2	Quadratic	0.997	0.998	0.966	0.995	0.465
	Exponential	0.996	0.991	0.996	0.995	0.999
	Clustering	1.000	0.999	0.988	0.999	0.846
8	Quadratic	0.991	0.988	0.973	0.984	0.757
	Exponential	0.994	0.989	0.991	0.992	0.981
	Clustering	0.990	0.984	0.979	0.983	0.259

If the practitioner experiences nonconvergence of the ANLVM with a particular application of linear regression, it is suggested that the ceiling be raised on the number of iterations allowed, and/or that a broader grid of initial parameter values be searched. Nonconvergence may also provide a hint that the heteroskedastic function $g(\cdot)$ has been misspecified.

5.5 Coverage Probabilities of Confidence Intervals

A MC simulation experiment was conducted to obtain empirical estimates of the coverage probabilities of the bootstrap confidence interval estimates discussed in §3.4. The experimental factors considered are as outlined in Table 5.32.

Table 5.32: Settings for Monte Carlo Simulation Experiment to Evaluate Coverage Probabilities of Bootstrap CIs

Factor	Levels
Sample Size n	20
ALVM Method	Clustering; Polynomial with L_2-Norm Penalty
Bootstrap Resampling Method	Pairs; Wild $(f_i(e_i)=e_i/(1-h_{ii})^{1/2})$
Interval Method	Percentile; BCa; Normal
Expansion Adjustment	No; Yes

In every case, the number of MC replications was $R=10^3$, while the number of bootstrap samples drawn was $B=10^3$. The DGP had only one covariate, which was generated from $U(0,3)$, while errors were generated independently from a normal distribution with zero mean and variance $\omega_i=(1+x_i)^2$ (additive heteroskedasticity). Nominal confidence level was 0.95 in every instance.

Since preliminary simulations found that the independent intervals approach discussed in §3.4.3 yielded negligible improvements in coverage probability despite heavy computational cost, this approach was not included in the main experiment. Moreover, while ideally hyperparameters such as λ (in the penalised polynomial and thin-plate spline ALVMs) should be re-tuned when the model is fitted to each bootstrap regression, this would be computationally very expensive. Thus, the hyperparameter values selected from the full data set are also used with every bootstrap sample. The same is true of feature selection results.

The averaged-out coverage probability estimate is,

$$\hat{\pi}_{\text{cover}}=\frac{1}{nR}\sum_{i=1}^{n}\sum_{r=1}^{R}I\left(\hat{\omega}_{i,\text{lo}}^{(r)}\leq\omega_i\leq\hat{\omega}_{i,\text{up}}^{(r)}\right), \tag{5.33}$$

where $\hat{\omega}_{i,\text{lo}}^{(r)}$ and $\hat{\omega}_{i,\text{up}}^{(r)}$ are the lower and upper bootstrap confidence limits for the rth Monte Carlo replication.

Table 5.33 shows coverage probabilities for three types of bootstrap confidence intervals computed from the clustering ALVM for a DGP with a sample size of $n = 20$, $p = 2$, and additive (quadratic) heteroskedasticity.

Table 5.33: Estimated Averaged-Out Coverage Probabilities of Bootstrap Confidence Intervals for ω_i, Clustering ALVM

Bootstrap	Interval Type	Expanded	$\hat{\pi}_{\text{cover}}$	SE($\hat{\pi}_{\text{cover}}$)
Pairs	Percentile	No	0.927	0.00400
	Percentile	Yes	0.941	0.00366
	BCa	No	0.804	0.00529
	BCa	Yes	0.844	0.00489
	Normal	No	0.917	0.00428
	Normal	Yes	0.932	0.00404
Wild-HC2	Percentile	No	0.550	0.00743
	Percentile	Yes	0.588	0.00747
	BCa	No	0.474	0.00746
	BCa	Yes	0.498	0.00748
	Normal	No	0.483	0.00708
	Normal	Yes	0.524	0.00711

While the pairs bootstrap coverage probabilities are reasonably good, the wild bootstrap with coverage probabilities are so low as to make the intervals useless. This is surprising, since the same DGP was used to compare the pairs bootstrap and the wild bootstrap for the HCCME described in §2.3.10. Both bootstrap methods performed well, and about equally so, for estimating the standard errors of the elements of $\hat{\beta}$ (results not shown).

Table 5.34 and Table 5.35 show the coverage probabilities for the same intervals computed from the same DGP, but with the linear and polynomial (L_2-norm) ALVMs, respectively. The results of the clustering and linear ALVMs are similar, while coverage probabilities are poorer for the polynomial ALVM.

Table 5.34: Estimated Averaged-Out Coverage Probabilities of Bootstrap Confidence Intervals for ω_i, Linear ALVM

Bootstrap	Interval Type	Expanded	$\hat{\pi}_{\text{cover}}$	SE($\hat{\pi}_{\text{cover}}$)
Pairs	Percentile	No	0.851	0.00533
	Percentile	Yes	0.876	0.00498
	BCa	No	0.824	0.00505
	BCa	Yes	0.855	0.00456
	Normal	No	0.900	0.00542
	Normal	Yes	0.911	0.00509
Wild-HC2	Percentile	No	0.528	0.01020
	Percentile	Yes	0.566	0.01010
	BCa	No	0.492	0.00925
	BCa	Yes	0.525	0.00940
	Normal	No	0.502	0.01010
	Normal	Yes	0.540	0.01020

Table 5.35: Estimated Averaged-Out Coverage Probabilities of Bootstrap Confidence Intervals for ω_i, Polynomial (L_2) ALVM

Bootstrap	Interval Type	Expanded	$\hat{\pi}_{\text{cover}}$	SE($\hat{\pi}_{\text{cover}}$)
Pairs	Percentile	No	0.754	0.00664
	Percentile	Yes	0.790	0.00630
	BCa	No	0.738	0.00698
	BCa	Yes	0.765	0.00680
	Normal	No	0.810	0.00722
	Normal	Yes	0.834	0.00665
Wild-HC2	Percentile	No	0.387	0.00919
	Percentile	Yes	0.418	0.00933
	BCa	No	0.389	0.00897
	BCa	Yes	0.412	0.00902
	Normal	No	0.359	0.00865
	Normal	Yes	0.391	0.00881

Surprisingly, the naïve normal interval outperforms the percentile and BCa intervals in terms of average coverage probability for both the linear and polynomial models.

Figure 5.9 shows how the MC estimate of bootstrap coverage probability relates to error variance magnitude ω_i for the different bootstrap CI methods, using the same DGP and clustering ALVM used to generate Table 5.33.

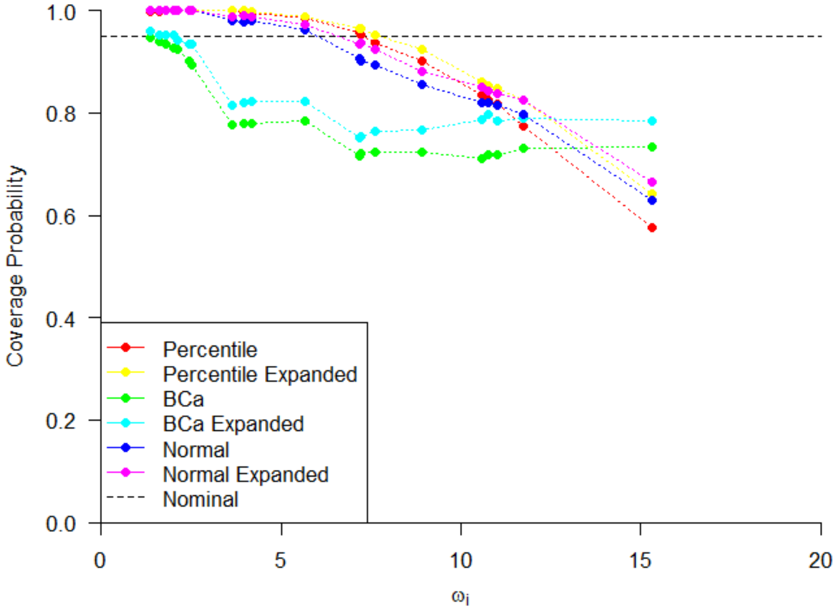


Figure 5.9: Estimated Bootstrap Confidence Interval Coverage Probability vs. Error Variance

It is evident that coverage probability tends to decline with error variance, and is therefore above the nominal confidence level for relatively small ω_i and falls far below the nominal confidence level for the largest ω_i. The BCa intervals tend to be consistently below the nominal level, and do not seem to be successful in improving on the coverage probability of the percentile intervals, as they are designed to do. Indeed, the BCa intervals' coverage probabilities are worse than those of the naïve normal bootstrap intervals. The expansion technique does, however, appear to be successful in improving coverage probabilities for all three interval methods.

Future research could exploit the apparent relationship between variance magnitude and CI coverage probability by making adjustments to the confidence limits based on the relative magnitudes of the point estimates.

5.6 Illustrations Using Real-World Data Sets

5.6.1 Fuel Economy of Cars

The `mtcars` data set in the R **datasets** package was extracted from the 1974 *Motor Trend* magazine, and contains observations on fuel consumption (measured in miles per US gallon) and ten predictors. There are 32 observations. Consider a linear regression model fitted to this data using OLS with fuel consumption as the response and weight (in thousands of pounds) and quarter-mile time (measured in seconds) as predictors. The coefficients table is shown in Table 5.36.

Table 5.36: Coefficients Table for Linear Model Fitted to `mtcars` Data

	Estimate $\hat{\beta}_j$	$\widehat{\text{SE}}(\hat{\beta}_j)$	t Statistic	Significance p-Value
(Intercept)	19.7500	5.252	3.760	0.000765
qsec	0.9292	0.265	3.506	0.001500
wt	-5.0480	0.484	-10.430	0.000000

A plot of the squared OLS residuals against each of the covariates (generated using the `hetplot` function from **skedastic**, discussed in §4.2.3) is shown in Figure 5.10. From the left panel, it appears that there may be heteroskedasticity linked to the `qsec` variable, since the e_i^2 are much more spread out for cars with a quarter-mile time of around 20 seconds than for those with lower quarter-mile times. Possibly, the error variance increases with `qsec`. In the right panel, if the point at upper right were ignored, it would appear as though the spread in the e_i^2 decreases with car weight. However, the point at upper right contradicts this pattern, and could suggest a quadratic relationship between `wt` and error variance, or may just be an outlier. With such a small sample size, there is a high risk of 'detecting' spurious patterns from a graph.

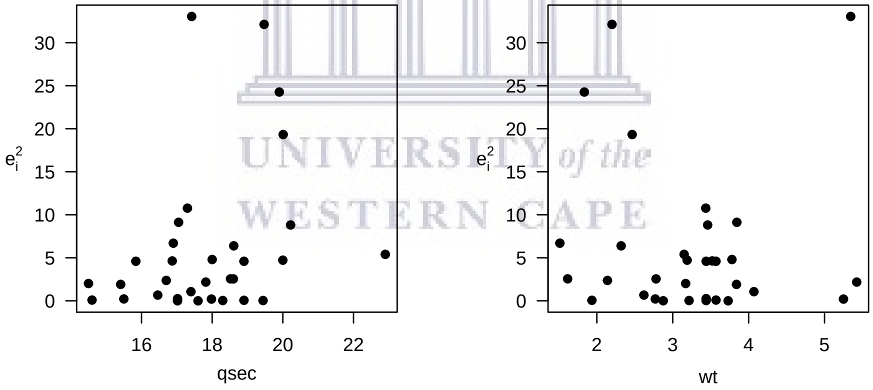


Figure 5.10: Heteroskedasticity Plots for the `wt` and `qsec` Variables in the `mtcars` Linear Model

For interest's sake, one can conduct heteroskedasticity tests on the model. Firstly, consider the three omnibus tests with the best AEPS for $n \approx 30$, according to Figure 5.3. The p-values for these three tests are shown in Table 5.37, and they unanimously find an absence of evidence for heteroskedasticity at the 5% significance level.

Table 5.37: Omnibus Heteroskedasticity Tests Run on `mtcars` Linear Model

Heteroskedasticity Test	p-Value
Glejser (1969)	0.126
Cook and Weisberg (1983)	0.209
Verbyla (1993)	0.164

Secondly, consider the deflator-based tests with the best AEPS for $n \approx 30$, according to Figure 5.2. The p-values for these four tests are shown in Table 5.38, when using both car weight and quarter-mile time as the deflator. Here, none of the tests detect heteroskedasticity linked to car weight at a 5% significance level, but two of the four detect heteroskedasticity linked to quarter-mile time. A two-tailed test has been used in the tests of Honda (1989) and Carapeto and Holt (2003) (which allow for this),[126] with two-sided p-values computed using the method of Kulinskaya (2008) (see §4.2.2). This partly explains why these two tests yielded higher p-values with `qsec` as the deflator.

```
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
```

[126] If one adopted a directional alternative solely on the basis of the apparent pattern in Figure 5.10, and not on any *a priori* theoretical grounds, this would increase the power but also the size of the test.

```
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
```

```
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
```

```
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
```

Table 5.38: Deflator-Based Heteroskedasticity Tests Run on `mtcars` Linear Model

Heteroskedasticity Test	p-Value (`qsec`)	p-Value (`wt`)
Evans and King's (1988) GLS test	0.00967	0.686
Honda (1989)	0.05380	0.838
Szroeter (1978)	0.02340	0.774
Carapeto and Holt (2003)	0.18500	0.631

With some evidence in hand that there is heteroskedasticity in the model, an ALVM can be fitted. Feature selection was performed using QGCV on the linear ALVM, CV on the linear ALVM, and heteroskedasticity testing using Evans and King's (1988) GLS test. In each case, the `qsec` variable was selected for inclusion in the ALVM while the `wt` variable was not.

For the clustering ALVM, the number of clusters n_c was chosen using five-fold CV and using the elbow method with SWD criterion; both methods arrived at $n_c = 6$. The L_2-norm penalised polynomial ALVM did not shrink any of the degree-two polynomial coefficients to zero, whereas the L_1-norm penalised polynomial ALVM shrank all of the degree-two polynomial coefficients to zero except for the intercept and the linear `qsec` term. The variance estimates $\hat{\omega}_i$ are plotted against the corresponding quarter-mile times in Figure 5.11. With the exception of the homoskedastic case, all of the models reflect an increasing trend of the error variances with `qsec`.

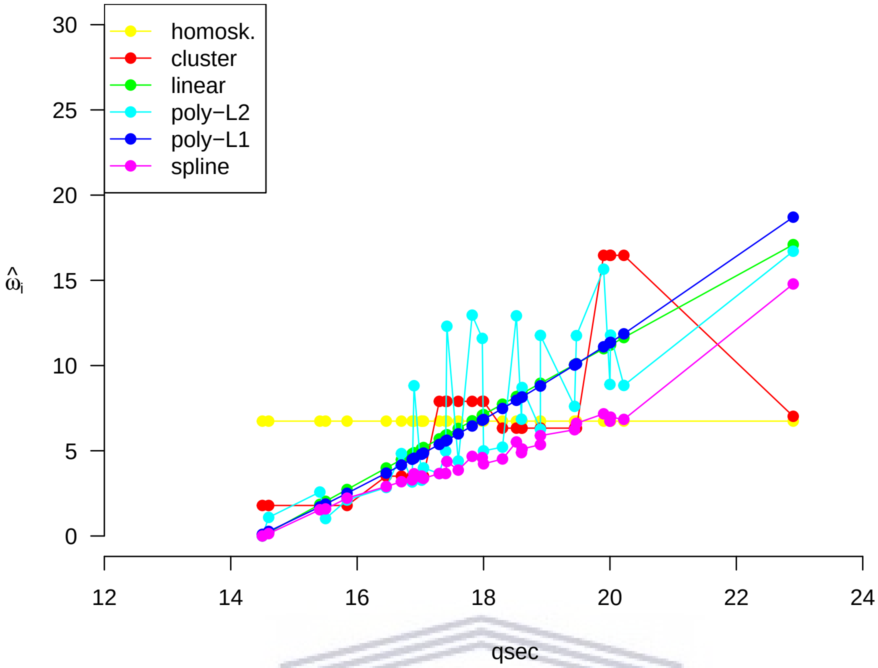


Figure 5.11: Variance Estimates for `mtcars` Linear Model vs. `qsec` Values

The variance estimates can be used to compute FWLS estimates of $\boldsymbol{\beta}$ and perform quasi-t-tests of significance for the elements of $\boldsymbol{\beta}$. The results are shown in Table 5.39. The changes in the coefficient estimates are noteworthy. All of the ALVM-based FWLS estimates of the intercept decreased compared to the OLS estimate. All of the ALVM-based FWLS estimates of the `qsec` partial slope coefficient increased in magnitude relative to the OLS estimate. As for inferences on the coefficients, the decisions about significance of the coefficients do not change, but the fact that the quasi-t-test significance p-value for the `qsec` coefficient has more than doubled in some models relative to the classical t-test p-values (Table 5.36) illustrates that the robust approach can make a difference.

Table 5.39: FWLS Coefficient Estimates (Quasi-t-Test Significance p-Values) for `mtcars` Linear Model by ALVM

	$\hat{\beta}_1$ (Intercept)	$\hat{\beta}_2$ (qsec)	$\hat{\beta}_3$ (wt)
Homoskedastic	19.7 (7.65×10^{-4})	0.929 (1.50×10^{-3})	-5.05 (2.52×10^{-11})
Clustering	17.3 (2.85×10^{-4})	1.1 (1.12×10^{-3})	-5.23 (4.89×10^{-11})
Linear	10.8 (1.05×10^{-3})	1.35 (3.31×10^{-3})	-4.57 (3.03×10^{-11})
L_2-Norm Pen. Poly	13.7 (1.22×10^{-3})	1.24 (3.00×10^{-3})	-5.02 (3.64×10^{-9})
L_1-Norm Pen. Poly	12.7 (1.26×10^{-3})	1.27 (4.03×10^{-3})	-4.75 (2.01×10^{-11})
Thin-Plate Spline	10.9 (2.65×10^{-4})	1.33 (1.02×10^{-3})	-4.55 (3.11×10^{-13})

5.6.2 Per Capita Expenditure on Public Schools

The **sandwich** package in R (Zeileis and Hothorn 2002, Zeileis 2004) contains a data set called `PublicSchools` that is also discussed as an example of heteroskedasticity in the package vignette. This data set is taken from United States Department of Commerce (1979) and is also discussed in Greene (2012) and in Cribari-Neto (2004). The data contains $n = 50$ observations of two variables, namely the per capita income in each US state (in 1978 US dollars) and the per capita expenditure on education, in US dollars. Following Zeileis and Hothorn (2002), the income variable is transformed into units of 10 000s of 1978 US dollars. A scatter plot of expenditure vs. income is shown in Figure 5.12. The outlying point at upper right represents the state of Alaska.

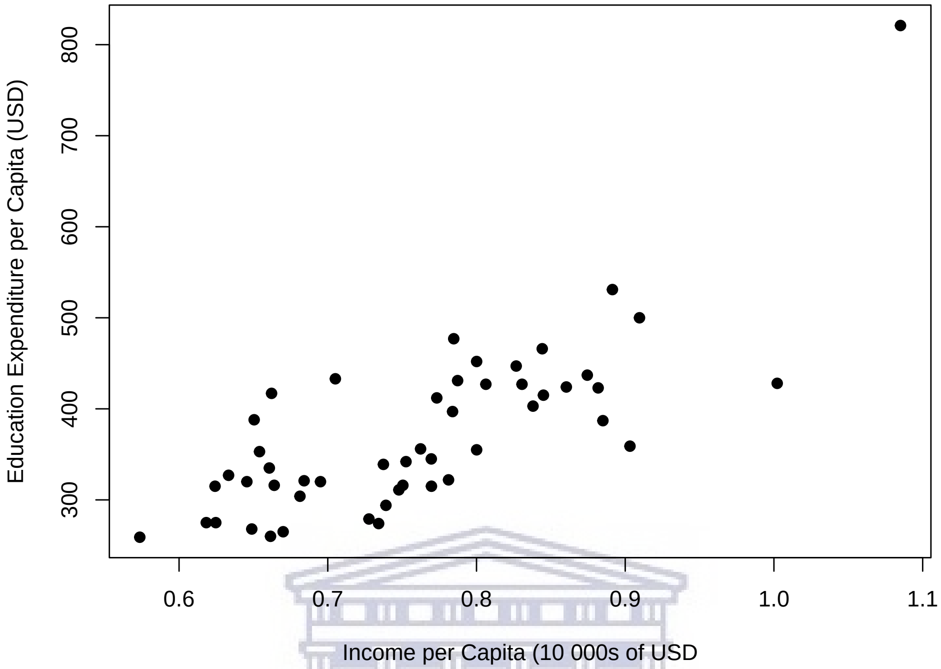


Figure 5.12: Scatter Plot of Per Capita Education Expenditure vs. Per Capita Income in US States, 1978

Table 5.40 gives the coefficients table for the linear regression model fitted to the Public Schools dataset.

Table 5.40: Coefficients Table for Linear Model Fitted to `PublicSchools` Data

	Estimate $\hat{\beta}_j$	$\widehat{\mathrm{SE}}(\hat{\beta}_j)$	t Statistic	Significance p-Value
(Intercept)	832.9	327.3	2.545	0.014280
Income	-1834.0	829.0	-2.213	0.031820
Income.Sq	1587.0	519.1	3.057	0.003677

A plot of the squared OLS residuals against the covariate (generated using the `hetplot` function from **skedastic**, discussed in §4.2.3) is shown in Figure 5.13. The plot does not provide compelling evidence for heteroskedasticity linked to income, as there are some large e_i^2 at various income levels.

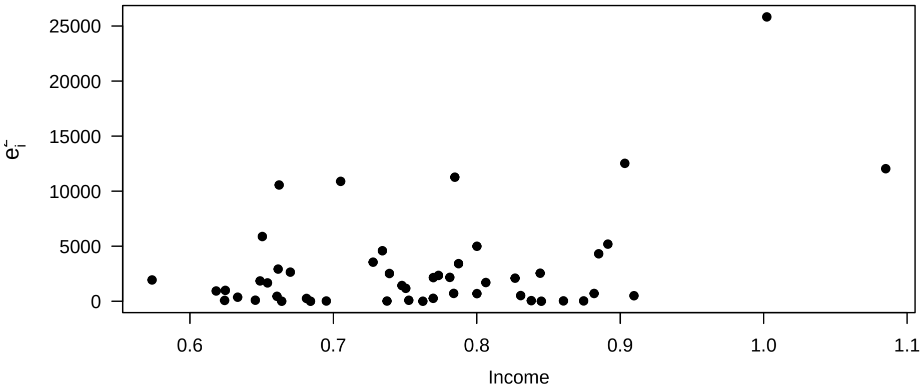


Figure 5.13: Heteroskedasticity Plot for the `Income` Variable in the `PublicSchools` Linear Model

For interest's sake, one can conduct heteroskedasticity tests on the model. As there is only one covariate in this case, there is no need to distinguish between omnibus and deflator tests. Table 5.41 shows results from a few heteroskedasticity tests. With the exception of Carapeto and Holt's (2003) test, all the tests agree at the 5% significance level that there is heteroskedasticity linked to income.

```
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
```

```
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
```

```
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
```

```
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
```

```
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
## Warning in do.call(‘::‘, args = list("CompQuadForm", algorithm))(q = 0, :  Note that Qq +
abserr is positive.
```

Table 5.41: Heteroskedasticity Tests Run on `PublicSchools` Linear Model

Heteroskedasticity Test	p-Value
Glejser (1969)	2.54×10^{-3}
Cook and Weisberg (1983)	7.86×10^{-5}
Verbyla (1993)	2.88×10^{-10}
Evans and King’s (1988) GLS test	2.24×10^{-2}
Honda (1989)	8.40×10^{-4}
Szroeter (1978)	1.83×10^{-2}
Carapeto and Holt (2003)	4.45×10^{-1}

With some evidence in hand that there is heteroskedasticity in the model, an ALVM can now be fitted. For the clustering ALVM, the number of clusters n_c was chosen to be $n_c = 5$ using the elbow method with SWD criterion. The L_2-norm penalised polynomial ALVM shrank the $\hat{\beta}_j$ coefficient estimates of both the linear and quadratic terms close to zero (0.130 and 0.232, respectively). The L_1-norm penalised polynomial ALVM did not experience much shrinkage and had far larger coefficients for these two terms (-2.92×10^5 and 2.07×10^5, respectively). The variance estimates $\hat{\omega}_i$ are plotted against the corresponding income values in Figure 5.14.

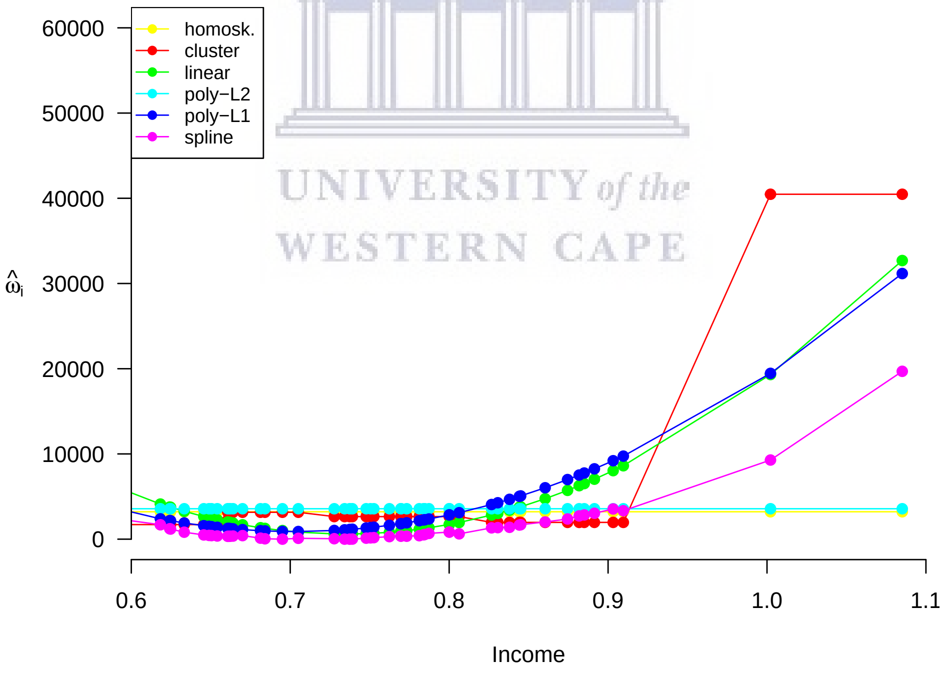


Figure 5.14: Variance Estimates for `PublicSchools` Linear Model vs. `Income` Values

With the exception of the homoskedastic case and the L_2-norm penalised polynomial (which is nearly homoskedastic), all of the model fits reflect a general increasing trend of the error variances with income. The dotted lines represent 95% confidence limits for the ω_i, computed using the naïve standard normal bootstrap method. Due to the small sample size of $n = 50$, the confidence bands are fairly wide.[127]

The variance estimates were used to compute FWLS estimates of $\boldsymbol{\beta}$ and perform quasi-t-tests of significance for the elements of $\boldsymbol{\beta}$. The results are shown in Table 5.42. The L_2-norm penalised polynomial results are nearly identical to the homoskedastic results, due to the aforementioned shrinkage of coefficients. The spline ALVM fit had many variance estimates on the constraint boundary $0^+ = 10^{-10}$; the resulting massive weights meant that the WLS routine returned an `NA` value for the coefficient estimate of the income squared term. Besides these two anomalies, the other three ALVMs (clustering, linear, and L_1-norm penalised polynomial) agree with the homoskedastic model in the signs of the coefficient estimates, while the magnitudes have changed to varying degrees.

Table 5.42: FWLS Coefficient Estimates (Quasi-t-Test Significance p-Values) for `PublicSchools` Linear Model by ALVM

	$\hat{\beta}_1$ (Intercept)	$\hat{\beta}_2$ (Income)	$\hat{\beta}_3$ (Income Squared)
Homoskedastic	833 (1.43×10^{-2})	-1830 (3.18×10^{-2})	1590 (3.68×10^{-3})
Clustering	374 (2.69×10^{-1})	-602 (3.69×10^{-1})	775 (2.48×10^{-1})
Linear	1100 (2.30×10^{-1})	-2580 (3.22×10^{-1})	2100 (1.97×10^{-1})
L_2-Norm Pen. Poly.	833 (1.97×10^{-2})	-1830 (4.12×10^{-2})	1590 (5.64×10^{-3})
L_1-Norm Pen. Poly.	897 (2.14×10^{-1})	-1990 (3.07×10^{-1})	1680 (1.85×10^{-1})
Thin-Plate Spline	7.78 (1.09×10^{-1})	449 (1.88×10^{-1})	NA (8.92×10^{-2})

Looking at the quasi-t-test significance p-values, an interesting phenomenon is observed: whereas all three β_j are statistically significant at the 5% level according to the classical t-test, the heteroskedasticity-robust quasi-t-tests—with the exception of the L_2-norm penalised polynomial, as already discussed—all agree that *none* of the three coefficients are statistically significant at the 5% level. In this respect, the quasi-t-tests agree with the finding of Zeileis and Hothorn (2002), who ran quasi-t-tests on this model after using the HC4 HCCME to estimate the error variances, and likewise found that the classical t-test had spuriously found the model coefficients to be significant. Zeileis and Hothorn (2002) argued that this was due to the classical homoskedastic t-test placing too much weight on the outlying Alaska observation, which arguably has a large error variance: its leverage score h_{ii} is 0.65, whereas all other observations but one have a leverage score less than 0.1.

If the `PublicSchools` OLS model is fitted with only the linear income term and the quadratic term is dropped, the coefficient of the linear term is highly significant when tested either using the classical t-test or using a quasi-t-test based on an HCCME- or ALVM-based standard error estimate. This demonstrates that using an ALVM has enabled the practitioner to avoid including a probably spurious quadratic term in the linear regression model.

5.6.3 Boston House Values

The Boston housing data set (Harrison and Rubinfeld 1978) contains data on 506 census tracts in Boston, USA from the 1970 census. It is available in R in the `BostonHousing2` object in the **mlbench** package (Leisch and Dimitriadou 2010). The 14 variables of interest are shown in Table 5.43. `cmedv` is the response variable and the other 13 variables are features (12 numerical features and one categorical feature). This data set has been repeatedly used as an empirical example in statistical methodological work, including concerning heteroskedasticity (Gilley and Pace 1996, Radchenko and James 2011, Cho and Fryzlewicz 2012, Cheng 2012, Simlai 2014, Miller and Startz 2019, e.g.,); indeed, it has been called a ‘popular proving ground for machine learning’ (Chen 2021, p. 2).

[127] Confidence limits for the spline model are not shown, as R would repeatedly freeze when running the spline ALVM on the bootstrap linear models.

Table 5.43: Description of Variables in Boston Housing Data Set

Variable Name	Variable Description
`cmedv`	corrected median value of owner-occupied homes in USD 1000s
`crim`	per capita crime rate by town
`zn`	proportion of residential land zoned for lots over 25 000 sq feet
`indus`	proportion of non-retail business acres per town
`chas`	Charles River (=1 if tract bounds river; 0 otherwise)
`nox`	nitric oxides concentration (parts per 10 million)
`rm`	average number of rooms per dwelling
`age`	proportion of owner-occupied units built prior to 1940
`dis`	weighted distances to five Boston employment centres
`rad`	index of accessibility to radial highways
`tax`	full-value property-tax rate per USD 10 000
`ptratio`	pupil-teacher ratio by town
`b`	$1000(B - 0.63)^2$ where B is the proportion of black residents per town
`lstat`	percentage of lower status of the population

Results of a linear regression model fitted to the data by OLS are shown in Table 5.44. The classical t-tests find eleven of the thirteen predictors to be statistically significant at 5% level; the exceptions are **`indus`** and **`age`**.

Table 5.44: Coefficients Table for Linear Model Fitted to `BostonHousing2` Data

	Estimate $\hat{\beta}_j$	$\widehat{\text{SE}}(\hat{\beta}_j)$	t Statistic	Significance p-Value
(Intercept)	3.64×10^{1}	5.058000	7.19100	2.40×10^{-12}
crim	-1.06×10^{-1}	0.032570	-3.26100	1.19×10^{-3}
zn	4.77×10^{-2}	0.013600	3.50800	4.93×10^{-4}
indus	2.32×10^{-2}	0.060940	0.38150	7.03×10^{-1}
chas1	2.69×10^{0}	0.853900	3.15200	1.72×10^{-3}
nox	-1.77×10^{1}	3.785000	-4.68700	3.59×10^{-6}
rm	3.79×10^{0}	0.414200	9.14900	1.52×10^{-18}
age	5.75×10^{-4}	0.013090	0.04392	9.65×10^{-1}
dis	-1.5×10^{0}	0.197700	-7.59800	1.53×10^{-13}
rad	3.04×10^{-1}	0.065750	4.62000	4.91×10^{-6}
tax	-1.27×10^{-2}	0.003727	-3.40900	7.06×10^{-4}
ptratio	-9.24×10^{-1}	0.129700	-7.12600	3.70×10^{-12}
b	9.23×10^{-3}	0.002662	3.46700	5.73×10^{-4}
lstat	-5.31×10^{-1}	0.050260	-10.56000	1.26×10^{-23}

A heteroskedasticity plot generated using **`hetplot`** (see §4.2.3) can be seen in Figure 5.15, which plots the squared BLUS residuals (denoted on the plot by $\tilde{e}_i^2$) against two of the covariates, **`rm`** and **`lstat`**. The reason for considering BLUS residuals rather than OLS residuals is that the BLUS residuals are mutually independent under Assumptions A1-A5, unlike the OLS residuals (see §1.1.7.5). Figure 5.15 suggests a nonmonotonic, U-shaped relationship between the `lstat` variable and the error variance.

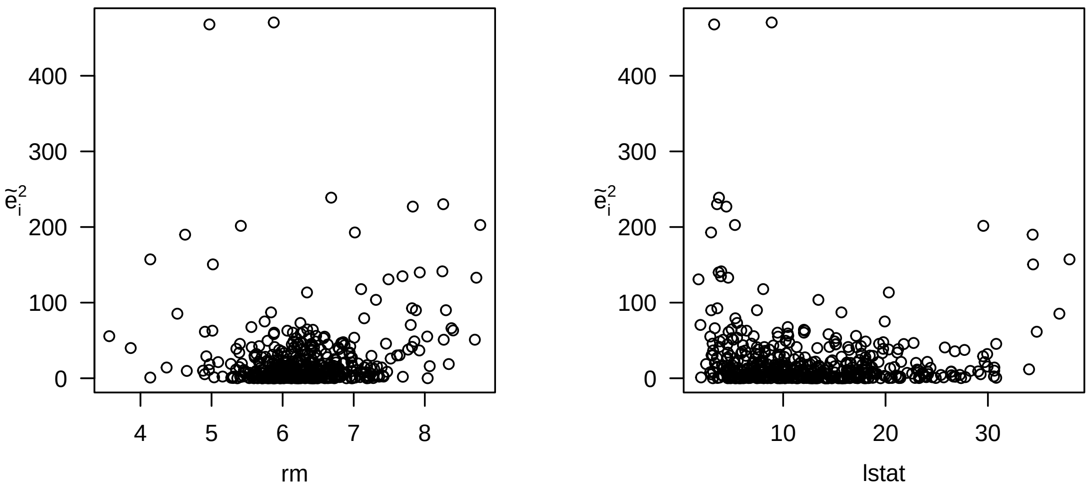


Figure 5.15: Plot of Squared BLUS Residuals vs. `rm` and `lstat` Explanatory Variables, Boston Housing Data

To perform heteroskedasticity diagnostics, some omnibus tests are first attempted. As Table 5.45 shows, the evidence for heteroskedasticity is overwhelming.

```
## Error in white_lm(bostonlm):  could not find function "white_lm"
## Error in signif(x, sigdig):  non-numeric argument to mathematical function
```

Table 5.45: Omnibus Heteroskedasticity Tests Run on `BostonHousing2` Linear Model

Heteroskedasticity Test	p-Value
Glejser (1969)	2.54×10^{-3}
Cook and Weisberg (1983)	7.86×10^{-5}
Verbyla (1993)	2.88×10^{-10}
Evans and King's (1988) GLS test	2.24×10^{-2}
Honda (1989)	8.40×10^{-4}
Szroeter (1978)	1.83×10^{-2}
Carapeto and Holt (2003)	4.45×10^{-1}

Deflator-based heteroskedasticity tests are next undertaken with each predictor considered in turn as the deflator, with the exception of the categorical `chas1` variable.[128] A Bonferroni correction results in a significance level of 0.05/12 being used. In addition to four heteroskedasticity tests from the literature (Goldfeld and Quandt 1965, Szroeter 1978, Evans and King 1988, Honda 1989), the ALVM-based test described in §3.5 is shown, based on the L_2-norm penalised polynomial model (to detect nonmonotonic heteroskedasticity such as that seen in Figure 5.15); in this case, the p-values were computed using the method of Godfrey and Orme (1999), described in §2.1.23.2. Results are shown in Table 5.46. At the 0.004167 significance level, the four tests do not unanimously agree on a finding of heteroskedasticity for most of the covariates; this is only the case for `rad`, `nox`, `age`, and `tax`. This inconsistency is probably due in large part to the tests' varying abilities to detect different heteroskedastic patterns (including nonmonotonic).

[128] A BAMSET test was run with the two values of `chas1` being the subsets. No heteroskedasticity linked to this variable was detected, with a p-value of 1.

Table 5.46: Deflator-Based Heteroskedasticity Tests Run on `BostonHousing2` Linear Model

Deflator	p-Values				
	Goldfeld and Quandt (1965)	Szroeter (1978)	Evans and King's (1988) GLS	Honda (1989)	ALVM Test
crim	5.06×10^{-14}	2.35×10^{-13}	4.07×10^{-11}	1.43×10^{-3}	2.60×10^{-3}
zn	4.04×10^{-2}	6.52×10^{-1}	8.93×10^{-4}	0	7.56×10^{-2}
indus	9.51×10^{-5}	9.65×10^{-4}	2.66×10^{-4}	2.42×10^{-14}	1.34×10^{-2}
nox	4.44×10^{-16}	4.42×10^{-10}	1.23×10^{-11}	7.77×10^{-8}	2.00×10^{-4}
rm	3.98×10^{-1}	6.54×10^{-2}	6.31×10^{-1}	1.13×10^{-1}	0
age	0	3.78×10^{-12}	2.28×10^{-13}	4.33×10^{-13}	0
dis	2.24×10^{-20}	1.00×10^{0}	1.00×10^{0}	0	0
rad	7.99×10^{-15}	2.39×10^{-10}	1.29×10^{-10}	2.10×10^{-13}	0
tax	7.60×10^{-11}	6.39×10^{-8}	7.05×10^{-8}	0	0
ptratio	4.44×10^{-1}	3.33×10^{-1}	2.13×10^{-2}	1.04×10^{-5}	2.30×10^{-1}
b	1.78×10^{-11}	1.00×10^{0}	9.97×10^{-1}	5.41×10^{-1}	9.34×10^{-2}
lstat	1.53×10^{-8}	1.00×10^{0}	1.00×10^{0}	1.56×10^{-2}	0

Six ALVMs were fitted to the model: homoskedastic, clustering (once with n_c chosen by the elbow method with SWD criterion and once with n_c chosen by five-fold CV), linear, and penalised polynomial (with L_2- and L_1-norm penalties). No spline ALVM was fitted due to the high dimensionality. For the clustering and linear ALVMs, feature selection was conducted using the Honda (1989) test at 5% significance level. This feature selection procedure resulted in eight of the twelve numerical variables being selected: **`crim`**, **`zn`**, **`indus`**, **`nox`**, **`age`**, **`dis`**, **`rad`**, **`tax`**, **`ptratio`**, and **`lstat`**. When n_c for the clustering ALVM was chosen by the elbow method, it was set to 18; when chosen by five-fold CV, it was only 2.

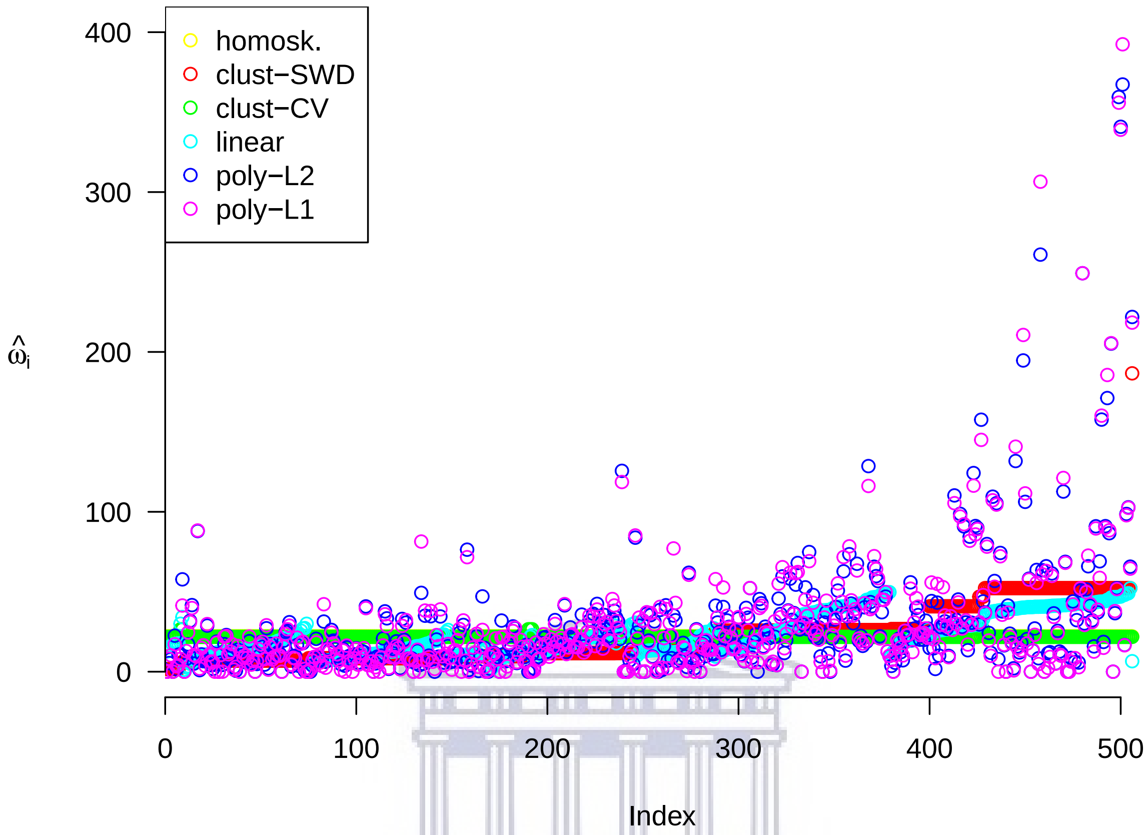


Figure 5.16: Variance Estimates for `BostonHousing2` Linear Model, Ordered by Squared OLS Residuals

In Figure 5.16, the variance estimates $\hat{\omega}_i$ under each ALVM are shown as points. In this case, the points have been ordered horizontally, not by any deflator, but in increasing order of the $\hat{\omega}_i$ from the clustering ALVM with number of clusters n_c chosen by the elbow method with SWD criterion. It is evident that the variance estimates from all the models (except, of course, for the homoskedastic case) are positively correlated with those of the clustering ALVM. It is just that the rate of increase varies; it tends to be highest with the polynomial models.

Table 5.47: FWLS Coefficient Estimates (Quasi-t-Test Significance p-Values) for `BostonHousing2` Linear Model by ALVM

	Homoskedastic	Clustering (SWD)	Clustering (CV)	Linear	L_2-Norm Pen. Poly	L_1-Norm Pen. Poly
(Intercept)	36.4 (3.66×10^{-4})	14.5 (6.72×10^{-4})	36.2 (3.57×10^{-4})	10.1 (5.69×10^{-4})	56.4 (3.65×10^{-3})	34.7 (4.28×10^{-3})
crim	-0.106 (1.72×10^{-2})	-0.116 (1.33×10^{-2})	-0.105 (2.28×10^{-2})	-0.12 (2.24×10^{-2})	-0.108 (1.66×10^{-2})	-0.118 (1.92×10^{-2})
zn	0.0477 (1.27×10^{-2})	0.036 (6.08×10^{-3})	0.0476 (1.23×10^{-2})	0.0374 (6.14×10^{-3})	0.0666 (1.39×10^{-2})	0.0846 (1.54×10^{-2})
indus	0.0233 (7.16×10^{-1})	-0.0074 (6.57×10^{-1})	0.0243 (7.14×10^{-1})	-0.067 (6.94×10^{-1})	-0.0201 (7.03×10^{-1})	0.0844 (7.04×10^{-1})
chas1	2.69 (1.98×10^{-2})	1.96 (2.33×10^{-2})	2.69 (1.92×10^{-2})	2.08 (2.53×10^{-2})	-2.12 (9.33×10^{-2})	0.18 (9.53×10^{-2})
nox	-17.7 (3.37×10^{-3})	-10.1 (3.47×10^{-3})	-17.7 (3.26×10^{-3})	-2.18 (3.72×10^{-3})	-21.7 (8.05×10^{-3})	-13.6 (8.13×10^{-3})
rm	3.79 (9.59×10^{-5})	5.58 (2.72×10^{-4})	3.8 (9.32×10^{-5})	5.34 (2.10×10^{-4})	0.781 (3.15×10^{-3})	2.63 (4.06×10^{-3})
age	0.000575 (9.66×10^{-1})	-0.0283 (9.68×10^{-1})	0.000586 (9.66×10^{-1})	-0.0575 (9.64×10^{-1})	-0.0496 (9.75×10^{-1})	-0.0401 (9.75×10^{-1})
dis	-1.5 (2.71×10^{-4})	-1.11 (1.47×10^{-4})	-1.5 (2.61×10^{-4})	-1.12 (1.61×10^{-4})	-1.79 (4.52×10^{-4})	-1.45 (5.53×10^{-4})
rad	0.304 (3.62×10^{-3})	0.226 (1.65×10^{-3})	0.303 (3.55×10^{-3})	0.187 (2.47×10^{-3})	0.294 (4.31×10^{-3})	0.221 (4.18×10^{-3})
tax	-0.0127 (1.43×10^{-2})	-0.0106 (2.28×10^{-3})	-0.0127 (1.39×10^{-2})	-0.009 (6.41×10^{-3})	-0.0144 (6.92×10^{-3})	-0.0125 (5.69×10^{-3})
ptratio	-0.924 (3.84×10^{-4})	-0.725 (2.21×10^{-4})	-0.922 (3.70×10^{-4})	-0.466 (2.23×10^{-4})	-0.692 (4.70×10^{-4})	-0.586 (5.22×10^{-4})
b	0.00923 (1.34×10^{-2})	0.00981 (3.02×10^{-2})	0.00938 (1.35×10^{-2})	0.01 (3.40×10^{-2})	0.00469 (3.27×10^{-2})	0.00509 (3.01×10^{-2})
lstat	-0.531 (4.25×10^{-5})	-0.289 (1.40×10^{-4})	-0.533 (4.21×10^{-5})	-0.33 (8.71×10^{-5})	-0.273 (1.94×10^{-3})	-0.31 (2.13×10^{-3})

Table 5.47 shows p-values for quasi-t significance tests on each coefficient in the Boston housing model, based on the various ALVMs. At 5% significance level, the only change in decision in comparison to Table 5.44 is that the categorical predictor `chas1` (proximity to Charles River) is no longer significant, according to the two polynomial models. Interestingly, this agrees with the finding of Cheng (2012), who likewise found in his robust significance tests that this coefficient is not significant. Cheng (2012) also found that the the `crim`, `zn`, `rad`, and `ptratio` coefficients were not significant. However, his method involved adjusting for both outliers and heteroskedasticity, and not only for heteroskedasticity as with the ALVMs.

5.7 Chapter Summary

The results presented in this chapter have served as a fairly thorough, albeit not exhaustive, investigation into the empirical performance of the methods developed in Chapter 3 in comparison to existing methods in the literature that had been reviewed in Chapter 2. This empirical work was made possible by the functions programmed for the **skedastic** R package as described in Chapter 4.

In §5.1, a simulation study of the performance of heteroskedasticity tests was undertaken using an AEPS metric not used in any previous simulation study of this kind. The simulation included several existing heteroskedasticity tests that have generally not been included in past simulation studies of heteroskedasticity testing (due in part to computational challenges that have now been overcome through functions such as `pRQF` in the **skedastic** package, as discussed in §4.2.5). It also included the new heteroskedasticity test proposed in §3.5.

The new ALVM-based heteroskedasticity test did not out-compete existing heteroskedasticity testing methods. However, some novel results did emerge from this simulation, in that little-known methods such as Evans and King's (1988) GLS test and Verbyla's (1993) test were found to outperform more famous methods such as Goldfeld and Quandt's (1965) test, Breusch and Pagan's (1979) test, and White's (1980) test.

Section 5.2 described the design of the most important MC simulation experiment, which had the purpose of evaluating the performance of the new auxiliary variance models introduced in §3.2. The experimental factors and factor combinations to be used were outlined, along with the metrics to be used for evaluating the performance of the models. These are summarised again in Table 5.48. The first two metrics address the variance estimation problem directly, while the last two metrics focus on the ends for which heteroskedastic error variance estimation is usually sought in practice: estimation of $\boldsymbol{\beta}$ and inference on elements of $\boldsymbol{\beta}$.

Techniques for estimating the standard errors of MC estimates were discussed next, and lastly, a scheme was introduced for reporting the metrics in relative terms to facilitate comparison across numerous different models.

Table 5.48: Summary of Four Key Metrics for Heteroskedastic Variance Estimation Performance

Metric	Description	Equation Where Defined
$\overline{\text{MSE}}_{\text{ust}}(\hat{\boldsymbol{\omega}})$	Unstandardised MSE for Estimating Individual Error Variances	(5.5)
$\overline{\text{MSE}}_{\text{st}}(\hat{\boldsymbol{\omega}})$	Standardised MSE for Estimating Individual Error Variances	(5.12)
$\text{MSE}(\hat{\boldsymbol{\beta}}_{\text{FWLS}})$	MSE for Estimating $\boldsymbol{\beta}$ Using FWLS	(5.13)
$\text{MSE}(\text{SE}(\hat{\boldsymbol{\beta}})$	MSE for Estimating Standard Errors of OLS Estimator for Purposes of Inference on β_j	(5.14)

Section 5.3 contains the 'meat' of this chapter: the empirical results of the MC simulation to evaluate the performance of the auxiliary variance models. For the simulations involving linear regression models with only one explanatory variable, results were presented in graphical form (with the metrics $\overline{\text{MSE}}_{\text{ust}}(\hat{\boldsymbol{\omega}})$ and $\overline{\text{MSE}}_{\text{st}}(\hat{\boldsymbol{\omega}})$ also decomposed into squared bias and variance components). For all four settings of number of covariates (one, two, eight, and sixteen), results on the four key performance metrics summarised in Table 5.48 were displayed in tabular form with discussion following. Additional sets of results were relegated to Appendix E.

These simulations have provided empirical evidence that the ALVMs are competitive according to several different metrics relative to existing methods, and outperform existing methods under certain conditions. Comparing the ALVMs themselves, there is no clear overall winner. Each has a niche in terms of circumstances where it appears to perform well. The linear and clustering ALVMs seem to perform well in wider circumstances than the penalised polynomial and thin-plate spline ALVMs, which makes them attractive given their simplicity and low computation time. There is evidence that the clustering ALVM can be particularly effective for large data sets.

The ANLVMs that require specification of the form of the heteroskedastic function $g(\cdot)$ are more effective than any ALVM when $g(\cdot)$ is correctly specified. This approach can be recommended when the form of heteroskedastic function is known. The clustering ANLVM showed a similar level of performance to the ALVM in most cases but had markedly better metrics in certain instances.

§5.4 looks at several other aspects of the performance of the auxiliary variance models, namely:

- whether the feature selection techniques proposed in §3.3.3 are effective;
- whether the performance of ALVMs is stable across different randomly generated design matrices $\boldsymbol{X}$; and
- whether the optimisation routine used to fit ANLVMs achieves satisfactory convergence rates.

The methods were generally found to be satisfactory in all respects.

In §5.5, an investigation was conducted into the empirical coverage probabilities achieved by the bootstrap methods discussed in §3.4 for obtaining approximate $(1-\alpha)100\%$ confidence intervals for error variances ω_i. The percentile interval and naïve standard normal interval were found to achieve reasonably good coverage in the linear and clustering ALVMs. The BCa modification to the percentile interval was found not to improve coverage but actually to detract from it. Another surprising result was that good coverage was achieved only by using the 'bootstrapping pairs' method for bootstrapping a heteroskedastic linear regression model. The vaunted 'wild bootstrap' method yielded abysmal coverage probabilities for reasons that are unclear.

Finally, for illustration purposes, ALVMs were applied to three real data sets. These were a small data set ($n = 32$) on the fuel economy of cars involving two covariates, another small data set ($n = 50$) on expenditure on public schools involving one covariate with a possible quadratic relationship to the response, and a larger data set ($n = 506$) on house values in Boston, USA with fourteen explanatory variables.

6 Conclusion

6.1 Summary of Research and Contributions

6.1.1 Introduction

Chapter 1 sets the stage for the study by introducing the linear regression model and its classical assumptions, together with important notation, terminology, and statistical results. Of particular interest, in this research, is the homoskedasticity assumption A2 and its violation, heteroskedasticity. Statistical theory covered in Chapter 1 included estimation of and inference on the gradient parameters $\boldsymbol{\beta}$ under both homoskedasticity and heteroskedasticity, statistical properties of the OLS residuals, other important kinds of model residuals, and a discussion of leverage and influence in the linear model. While most of the theory covered in this chapter is long-established and well-known, it still represents a contribution inasmuch as standard treatments of the linear regression model and its assumptions seldom offer the level of detail provided here, particularly on the statistical properties of the OLS residuals under heteroskedasticity.

6.1.2 Literature Review

Next, in Chapter 2, a review of literature was provided concerning handling of heteroskedasticity in the linear regression model. Specifically, existing methods of heteroskedasticity testing, Feasible Weighted Least Squares, Heteroskedasticity-Consistent Covariance Matrix Estimators, and heteroskedasticity-robust inference on model coefficients $\boldsymbol{\beta}$ were described and discussed, along with implementation of these methods in statistical software, especially R. This chapter is, as far as the author is aware, the most thorough review of heteroskedasticity testing methods and HCCMEs published to date. The literature review highlighted an interesting paradox. On the one hand, the stock of heteroskedasticity testing as an important tool for the linear regression practitioner has declined over the past two decades, due to several studies that advised against its use in an adaptive approach to inference. On the other hand, new heteroskedasticity testing methods have continued to appear in the literature, and 'classical' methods like Breusch and Pagan's (1979) test and White's (1980) test continue to rack up citations.

6.1.3 Methodology

Chapter 3 begins with a theoretical treatment on the statistical properties of the squared OLS residuals. Some of the results presented had not been found by the author in any previous literature. Rigorous treatments of the linear regression model typically offer statistical results on the OLS residuals, but not their squares, despite the particular importance of the latter for detecting and modelling heteroskedasticity. The statistical results derived for the squared OLS residuals include expressions for their variances and covariances (both in scalar and in matrix form, and both under homoskedasticity and under heteroskedasticity) and their marginal distributions and pairwise (bivariate) joint distributions—again, both under homoskedasticity and under heteroskedasticity. A strategy is also suggested for approximately computing the PDF of the joint distribution of all n squared OLS residuals, which however turns out to be degenerate. Thus, while these statistical results may represent a new contribution, they do not at the moment provide for new estimation or inferential methods, such as ML estimators or LR tests. The expectation and variance-covariance matrix of the squared WLS residuals under heteroskedasticity were also shown to have a form analogous to those of the squared OLS residuals, but in terms of $\boldsymbol{M_\Omega}$ rather than $\boldsymbol{M}$.

§3.1.4 highlighted a shortcoming of the existing HCCMEs and modelling-based FWLS methods described in §2: they take the squared OLS residuals e_i^2 as proxies for the corresponding unknown error variances ω_i, when in fact the former are biased estimators of the latter. What is more, the bias correction factors c_i used in the various HCCMEs may, for certain observations in certain instances, 'correct' in the wrong direction and thus increase this bias. Awareness of this shortcoming was the motivating point of departure for the most significant methodological contribution of this research project: the auxiliary variance models introduced in §3.2.

The new model with general model equation (3.34), unlike all previous auxiliary regression models proposed for estimation of heteroskedastic error variances, is correctly specified in terms of its mean function. A further advantage is that the variance-covariance matrix of this model's errors is known in terms of the parameter vector $\boldsymbol{\omega}$ that occurs in the mean function. An obvious shortcoming of the model, as specified initially in (3.34), is that there are n parameters to be estimated from n observations. However, in §3.2.2-3.2.4, several strategies were proposed for reducing the number of parameters to be estimated. All of these rested on the assumption that the error variance parameters ω_i are in fact related to some design variables $\boldsymbol{Z}'_{i\cdot}$ through a continuous, differentiable function $g(\cdot)$. The form of $g(\cdot)$ could be specified by assumption (as in the linear ALVM and the quadratic and exponential ANLVMs) or estimated within certain restrictions (as in the penalised polynomial ALVMs and regression spline ALVMs). Alternatively, an agglomerative hierarchical clustering algorithm could be used to group points that are proximal in the covariate space, on the premise that proximity implies (approximately) equal variances, by the differentiability of $g(\cdot)$. This leads to the clustering ALVM (and ANLVM).

Section 3.3.1 discussed methods for fitting the ALVMs and ANLVMs that had been proposed. Fitting the ALVMs that do not have penalty terms (basic, linear, and clustering ALVMs) entails solving an Inequality-Constrained Least Squares problem. When there is a square penalty matrix with the same dimensions as the parameter vector $\boldsymbol{\gamma}$ $(q \times q)$, as in the L_2-norm penalised polynomial and thin-plate spline ALVMs, the fitting problem is an Inequality-Constrained Ridge Regression problem. Both an ICLS problem and ICRR problem are special cases of a Quadratic Programming problem, and the estimation problem for the L_1-norm penalised polynomial ALVM can also be expressed as a QP problem. Thus, QP is a unifying approach to fitting all of the ALVMs proposed herein.

In the case of ANLVMs, MQL estimation, as described in §3.3.1.4, can be used to fit the model. A strength of this approach is that it takes into account the known form of the model errors' variance-covariance matrix; a weakness is that convergence of the associated Gauss-Newton optimisation algorithms is not guaranteed, and the solution may be sensitive to initial values. Use of a grid of initial values can mitigate these issues.

Section 3.3.1.5 discussed how the known form of the ALVM errors' variance-covariance matrix could potentially be used to improve model estimation by using the initial ICLS or ICRR estimates in a generalised (specifically, FICGLS) two-step or iterative procedure. However, results reported in Chapter 5 are based only on a single QP step and not on generalised procedures.

Some of the ALVMs involve hyperparameters. Of particular importance are the λ penalty intensity parameter in the penalised polynomial and smoothing and thin-plate spline ALVMs, and the n_c parameter (number of clusters) in the clustering ALVM. §3.3.2 discusses strategies for tuning these hyperparameters, using K-fold cross-validation, quasi-generalised cross-validation, and (in the case of n_c) an elbow method based on a criterion such as SWD. Applying K-fold CV to an ALVM is non-trivial. Two techniques are proposed for doing so, called the test set OLS technique and the partitioning of residuals technique. The former is theoretically more sound and was therefore used for the simulations reported in Chapter 5.

All of the ALVMs rely on correct specification of the matrix $\boldsymbol{Z}$ of features that are related through a function $g(\cdot)$ to the linear regression model error variances. In the absence of extraneous information, it might be reasonable to assume that these predictors are a subset of those in the feature matrix $\boldsymbol{X}$ from the original linear model. Even then, the question remains, which subset? To this end, §3.3.3 discusses several feature selection techniques that can be used to attempt to answer this question. One technique is the shrinkage penalty that is built into the penalised polynomial ALVM. Particularly with the L_1-norm penalty (the LASSO-type ALVM), the sparsity properties tend to result in coefficients of unimportant features being shrunk to zero, which is tantamount to non-selection of those features. A second feature selection technique entails conducting a deflator-based heteroskedasticity test with each feature in turn serving as the deflator. The effectiveness of this technique depends, of course, on the power and size of the test. The significance level therefore becomes like an additional hyperparameter of the model, with a lower significance level representing a more conservative approach to feature selection. A third feature selection technique is BSS, using either an exhaustive search or a greedy search, with K-fold CV or QGCV used to compute the loss function.

Some statistical results on the variance estimators were discussed briefly in §3.3.4. In the case of the ALVMs, the gist of the discussion was that obtaining analytical results on the variance estimators seems intractable based on existing theory on statistical properties of ICLS and ICRR estimators. Thus bootstrap methods seemed to be the best way forward for obtaining interval estimates.

Several nonparametric bootstrap methods for obtaining confidence intervals for individual error variances ω_i were proposed in §3.4. Firstly, two methods of nonparametric bootstrap resampling of heteroskedastic linear regression models were discussed, namely bootstrapping pairs and the wild bootstrap. After the ALVM has been fitted to the bootstrapped regression models, how should one compute the confidence intervals? Three methods were discussed: a naïve normal interval, the percentile interval, and the BCa interval. A multivariate extension of the percentile interval by Olive (2018) was also discussed, which could be used if one is interested in a confidence *region* for $\boldsymbol{\omega}$.

Rounding off the methodology chapter was a proposed new heteroskedasticity test based on an ALVM

(§3.5). The test statistic was constructed as a ratio of two sums of squares in the squared OLS residuals. In the numerator, the e_i^2 are compared with their expectations under homoskedasticity, but with the common variance ω replaced by an unbiased estimator. In the denominator, the e_i^2 are compared with their expectations under heteroskedasticity, but with the error variances ω_i replaced by their ALVM estimators. No exact or asymptotic null distribution for the test statistic was offered; p-values can however be computed using simulation-based methods such as those of Godfrey and Orme (1999) or Dufour et al. (2004), which had been described in the literature review. There would be some redundancy in using an ALVM to test for heteroskedasticity if a heteroskedasticity test were used for feature selection within the ALVM fitting procedure.

The new methods proposed in this chapter, of which the ALVMs are the most important, represent a new approach to modelling and handling heteroskedasticity in the linear regression model. Their main theoretical justification is that the mean function of the auxiliary regression is correctly specified. That the variance-covariance matrix of the auxiliary regression model is of known form (in terms of the same parameters $\boldsymbol{\gamma}$ that appear in its mean function) is also advantageous.

6.1.4 Software Implementation

In Chapter 4, a new R package called **skedastic**, developed specifically for this research, was described in detail. The package's functionality includes implementation of many of the methods discussed in Chapter 2 (such as heteroskedasticity tests and HCCMEs), as well as implementation of the most important new methods proposed in Chapter 3: the ALVMs and bootstrap confidence intervals based on them. Many of the existing methods programmed in **skedastic** had not previously been made available in statistical software, to the author's knowledge. Various supporting functions were also described, some of them (such as `pRQF` and `twosidedpval`) providing value to statistics practitioners well beyond the confines of the problem of heteroskedasticity in linear regression. This package therefore represents a significant contribution in its own right.

6.1.5 Results

Since the Methodology chapter was not able to offer much else in the way of rigorous statistical proofs of the validity or optimality of the proposed new models and methods, empirical evidence of their effectiveness is required. Chapter 5 sought to address this need, primarily by means of MC simulations run in R software.

In §5.1, Lloyd's (2005) AEPS metric was used to compare the performance of numerous existing heteroskedasticity tests, along with the new ALVM-based tests that had been proposed in §3.5. These simulation experiments made an original contribution in that the heteroskedasticity testing methods found to be most effective—at least under these experimental conditions—were not the most popular tests (such as Breusch and Pagan's (1979) and White's (1980)). Instead, lesser-known methods such as Evans and King's (1988) and Verbyla's (1993)—which have usually not even been *considered* in previous MC power studies of heteroskedasticity tests—proved to be most effective. The newly developed ALVM-based tests performed reasonably well, albeit not as well as some other methods.

Section 5.2 described the design of the main MC experiment, the purpose of which was to evaluate the performance of the new ALVMs as tools for estimation and inference in the linear regression model, relative to other existing methods (particularly, HCCMEs and Miller and Startz's (2019) SVR model). Four performance metrics were defined. The first two looked at the models' MSE for estimating the error variances ω_i as an end in itself; one was unstandardised relative to the different magnitudes of the ω_i and the other was standardised. Two other metrics were introduced on the grounds that the error variances are usually not of primary interest to the linear regression practitioner, but are useful for enabling robust estimation of and inference on the coefficient vector $\boldsymbol{\beta}$. Accordingly, the third metric looked at the models' MSE for estimating $\boldsymbol{\beta}$ using FWLS with the weights being the reciprocals of the ALVM error variance estimates, $\hat{\omega}_i^{-1}$. The fourth metric looked at the models' MSE for estimating $\mathrm{SE}(\hat{\boldsymbol{\beta}}_{\mathrm{OLS}})$ (elementwise), since these standard errors form the denominator of the quasi-t test statistic used in robust significance tests on the coefficients β_j.

The results of these simulations, run with four different dimensionalities (one covariate; two covariates; eight covariates; sixteen covariates) were reported and discussed in §5.3. In the one-covariate (simple linear regression) case, simulation results were shown in graphical form and the MSEs metrics were also broken down into squared bias and variance components. With all four simulations, the results for each of the four metrics were also reported in tabular form. The tables expressed the estimated MSEs in relative terms, with the lowest estimated MSEs reported as 1 (highlighted in green) and the others as multiples thereof. Yellow colour was used to highlight MSE estimates that were not inferior to the best one by a statistically significant margin. Overall, the results of these simulations were a mixed bag. The new ALVMs did not *always* outperform the existing methods, but they did outperform them in some circumstances for some metrics. Thus, these simulations provide evidence that the new ALVMs are viable and competitive statistical methods that should be given serious consideration by practitioners of linear regression.

A limited extension of the simulation to other factor combinations was also made. In the simple linear regression case, sample sizes of $n = 20$ and $n = 1000$ were used in addition to the default $n = 100$. Moreover, for simple linear regression with $n = 100$, a nonmonotonic (sinusoidal) heteroskedastic DGP was used, and in another instance, non-normal errors (specifically, Laplace- and uniform-distributed) were used. In the two-covariate and eight-covariate cases, a setting involving multicollinearity among the predictors was tried. Results tables for these extensions of the experiment are all found in Appendix E, but they are discussed in §5.3.

Results on ANLVMs were also reported under some of the experimental conditions. The quadratic and exponential ANLVMs were found to perform very well—better than any other method—when their specification of the heteroskedastic function $g(\cdot)$ was correct. The performance of the clustering ANLVM was comparable to that of the clustering ALVM and, in some instances, better.

§5.4 supplemented these model performance simulations by looking in more detail at the performance of certain aspects of the ALVMs. §5.4.1 looked at the effectiveness of the different feature selection techniques used in conjunction with ALVMs (as discussed in §3.3.3). Conducting an Evans-King GLS test of heteroskedasticity (with each covariate in turn serving as the deflator) was found to be the most successful feature selection method overall, although best subset selection using QGCV based on the linear ALVM was also competitive.

Section 5.4.2 looked at the stability of the ALVM estimates relative to the form of the design matrix $\boldsymbol{X}$. Here, the ALVMs were found for the most part to produce stable results across different randomly generated design matrices. The caveat is that the linear ALVM estimates can be highly sensitive to the form of $\boldsymbol{X}$ when there is extreme heteroskedasticity linked to the covariates.

Convergence rates for the Gauss-Newton optimisation routine used to implement MQL estimation for fitting of the ANLVMs were also reported, in §5.4.3, and were found to be very high except under multiplicative heteroskedasticity linked to multiple covariates (which would be heteroskedasticity of an extreme magnitude).

Section 5.5 reported on MC simulations conducted to evaluate the coverage probabilities of approximate CIs for the ω_i based on the bootstrap techniques described in §3.4. One important finding from these simulations was that the intervals based on the pairs bootstrap were far superior to those based on the wild bootstrap. Indeed, the coverage probabilities for the intervals based on the wild bootstrap were so low as to render these intervals meaningless. Moreover, the percentile interval and naïve normal interval performed reasonably well in terms of coverage, but the BCa interval—ostensibly an improvement on the percentile interval—failed to improve the coverage probabilities in this instance.

Finally, §5.6 illustrated the application of ALVMs to three real-world data sets: the 'mtcars' fuel economy data set, a per capita expenditure on public schools data set, and the well-known Boston house values data set.

6.2 Achievement of Research Objectives

The first research objective was to review and catalogue the many existing heteroskedasticity testing methods in the literature. This was achieved, as evidenced in §2.1. The second research objective was to program these heteroskedasticity testing methods and make them accessible to practitioners via a package in R statistical software. This has been achieved by the **skedastic** R package developed for this research, as has been discussed in §4.1. The third research objective was to evaluate the role (if any) of heteroskedasticity tests in handling the problem of heteroskedasticity in the linear model. The literature review found (§2.4.3) that an adaptive approach to inference in linear regression has largely been discredited, which seemed to have curtailed the relevance of heteroskedasticity testing. However, this study has identified a new role for heteroskedasticity testing, namely as a feature selection technique within the auxiliary variance models developed herein.

The fourth objective was to develop a new method of handling heteroskedasticity in the linear model by direct estimation of the error variances using a suitable auxiliary model. This objective was achieved by the development of the ALVM and the related ANLVM, with its several forms, in the Methodology chapter. This development consisted not only of specifying the model but of providing viable methods of reducing the number of parameters to be estimated, of fitting the model, and of tuning the model's hyperparameters (where applicable).

The fifth objective was to show empirically, using MC simulations, that the new methods (ALVMs and ANLVMs) perform well relative to existing methods in terms of meaningful performance metrics. This objective was achieved by the simulations described and reported on in the Chapter 5, particularly in §5.2 and §5.3. The sixth objective was to show empirically, using MC simulations, that the new methods are robust in certain respects. This objective was achieved by showing that the ALVMs are reasonably stable across different design matrices (§5.4.2) and still perform reasonably well in the presence of multicollinearity (Appendix E).[129]

The seventh and final objective was to make the new method(s) accessible to practitioners via a package in R statistical software. This too has been achieved via the **skedastic** R package, as described in detail in Chapter 4 (particularly §4.4).

The research objectives have thus all been achieved, and in doing so a viable solution has been produced to the research problem, inasmuch as the new auxiliary variance models offer a unified approach to handling heteroskedasticity in the linear regression model, one that is accessible to practitioners via a package in R software.

6.3 Possible Directions for Future Research

This study has opened up a number of possible avenues for further research. A few of these will be briefly outlined here.

6.3.1 Maximum Likelihood Methods Based on Multivariate Gamma Distribution

It was discussed in §3.1.3 that the joint distribution of the squared OLS residuals, both under homoskedasticity and under heteroskedasticity, is a multivariate Gamma distribution and, more specifically, a generalisation of the bivariate Gamma distribution due to Kibble (1941). This multivariate Gamma distribution is problematic as a tool for distribution-based methods for handling heteroskedasticity such as ML estimation of error variances or LR tests for heteroskedasticity. It is problematic not only because (3.24) is difficult to compute, but also because the joint distribution is in this instance degenerate (having a singular variance-covariance matrix). Further research undertaken to overcome these difficulties—through numerical methods and dimension reduction, for instance—may facilitate the development of new ML estimation techniques and/or LR tests.

6.3.2 Other Ways of Specifying an Auxiliary Nonlinear Variance Model

The Methodology chapter of this study introduced ANLVMs, based on a parametric specification of $\boldsymbol{\omega}$ in (3.34) that related $\boldsymbol{\omega}$ to the auxiliary covariate matrix $\boldsymbol{Z}$ in terms of a nonlinear function of parameters $\boldsymbol{\gamma}$. Another nonlinear form of the auxiliary regression model could be arrived at by using the logarithm of the squared OLS residuals as the response, as discussed in Appendix C.2. This type of ANLVM merits further exploration.

Still another specification of an ANLVM can be constructed from the results given previously in §1.1.7.4. It can be shown using the same steps used in Appendix C.1.1 to derive the results in §3.1.2 that the following results hold (under A1, A3-A5) concerning the squared WLS residuals $\boldsymbol{e}_{\text{WLS}} \circ \boldsymbol{e}_{\text{WLS}}$:

$$\mathrm{E}(\boldsymbol{e}_{\text{WLS}} \circ \boldsymbol{e}_{\text{WLS}}) = \operatorname{diag}(\boldsymbol{M}_{\boldsymbol{\Omega}}\boldsymbol{\Omega}\boldsymbol{M}_{\boldsymbol{\Omega}}) = (\boldsymbol{M}_{\boldsymbol{\Omega}} \circ \boldsymbol{M}_{\boldsymbol{\Omega}})\,\boldsymbol{\omega} \tag{6.1}$$

and

$$\mathrm{Cov}(\boldsymbol{e}_{\text{WLS}} \circ \boldsymbol{e}_{\text{WLS}}) = 2\,(\boldsymbol{M}_{\boldsymbol{\Omega}}\boldsymbol{\Omega}\boldsymbol{M}_{\boldsymbol{\Omega}}) \circ (\boldsymbol{M}_{\boldsymbol{\Omega}}\boldsymbol{\Omega}\boldsymbol{M}_{\boldsymbol{\Omega}})\,. \tag{6.2}$$

Equations (6.1) and (6.2) can be recognised as having the same form as (3.10) and (3.11), but with $\boldsymbol{M}$ replaced with $\boldsymbol{M}_{\boldsymbol{\Omega}}$. Thus, an auxiliary regression model can be constructed with the same form as (3.34), but with $\boldsymbol{e} \circ \boldsymbol{e}$ replaced by $\boldsymbol{e}_{\text{WLS}} \circ \boldsymbol{e}_{\text{WLS}}$ and $\boldsymbol{M} \circ \boldsymbol{M}$ replaced by $\boldsymbol{M}_{\boldsymbol{\Omega}} \circ \boldsymbol{M}_{\boldsymbol{\Omega}}$. This model is nonlinear in $\boldsymbol{\omega}$, since $\boldsymbol{M}_{\boldsymbol{\Omega}}$ depends on $\boldsymbol{\omega}$. Thus, it is not possible to develop an ALVM based on the squared WLS residuals, but an ANLVM could be developed. The caveat, of course, is that it is not possible to compute the $\boldsymbol{e}_{\text{WLS}}$ without knowing $\boldsymbol{\Omega}^{-1}$, and thus using this model would require a preliminary estimate of $\boldsymbol{\omega}$—perhaps obtained from an ALVM based on the OLS squared residuals. It is certainly worth exploring a generalisation of the auxiliary variance models proposed in this study (which were built around the squared OLS residuals) to the WLS case, or rather the FWLS case, using the multi-step FICGLS fitting procedure outlined in §3.3.1.5.

[129] These robustness checks have not been performed on the ANLVMs as yet.

6.3.3 Further Exploration of Generalised and Weight-Based Estimation of Auxiliary Linear Variance Models

In §3.3.1.5, GLS-based estimation procedures for the ALVMs were discussed on the grounds that the variance-covariance matrix of the ALVM errors $\boldsymbol{u}$ is known in terms of $\boldsymbol{\omega}$ (see (3.3) and (3.8) for these results under homoskedasticity and heteroskedasticity, respectively). It was believed that this known form of the variance-covariance matrix would result in multi-step FICGLS producing more accurate estimates than a single-step ICLS procedure that does not take into account the covariance structure of the errors. However, preliminary simulations found that performing the additional steps for FICGLS, as described in §3.3.1.5, made virtually no difference in the error variance estimates $\hat{\omega}_i$ or to the performance of the ALVMs, and therefore for reasons of computational cost generalised estimation procedures were not used in the main simulations in the Results and Discussion chapter.

Nevertheless, there seems to be great potential in using FICGLS rather than ICLS to fit the ALVMs, and further research may be able to identify the reasons why the generalised estimation procedures outlined in §3.3.1.5 did not yield significant improvements, and rectify this.

Two weighting procedures were also proposed in §3.3.1.5 to potentially improve on the estimation of the ALVMs in specific circumstances. It remains to explore the effectiveness of these weighting methods.

6.3.4 Improvement of Hyperparameter Tuning Using Cross-Validation

Two techniques for predicting ALVM responses for purposes of K-fold CV were discussed in §3.3.2.1, but only one technique—the test fold OLS technique—was used to generate results. Further research can be conducted into the relative performance of these two techniques in terms of bias vs. variance.

6.3.5 Improvement of Bootstrap Confidence Intervals for Error Variances

It was noted in §5.5 that, surprisingly, the wild bootstrap technique resulted in very poor coverage probabilities in confidence intervals constructed for individual error variances ω_i based on ALVMs. It is not clear why the wild bootstrap should have been so far inferior to the pairs bootstrap in this respect, since the wild bootstrap is a well-established method for bootstrapping of heteroskedastic linear regression models. Indeed, as a check, the same DGPs used to estimate the CI coverage probabilities were also used to estimate the SEs of $\hat{\boldsymbol{\beta}}_{\text{OLS}}$ (as in the wild bootstrap HCCME discussed in §2.3.10). The wild bootstrap and pairs bootstrap performed about equally well in estimating these SEs, so it remains a mystery why the wild bootstrap is so ineffective for ALVM-based interval estimation of the error variances ω_i. Further research may solve this mystery and result in effective bootstrap CI methods for the error variances based on the wild bootstrap.

Equally surprising was that the BCa method, which is designed to improve on coverage probability relative to the bootstrap percentile interval, actually took the coverage probabilities further away from the nominal 95%. Further research could be used to identify the reasons for this and lead to improved interval estimation of error variances using bootstrap methods built on ALVMs.

6.3.6 Extension of Empirical Work to Larger Data Sets

The largest sample size ($n = 1000$) and the largest design dimensionality ($p = 17$) considered in the simulations reported in Chapter 5 are still small in comparison to the sizes of data sets often being analysed today. Computing resources were a constraint on the present study, but it would certainly be worthwhile to explore the empirical performance of the ALVMs and ANLVMs in connection with linear regression models fitted to much larger data sets.

With this outline of possible future research avenues in hand, it may be appropriate to close with a famous aphorism from Winston Churchill: ‘Now this is not the end. It is not even the beginning of the end. But it is, perhaps, the end of the beginning’ (Churchill 1943, p. 266).

Appendices

A Proofs of Some Elementary Statistical Results on the Linear Regression Model

A.1 Derivation of the Ordinary Least Squares Estimator

Let $\hat{\boldsymbol{\beta}}$ be a candidate estimator of $\boldsymbol{\beta}$. The objective function to be minimised is the sum of squared residuals.

$$\begin{aligned}
SS_{\text{residual}}(\boldsymbol{\beta}) &= (\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}})'(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}) \\
&= (\boldsymbol{y}' - \hat{\boldsymbol{\beta}}'\boldsymbol{X}')(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}) \\
&= \boldsymbol{y}'\boldsymbol{y} - \hat{\boldsymbol{\beta}}'\boldsymbol{X}'\boldsymbol{y} - \boldsymbol{y}'\boldsymbol{X}\hat{\boldsymbol{\beta}} + \hat{\boldsymbol{\beta}}'\boldsymbol{X}'\boldsymbol{X}\hat{\boldsymbol{\beta}} \\
&= \boldsymbol{y}'\boldsymbol{y} - 2\hat{\boldsymbol{\beta}}'\boldsymbol{X}'\boldsymbol{y} + \hat{\boldsymbol{\beta}}'\boldsymbol{X}'\boldsymbol{X}\hat{\boldsymbol{\beta}} \text{ (since } \boldsymbol{y}'\boldsymbol{X}\hat{\boldsymbol{\beta}} = \left(\hat{\boldsymbol{\beta}}'\boldsymbol{X}'\boldsymbol{y}\right)' \text{ is a scalar)} \\
\frac{\partial}{\partial\hat{\boldsymbol{\beta}}} SS_{\text{residual}} &= -2\boldsymbol{X}'\boldsymbol{y} + 2\boldsymbol{X}'\boldsymbol{X}\hat{\boldsymbol{\beta}} = 0 \\
\boldsymbol{X}'\boldsymbol{X}\hat{\boldsymbol{\beta}} &= \boldsymbol{X}'\boldsymbol{y} \\
\hat{\boldsymbol{\beta}} &= (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{y} \text{ (note: } \boldsymbol{X}'\boldsymbol{X} \text{ is invertible by A4).}
\end{aligned}$$

Note that $\dfrac{\partial^2}{\partial\hat{\boldsymbol{\beta}}^2} SS_{\text{residual}} = 2\boldsymbol{X}'\boldsymbol{X}$ is positive definite,[130] so by the second derivative test, the above critical point is a minimum.

A.2 A Proof of the Gauss-Markov Theorem

The Gauss-Markov Theorem states that, under assumptions A1-A4, $\hat{\boldsymbol{\beta}}_{\text{OLS}}$ is the Best Linear Unbiased Estimator (BLUE) of $\boldsymbol{\beta}$. This can be proven as follows. Let $\hat{\boldsymbol{\beta}} = \boldsymbol{A}\boldsymbol{y}$ be a linear unbiased estimator of $\boldsymbol{\beta}$, where $\boldsymbol{A}$ depends only on $\boldsymbol{X}$. The conditional expectation of the estimator can be expressed as follows:

$$\begin{aligned}
\mathrm{E}\left(\hat{\boldsymbol{\beta}}\right) &= \boldsymbol{A}\,\mathrm{E}(\boldsymbol{y}) \\
&= \boldsymbol{A}\boldsymbol{X}\boldsymbol{\beta} \text{ (by A1).}
\end{aligned}$$

Thus, it follows from the unbiasedness of $\hat{\boldsymbol{\beta}}$ that $\boldsymbol{A}\boldsymbol{X} = \boldsymbol{I}_p$ (which, notably, is satisfied when $\boldsymbol{A} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'$). Now, by A1 and A2, the conditional covariance matrix of the estimator is

$$\mathrm{Cov}(\hat{\boldsymbol{\beta}}) = \boldsymbol{A}\omega\boldsymbol{I}_p\boldsymbol{A}' - \boldsymbol{\beta}\boldsymbol{\beta}' = \omega\boldsymbol{A}\boldsymbol{A}' - \boldsymbol{\beta}\boldsymbol{\beta}'.$$

$\mathrm{Cov}(\hat{\boldsymbol{\beta}})$ can be written as $c(\boldsymbol{A}) - \boldsymbol{\beta}\boldsymbol{\beta}'$, where $c(\boldsymbol{A}) = \omega\boldsymbol{A}\boldsymbol{A}'$. Since $\boldsymbol{\beta}\boldsymbol{\beta}'$ is constant (not depending on $\boldsymbol{X}$), what one seeks to minimise are the diagonal elements of $c(\boldsymbol{A})$. Suppose, without loss of generality, that $\boldsymbol{A} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}' + \boldsymbol{B}$, where $\boldsymbol{B}$ is some real-valued matrix of the same dimensions as $\boldsymbol{A}$. The unbiasedness condition then implies that $\boldsymbol{B}\boldsymbol{X} = \boldsymbol{0}$, and one can proceed as follows:

$$\begin{aligned}
c(\boldsymbol{A}) &\propto \left((\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}' + \boldsymbol{B}\right)\left((\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}' + \boldsymbol{B}\right)' \\
&= \left((\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}' + \boldsymbol{B}\right)\left(\boldsymbol{B}' + \boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\right) \text{ (since } \boldsymbol{X}'\boldsymbol{X} \text{ is symmetric and thus also } (\boldsymbol{X}'\boldsymbol{X})^{-1}) \\
&= (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1} + (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{B}' + \boldsymbol{B}\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1} + \boldsymbol{B}\boldsymbol{B}' \\
&= (\boldsymbol{X}'\boldsymbol{X})^{-1} + \left(\boldsymbol{B}\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\right)' + \boldsymbol{B}\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1} + \boldsymbol{B}\boldsymbol{B}' \\
&= (\boldsymbol{X}'\boldsymbol{X})^{-1} + \boldsymbol{0}'_{p\times p} + \boldsymbol{0}_{p\times p} + \boldsymbol{B}\boldsymbol{B}' \text{ (by unbiasedness).}
\end{aligned}$$

[130] This is true because $\boldsymbol{X}$ has full column rank, which means that $\boldsymbol{u} := \boldsymbol{X}\boldsymbol{v} \neq 0$ for any nonzero vector $\boldsymbol{v}$. Therefore, $\boldsymbol{v}'\boldsymbol{X}'\boldsymbol{X}\boldsymbol{v} = \boldsymbol{u}'\boldsymbol{u} = \displaystyle\sum_{i=1}^{n} u_i^2 > 0$.

The diagonal elements of $\boldsymbol{B}\boldsymbol{B}'$ are nonnegative since they are all sums of squared elements of $\boldsymbol{B}$. Thus, the smallest possible diagonal values of $\mathrm{Cov}(\hat{\beta})$ occur when $\boldsymbol{B} = \mathbf{0}_{p\times n}$, which implies that $\boldsymbol{A} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'$, i.e., that $\hat{\beta} = \hat{\beta}_{\mathrm{OLS}}$. The theorem is therefore proven.[131]

A.3 Maximum Likelihood Estimator of Linear Regression Parameters under Classical Model Assumptions

Under A1-A5 the likelihood function for the parameter vector $\boldsymbol{\theta}_0 = [\beta', \omega]'$ can be derived from the Gaussian PDF as

$$L_0(\boldsymbol{\theta}_0) = (2\pi)^{-n/2}(\omega)^{-n/2}\exp\left\{-\frac{1}{2\omega}(\boldsymbol{y}-\boldsymbol{X}\beta)'(\boldsymbol{y}-\boldsymbol{X}\beta)\right\}. \tag{A.1}$$

Note the following matrix derivative identity (Petersen and Pedersen 2012, p. 11):

$$\frac{\partial}{\partial \boldsymbol{x}}\left[(\boldsymbol{B}\boldsymbol{x}+\boldsymbol{b})'\boldsymbol{C}(\boldsymbol{B}\boldsymbol{x}+\boldsymbol{b})\right] = \boldsymbol{B}'(\boldsymbol{C}+\boldsymbol{C}')(\boldsymbol{B}\boldsymbol{x}+\boldsymbol{b}). \tag{A.2}$$

From (A.2), it follows that

$$\frac{\partial}{\partial \beta}\left[(\boldsymbol{y}-\boldsymbol{X}\beta)'(\boldsymbol{y}-\boldsymbol{X}\beta)\right] = -\boldsymbol{X}'(\boldsymbol{I}_n+\boldsymbol{I}_n)(\boldsymbol{y}-\boldsymbol{X}\beta) = -2\boldsymbol{X}'(\boldsymbol{y}-\boldsymbol{X}\beta).$$

Taking $\ell_0 = \log L_0$, differentiating with respect to the parameters, and setting the derivatives equal to zero, one can derive the Maximum Likelihood (ML) estimator of $\boldsymbol{\theta}_0$ as follows:

$$\begin{aligned}
\ell_0(\boldsymbol{\theta}_0) &= -\frac{n}{2}\log(2\pi) - \frac{n}{2}\log(\omega) - \frac{1}{2\omega}(\boldsymbol{y}-\boldsymbol{X}\beta)'(\boldsymbol{y}-\boldsymbol{X}\beta)\\
\frac{\partial \ell_0}{\partial \omega} &= -\frac{n}{2\omega} + \frac{1}{2\omega^2}(\boldsymbol{y}-\boldsymbol{X}\beta)'(\boldsymbol{y}-\boldsymbol{X}\beta) = 0\\
\frac{n}{2\omega} &= \frac{1}{2\omega^2}(\boldsymbol{y}-\boldsymbol{X}\beta)'(\boldsymbol{y}-\boldsymbol{X}\beta)\\
\omega &= \frac{1}{n}(\boldsymbol{y}-\boldsymbol{X}\beta)'(\boldsymbol{y}-\boldsymbol{X}\beta)\\
\frac{\partial \ell_0}{\partial \beta} &= -\frac{1}{2\omega}\left(-2\boldsymbol{X}'(\boldsymbol{y}-\boldsymbol{X}\beta)\right) = \mathbf{0}_{p\times 1}\\
\boldsymbol{X}'\boldsymbol{y} - \boldsymbol{X}'\boldsymbol{X}\beta &= \mathbf{0}_{p\times 1}\\
\beta &= \left(\boldsymbol{X}'\boldsymbol{X}\right)^{-1}\boldsymbol{X}'\boldsymbol{y}.
\end{aligned}$$

Thus, the ML estimators of $\boldsymbol{\theta}_0$ are

$$\hat{\boldsymbol{\theta}}_{0,\mathrm{MLE}} = \begin{bmatrix}\hat{\beta}_{\mathrm{MLE}}\\ \bar{\omega}\end{bmatrix} = \begin{bmatrix}\left(\boldsymbol{X}'\boldsymbol{X}\right)^{-1}\boldsymbol{X}'\boldsymbol{y}\\ \frac{1}{n}(\boldsymbol{y}-\boldsymbol{X}\hat{\beta})'(\boldsymbol{y}-\boldsymbol{X}\hat{\beta})\end{bmatrix} = \begin{bmatrix}\left(\boldsymbol{X}'\boldsymbol{X}\right)^{-1}\boldsymbol{X}'\boldsymbol{y}\\ \frac{1}{n}\boldsymbol{e}'\boldsymbol{e}\end{bmatrix},$$

where $\boldsymbol{e}$ are the OLS residuals.

B A Proof of a Generalisation of the Gauss-Markov Theorem

This theorem asserts that, under assumptions A1 and A3-A4, the weighted least squares estimator $\hat{\beta}_{\mathrm{WLS}}$ is the Best Linear Unbiased Estimator (BLUE) of β. First, let $\hat{\beta} = \boldsymbol{A}\boldsymbol{y}$ be a linear unbiased estimator of β, where $\boldsymbol{A}$ depends only on $\boldsymbol{X}$. As before, the unbiasedness property implies that $\boldsymbol{A}\boldsymbol{X} = \boldsymbol{I}_p$ (observe that this is satisfied when $\boldsymbol{A} = \left(\boldsymbol{X}'\boldsymbol{W}\boldsymbol{X}\right)^{-1}\boldsymbol{X}'\boldsymbol{W}$). The conditional covariance matrix of the estimator in this case is

$$\mathrm{Cov}(\hat{\beta}) = \boldsymbol{A}\boldsymbol{\Omega}\boldsymbol{A}' - \beta\beta'.$$

[131] This proof is sketched in Heij et al. (2004, p. 127), for instance. For a slightly different proof, see Rencher and Schaalje (2008, p. 147).

One can write $\text{Cov}(\hat{\boldsymbol{\beta}})$ as $c(\boldsymbol{A}) - \boldsymbol{\beta}\boldsymbol{\beta}'$, where $c(\boldsymbol{A}) = \boldsymbol{A}\boldsymbol{\Omega}\boldsymbol{A}'$. The BLUE of $\boldsymbol{\beta}$ results from the choice of $\boldsymbol{A}$ that minimises the diagonal elements of $c(\boldsymbol{A})$. Suppose, without loss of generality, that $\boldsymbol{A} = (\boldsymbol{X}'\boldsymbol{\Omega}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{\Omega}^{-1} + \boldsymbol{B}$. The unbiasedness condition implies that $\boldsymbol{B}\boldsymbol{X} = \boldsymbol{0}$. One can then proceed as follows:

$$\begin{aligned}
c(\boldsymbol{A}) = \boldsymbol{A}\boldsymbol{\Omega}\boldsymbol{A}' &= \left[\left(\boldsymbol{X}'\boldsymbol{\Omega}^{-1}\boldsymbol{B}\right)^{-1}\boldsymbol{X}'\boldsymbol{\Omega}^{-1} + \boldsymbol{B}\right]\boldsymbol{\Omega}\left[\left(\boldsymbol{X}'\boldsymbol{\Omega}^{-1}\boldsymbol{B}\right)^{-1}\boldsymbol{X}'\boldsymbol{\Omega}^{-1} + \boldsymbol{B}\right]' \\
&= \left[\left(\boldsymbol{X}'\boldsymbol{\Omega}^{-1}\boldsymbol{B}\right)^{-1}\boldsymbol{X}'\boldsymbol{\Omega}^{-1} + \boldsymbol{B}\right]\boldsymbol{\Omega}\left[\boldsymbol{\Omega}^{-1}\boldsymbol{X}\left(\boldsymbol{X}'\boldsymbol{\Omega}^{-1}\boldsymbol{B}\right)^{-1} + \boldsymbol{B}'\right] \\
&= \left(\boldsymbol{X}'\boldsymbol{\Omega}^{-1}\boldsymbol{X}\right)^{-1}\boldsymbol{X}'\boldsymbol{\Omega}^{-1}\boldsymbol{\Omega}\boldsymbol{\Omega}^{-1}\boldsymbol{X}\left(\boldsymbol{X}'\boldsymbol{\Omega}^{-1}\boldsymbol{X}\right)^{-1} + \boldsymbol{B}\boldsymbol{\Omega}\boldsymbol{B}' + \underbrace{\text{cross-terms}}_{=\boldsymbol{0}\text{ since }\boldsymbol{B}\boldsymbol{X}=\boldsymbol{0}} \\
&= \left(\boldsymbol{X}'\boldsymbol{\Omega}^{-1}\boldsymbol{X}\right)^{-1} + \boldsymbol{B}\boldsymbol{\Omega}\boldsymbol{B}'.
\end{aligned}$$

The ith diagonal element of $\boldsymbol{B}\boldsymbol{\Omega}\boldsymbol{B}'$ is $\displaystyle\sum_{j=1}^{n} b_{ij}^2\omega_j \geq 0$; thus $c(\boldsymbol{A})$ is minimised when $\boldsymbol{B} = \boldsymbol{0}$. The theorem is therefore proven.

B.1 Maximum Likelihood Estimation of Linear Regression Parameters under Heteroskedasticity

The likelihood function for the parameter vector $\boldsymbol{\theta}_1 = [\boldsymbol{\beta}', \boldsymbol{\omega}']'$ is given by (B.1).

$$\begin{aligned}
L_1(\boldsymbol{\theta}_1) &= (2\pi)^{-n/2}\det(\boldsymbol{\Omega})^{-1/2}\exp\left\{-\frac{1}{2}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})'\boldsymbol{\Omega}^{-1}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})\right\} \\
&= (2\pi)^{-n/2}\left(\prod_{i=1}^{n}\omega_i\right)^{-1/2}\exp\left\{-\frac{1}{2}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})'\boldsymbol{\Omega}^{-1}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})\right\}. \qquad \text{(B.1)}
\end{aligned}$$

Applying (A.2),

$$\frac{\partial}{\partial\boldsymbol{\beta}}\left[(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})'\boldsymbol{\Omega}^{-1}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})\right] = -\boldsymbol{X}'(\boldsymbol{\Omega}^{-1} + \boldsymbol{\Omega}^{-1})(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) = -2\boldsymbol{X}'\boldsymbol{\Omega}^{-1}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}).$$

Another useful matrix identity (Petersen and Pedersen 2012, p. 9) is,

$$\frac{\partial\boldsymbol{A}^{-1}}{\partial x} = -\boldsymbol{A}^{-1}\frac{\partial\boldsymbol{A}}{\partial x}\boldsymbol{A}^{-1}. \qquad \text{(B.2)}$$

From (B.2), it follows that

$$\frac{\partial\boldsymbol{\Omega}^{-1}}{\partial\omega_i} = -\boldsymbol{\Omega}^{-1}\frac{\partial\boldsymbol{\Omega}}{\partial\omega_i}\boldsymbol{\Omega}^{-1}.$$

This will be an $n \times n$ matrix with $-\omega_i^{-2}$ as its ith diagonal element and all other elements zero. Thus, take $\ell_1(\boldsymbol{\theta}_1) = \log L_1(\boldsymbol{\theta}_1)$ and differentiate with respect to the parameters as follows.

$$\begin{aligned}
\ell_1(\boldsymbol{\theta}_1) &= -\frac{n}{2}\log(2\pi) - \frac{1}{2}\sum_{i=1}^{n}\log\omega_i - \frac{1}{2}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})'\boldsymbol{\Omega}^{-1}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) \\
\frac{\partial\ell_1}{\partial\omega_i} &= -\frac{1}{2\omega_i} + \frac{1}{2}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})'\boldsymbol{\Omega}^{-1}\frac{\partial\boldsymbol{\Omega}}{\partial\omega_i}\boldsymbol{\Omega}^{-1}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) = 0 \\
\frac{1}{2\omega_i} &= \frac{(y_i - \boldsymbol{X}_{i.}'\boldsymbol{\beta})^2}{2\omega_i^2} \\
\omega_i &= (y_i - \boldsymbol{X}_{i.}'\boldsymbol{\beta})^2 \\
\frac{\partial\ell_1}{\partial\boldsymbol{\beta}} &= -\frac{1}{2}\left[-2\boldsymbol{X}'\boldsymbol{\Omega}^{-1}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})\right] = \boldsymbol{0}_{p\times 1} \\
\boldsymbol{X}'\boldsymbol{\Omega}^{-1}\boldsymbol{y} &= \boldsymbol{X}'\boldsymbol{\Omega}^{-1}\boldsymbol{X}\boldsymbol{\beta} \\
\boldsymbol{\beta} &= \left(\boldsymbol{X}'\boldsymbol{\Omega}^{-1}\boldsymbol{X}\right)^{-1}\boldsymbol{X}'\boldsymbol{\Omega}^{-1}\boldsymbol{y}.
\end{aligned}$$

This leads directly to (1.9).

B.2 Proof that the Ordinary Least Squares Residual Vector is a Best Linear Unbiased Predictor of the Random Error Vector

Here is given a proof that the OLS residual vector $\boldsymbol{e}$ is the Best Linear Unbiased Predictor (BLUP) of the random error vector $\boldsymbol{\epsilon}$.[132] Let $\hat{\boldsymbol{\epsilon}} = \boldsymbol{A}\boldsymbol{y}$ be an unbiased linear predictor of $\boldsymbol{\epsilon}$, where $\boldsymbol{A}$ is an $n \times n$ matrix that depends only on $\boldsymbol{X}$.[133] Since it follows from A1 that $\mathrm{E}(\boldsymbol{y}) = \boldsymbol{X}\boldsymbol{\beta}$, the unbiasedness property of $\hat{\boldsymbol{\epsilon}}$ ($\mathrm{E}(\hat{\boldsymbol{\epsilon}}) = \mathrm{E}(\boldsymbol{\epsilon}) = \boldsymbol{0}$) implies that $\boldsymbol{A}\boldsymbol{X} = \boldsymbol{0}$. This, in turn, implies that $\hat{\boldsymbol{\epsilon}} = \boldsymbol{A}\boldsymbol{\epsilon}$. Hence,

$$\mathrm{Cov}(\hat{\boldsymbol{\epsilon}}) = \mathrm{E}(\boldsymbol{A}\boldsymbol{\epsilon}\boldsymbol{\epsilon}'\boldsymbol{A}') = \omega\boldsymbol{A}\boldsymbol{A}'.$$

Suppose, without loss of generality, that $\boldsymbol{A} = \boldsymbol{M} + \boldsymbol{B}$. Then, since $\boldsymbol{A}\boldsymbol{X} = 0$ (by unbiasedness) and $\boldsymbol{M}\boldsymbol{X} = 0$ (from the definition of $\boldsymbol{M}$), it follows that $\boldsymbol{B}\boldsymbol{X} = 0$ as well. Furthermore, this implies that,

$$\boldsymbol{B}\boldsymbol{M} = \boldsymbol{B}(\boldsymbol{I}_n - \boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}') = \boldsymbol{B},$$

and similarly that $\boldsymbol{M}\boldsymbol{B}' = \boldsymbol{B}'$. Now, the BLUP of $\boldsymbol{\epsilon}$ will be that vector $\hat{\boldsymbol{\epsilon}}$ with associated matrix $\boldsymbol{A}$ that minimises the mean squared prediction error $\mathrm{E}\left[(\hat{\boldsymbol{\epsilon}} - \boldsymbol{\epsilon})'(\hat{\boldsymbol{\epsilon}} - \boldsymbol{\epsilon})\right]$. But,

$$\begin{aligned}
\mathrm{E}\left[(\hat{\boldsymbol{\epsilon}} - \boldsymbol{\epsilon})'(\hat{\boldsymbol{\epsilon}} - \boldsymbol{\epsilon})\right] &= \mathrm{E}\left[\hat{\boldsymbol{\epsilon}}'\hat{\boldsymbol{\epsilon}} - \boldsymbol{\epsilon}'\hat{\boldsymbol{\epsilon}} - \hat{\boldsymbol{\epsilon}}'\boldsymbol{\epsilon} + \boldsymbol{\epsilon}'\boldsymbol{\epsilon}\right] \\
&= \mathrm{E}\left[\boldsymbol{\epsilon}'\boldsymbol{A}'\boldsymbol{A}\boldsymbol{\epsilon} - 2\boldsymbol{\epsilon}'\boldsymbol{A}\boldsymbol{\epsilon} + \boldsymbol{\epsilon}'\boldsymbol{\epsilon}\right] \\
&= \omega\left[\mathrm{tr}(\boldsymbol{A}'\boldsymbol{A}) - 2\,\mathrm{tr}(\boldsymbol{A}) + n\right] \quad (\text{since } \mathrm{E}(\boldsymbol{\epsilon}) = 0).
\end{aligned}$$

Thus, the optimal choice of $\boldsymbol{A}$ will be that matrix that minimises $\mathrm{tr}(\boldsymbol{A}'\boldsymbol{A}) - 2\,\mathrm{tr}(\boldsymbol{A})$. Now, substituting $\boldsymbol{A} = \boldsymbol{M} + \boldsymbol{B}$ and using the properties of the trace operator (see Petersen and Pedersen 2012, p. 6),

$$\begin{aligned}
&\mathrm{tr}\left[(\boldsymbol{M} + \boldsymbol{B})'(\boldsymbol{M} + \boldsymbol{B})\right] - 2\,\mathrm{tr}(\boldsymbol{M} + \boldsymbol{B}) \\
&= \mathrm{tr}\left[\boldsymbol{B}'\boldsymbol{M} + \boldsymbol{M}\boldsymbol{B} + \boldsymbol{M} + \boldsymbol{B}'\boldsymbol{B}\right] - 2\,\mathrm{tr}(\boldsymbol{M} + \boldsymbol{B}) \text{ (by symmetry and idempotence of } \boldsymbol{M}) \\
&= \mathrm{tr}(\boldsymbol{B}'\boldsymbol{M}) + \mathrm{tr}(\boldsymbol{M}\boldsymbol{B}) + \mathrm{tr}(\boldsymbol{M}) + \mathrm{tr}(\boldsymbol{B}'\boldsymbol{B}) - 2\,\mathrm{tr}(\boldsymbol{M}) - 2\,\mathrm{tr}(\boldsymbol{B}) \\
&= \mathrm{tr}(\boldsymbol{M}\boldsymbol{B}') + \mathrm{tr}(\boldsymbol{B}\boldsymbol{M}) - 2\,\mathrm{tr}(\boldsymbol{B}) - \mathrm{tr}(\boldsymbol{M}) + \mathrm{tr}(\boldsymbol{B}'\boldsymbol{B}) \\
&= \mathrm{tr}(\boldsymbol{B}') + \mathrm{tr}(\boldsymbol{B}) - 2\,\mathrm{tr}(\boldsymbol{B}) - \mathrm{tr}(\boldsymbol{M}) + \mathrm{tr}(\boldsymbol{B}'\boldsymbol{B}) \\
&= -\,\mathrm{tr}(\boldsymbol{M}) + \sum_{i=1}^{n} b_{ii}^2,
\end{aligned}$$

where b_{ii} is the ith diagonal element of $\boldsymbol{B}$, $i = 1, 2, \ldots, n$. Since $\sum_{i=1}^{n} b_{ii}^2 \geq 0$, it follows that the mean squared error is minimised when $\boldsymbol{B} = \boldsymbol{0}_{n\times n}$, i.e. when $\boldsymbol{A} = \boldsymbol{M}$.[134] Thus, $\hat{\boldsymbol{\epsilon}} = \boldsymbol{M}\boldsymbol{y} = \boldsymbol{e}$ is the BLUP of $\boldsymbol{\epsilon}$.

B.3 Derivation of the Distributions of Certain Random Vectors pertaining to the Linear Regression Model under Classical Assumptions

$$\begin{aligned}
M_{\hat{\boldsymbol{\beta}}_{\mathrm{OLS}}}(\boldsymbol{t}) &= M_{\boldsymbol{y}}(\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{t}) \\
&= \exp\left\{\boldsymbol{\beta}'\boldsymbol{X}'\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{t} + \frac{\omega}{2}(\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{t})'\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{t}\right\} \\
&= \exp\left\{\boldsymbol{\beta}'\boldsymbol{t} + \frac{\omega}{2}\boldsymbol{t}'(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{t}\right\} \\
&= \exp\left\{\boldsymbol{\beta}'\boldsymbol{t} + \frac{1}{2}\boldsymbol{t}'\omega(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{t}\right\},
\end{aligned}$$

[132] For an alternative proof, using Lagrange multipliers, see (Theil 1965, p. 1069).

[133] Following Henderson (1975), a statistical prediction $\hat{\boldsymbol{\epsilon}}$ of the random variable $\boldsymbol{\epsilon}$ is said to be unbiased if $\mathrm{E}(\hat{\boldsymbol{\epsilon}}) = \mathrm{E}(\boldsymbol{\epsilon})$.

[134] Since $\mathrm{tr}(M) = n - p$ (see §3.1.1), the mean squared prediction error of the OLS residuals is in fact ωp. (Theil 1965, p. 1069) notes that, since $\mathrm{E}(\boldsymbol{\epsilon}'\boldsymbol{\epsilon}) = n\omega$, by dividing the mean squared prediction error by this mean squared model error, one obtains the 'average inaccuracy' of the predictions to be p/n.

from which it follows that, under A1-A5,

$$\hat{\boldsymbol{\beta}} \sim N(\boldsymbol{\beta}, \omega(\boldsymbol{X}'\boldsymbol{X})^{-1}).$$

Similarly, recalling (as stated in 1.1) that $\hat{\boldsymbol{y}} = \boldsymbol{H}\boldsymbol{y}$,

$$\begin{aligned}
M_{\hat{\boldsymbol{y}}}(\boldsymbol{t}) &= M_{\boldsymbol{y}}(\boldsymbol{H}'\boldsymbol{t}) \\
&= \exp\left\{\boldsymbol{\beta}'\boldsymbol{X}'\boldsymbol{H}'\boldsymbol{t} + \frac{\omega}{2}(\boldsymbol{H}'\boldsymbol{t})'(\boldsymbol{H}'\boldsymbol{t})\right\} \\
&= \exp\left\{\boldsymbol{\beta}'\boldsymbol{X}'\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{t} + \frac{\omega}{2}\boldsymbol{t}'\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{t}\right\} \\
&= \exp\left\{(\boldsymbol{X}\boldsymbol{\beta})'\boldsymbol{t} + \frac{1}{2}\boldsymbol{t}'\omega\boldsymbol{H}\boldsymbol{t}\right\},
\end{aligned}$$

from which it follows that

$$\hat{\boldsymbol{y}} \sim N(\boldsymbol{X}\boldsymbol{\beta}, \omega\boldsymbol{H}).$$

Finally, recalling that $\boldsymbol{e} = \boldsymbol{M}\boldsymbol{\epsilon}$,

$$\begin{aligned}
M_{\boldsymbol{e}}(\boldsymbol{t}) &= M_{\boldsymbol{\epsilon}}(\boldsymbol{M}\boldsymbol{t}) \text{ (since } \boldsymbol{M} \text{ is symmetric)} \\
&= \exp\left\{\boldsymbol{0}'\boldsymbol{M}'\boldsymbol{t} + \frac{\omega}{2}(\boldsymbol{M}'\boldsymbol{t})'(\boldsymbol{M}'\boldsymbol{t})\right\} \\
&= \exp\left\{\boldsymbol{0}'\boldsymbol{t} + \frac{\omega}{2}\boldsymbol{t}'\boldsymbol{M}'\boldsymbol{M}\boldsymbol{t}\right\} \\
&= \exp\left\{\boldsymbol{0}'\boldsymbol{t} + \frac{1}{2}\boldsymbol{t}'\omega\boldsymbol{M}\boldsymbol{t}\right\} \text{ (using the symmetry and idempotence properties of } \boldsymbol{M}\text{)},
\end{aligned}$$

from which it follows that

$$\boldsymbol{e} \sim N(\boldsymbol{0}, \omega\boldsymbol{M}).$$

B.4 Proofs of Two Results Necessary for Construction of Exact t-Tests for Inference on Linear Regression Parameters under the Classical Assumptions

It is first proven that, under A1-A5, $\omega^{-1}\boldsymbol{e}'\boldsymbol{e} \sim \chi^2(n-p)$. This result is an instance of the following result (B.3) with $\boldsymbol{u} = \omega^{-1/2}\boldsymbol{\epsilon}$, $\boldsymbol{A} = \boldsymbol{M}$, and $r = n - p$. For any n-vector $\boldsymbol{u}$ of iid standard normal random variables and any symmetric, idempotent $n \times n$ matrix $\boldsymbol{A}$ with $r = \text{rank}(\boldsymbol{A}) = \text{tr}(\boldsymbol{A})$,[135] it follows that

$$\boldsymbol{u}'\boldsymbol{A}\boldsymbol{u} \sim \chi^2(r). \tag{B.3}$$

A proof of (B.3) is sketched as follows (see Heij et al. 2004). An idempotent $n \times n$ matrix of rank r has r unit eigenvalues and $n - r$ zero eigenvalues. By symmetry and idempotence of $\boldsymbol{A}$, it has singular value decomposition $\boldsymbol{V}\boldsymbol{D}\boldsymbol{V}'$, where the columns of $\boldsymbol{V}$ are the eigenvectors of $\boldsymbol{A}$ and $\boldsymbol{D}$ is a diagonal matrix with the eigenvalues of $\boldsymbol{A}$ on its diagonal. Let $\boldsymbol{V}_1$ be $\boldsymbol{V}$ with the $n - r$ columns corresponding to zero eigenvalues of $\boldsymbol{A}$ removed, and $\boldsymbol{D}_1$ be the $r \times r$ identity matrix formed by removing the rows and columns with zero eigenvalues on the diagonal. $\boldsymbol{V}_1$ is then an $n \times r$ matrix with orthonormal columns. Since the deleted portions of $\boldsymbol{V}$ and $\boldsymbol{D}$ contribute nothing to $\boldsymbol{A}$, it follows that one can write $\boldsymbol{A} = \boldsymbol{V}_1\boldsymbol{D}_1\boldsymbol{V}_1' = \boldsymbol{V}_1\boldsymbol{V}_1'$. It can be shown (e.g., using MGFs as in (1.28)) that $\boldsymbol{V}_1'\boldsymbol{u} \sim N(\boldsymbol{0}, \boldsymbol{V}_1'\boldsymbol{V}_1)$, but since $\boldsymbol{V}_1$ has orthonormal columns, $\boldsymbol{V}_1'\boldsymbol{V}_1 = \boldsymbol{I}_n$. Thus, $\boldsymbol{u}'\boldsymbol{A}\boldsymbol{u} = \boldsymbol{u}'\boldsymbol{V}_1\boldsymbol{V}_1'\boldsymbol{u} = (\boldsymbol{V}_1'\boldsymbol{u})'(\boldsymbol{V}_1'\boldsymbol{u})$ is the sum of squares of r independent standard normal random variables, and thus has a chi-square distribution with r degrees of freedom.

The independence of $\hat{\boldsymbol{\beta}}$ and $\boldsymbol{e}$ follows from the property that $\text{Cov}(\hat{\boldsymbol{\beta}}, \boldsymbol{e}) = \boldsymbol{0}$, together with the property that normally distributed random vectors are independent if and only if they are uncorrelated. The covariance of $\hat{\boldsymbol{\beta}}$ and $\boldsymbol{e}$ can be derived as follows.

[135] Idempotent matrices have the property that their rank equals their trace (Petersen and Pedersen 2012).

$$
\begin{aligned}
\operatorname{Cov}(\hat{\boldsymbol{\beta}}, \boldsymbol{e}) &= \mathrm{E}\left[(\hat{\boldsymbol{\beta}} - \mathrm{E}(\hat{\boldsymbol{\beta}}))(\boldsymbol{e} - \mathrm{E}(\boldsymbol{e}))'\right] \\
&= \mathrm{E}\left[(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})\boldsymbol{e}'\right] = \mathrm{E}\left[\left((\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{\epsilon}\right)(\boldsymbol{M}\boldsymbol{\epsilon})'\right] \\
&= \mathrm{E}\left[(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{\epsilon}\boldsymbol{\epsilon}'(\boldsymbol{I}_n - \boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}')\right] \\
&= (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\,\mathrm{E}(\boldsymbol{\epsilon}\boldsymbol{\epsilon}') - (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\,\mathrm{E}(\boldsymbol{\epsilon}\boldsymbol{\epsilon}')\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}' \\
&= \omega(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}' - \omega(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}' = \boldsymbol{0} \text{ (by A1-A4).}
\end{aligned}
$$

B.5 Derivation of the Distributions of Certain Random Vectors pertaining to the Linear Regression Model under Heteroskedasticity

$$
\begin{aligned}
M_{\hat{\boldsymbol{\beta}}_{\mathrm{OLS}}}(\boldsymbol{t}) &= M_{\boldsymbol{y}}(\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{t}) \\
&= \exp\left\{\boldsymbol{\beta}'\boldsymbol{X}'\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{t} + \frac{1}{2}\left(\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{t}\right)'\boldsymbol{\Omega}\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{t}\right\} \\
&= \exp\left\{\boldsymbol{\beta}'\boldsymbol{t} + \frac{1}{2}\boldsymbol{t}'(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{\Omega}\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{t}\right\},
\end{aligned}
$$

which implies that

$$\hat{\boldsymbol{\beta}}_{\mathrm{OLS}} \sim N(\boldsymbol{\beta}, (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{\Omega}\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}).$$

This result aligns with those stated in §1.1.6.1. Then,

$$
\begin{aligned}
M_{\hat{\boldsymbol{y}}}(\boldsymbol{t}) &= M_{\boldsymbol{y}}(\boldsymbol{H}\boldsymbol{t}) \\
&= \exp\left\{\boldsymbol{\beta}'\boldsymbol{X}'\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{t} + \frac{1}{2}(\boldsymbol{H}\boldsymbol{t})'\boldsymbol{\Omega}\boldsymbol{H}\boldsymbol{t}\right\} \\
&= \exp\left\{\boldsymbol{\beta}'\boldsymbol{X}'\boldsymbol{t} + \frac{1}{2}\boldsymbol{t}'\boldsymbol{H}\boldsymbol{\Omega}\boldsymbol{H}\boldsymbol{t}\right\},
\end{aligned}
$$

implying that

$$\hat{\boldsymbol{y}} \sim N(\boldsymbol{X}\boldsymbol{\beta}, \boldsymbol{H}\boldsymbol{\Omega}\boldsymbol{H}).$$

Furthermore,

$$
\begin{aligned}
M_{\boldsymbol{e}}(\boldsymbol{t}) &= M_{\boldsymbol{\epsilon}}(\boldsymbol{M}\boldsymbol{t}) \\
&= \exp\left\{\boldsymbol{0}'\boldsymbol{M}\boldsymbol{t} + \frac{1}{2}(\boldsymbol{M}\boldsymbol{t})'\boldsymbol{\Omega}\boldsymbol{M}\boldsymbol{t}\right\} \\
&= \exp\left\{\boldsymbol{0}'\boldsymbol{t} + \frac{1}{2}\boldsymbol{t}'\boldsymbol{M}\boldsymbol{\Omega}\boldsymbol{M}\boldsymbol{t}\right\},
\end{aligned}
$$

implying that

$$\boldsymbol{e} \sim N(\boldsymbol{0}, \boldsymbol{M}\boldsymbol{\Omega}\boldsymbol{M}).$$

C Some Theoretical Results pertaining to the Methodology Chapter

C.1 Further Theoretical Results on Squared Ordinary Least Squares Residuals

C.1.1 Derivation of Covariances of Squared Ordinary Least Squares Residuals

The following is a derivation of the variance-covariance matrix of the squared OLS residuals under assumptions A1-A5 as given in (3.3). The derivation under heteroskedasticity (where A2 is relaxed) is not shown but follows the same steps. Working from the definition of a variance-covariance matrix,

$$\begin{aligned}\operatorname{Cov}(\boldsymbol{e}\circ\boldsymbol{e}) &= \operatorname{E}\left[(\boldsymbol{e}\circ\boldsymbol{e}-\omega\operatorname{diag}(\boldsymbol{M}))\,(\boldsymbol{e}\circ\boldsymbol{e}-\omega\operatorname{diag}(\boldsymbol{M}))'\right]\\ &= \operatorname{E}\left[(\boldsymbol{e}\circ\boldsymbol{e})(\boldsymbol{e}\circ\boldsymbol{e})'\right]-\omega^2\operatorname{diag}(\boldsymbol{M})(\operatorname{diag}(\boldsymbol{M}))'.\end{aligned}$$

Now, the (i,j)th element of the $n\times n$ matrix $\omega^2\operatorname{diag}(\boldsymbol{M})(\operatorname{diag}(\boldsymbol{M}))'$ is $\omega^2 m_{ii}m_{jj}$ (consequently the ith diagonal element is $\omega^2 m_{ii}^2$). It remains to find the elements of $\operatorname{E}[(\boldsymbol{e}\circ\boldsymbol{e})(\boldsymbol{e}\circ\boldsymbol{e})']$. Now, $\boldsymbol{e}\circ\boldsymbol{e}=\operatorname{diag}(\boldsymbol{e}\boldsymbol{e}')=\operatorname{diag}(\boldsymbol{M}\boldsymbol{\epsilon}\boldsymbol{\epsilon}'\boldsymbol{M})$, and therefore it is necessary to find $\operatorname{E}[\operatorname{diag}(\boldsymbol{M}\boldsymbol{\epsilon}\boldsymbol{\epsilon}'\boldsymbol{M})\operatorname{diag}(\boldsymbol{M}\boldsymbol{\epsilon}\boldsymbol{\epsilon}'\boldsymbol{M})']$. The diagonal and off-diagonal elements of this matrix and their expectations are considered separately.

First, the ith diagonal element of $\operatorname{diag}(\boldsymbol{M}\boldsymbol{\epsilon}\boldsymbol{\epsilon}'\boldsymbol{M})\operatorname{diag}(\boldsymbol{M}\boldsymbol{\epsilon}\boldsymbol{\epsilon}'\boldsymbol{M})'$ is given by,

$$\begin{aligned}&\left(\sum_k m_{ik}^2\epsilon_k^2+2\sum_{k<\ell}\sum m_{ik}m_{i\ell}\epsilon_k\epsilon_\ell\right)^2\\ &=\underbrace{\sum_k m_{ik}^2\epsilon_k^2\sum_\ell m_{i\ell}^2\epsilon_\ell^2}_{\text{Term 1}}+\underbrace{4\left(\sum_k m_{ik}^2\epsilon_k^2\right)\left(\sum_{p<q}\sum m_{ip}m_{iq}\epsilon_p\epsilon_q\right)}_{\text{Term 2}}\\ &\quad+\underbrace{4\sum_{k<\ell}\sum m_{ik}m_{i\ell}\epsilon_k\epsilon_\ell\sum_{p<q}\sum m_{ip}m_{iq}\epsilon_p\epsilon_q}_{\text{Term 3}}.\end{aligned}$$

Taking expectation, observe that under A1-A5, $\operatorname{E}(\epsilon_i)=\operatorname{E}(\epsilon_i^3)=0$, $\operatorname{E}(\epsilon_i^2)=\omega$, and $\operatorname{E}(\epsilon_i^4)=3\omega^2$. Moreover, by A3, any product moment $\operatorname{E}(\epsilon_i^r\epsilon_j^s)$ will be 0 if at least one of r or s is odd, for $r,s\in\{1,2,3,4\}$. From this, it follows that Term 2 above has conditional expectation 0.

The expectation of Term 1 is

$$\begin{aligned}&\operatorname{E}\left[\sum_k m_{ik}^2\epsilon_k^2\sum_\ell m_{i\ell}^2\epsilon_\ell^2\right]\\ &=\sum_k m_{ik}^4\operatorname{E}(\epsilon_k^4)+2\sum_{k<\ell}\sum m_{ik}^2m_{i\ell}^2\underbrace{\operatorname{E}(\epsilon_k^2\epsilon_\ell^2)}_{=\operatorname{E}(\epsilon_k^2)\operatorname{E}(\epsilon_\ell^2)\text{ by independence}}\\ &=3\omega^2\sum_k m_{ik}^4+2\omega^2\sum_{k<\ell}\sum m_{ik}^2m_{i\ell}^2.\end{aligned}$$

The expectation of Term 3 is

$$\begin{aligned}&\operatorname{E}\left[4\sum_{k<\ell}\sum m_{ik}m_{i\ell}\epsilon_k\epsilon_\ell\sum_{p<q}\sum m_{ip}m_{iq}\epsilon_p\epsilon_q\right]\\ &=4\sum_{k<\ell}\sum m_{ik}^2m_{i\ell}^2\operatorname{E}(\epsilon_k^2)\operatorname{E}(\epsilon_\ell^2)+\text{ cross-terms with conditional expectation 0}\\ &=4\omega^2\sum_{k<\ell}\sum m_{ik}^2m_{i\ell}^2.\end{aligned}$$

Combining the above with earlier observations, the ith diagonal element of $\mathrm{Cov}(\boldsymbol{e} \circ \boldsymbol{e})$ is

$$\begin{aligned}
&3\omega^2 \sum_k m_{ik}^4 + 6\omega^2 \sum\sum_{k<\ell} m_{ik}^2 m_{i\ell}^2 - \omega^2 m_{ii}^2 \\
&= 3\omega^2 \left(\sum_k m_{ik}^4 + 2 \sum\sum_{k<\ell} m_{ik}^2 m_{i\ell}^2 \right) - \omega^2 m_{ii}^2. \\
\text{But } &\sum_k m_{ik}^4 + 2 \sum\sum_{k<\ell} m_{ik}^2 m_{i\ell}^2 = \left(\sum_k m_{ik}^2 \right) \left(\sum_\ell m_{i\ell}^2 \right) = m_{ii}^2 \\
&\therefore \; 3\omega^2 m_{ii}^2 - \omega^2 m_{ii}^2 \\
&= 2\omega^2 m_{ii}^2. \qquad \text{(C.1)}
\end{aligned}$$

Now, consider the off-diagonal elements of $\mathrm{Cov}(\boldsymbol{e} \circ \boldsymbol{e})$. The (i,j)th element of $\mathrm{diag}(\boldsymbol{M\epsilon\epsilon' M})\,\mathrm{diag}(\boldsymbol{M\epsilon\epsilon' M})'$, $i \neq j$, is given by,

$$\begin{aligned}
&\left(\sum_k m_{ik}^2 \epsilon_k^2 + 2 \sum\sum_{k<\ell} m_{ik} m_{i\ell} \epsilon_k \epsilon_\ell \right) \left(\sum_k m_{jk}^2 \epsilon_k^2 + 2 \sum\sum_{k<\ell} m_{jk} m_{j\ell} \epsilon_k \epsilon_\ell \right) \\
&= \underbrace{\sum_k m_{ik}^2 \epsilon_k^2 \sum_\ell m_{j\ell}^2 \epsilon_\ell^2}_{\text{Term 1}} + \underbrace{2 \left(\sum_k m_{ik}^2 \epsilon_k^2 \right) \left(\sum\sum_{p<q} m_{ip} m_{iq} \epsilon_p \epsilon_q \right)}_{\text{Term 2A}} \\
&+ \underbrace{2 \left(\sum_k m_{jk}^2 \epsilon_k^2 \right) \left(\sum\sum_{p<q} m_{jp} m_{jq} \epsilon_p \epsilon_q \right)}_{\text{Term 2B}} + \underbrace{4 \left(\sum\sum_{k<\ell} m_{ik} m_{i\ell} \epsilon_k \epsilon_\ell \right) \left(\sum\sum_{p<q} m_{jp} m_{jq} \epsilon_p \epsilon_q \right)}_{\text{Term 3}}
\end{aligned}$$

Terms 2A and 2B have expectation 0 since all terms of these expressions contain an odd power of the disturbance. The expectation of Term 1 is

$$\begin{aligned}
&\mathrm{E}\left[\sum_k m_{ik}^2 \epsilon_k^2 \sum_\ell m_{j\ell}^2 \epsilon_\ell^2 \right] \\
&= \sum_k m_{ik}^2 m_{jk}^2 \,\mathrm{E}(\epsilon_k^4) + 2 \sum\sum_{k<\ell} m_{ik}^2 m_{j\ell}^2 \,\mathrm{E}(\epsilon_k^2)\,\mathrm{E}(\epsilon_\ell^2) \\
&= 3\omega^2 \sum_k m_{ik}^2 m_{kj}^2 + 2\omega^2 \sum\sum_{k<\ell} m_{ik}^2 m_{j\ell}^2.
\end{aligned}$$

The expectation of Term 3 is

$$\begin{aligned}
&\mathrm{E}\left[4 \left(\sum\sum_{k<\ell} m_{ik} m_{i\ell} \epsilon_k \epsilon_\ell \right) \left(\sum\sum_{p<q} m_{jp} m_{jq} \epsilon_p \epsilon_q \right) \right] \\
&= 4 \sum\sum_{k<\ell} m_{ik} m_{i\ell} m_{jk} m_{j\ell} \,\mathrm{E}(\epsilon_k^2)\,\mathrm{E}(\epsilon_\ell^2) + \text{ cross-terms with expectation 0} \\
&= 4\omega^2 \sum\sum_{k<\ell} m_{ik} m_{i\ell} m_{jk} m_{j\ell}.
\end{aligned}$$

Hence, combining these results, the (i,j)th element of $\mathrm{Cov}(\boldsymbol{e} \circ \boldsymbol{e})$ is

$$
\begin{aligned}
&3\omega^2 \sum_k m_{ik}^2 m_{kj}^2 + 2\omega^2 \sum\sum_{k<\ell} m_{ik}^2 m_{j\ell}^2 + 4\omega^2 \sum\sum_{k<\ell} m_{ik} m_{i\ell} m_{jk} m_{j\ell} - \omega^2 m_{ii} m_{jj} \\
&= \omega^2 \left(\sum_k m_{ik}^2 m_{kj}^2 + 2 \sum\sum_{k<\ell} m_{ik}^2 m_{j\ell}^2 \right) \\
&+ 2\omega^2 \left(\sum_k m_{ik}^2 m_{kj}^2 + 2 \sum\sum_{k<\ell} m_{ik} m_{i\ell} m_{jk} m_{j\ell} \right) - \omega^2 m_{ii} m_{jj}. \\
&\text{But } \sum_k m_{ik}^2 m_{kj}^2 + 2 \sum\sum_{k<\ell} m_{ik}^2 m_{j\ell}^2 = \left(\sum_k m_{ik}^2 \right) \left(\sum_\ell m_{j\ell}^2 \right) = m_{ii} m_{jj}, \text{ and} \\
&\sum_k m_{ik}^2 m_{kj}^2 + 2 \sum\sum_{k<\ell} m_{ik} m_{i\ell} m_{jk} m_{j\ell} = \left(\sum_k m_{ik} m_{kj} \right) \left(\sum_\ell m_{i\ell} m_{\ell j} \right) = m_{ij}^2. \\
&\therefore\ \omega^2 m_{ii} m_{jj} + 2\omega^2 m_{ij}^2 - \omega^2 m_{ii} m_{jj} = 2\omega^2 m_{ij}^2.
\end{aligned}
$$

Thus, the covariance of any two squared OLS residuals e_i^2, e_j^2 can be written as

$$\operatorname{Cov}(e_i^2, e_j^2) = 2\omega^2 m_{ij}^2,$$

and (3.3) is proven. This leads directly to an expression for the variance-covariance matrix of the squared OLS residual vector,

$$\operatorname{Cov}(\boldsymbol{e} \circ \boldsymbol{e}) = 2\omega^2 (\boldsymbol{M} \circ \boldsymbol{M}).$$

C.1.2 Some Properties of the Mean Squared Error of the Squared Ordinary Least Squares Residuals

This is a brief extension of the discussion in §3.1.4 of the bias properties of the OLS residuals, taken as estimators of the error variances ω_i, under heteroskedasticity.

From (3.11), the mean squared error of the e_i^2 as estimators of the error variances is given by (C.2).

$$
\begin{aligned}
\operatorname{MSE}(e_i^2) &= \operatorname{Var}(e_i^2) + \left[\operatorname{Bias}(e_i^2)\right]^2 \\
&= 2\left[\sum_{k=1}^{n} \omega_k m_{ik}^2\right]^2 + \left[\sum_{k=1}^{n} \omega_k m_{ik}^2 - \omega_i\right]^2 \\
&= 3\left[\sum_{k=1}^{n} \omega_k m_{ik}^2\right]^2 - 2\omega_i \sum_{k=1}^{n} \omega_k m_{ik}^2 + \omega_i^2 \\
&= 3\left[\sum_{k=1}^{n} \omega_i (1-h_{ii})^2 + \sum_{k\neq i} \omega_k h_{ik}^2\right]^2 - 2\omega_i \left[\omega_i (1-h_{ii})^2 + \sum_{k\neq i} \omega_k h_{ik}^2\right] + \omega_i^2. \qquad \text{(C.2)}
\end{aligned}
$$

It is easy to show that

$$\frac{\partial}{\partial h_{ii}} \operatorname{MSE}(e_i^2) = 4\omega_i (1-h_{ii}) \left[-3\sum_{k\neq i} \omega_k h_{ik}^2 - \omega_i \left(3(1-h_{ii})^2 - 1\right)\right]. \qquad \text{(C.3)}$$

Since $0 \leq h_{ii} \leq 1$, and since $3(1-h_{ii})^2 - 1 > 0$ for all $h_{ii} < 1 - \dfrac{\sqrt{3}}{3} \approx 0.423$, it follows that the MSE strictly decreases with leverage up to about $h_{ii} = 0.423$. Thereafter, it is theoretically possible that the derivative could become positive if the (now-positive) $-\omega_i \left(3(1-h_{ii})^2 - 1\right)$ term dominates the $-3\sum_{k\neq i} \omega_k h_{ik}^2$ term.

However, 0.423 is already a very high leverage value, especially if n is large, so it remains true under heteroskedasticity that the OLS squared residuals are better estimators of the corresponding error variances for high-leverage points than for low-leverage points.

C.2 An Auxiliary Nonlinear Variance Model Built on Logarithms of Squared Ordinary Least Squares Residuals

Taking the logarithm of the response is a widely used technique in regression modelling. In this appendix, an alternative approach to constructing an ANLVM based on the logarithms of the squared OLS residuals is presented.

C.2.1 Logarithms of Squared Ordinary Least Squares Residuals under Homoskedasticity

Consider the natural logarithms of the squared OLS residuals, $\log e_i^2$, under A1-A5. Deriving exact analytical results on the moments is very difficult, but by making use of a second-order Taylor expansion about $\mathrm{E}(e_i^2)$, one obtains

$$\mathrm{E}\left(\log e_i^2\right) \approx \log\left(\omega m_{ii}\right) - 1, \tag{C.4}$$

$$\mathrm{Var}\left(\log e_i^2\right) \approx 1, \text{ and} \tag{C.5}$$

$$\begin{aligned}\mathrm{Cov}\left(\log e_i^2, \log e_j^2\right) &\approx \frac{2m_{ij}^2}{m_{ii}m_{jj}} - 1 \\ &= 2\,\mathrm{Corr}(e_i^2, e_j^2) - 1.\end{aligned} \tag{C.6}$$

Proof:

Let U and V be random variables with expectations μ_U and μ_V, respectively, and let $f : \mathbb{R} \to \mathbb{R}$ be a twice-differentiable, real-valued function. First, derive the second-order Taylor series approximation of $\mathrm{E}\left[f(U)\right]$ about μ_U:

$$\begin{aligned}\mathrm{E}\left[f(U)\right] &\approx \mathrm{E}\left[f(\mu_U) + f'(\mu_U)(U-\mu_U) + \frac{f''(\mu_U)}{2!}(U-\mu_U)^2\right] \\ &= f(\mu_U) + f'(\mu_U)\left(\mathrm{E}(U) - \mu_U\right) + \frac{f''(\mu_U)}{2}\mathrm{Var}(U) \\ &= f(\mu_U) + \frac{1}{2}f''(\mu_U)\,\mathrm{Var}(U).\end{aligned}$$

Then, since $\mathrm{E}(U) = \mu_U$, substituting $U = e_i^2$ and $f(U) = \log U$ yields

$$\mathrm{E}\left[\log e_i^2\right] \approx \log\left[\mathrm{E}(e_i^2)\right] - \frac{\mathrm{Var}(e_i^2)}{2\left[\mathrm{E}(e_i^2)\right]^2}.$$

Now, substituting in the results from (1.11) and (C.1),

$$\begin{aligned}\mathrm{E}\left(\log e_i^2\right) &\approx \log\left(\omega m_{ii}\right) - \frac{2\omega^2 m_{ii}^2}{2\left(\omega m_{ii}\right)^2} \\ &= \log\left(\omega m_{ii}\right) - 1.\end{aligned}$$

Thus, (C.4) is proven.[136] Next, derive the second-order Taylor series approximation of $\mathrm{E}\left[g(U,V)\right]$ about

[136] A third-order Taylor expansion for $\mathrm{E}\left(\log e_i^2\right)$ can be achieved by adding the term $\frac{f^{(3)}(\mu_U)}{6}\mathrm{E}\left[(U-\mu_U)^3\right]$, which in this case is $\frac{1}{3}\mathrm{E}\left(e_i^2\right)^{-3}\mathrm{E}\left[\left(e_i^2 - \mathrm{E}(e_i^2)\right)^3\right]$. Under the assumptions A1-A5, using the fact that a normally distributed random variable X with mean a and variance b^2 has $\mathrm{E}(X^6) = a^6 + 15a^4b^2 + 45a^2b^4 + 15b^6$, result (1.31) implies that this term reduces to $\frac{8}{3}$.

(μ_U, μ_V), where $g(U,V) = f(U)f(V)$. Noting that $g_U(U,V) = f'(U)f(V)$ and $g_V(U,V) = f(U)f'(V)$, proceed as follows:

$$\begin{aligned}
\mathrm{E}\left[g(U,V)\right] \approx \mathrm{E}\Big[& f(\mu_U)f(\mu_V) + (U-\mu_U)f'(\mu_U)f(\mu_V) + (V-\mu_V)f(\mu_U)f'(\mu_V) \\
& + \frac{1}{2}\Big((U-\mu_U)^2 f''(\mu_U)f(\mu_V) + 2(U-\mu_U)(V-\mu_V)f'(\mu_U)f'(\mu_V) + (V-\mu_V)^2 f(\mu_U)f''(\mu_V)\Big)\Big] \\
& = f(\mu_U)f(\mu_V) + \frac{1}{2}f''(\mu_U)f(\mu_V)\operatorname{Var}(U) + f'(\mu_U)f'(\mu_V)\operatorname{Cov}(U,V) + \frac{1}{2}f(\mu_U)f''(\mu_V)\operatorname{Var}(V).
\end{aligned}$$

Therefore,

$$\begin{aligned}
\operatorname{Cov}\left[f(U), f(V)\right] &= \mathrm{E}\left[g(U,V)\right] - \mathrm{E}\left[f(U)\right]\mathrm{E}\left[f(V)\right] \\
&= f(\mu_U)f(\mu_V) + f'(\mu_U)f'(\mu_V)\operatorname{Cov}(U,V) + \frac{1}{2}f''(\mu_U)f(\mu_V)\operatorname{Var}(U) \\
&+ \frac{1}{2}f(\mu_U)f''(\mu_V)\operatorname{Var}(V) - \left[f(\mu_U) + \frac{1}{2}f''(\mu_U)\operatorname{Var}(U)\right]\left[f(\mu_V) + \frac{1}{2}f''(\mu_V)\operatorname{Var}(V)\right] \\
&= f(\mu_U)f(\mu_V) + f'(\mu_U)f'(\mu_V)\operatorname{Cov}(U,V) + \frac{1}{2}f''(\mu_U)f(\mu_V)\operatorname{Var}(U) \\
&+ \frac{1}{2}f(\mu_U)f''(\mu_V)\operatorname{Var}(V) - f(\mu_U)f(\mu_V) - \frac{1}{2}f''(\mu_U)f(\mu_V)\operatorname{Var}(U) \\
&- \frac{1}{2}f(\mu_U)f''(\mu_V)\operatorname{Var}(V) - \frac{1}{4}f''(\mu_U)f''(\mu_V)\operatorname{Var}(U)\operatorname{Var}(V) \\
&= f'(\mu_U)f'(\mu_V)\operatorname{Cov}(U,V) - \frac{1}{4}f''(\mu_U)f''(\mu_V)\operatorname{Var}(U)\operatorname{Var}(V).
\end{aligned}$$

Then, substituting for U and $f(\cdot)$ as before, along with $V = e_j^2$, it follows that

$$\operatorname{Cov}\left(\log e_i^2, \log e_j^2\right) = \frac{\operatorname{Cov}(e_i^2, e_j^2)}{\mathrm{E}(e_i^2)\,\mathrm{E}(e_j^2)} - \frac{\operatorname{Var}(e_i^2)\operatorname{Var}(e_j^2)}{4\left[\mathrm{E}(e_i^2)\right]^2\left[\mathrm{E}(e_j^2)\right]^2}.$$

Again, substituting the results from (1.11), (C.1), (3.3), and (3.5),

$$\begin{aligned}
\operatorname{Cov}(\log e_i^2, \log e_j^2) &\approx \frac{2\omega^2 m_{ij}^2}{\omega m_{ii}\omega m_{jj}} - \frac{\left(2\omega^2 m_{ii}^2\right)\left(2\omega^2 m_{jj}^2\right)}{4\left[\omega m_{ii}\right]^2\left[\omega m_{jj}^2\right]^2} \\
&= \frac{2m_{ij}^2}{m_{ii}m_{jj}} - 1 \\
&= 2\operatorname{Corr}(e_i^2, e_j^2) - 1,
\end{aligned}$$

which proves (C.6). Letting $j = i$ in the above expression, it is obvious that the expression reduces to 1, thus proving (C.5).

In matrix notation, the results (C.4), (C.5), and (C.6) can be expressed thus:

$$\mathrm{E}\left(\log e_i^2\right) \approx \log\left[\omega\operatorname{diag}(\boldsymbol{M})\right] - \mathbf{1}_n, \text{ and} \tag{C.7}$$

$$\operatorname{Cov}\left(\log e_i^2, \log e_j^2\right) \approx 2\boldsymbol{M}_{\text{diag}}^{-1}\boldsymbol{M} \circ \boldsymbol{M}\boldsymbol{M}_{\text{diag}}^{-1} - \mathbf{1}_{n\times n}, \tag{C.8}$$

where log is applied to a vector elementwise, $\boldsymbol{M}_{\text{diag}}$ is a diagonal $n \times n$ matrix with $\operatorname{diag}(\boldsymbol{M})$ as its diagonal, $\mathbf{1}_{n\times n}$ is an $n \times n$ unit matrix, and $\mathbf{1}_n$ is a unit n-vector.

C.2.2 Logarithms of Squared Ordinary Least Squares Residuals under Heteroskedasticity

Proceeding as in §C.2.1, but under heteroskedasticity, one obtains

$$\mathrm{E}\left(\log e_i^2\right) \approx \log\left\{\sum_{k=1}^{n} \omega_k m_{ik}^2\right\} - 1, \tag{C.9}$$

$$\mathrm{Var}\left(\log e_i^2\right) \approx 1, \text{ and} \tag{C.10}$$

$$\begin{aligned} \mathrm{Cov}\left(\log e_i^2, \log e_j^2\right) &\approx \frac{2\left(\displaystyle\sum_{k=1}^{n} \omega_k m_{ik} m_{jk}\right)^2}{\displaystyle\sum_{k=1}^{n} \omega_k m_{ik}^2 \sum_{\ell=1}^{n} \omega_\ell m_{j\ell}^2} - 1 \qquad (\mathrm{C.11}) \\ &= 2\,\mathrm{Corr}(e_i^2, e_j^2) - 1. \end{aligned}$$

Proof:

Using the Taylor series derivation for $\mathrm{E}\left(\log e_i^2\right)$ from the previous section, and substituting in the results from (1.15) and (3.12),

$$\begin{aligned} \mathrm{E}\left(\log e_i^2\right) &\approx \log\left\{\sum_{k=1}^{n} \omega_k m_{ik}^2\right\} - \frac{2\left(\displaystyle\sum_{k=1}^{n} \omega_k m_{ik}^2\right)^2}{2\left[\displaystyle\sum_{k=1}^{n} \omega_k m_{ik}^2\right]^2} \\ &= \log\left\{\sum_{k=1}^{n} \omega_k m_{ik}^2\right\} - 1. \end{aligned}$$

Thus, (C.9) is proven. Next, using the Taylor series derivation for $\mathrm{Cov}\left(\log e_i^2, \log e_j^2\right)$ from the previous section, and substituting the results from (1.15), (3.12), (3.13), and (C.10),

$$\begin{aligned} \mathrm{Cov}(\log e_i^2, \log e_j^2) &\approx \frac{2\left(\displaystyle\sum_{k=1}^{n} \omega_k m_{ik} m_{jk}\right)^2}{\displaystyle\sum_{k=1}^{n} \omega_k m_{ik}^2 \sum_{\ell=1}^{n} \omega_\ell m_{j\ell}^2} - \frac{4\left(\displaystyle\sum_{k=1}^{n} \omega_k m_{ik}^2\right)^2 \left(\displaystyle\sum_{\ell=1}^{n} \omega_\ell m_{j\ell}^2\right)^2}{4\left[\displaystyle\sum_{k=1}^{n} \omega_k m_{ik}^2\right]^2 \left[\displaystyle\sum_{\ell=1}^{n} \omega_\ell m_{j\ell}^2\right]^2} \\ &= \frac{2\left(\displaystyle\sum_{k=1}^{n} \omega_k m_{ik} m_{jk}\right)^2}{\displaystyle\sum_{k=1}^{n} \omega_k m_{ik}^2 \sum_{\ell=1}^{n} \omega_\ell m_{j\ell}^2} - 1, \end{aligned}$$

which proves (C.11). Letting $j = i$ in the above expression, it is obvious that the expression reduces to 1, thus proving (C.10).

In matrix notation, the results (C.9), (C.10), and (C.11) can be expressed thus:

$$\mathrm{E}\left(\log e_i^2\right) \approx \log\left[\mathrm{diag}(\boldsymbol{M\Omega M})\right] - \mathbf{1}_n, \text{ and} \tag{C.12}$$

$$\mathrm{Cov}\left(\log e_i^2, \log e_j^2\right) \approx 2\left(\boldsymbol{M\Omega M}\right)_{\mathrm{diag}}^{-1}\left[(\boldsymbol{M\Omega M}) \circ (\boldsymbol{M\Omega M})\right]\left(\boldsymbol{M\Omega M}\right)_{\mathrm{diag}}^{-1} - \mathbf{1}_{n\times n}, \tag{C.13}$$

where $\left(\boldsymbol{M\Omega M}\right)_{\mathrm{diag}}$ is a diagonal $n \times n$ matrix with $\mathrm{diag}(\boldsymbol{M\Omega M})$ as its diagonal.

C.2.3 Constructing an Auxiliary Nonlinear Variance Model Based on Taylor Series Approximations

The approximate expectation and variance-covariance matrix of the logs of the squared OLS residuals—given in scalar form in (C.9) and (C.11) and in matrix form in (C.12) and (C.13)—suggest the model equation,

$$\log e_i^2 + 1 = \log\left\{\sum_{k=1}^{n} \omega_k m_{ik}^2\right\} + v_i, i = 1, 2, \ldots, n, \tag{C.14}$$

or, alternatively,

$$\log\{\boldsymbol{e} \circ \boldsymbol{e}\} + \boldsymbol{1}_n = \log\left[\operatorname{diag}(\boldsymbol{M\Omega M})\right] + \boldsymbol{v}. \tag{C.15}$$

Whether the log transformation of the response improves the model will depend on the bias-variance trade-off. The log transformation introduces bias inasmuch as the conditional mean function is now a second-order Taylor series approximation about $\mathrm{E}(e_i^2)$ rather than exact as in the original model. However, in many cases the log transformation will reduce the variance of the model errors, as can be seen by comparing (3.12) with (C.10). A further downside of the log-transformed models is that they are no longer linear in $\boldsymbol{\omega}$. Thus, regardless of how one might reparametrise $\boldsymbol{\omega}$ to reduce the number of parameters to be estimated, the model is an ANLVM and not an ALVM, an estimation method such as quasi-likelihood must be used. Investigating the viability of these log-based ANLVMs is an area for further research.

C.3 Further Details on the Thin-Plate Spline Auxiliary Linear Variance Model

Referring to (3.53), provided that the technical restriction $2m > p'$ is imposed, it can be shown (Wood 2003) that the solution to (3.52) has the form

$$\hat{g}(\boldsymbol{X}) = \sum_{i=1}^{n} \delta_i \eta_{md}(| - \boldsymbol{X}'_{i\cdot}||) + \sum_{j=1}^{M} \alpha_j \phi_j(\boldsymbol{X}), \tag{C.16}$$

where $\boldsymbol{\delta}$ and $\boldsymbol{\alpha}$ are coefficients to be estimated, δ being subject to the linear constraints $\boldsymbol{T}'\boldsymbol{\delta} = 0$ where $T_{ij} = \phi_j(\boldsymbol{X}'_{i\cdot})$, $M = \binom{m+d-1}{d}$, and

$$\eta_{md}(r) = \begin{cases} \dfrac{(-1)^{m+1+d/2}}{2^{2m-1}\pi^{d/2}(m-1)!(m-d/2)!} r^{2m-d}\log(r), & d \text{ even} \\ \dfrac{\Gamma(d/2-m)}{2^{2m}\pi^{d/2}(m-1)!} r^{2m-d}, & d \text{ odd} \end{cases}. \tag{C.17}$$

Defining matrix $\boldsymbol{E}$ by $E_{ij} = \eta_{m,p'}(||\boldsymbol{x}_i - \boldsymbol{x}_j||)$, the spline estimation problem becomes

$$\begin{aligned} &\arg\min_{\boldsymbol{\delta},\boldsymbol{\alpha}} ||\boldsymbol{y} - \boldsymbol{E\delta} - \boldsymbol{T\alpha}||_2^2 + \lambda\boldsymbol{\delta}'\boldsymbol{E\delta}, \\ &\text{subject to } \boldsymbol{T}'\boldsymbol{\delta} = \boldsymbol{0}. \end{aligned} \tag{C.18}$$

Due to the high computational cost of the optimisation problem (C.18), Wood (2003) proposes to use a truncated q-dimensional basis for the $\boldsymbol{\delta}$ parameter space, where $q > M$, constructed through eigen-decomposition of $\boldsymbol{E}$. This provides an approximate solution at much-reduced computational cost.

The equality constraint from (C.18) falls away, and the estimation problem becomes

$$\arg\min_{\tilde{\boldsymbol{\delta}},\boldsymbol{\alpha}} \left|\left|\boldsymbol{y} - \boldsymbol{U}_k\boldsymbol{D}_k\boldsymbol{Z}_k\tilde{\boldsymbol{\delta}} - \boldsymbol{T\alpha}\right|\right|_2^2 + \lambda\tilde{\boldsymbol{\delta}}'\boldsymbol{Z}_k'\boldsymbol{D}_k\boldsymbol{Z}_k\tilde{\boldsymbol{\delta}}, \tag{C.19}$$

where $\boldsymbol{U}_k$ is a submatrix of $\boldsymbol{U}$ (whose columns are eigenvectors of $\boldsymbol{E}$), $\boldsymbol{D}_k$ is a submatrix of $\boldsymbol{D}$ (a diagonal matrix of eigenvalues of $\boldsymbol{E}$), and $\boldsymbol{Z}_k$ is an orthogonal column basis that maps $\tilde{\boldsymbol{\delta}}$ onto $\boldsymbol{\delta}_k$, a subvector of $\boldsymbol{\delta}$. Clearly, the thin-plate spline model is linear in the combined parameter vector $\boldsymbol{\gamma} = [\boldsymbol{\delta}', \boldsymbol{\phi}']'$, which has dimensionality q. By augmenting these matrices with zeroes, one can obtain the linear predictor matrix $\boldsymbol{L}$ and the $q \times q$ penalty matrix $\boldsymbol{P}$. Introducing our $\boldsymbol{M} \circ \boldsymbol{M}$ term and linear inequality constraint, the estimation problem takes the form of (3.54).

C.4 Derivation of the Expectation of the Squared Wild Bootstrap Residual Vector

Here is given a derivation of the expectation of the squared wild bootstrap residual vector, as stated in (3.100), under the condition that $\boldsymbol{F}(\boldsymbol{e}) = \operatorname{diag}\{\boldsymbol{e}\}$.

First,

$$\begin{aligned}
\boldsymbol{e}^{(b)} &= \boldsymbol{y}^{(b)} - \boldsymbol{X}\hat{\boldsymbol{\beta}}^{(b)} \\
&= \boldsymbol{X}\left(\hat{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}^{(b)}\right) + \boldsymbol{F}(\boldsymbol{e})\boldsymbol{r}^{(b)} \\
&= \boldsymbol{H}\left(\boldsymbol{y} - \boldsymbol{y}^{(b)}\right) + \boldsymbol{F}(\boldsymbol{e})\boldsymbol{r}^{(b)} \\
&= \boldsymbol{H}\left(\boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} - \boldsymbol{X}\hat{\boldsymbol{\beta}} - \boldsymbol{F}(\boldsymbol{e})\boldsymbol{r}^{(b)}\right) + \boldsymbol{F}(\boldsymbol{e})\boldsymbol{r}^{(b)} \\
&= \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{H}\boldsymbol{\epsilon} - \boldsymbol{X}\hat{\boldsymbol{\beta}} + \boldsymbol{M}\boldsymbol{F}(\boldsymbol{e})\boldsymbol{r}^{(b)} \\
&= \boldsymbol{X}\left(\boldsymbol{\beta} - (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{y}\right) + \boldsymbol{H}\boldsymbol{\epsilon} + \boldsymbol{M}\boldsymbol{F}(\boldsymbol{e})\boldsymbol{r}^{(b)} \\
&= \boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{H}\left(\boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}\right) + \boldsymbol{H}\boldsymbol{\epsilon} + \boldsymbol{M}\boldsymbol{F}(\boldsymbol{e})\boldsymbol{r}^{(b)} \\
&= \boldsymbol{M}\boldsymbol{F}(\boldsymbol{e})\boldsymbol{r}^{(b)}.
\end{aligned}$$

It follows that the expectation of $\boldsymbol{e}^{(b)}$ is given by,

$$\begin{aligned}
\mathrm{E}(\boldsymbol{e}^{(b)}) &= \mathrm{E}\left[\boldsymbol{M}\boldsymbol{F}(\boldsymbol{e})\boldsymbol{r}^{(b)}\right] \\
&= \boldsymbol{M}\,\mathrm{E}\left[\boldsymbol{F}(\boldsymbol{e})\boldsymbol{r}^{(b)}\right] \\
&= \boldsymbol{M}\,\mathrm{E}\left[\boldsymbol{F}(\boldsymbol{e})\right]\mathrm{E}\left[\boldsymbol{r}^{(b)}\right] \\
&= \boldsymbol{0}.
\end{aligned}$$

The independence of $\boldsymbol{F}(\boldsymbol{e})$ and $\boldsymbol{r}^{(b)}$ and the zero expectation of the latter both follow from the definition of $\boldsymbol{r}^{(b)}$. Now,

$$\begin{aligned}
\mathrm{Cov}(\boldsymbol{e}^{(b)}) &= \mathrm{E}\left[\left(\boldsymbol{M}\boldsymbol{F}(\boldsymbol{e})\boldsymbol{r}^{(b)} - \mathrm{E}(\boldsymbol{M}\boldsymbol{F}(\boldsymbol{e})\boldsymbol{r}^{(b)})\right)\left(\boldsymbol{M}\boldsymbol{F}(\boldsymbol{e})\boldsymbol{r}^{(b)} - \mathrm{E}(\boldsymbol{M}\boldsymbol{F}(\boldsymbol{e})\boldsymbol{r}^{(b)})\right)'\right] \\
&= \mathrm{E}\left[\left(\boldsymbol{M}\boldsymbol{F}(\boldsymbol{e})\boldsymbol{r}^{(b)}\right)\left(\boldsymbol{M}\boldsymbol{F}(\boldsymbol{e})\boldsymbol{r}^{(b)}\right)'\right] \\
&= \mathrm{E}\left[\boldsymbol{M}\boldsymbol{F}(\boldsymbol{e})\boldsymbol{r}^{(b)}\boldsymbol{r}^{(b)\prime}\boldsymbol{F}(\boldsymbol{e})\boldsymbol{M}\right] = \boldsymbol{M}\,\mathrm{E}\left[\boldsymbol{F}(\boldsymbol{e})\boldsymbol{r}^{(b)}\boldsymbol{r}^{(b)\prime}\boldsymbol{F}(\boldsymbol{e})\right]\boldsymbol{M}.
\end{aligned}$$

The (i,j)th element of $\boldsymbol{F}(\boldsymbol{e})\boldsymbol{r}^{(b)}\boldsymbol{r}^{(b)\prime}\boldsymbol{F}(\boldsymbol{e})$ is $e_i r_i^{(b)} e_j r_j^{(b)}$. If $i \neq j$,

$$\begin{aligned}
\mathrm{E}\left[e_i e_j r_i^{(b)} r_j^{(b)}\right] &= \mathrm{E}\left[e_i e_j\right]\mathrm{E}(r_i^{(b)})\,\mathrm{E}(r_j^{(b)}) \\
&= 0.
\end{aligned}$$

The independence of $r_i^{(b)}$ and $r_j^{(b)}$ from each other and from e_i and e_j follow from the definition of $r_i^{(b)}$. Then, for the diagonal elements,

$$\mathrm{E}\left(f_i(e_i)^2 r_i^{(b)2}\right) = \mathrm{E}\left(f_i(e_i)^2\right)\underbrace{\mathrm{E}\left(r_i^{(b)2}\right)}_{=1} \text{ (by independence).}$$

It follows that, if $f_i(e_i) = e_i$, $\mathrm{E}\left[\boldsymbol{F}(\boldsymbol{e})\boldsymbol{r}^{(b)}\boldsymbol{r}^{(b)\prime}\boldsymbol{F}(\boldsymbol{e})\right]$ is a diagonal matrix with diagonal elements $\mathrm{E}(\boldsymbol{e} \circ \boldsymbol{e}) = (\boldsymbol{M} \circ \boldsymbol{M})\boldsymbol{\omega}$ (see (3.10)). Thus,

$$\mathrm{Cov}(\boldsymbol{e}^{(b)}) = \boldsymbol{M}\operatorname{diag}\left\{(\boldsymbol{M} \circ \boldsymbol{M})\boldsymbol{\omega}\right\}\boldsymbol{M},$$

which has diagonal elements $(\boldsymbol{M} \circ \boldsymbol{M})(\boldsymbol{M} \circ \boldsymbol{M})\boldsymbol{\omega}$. But the diagonal elements of $\mathrm{Cov}(\boldsymbol{e}^{(b)})$ are $\mathrm{Var}(e_i^{(b)})$, $i = 1, 2, \ldots, n$, which is equivalent to $\mathrm{E}(e_i^{(b)2})$, since $\mathrm{E}(\boldsymbol{e}^{(b)}) = \boldsymbol{0}$. Hence, $\mathrm{E}(\boldsymbol{e}^{(b)} \circ \boldsymbol{e}^{(b)}) = (\boldsymbol{M} \circ \boldsymbol{M})(\boldsymbol{M} \circ \boldsymbol{M})\boldsymbol{\omega}$.

D How to Access and Install the skedastic R Package

The **skedastic** R package developed for this research project (as discussed in Chapter 4) can be viewed on CRAN at https://cran.r-project.org/package=skedastic, or alternatively on Github at https://github.com/tjfarrar/skedastic. The version of the package currently on CRAN (version 2.0.1 at the time of writing), can be installed from within R software by running the code `install.packages("skedastic", dependencies = TRUE)`. The development version of the package can be installed from Github by running the code `devtools::install_github("tjfarrar/skedastic")` after installing the **devtools** package (Wickham et al. 2021).

E Additional Simulation Results on the Performance of the Auxiliary Variance Models

E.1 Linear Regression with One Covariate and $n = 20$ Observations

E.1.1 Auxiliary Linear Variance Model Results

Tables E.1-E.4 show results for a simulation like that discussed in §5.3.1, with one covariate generated from a $U(0, 3)$ distribution, but with a sample size of only $n = 20$ rather than $n = 100$.

Table E.1: (Relative) Unstandardised MSE-of-Variances Estimate (with Estimated Standard Error) for One-Covariate Linear Regression Model with $n = 20$

	Homosked.	Add. Het.	Mult. Het.
Model	$\boldsymbol{\mathcal{H}} = \emptyset$	$\boldsymbol{\mathcal{H}} = \{2\}$	$\boldsymbol{\mathcal{H}} = \{2\}$
HC3	22.9 (2.38×10^{-2})	8.02 (1.91×10^{0})	7.67 (2.14×10^{0})
HC4	16.1 (1.52×10^{-2})	4.21 (6.49×10^{-1})	3.45 (5.44×10^{-1})
HC6	8.1 (5.44×10^{-3})	3.62 (7.98×10^{-1})	3.7 (9.35×10^{-1})
Homoskedastic	1 (1.86×10^{-3})	1.31 (6.16×10^{-2})	1.42 (4.42×10^{-2})
Basic ALVM	22.1 (2.17×10^{-2})	7.16 (1.57×10^{0})	6.78 (1.62×10^{0})
Clustering ALVM	2.15 (5.75×10^{-3})	2.63 (8.57×10^{-1})	2.78 (1.01×10^{0})
Linear ALVM	1.54 (2.99×10^{-3})	1 (1.81×10^{-1})	1 (1.67×10^{-1})
L_2-Norm Pen. Poly. ALVM	1.93 (3.93×10^{-3})	1.29 (3.77×10^{-1})	1.34 (3.70×10^{-1})
L_1-Norm Pen. Poly. ALVM	2.55 (4.62×10^{-3})	1.58 (5.01×10^{-1})	1.56 (4.24×10^{-1})
Thin-Plate spline ALVM	4.46 (3.71×10^{-3})	1.65 (1.83×10^{-1})	1.59 (2.58×10^{-1})
Miller-Startz SVR	3.79 (1.85×10^{-3})	1.49 (7.72×10^{-2})	1.44 (7.25×10^{-2})

Table E.2: (Relative) Standardised MSE-of-Variances Estimate (with Estimated Standard Error) for One-Covariate Linear Regression Model with $n = 20$

Model	Homosked.	Add. Het.	Mult. Het.
	$\boldsymbol{\mathcal{H}} = \emptyset$	$\boldsymbol{\mathcal{H}} = \{2\}$	$\boldsymbol{\mathcal{H}} = \{2\}$
HC3	22.9 (2.38×10^{-2})	6.35 (2.45×10^{-2})	6.66 (2.57×10^{-2})
HC4	16.1 (1.52×10^{-2})	4.52 (1.65×10^{-2})	4.8 (1.76×10^{-2})
HC6	8.1 (5.44×10^{-3})	2.27 (5.76×10^{-3})	2.33 (6.13×10^{-3})
Homoskedastic	1 (1.86×10^{-3})	4 (2.27×10^{-2})	4.35 (2.62×10^{-2})
Basic ALVM	22.1 (2.17×10^{-2})	6.04 (2.25×10^{-2})	6.31 (2.32×10^{-2})
Clustering ALVM	2.15 (5.75×10^{-3})	1.55 (8.22×10^{-3})	1.44 (8.28×10^{-3})
Linear ALVM	1.54 (2.99×10^{-3})	1.14 (7.25×10^{-3})	1.29 (9.68×10^{-3})
L_2-Norm Pen. Poly. ALVM	1.93 (3.93×10^{-3})	1.51 (9.47×10^{-3})	1.49 (9.77×10^{-3})
L_1-Norm Pen. Poly. ALVM	2.55 (4.62×10^{-3})	1.17 (8.40×10^{-3})	1.2 (8.99×10^{-3})
Thin-Plate spline ALVM	4.46 (3.71×10^{-3})	1.46 (3.02×10^{-3})	1.58 (3.30×10^{-3})
Miller-Startz SVR	3.79 (1.85×10^{-3})	1 (1.74×10^{-3})	1 (1.74×10^{-3})

Table E.3: (Relative) MSE of FWLS Estimate of $\boldsymbol{\beta}$ (with Estimated Standard Error) for One-Covariate Linear Regression Model with $n = 20$

Model	Homosked.	Add. Het.	Mult. Het.
	$\boldsymbol{\mathcal{H}} = \emptyset$	$\boldsymbol{\mathcal{H}} = \{2\}$	$\boldsymbol{\mathcal{H}} = \{2\}$
OLS	1 (1.43×10^{-3})	1.15 (7.87×10^{-3})	1.25 (7.51×10^{-3})
HC3	1.08 (1.52×10^{-3})	1.14 (7.92×10^{-3})	1.21 (7.37×10^{-3})
HC4	1.1 (1.54×10^{-3})	1.14 (7.83×10^{-3})	1.24 (7.50×10^{-3})
HC6	1.15 (1.62×10^{-3})	1.19 (8.02×10^{-3})	1.31 (7.88×10^{-3})
Homoskedastic	1 (1.43×10^{-3})	1.15 (7.87×10^{-3})	1.25 (7.51×10^{-3})
Basic ALVM	2.74 (5.52×10^{-2})	2.1 (1.17×10^{-1})	1.98 (6.45×10^{-2})
Clustering ALVM	2.48 (1.92×10^{-2})	4.42 (1.52×10^{-1})	3.28 (8.79×10^{-2})
Linear ALVM	1.14 (1.88×10^{-3})	1.54 (1.27×10^{-2})	1.71 (1.21×10^{-2})
L_2-Norm Pen. Poly. ALVM	1.29 (5.35×10^{-3})	5.82 (1.45×10^{0})	11.2 (1.15×10^{0})
L_1-Norm Pen. Poly. ALVM	1.56 (2.44×10^{-2})	27.2 (5.20×10^{0})	19.8 (2.03×10^{0})
Thin-Plate spline ALVM	48.6 (7.13×10^{-1})	18.5 (5.29×10^{-1})	21.4 (4.56×10^{-1})
Miller-Startz SVR	1.07 (1.54×10^{-3})	1 (6.74×10^{-3})	1 (6.02×10^{-3})

Table E.4: (Relative) MSE of Standard Errors of $\hat{\beta}_{\text{OLS}}$ (with Estimated Standard Error) for One-Covariate Linear Regression Model with $n = 20$

	Homosked.	Add. Het.	Mult. Het.
Model	$\boldsymbol{\mathcal{H}} = \emptyset$	$\boldsymbol{\mathcal{H}} = \{2\}$	$\boldsymbol{\mathcal{H}} = \{2\}$
HC3	2.65 (1.34×10^{-4})	2.39 (1.88×10^{-3})	2.37 (2.19×10^{-3})
HC4	1.83 (6.43×10^{-5})	1.61 (7.95×10^{-4})	1.58 (8.21×10^{-4})
HC6	8.09 (1.31×10^{-4})	3.76 (1.18×10^{-3})	3.33 (1.30×10^{-3})
Homoskedastic	1 (4.38×10^{-5})	1.33 (3.16×10^{-4})	1.43 (3.50×10^{-4})
Basic ALVM	2.29 (9.97×10^{-5})	2.04 (1.36×10^{-3})	2.11 (1.64×10^{-3})
Clustering ALVM	1.36 (6.92×10^{-5})	1.79 (1.18×10^{-3})	1.81 (1.29×10^{-3})
Linear ALVM	1.19 (4.71×10^{-5})	1 (4.47×10^{-4})	1 (4.83×10^{-4})
L_2-Norm Pen. Poly. ALVM	1.42 (6.48×10^{-5})	1.29 (7.34×10^{-4})	1.3 (7.84×10^{-4})
L_1-Norm Pen. Poly. ALVM	1.72 (7.29×10^{-5})	1.45 (9.84×10^{-4})	1.4 (9.48×10^{-4})
Thin-Plate spline ALVM	6.33 (1.56×10^{-4})	3.77 (1.06×10^{-3})	3.65 (1.09×10^{-3})
Miller-Startz SVR	6.21 (1.24×10^{-4})	3.59 (8.38×10^{-4})	3.39 (8.60×10^{-4})

E.1.2 Auxiliary Nonlinear Variance Model Results

Table E.5: Relative Performance Metrics (with Estimated Standard Errors) for ANLVMs Fit to One-Covariate Linear Regression Model with $n = 20$

		Homosked.	Add. Het.	Mult. Het.
Metric	ANLVM	$\boldsymbol{\mathcal{H}} = \emptyset$	$\boldsymbol{\mathcal{H}} = \{2\}$	$\boldsymbol{\mathcal{H}} = \{2\}$
$\overline{\text{MSE}}_{\text{ust}}(\hat{\boldsymbol{\omega}})$	Quadratic	2.83 (4.36×10^{-2})	1.98 (8.66×10^{-1})	1.6 (6.45×10^{-1})
	Exponential	2.2 (1.18×10^{-2})	5.47 (3.69×10^{0})	4.67 (8.73×10^{0})
	Clustering	1.93 (1.22×10^{-2})	2.16 (6.91×10^{-1})	2.05 (5.80×10^{-1})
$\overline{\text{MSE}}_{\text{std}}(\hat{\boldsymbol{\omega}})$	Quadratic	2.83 (4.36×10^{-2})	1.5 (1.56×10^{-2})	1.42 (1.65×10^{-2})
	Exponential	2.2 (1.18×10^{-2})	1.64 (2.11×10^{-2})	1.22 (4.01×10^{-2})
	Clustering	1.93 (1.22×10^{-2})	1.47 (8.02×10^{-3})	1.38 (7.80×10^{-3})
$\text{MSE}(\hat{\boldsymbol{\beta}}_{\text{FWLS}})$	Quadratic	1.06 (1.43×10^{-3})	1.02 (7.87×10^{-3})	1.01 (7.51×10^{-3})
	Exponential	1.08 (1.52×10^{-3})	0.97 (7.92×10^{-3})	0.926 (7.37×10^{-3})
	Clustering	1.04 (1.54×10^{-3})	0.988 (7.83×10^{-3})	0.985 (7.50×10^{-3})
$\text{MSE}(\text{SE}(\hat{\boldsymbol{\beta}}))$	Quadratic	1.56 (1.34×10^{-4})	1.52 (1.88×10^{-3})	1.31 (2.19×10^{-3})
	Exponential	1.42 (6.43×10^{-5})	2.78 (7.95×10^{-4})	2 (8.21×10^{-4})
	Clustering	1.27 (1.31×10^{-4})	1.59 (1.18×10^{-3})	1.58 (1.30×10^{-3})

E.2 Linear Regression with One Covariate and $n = 1000$

E.2.1 Auxiliary Linear Variance Model Results

Tables E.6-E.9 show results for a simulation like that discussed in §5.3.1, with one covariate generated from a $U(0, 3)$ distribution, but with a sample size of $n = 1000$ rather than $n = 100$. Due to the increased computation time required, only $R = 10^3$ MC replications were used in this simulation rather than $R = 10^4$, and some slower models (LASSO polynomial ALVM, thin-plate spline ALVM) were not included.

Table E.6: (Relative) Unstandardised MSE-of-Variances Estimate (with Estimated Standard Error) for One-Covariate Linear Regression Model with $n = 1000$

Model	Homosked.	Add. Het.	Mult. Het.
	$\boldsymbol{\mathcal{H}} = \emptyset$	$\boldsymbol{\mathcal{H}} = \{2\}$	$\boldsymbol{\mathcal{H}} = \{2\}$
HC3	994 (7.76×10^{-3})	171 (7.29×10^{-1})	60.7 (8.65×10^{-1})
HC4	988 (7.64×10^{-3})	170 (7.21×10^{-1})	59.9 (8.23×10^{-1})
HC6	453 (1.09×10^{-4})	76.5 (2.30×10^{-2})	26.6 (3.20×10^{-2})
Homoskedastic	1 (8.71×10^{-5})	24 (6.69×10^{-3})	11.9 (6.69×10^{-3})
Basic ALVM	991 (7.61×10^{-3})	171 (7.03×10^{-1})	60.1 (8.06×10^{-1})
Clustering ALVM	3.09 (3.76×10^{-4})	2.48 (3.29×10^{-2})	1 (3.86×10^{-2})
Linear ALVM	1.54 (1.26×10^{-4})	1 (1.25×10^{-2})	2.06 (1.22×10^{-2})
L_2-Norm Pen. Poly. ALVM	188 (9.77×10^{-4})	33.5 (9.01×10^{-2})	12.2 (9.69×10^{-2})
Miller-Startz SVR	187 (1.02×10^{-3})	33.7 (8.92×10^{-2})	12.2 (1.01×10^{-1})

Table E.7: (Relative) Standardised MSE-of-Variances Estimate (with Estimated Standard Error) for One-Covariate Linear Regression Model with $n = 1000$

Model	Homosked.	Add. Het.	Mult. Het.
	$\boldsymbol{\mathcal{H}} = \emptyset$	$\boldsymbol{\mathcal{H}} = \{2\}$	$\boldsymbol{\mathcal{H}} = \{2\}$
HC3	994 (7.76×10^{-3})	63 (7.36×10^{-3})	62.6 (7.77×10^{-3})
HC4	988 (7.64×10^{-3})	62.4 (7.39×10^{-3})	62.3 (7.71×10^{-3})
HC6	453 (1.09×10^{-4})	28.9 (1.27×10^{-4})	28.9 (1.28×10^{-4})
Homoskedastic	1 (8.71×10^{-5})	93.9 (1.30×10^{-2})	117 (1.79×10^{-2})
Basic ALVM	991 (7.61×10^{-3})	63 (7.66×10^{-3})	62.2 (7.47×10^{-3})
Clustering ALVM	3.09 (3.76×10^{-4})	1 (3.54×10^{-4})	1 (3.30×10^{-4})
Linear ALVM	1.54 (1.26×10^{-4})	1.52 (3.65×10^{-4})	5.79 (1.63×10^{-3})
L_2-Norm Pen. Poly. ALVM	188 (9.77×10^{-4})	12.1 (1.00×10^{-3})	12.2 (9.83×10^{-4})
Miller-Startz SVR	187 (1.02×10^{-3})	12.2 (9.45×10^{-4})	12.1 (9.49×10^{-4})

Table E.8: (Relative) MSE of FWLS Estimate of $\boldsymbol{\beta}$ (with Estimated Standard Error) for One-Covariate Linear Regression Model with $n = 1000$

	Homosked.	Add. Het.	Mult. Het.
Model	$\boldsymbol{\mathcal{H}} = \emptyset$	$\boldsymbol{\mathcal{H}} = \{2\}$	$\boldsymbol{\mathcal{H}} = \{2\}$
OLS	1.09 (1.16×10^{-4})	1.49 (4.53×10^{-4})	1.75 (4.56×10^{-4})
HC3	1.06 (1.14×10^{-4})	1.59 (4.72×10^{-4})	1.69 (4.58×10^{-4})
HC4	1.02 (1.09×10^{-4})	1.65 (5.28×10^{-4})	1.79 (5.00×10^{-4})
HC6	1.09 (1.27×10^{-4})	1.47 (5.09×10^{-4})	1.75 (4.46×10^{-4})
Homoskedastic	1.09 (1.16×10^{-4})	1.49 (4.53×10^{-4})	1.75 (4.56×10^{-4})
Basic ALVM	1.1 (1.24×10^{-4})	1.6 (4.89×10^{-4})	1.83 (4.93×10^{-4})
Clustering ALVM	1 (1.08×10^{-4})	1 (3.11×10^{-4})	1 (2.61×10^{-4})
L_2-Norm Pen. Poly. ALVM	1.09 (1.16×10^{-4})	1.49 (4.53×10^{-4})	1.75 (4.56×10^{-4})
Miller-Startz SVR	1.08 (1.18×10^{-4})	74.1 (2.88×10^{-2})	118 (3.58×10^{-2})

Table E.9: (Relative) MSE of Standard Errors of $\hat{\beta}_{\text{OLS}}$ (with Estimated Standard Error) for One-Covariate Linear Regression Model with $n = 1000$

	Homosked.	Add. Het.	Mult. Het.
Model	$\boldsymbol{\mathcal{H}} = \emptyset$	$\boldsymbol{\mathcal{H}} = \{2\}$	$\boldsymbol{\mathcal{H}} = \{2\}$
HC3	2.26 (1.32×10^{-7})	1.04 (6.12×10^{-7})	1.05 (7.21×10^{-7})
HC4	2.21 (1.20×10^{-7})	1 (5.54×10^{-7})	1.05 (7.62×10^{-7})
HC6	1140 (1.48×10^{-6})	495 (8.77×10^{-6})	372 (9.77×10^{-6})
Homoskedastic	1 (5.82×10^{-8})	80.8 (6.48×10^{-6})	62.7 (5.77×10^{-6})
Basic ALVM	2.07 (1.26×10^{-7})	1.03 (5.78×10^{-7})	1 (6.72×10^{-7})
Clustering ALVM	1.24 (8.08×10^{-8})	1 (5.59×10^{-7})	1.05 (7.06×10^{-7})
Linear ALVM	1.35 (7.95×10^{-8})	1.53 (7.44×10^{-7})	2.64 (1.04×10^{-6})
L_2-Norm Pen. Poly. ALVM	285 (1.77×10^{-6})	134 (8.19×10^{-6})	110 (8.47×10^{-6})
Miller-Startz SVR	283 (1.83×10^{-6})	135 (7.96×10^{-6})	110 (8.76×10^{-6})

E.2.2 Auxiliary Nonlinear Variance Model Results

The ANLVM results in Table E.10 are based on only $R = 10^2$ MC replications due to the high computation time required for MQL estimation with $n = 1000$ observations.

Table E.10: Relative Performance Metrics (with Estimated Standard Errors) for ANLVMs Fit to One-Covariate Linear Regression Model with $n = 1000$

		Homosked.	Add. Het.	Mult. Het.
Metric	ANLVM	$\boldsymbol{\mathcal{H}} = \emptyset$	$\boldsymbol{\mathcal{H}} = \{2\}$	$\boldsymbol{\mathcal{H}} = \{2\}$
$\overline{\text{MSE}}_{\text{ust}}(\hat{\boldsymbol{\omega}})$	Quadratic	1.59 (4.30×10^{-4})	0.332 (3.14×10^{-2})	1.14 (7.68×10^{-2})
	Exponential	1.63 (4.51×10^{-4})	2.36 (1.44×10^{-1})	0.186 (6.26×10^{-2})
	Clustering	2.9 (9.57×10^{-4})	2.35 (1.09×10^{-1})	1.07 (1.74×10^{-1})
$\overline{\text{MSE}}_{\text{std}}(\hat{\boldsymbol{\omega}})$	Quadratic	1.59 (4.30×10^{-4})	0.129 (3.38×10^{-4})	0.927 (6.61×10^{-4})
	Exponential	1.63 (4.51×10^{-4})	0.822 (8.40×10^{-4})	0.133 (4.37×10^{-4})
	Clustering	2.9 (9.57×10^{-4})	0.946 (9.69×10^{-4})	1.02 (1.10×10^{-3})
$\text{MSE}(\hat{\boldsymbol{\beta}}_{\text{FWLS}})$	Quadratic	1.06 (1.16×10^{-4})	0.951 (4.53×10^{-4})	0.953 (4.56×10^{-4})
	Exponential	1.01 (1.14×10^{-4})	1.09 (4.72×10^{-4})	0.851 (4.58×10^{-4})
	Clustering	0.926 (1.09×10^{-4})	0.803 (5.28×10^{-4})	0.829 (5.00×10^{-4})
$\text{MSE}(\text{SE}(\hat{\boldsymbol{\beta}}))$	Quadratic	1.28 (1.32×10^{-7})	0.621 (6.12×10^{-7})	3.3 (7.21×10^{-7})
	Exponential	1.35 (1.20×10^{-7})	3.41 (5.54×10^{-7})	0.646 (7.62×10^{-7})
	Clustering	1.53 (1.48×10^{-6})	1.1 (8.77×10^{-6})	0.902 (9.77×10^{-6})

E.3 Linear Regression with One Covariate and Nonmonotonic Heteroskedasticity

The DGP for this simulation consisted of $n = 100$ observations of a single design variable generated from $U(0, 3)$. The heteroskedastic function was $g(x) = \left[\sin^2\left(\frac{2\pi x}{3}\right)\right] + \frac{1}{5}$, which is plotted in Figure E.1.

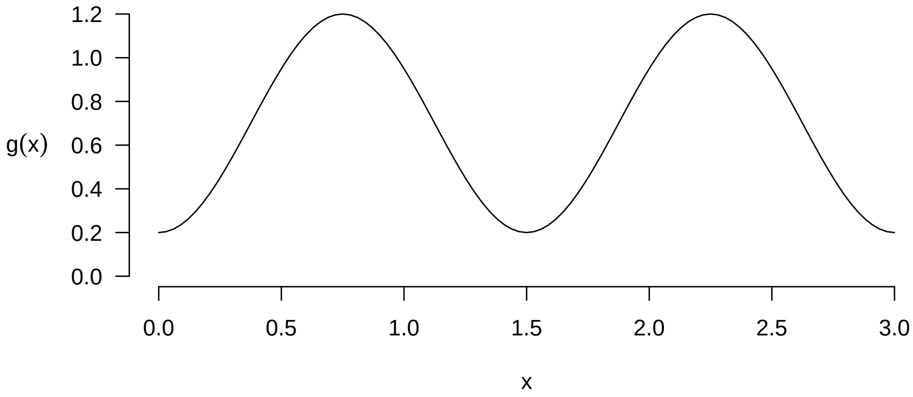
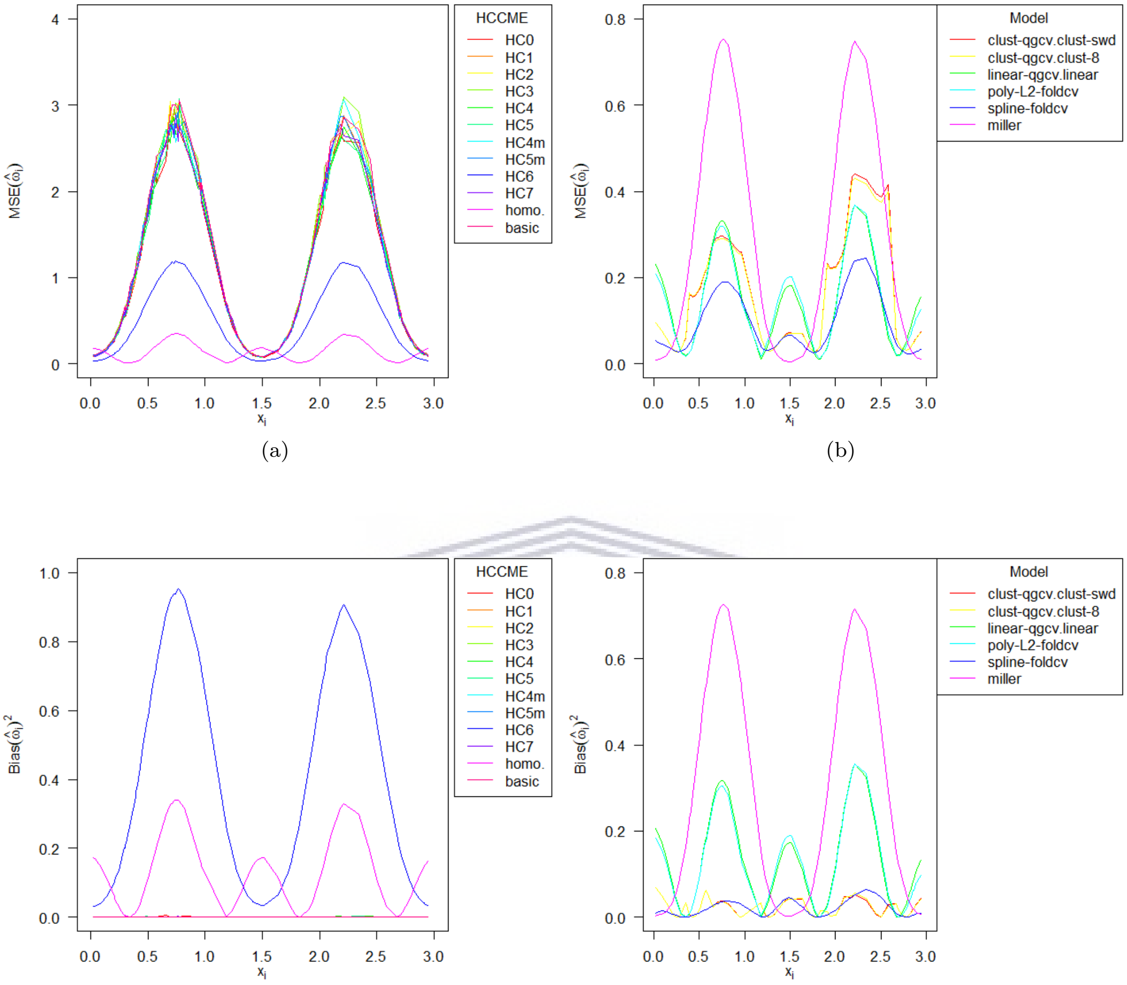


Figure E.1: Graph of $g(x) = \left[\sin^2\left(\frac{2\pi x}{3}\right)\right] + \frac{1}{5}$ for $x \in [0, 3]$

E.3.1 Auxiliary Linear Variance Model Results

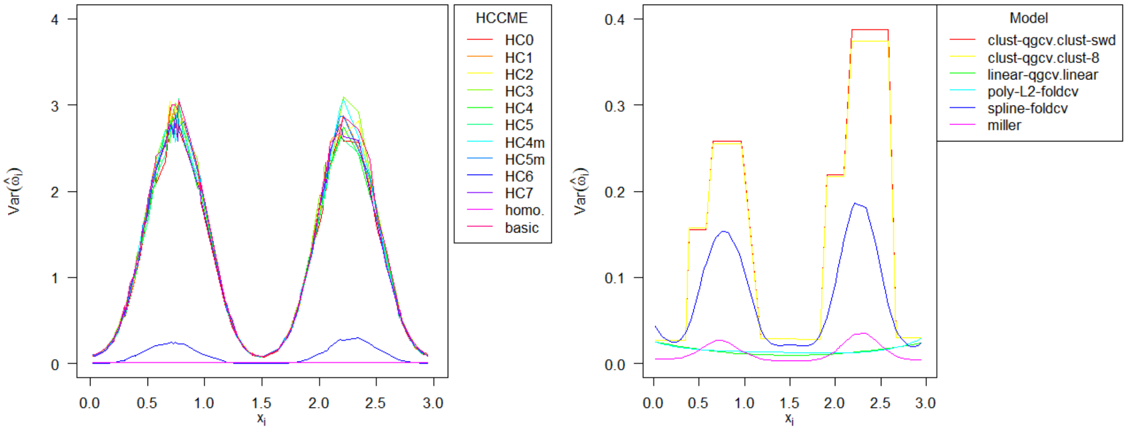


(a)

(b)

(c)

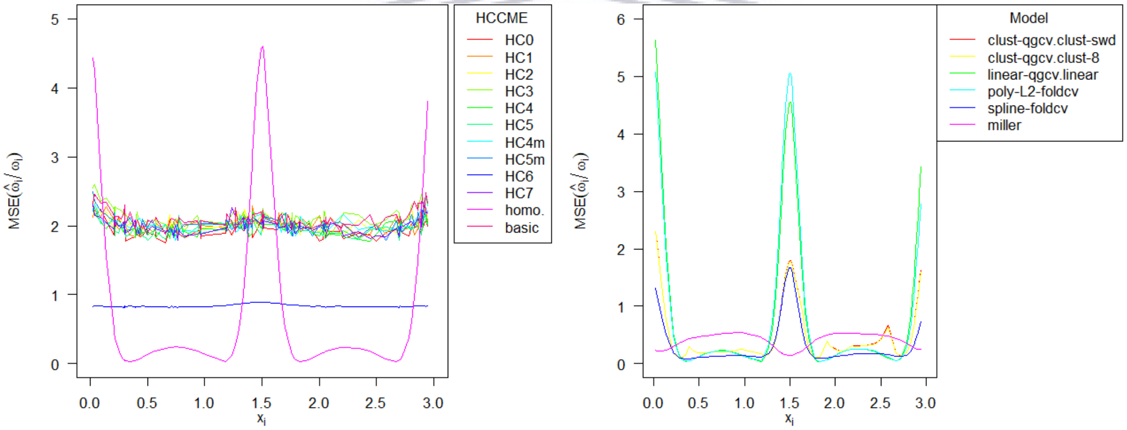
(d)



(e)

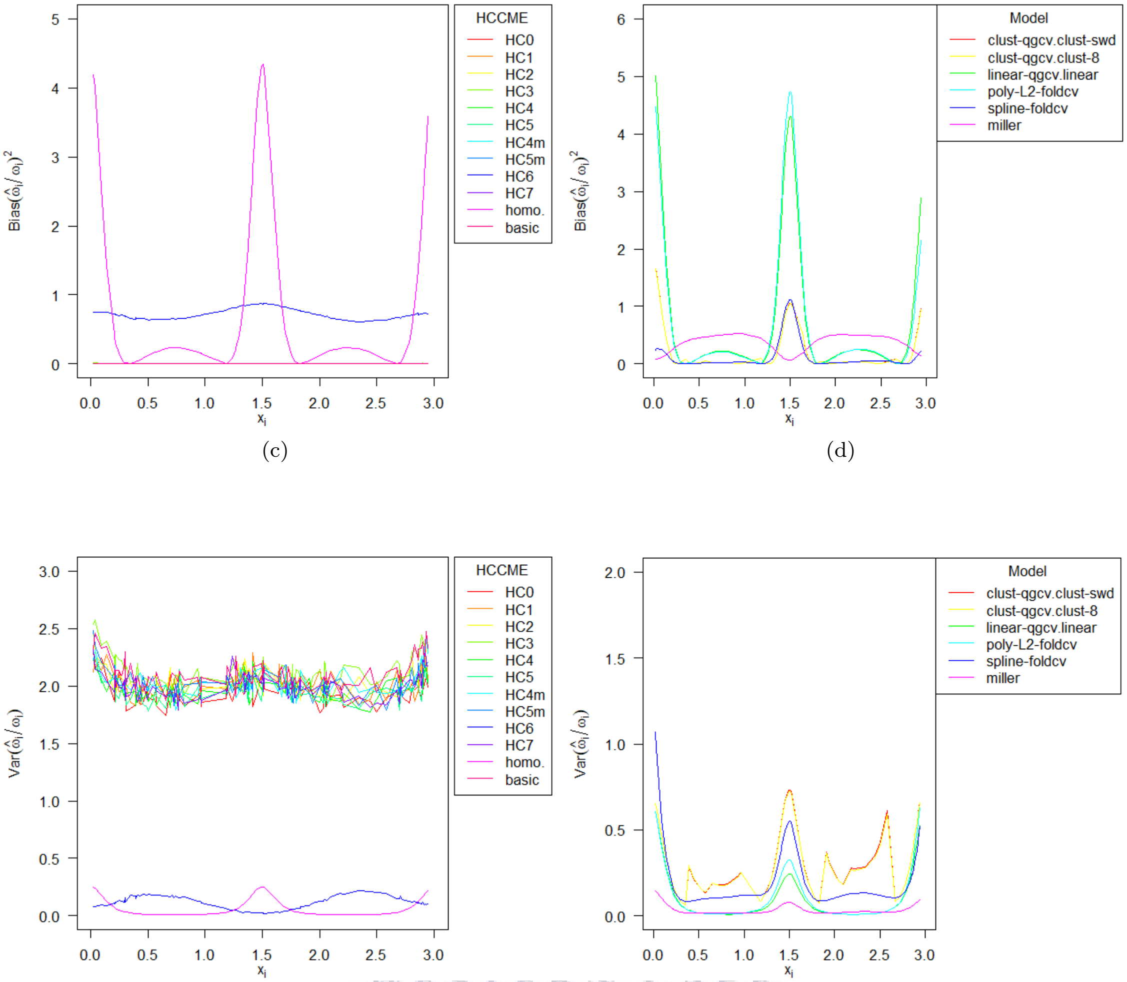
(f)

Figure E.2: Unstandardised MSE, Squared Bias, and Variance Metrics of HCCMEs (a, c, e) and ALVMs (b, d, f) for Simple Linear Regression Model with Nonmonotonic Heteroskedasticity



(a)

(b)



(c) (d)

(e) (f)

Figure E.3: Standardised MSE, Squared Bias, and Variance Metrics of HCCMEs (a, c, e) and ALVMs (b, d, f) for Simple Linear Regression Model with Nonmonotonic Heteroskedasticity

Table E.11: Estimated Metrics (with Estimated SE) for ALVMs Fitted under Nonmonotonic Heteroskedasticity

Model	$\overline{\text{MSE}}_{\text{ust}}(\hat{\omega})$	$\overline{\text{MSE}}_{\text{st}}(\hat{\omega})$	$\text{MSE}(\hat{\beta}_{\text{FWLS}})$	$\text{MSE}(\text{SE}(\hat{\beta}))$
OLS			1.13 (1.93×10^{-4})	
HC3	13.6 (5.32×10^{-3})	6.62 (8.12×10^{-3})	1.14 (1.91×10^{-4})	1.43 (2.19×10^{-6})
HC4	12.6 (4.75×10^{-3})	6.13 (7.41×10^{-3})	1.13 (1.95×10^{-4})	1.31 (1.92×10^{-6})
HC6	5.42 (5.18×10^{-4})	2.6 (5.81×10^{-4})	1.16 (1.93×10^{-4})	44.6 (9.02×10^{-6})
Homoskedastic	1.64 (1.53×10^{-4})	2.69 (4.27×10^{-3})	1.13 (1.93×10^{-4})	1 (1.52×10^{-6})
Basic ALVM	13.4 (5.19×10^{-3})	6.53 (7.91×10^{-3})	1.19 (2.00×10^{-4})	1.35 (2.01×10^{-6})
Clustering ALVM (n_c: SWD)	1.7 (7.54×10^{-4})	1.45 (3.34×10^{-3})	1.01 (1.73×10^{-4})	1.24 (1.90×10^{-6})
Clustering ALVM ($n_c = 8$)	1.68 (7.38×10^{-4})	1.44 (3.29×10^{-3})	1.04 (1.79×10^{-4})	1.24 (1.90×10^{-6})
Linear ALVM	1.68 (1.88×10^{-4})	2.77 (4.32×10^{-3})	1.15 (1.99×10^{-4})	1.22 (1.94×10^{-6})
L_2-Norm Pen. Poly. ALVM	1.67 (1.84×10^{-4})	2.75 (4.35×10^{-3})	1.13 (1.91×10^{-4})	1.18 (1.82×10^{-6})
Thin-Plate spline ALVM	1 (5.39×10^{-4})	1 (2.38×10^{-3})	2.24 (7.83×10^{-3})	1.52 (2.34×10^{-6})
Miller-Startz SVR	3.14 (5.55×10^{-4})	1.19 (8.62×10^{-4})	1 (1.69×10^{-4})	20.5 (9.31×10^{-6})

E.3.2 Auxiliary Nonlinear Variance Model Results

Table E.12: Estimated Metrics (with Estimated SE) for ANLVMs Fitted under Nonmonotonic Heteroskedasticity

ANLVM	$\overline{\text{MSE}}_{\text{ust}}(\hat{\omega})$	$\overline{\text{MSE}}_{\text{st}}(\hat{\omega})$	$\text{MSE}(\hat{\beta}_{\text{FWLS}})$	$\text{MSE}(\text{SE}(\hat{\beta}))$
Quadratic	1.73 (8.44×10^{-4})	2.84 (1.01×10^{-2})	1.17 (2.05×10^{-4})	1.52 (4.95×10^{-6})
Exponential	1.72 (2.50×10^{-4})	2.83 (4.66×10^{-3})	1.15 (1.92×10^{-4})	1.52 (2.65×10^{-6})
Clustering (n_c: SWD)	1.71 (4.98×10^{-4})	2.24 (4.00×10^{-3})	1.12 (1.91×10^{-4})	1.4 (2.36×10^{-6})
Clustering ($n_c = 8$)	1.72 (7.72×10^{-4})	1.47 (3.40×10^{-3})	1.05 (1.79×10^{-4})	1.29 (2.04×10^{-6})
Sq. Sinusoidal	1.45 (6.88×10^{-4})	2.59 (7.80×10^{-3})	1.06 (1.84×10^{-4})	0.903 (1.43×10^{-6})

Note: the 'Sq. Sinusoidal' ANLVM in the table corresponds to an ANLVM with $g(x) = \left[\sin^2\left(\frac{2\pi x}{3}\right)\right] + \frac{1}{5}$ (illustrated in Figure E.1)—the exact heteroskedastic function of the DGP.

E.4 Linear Regression with One Covariate and Non-Normal Errors

E.4.1 Auxiliary Linear Variance Model Results

Table E.13: (Relative) Unstandardised MSE-of-Variances Estimate (with Estimated Standard Error) for One-Covariate Linear Regression Model with Non-Normal Errors

	$\mathcal{H} = \emptyset$		$\mathcal{H} = \{2\}$			
	Homoskedastic		Additive Het.		Mult. Het.	
Model	Laplace	Uniform	Laplace	Uniform	Laplace	Uniform
HC3	97.7 (1.49×10^{-1})	102 (3.52×10^{-3})	40.5 (1.56×10^{1})	31.4 (3.82×10^{-1})	26.3 (1.99×10^{1})	19.3 (4.07×10^{-1})
HC4	85.2 (1.16×10^{-1})	95.5 (3.19×10^{-3})	33.6 (9.56×10^{0})	29.2 (3.10×10^{-1})	26 (1.39×10^{1})	17.6 (3.71×10^{-1})
HC6	24.6 (4.25×10^{-2})	90.6 (5.24×10^{-4})	14.5 (7.44×10^{0})	25.3 (8.29×10^{-2})	12.1 (1.12×10^{1})	14.6 (1.21×10^{-1})
Homoskedastic	1 (2.71×10^{-3})	1 (3.73×10^{-4})	2.79 (1.86×10^{-1})	11.2 (2.42×10^{-2})	2.69 (2.53×10^{-1})	9.65 (2.65×10^{-2})
Basic ALVM	92.9 (1.32×10^{-1})	101 (3.37×10^{-3})	38.1 (1.17×10^{1})	31.3 (3.46×10^{-1})	27.3 (1.99×10^{1})	19.5 (4.45×10^{-1})
Clustering ALVM	1.62 (5.07×10^{-3})	2.01 (7.93×10^{-4})	2.15 (1.04×10^{0})	1.92 (7.77×10^{-2})	1.59 (1.20×10^{0})	1.63 (8.98×10^{-2})
Linear ALVM	1.39 (3.59×10^{-3})	1.64 (5.64×10^{-4})	1 (5.19×10^{-1})	1 (5.47×10^{-2})	1.09 (5.59×10^{-1})	2.05 (5.77×10^{-2})
L_2-Norm Pen. Poly. ALVM	1.65 (4.04×10^{-3})	1.84 (6.00×10^{-4})	1.15 (4.67×10^{-1})	1.19 (7.23×10^{-2})	1 (5.77×10^{-1})	1.05 (9.40×10^{-2})
L_1-Norm Pen. Poly. ALVM	2.1 (4.74×10^{-3})	2.12 (5.76×10^{-4})	1.22 (5.16×10^{-1})	1.12 (8.48×10^{-2})	1.17 (7.25×10^{-1})	1 (9.75×10^{-2})
Thin-Plate spline ALVM	3.35 (5.73×10^{-3})	4.25 (8.94×10^{-4})	1.5 (4.75×10^{-1})	1.73 (1.02×10^{-1})	1.01 (4.22×10^{-1})	1.45 (1.18×10^{-1})
Miller-Startz SVR	11.2 (2.62×10^{-3})	23.6 (2.57×10^{-3})	4.75 (2.32×10^{-1})	9.26 (2.27×10^{-1})	3.49 (2.71×10^{-1})	6.39 (2.60×10^{-1})

Table E.14: (Relative) Standardised MSE-of-Variances Estimate (with Estimated Standard Error) for One-Covariate Linear Regression Model with Non-Normal Errors

	$\boldsymbol{\mathcal{H}} = \emptyset$		$\boldsymbol{\mathcal{H}} = \{2\}$			
	Homoskedastic		Additive Het.		Mult. Het.	
Model	Laplace	Uniform	Laplace	Uniform	Laplace	Uniform
HC3	97.7 (1.49×10^{-1})	102 (3.52×10^{-3})	19.2 (1.65×10^{-1})	15.6 (4.39×10^{-3})	15.2 (1.56×10^{-1})	11.9 (4.68×10^{-3})
HC4	85.2 (1.16×10^{-1})	95.5 (3.19×10^{-3})	16.5 (1.17×10^{-1})	14.7 (3.88×10^{-3})	14.5 (1.26×10^{-1})	11.1 (4.13×10^{-3})
HC6	24.6 (4.25×10^{-2})	90.6 (5.24×10^{-4})	4.74 (3.62×10^{-2})	13.7 (4.64×10^{-4})	3.99 (3.79×10^{-2})	10.4 (4.73×10^{-4})
Homoskedastic	1 (2.71×10^{-3})	1 (3.73×10^{-4})	11.6 (7.65×10^{-2})	47.7 (2.60×10^{-2})	13.4 (1.29×10^{-1})	49.1 (3.92×10^{-2})
Basic ALVM	92.9 (1.32×10^{-1})	101 (3.37×10^{-3})	18.4 (1.45×10^{-1})	15.5 (3.94×10^{-3})	15.7 (1.64×10^{-1})	11.8 (4.40×10^{-3})
Clustering ALVM	1.62 (5.07×10^{-3})	2.01 (7.93×10^{-4})	1.5 (2.43×10^{-2})	1.35 (1.17×10^{-3})	1.02 (1.75×10^{-2})	1 (1.08×10^{-3})
Linear ALVM	1.39 (3.59×10^{-3})	1.64 (5.64×10^{-4})	1 (2.38×10^{-2})	1 (1.09×10^{-3})	1.24 (2.30×10^{-2})	2.85 (4.07×10^{-3})
L_2-Norm Pen. Poly. ALVM	1.65 (4.04×10^{-3})	1.84 (6.00×10^{-4})	1.72 (2.64×10^{-2})	1.84 (4.46×10^{-3})	1.31 (2.51×10^{-2})	1.33 (3.73×10^{-3})
L_1-Norm Pen. Poly. ALVM	2.1 (4.74×10^{-3})	2.12 (5.76×10^{-4})	1.02 (1.13×10^{-2})	1.71 (3.56×10^{-3})	1.02 (1.60×10^{-2})	1.39 (3.48×10^{-3})
Thin-Plate spline ALVM	3.35 (5.73×10^{-3})	4.25 (8.94×10^{-4})	1.09 (1.33×10^{-2})	1.44 (2.73×10^{-3})	1 (8.28×10^{-3})	2.27 (4.96×10^{-3})
Miller-Startz SVR	11.2 (2.62×10^{-3})	23.6 (2.57×10^{-3})	2.15 (2.72×10^{-3})	3.85 (2.48×10^{-3})	1.77 (2.85×10^{-3})	2.99 (2.58×10^{-3})

Table E.15 (Relative) MSE of FWLS Estimate of $\boldsymbol{\beta}$ (with Estimated Standard Error) for One-Covariate Linear Regression Model with Non-Normal Errors

	$\boldsymbol{\mathcal{H}} = \emptyset$		$\boldsymbol{\mathcal{H}} = \{2\}$			
	Homoskedastic		Additive Het.		Mult. Het.	
Model	Laplace	Uniform	Laplace	Uniform	Laplace	Uniform
OLS	1.18 (1.14×10^{-3})	1 (9.63×10^{-4})	1.47 (4.88×10^{-3})	1.4 (4.59×10^{-3})	1.91 (5.13×10^{-3})	1.56 (4.75×10^{-3})
HC3	1.09 (9.56×10^{-4})	1.2 (1.17×10^{-3})	1.49 (4.76×10^{-3})	1.56 (5.36×10^{-3})	1.84 (5.06×10^{-3})	1.67 (5.10×10^{-3})
HC4	1.09 (1.01×10^{-3})	1.26 (1.25×10^{-3})	1.33 (4.62×10^{-3})	1.64 (5.35×10^{-3})	1.72 (4.98×10^{-3})	1.56 (4.63×10^{-3})
HC6	1.17 (9.67×10^{-4})	1.12 (1.08×10^{-3})	1.57 (5.28×10^{-3})	1.54 (4.92×10^{-3})	1.78 (4.75×10^{-3})	1.6 (4.71×10^{-3})
Homoskedastic	1.18 (1.14×10^{-3})	1 (9.63×10^{-4})	1.47 (4.88×10^{-3})	1.4 (4.59×10^{-3})	1.91 (5.13×10^{-3})	1.56 (4.75×10^{-3})
Basic ALVM	1.14 (1.02×10^{-3})	10.6 (1.60×10^{-1})	1.38 (4.87×10^{-3})	1.79 (2.06×10^{-2})	1.8 (4.81×10^{-3})	1.75 (6.68×10^{-3})
Clustering ALVM	1.18 (1.11×10^{-3})	1.15 (1.07×10^{-3})	1 (3.23×10^{-3})	1 (3.16×10^{-3})	1 (2.39×10^{-3})	1 (2.85×10^{-3})
Linear ALVM	1.06 (9.68×10^{-4})	1.14 (1.09×10^{-3})	6.08 (4.53×10^{-2})	7.19 (2.24×10^{-2})	9.95 (5.31×10^{-2})	11.1 (2.30×10^{-2})
L_2-Norm Pen. Poly. ALVM	1.31 (2.15×10^{-3})	1.06 (9.84×10^{-4})	483 (3.79×10^{1})	1.69 (9.13×10^{-3})	1820 (9.77×10^{1})	1.3 (9.89×10^{-3})
L_1-Norm Pen. Poly. ALVM	1.45 (2.48×10^{-3})	1.18 (1.11×10^{-3})	21.1 (8.22×10^{-1})	1.73 (9.62×10^{-3})	770 (4.26×10^{1})	1.62 (1.29×10^{-2})
Thin-Plate spline ALVM	369 (6.27×10^{0})	1.08 (1.08×10^{-3})	6180 (1.47×10^{2})	148 (6.22×10^{0})	13600 (1.90×10^{2})	8670 (1.49×10^{2})
Miller-Startz SVR	1.32 (1.21×10^{-3})	1.17 (1.08×10^{-3})	1.01 (3.41×10^{-3})	1.04 (3.55×10^{-3})	1.15 (3.00×10^{-3})	1.01 (2.93×10^{-3})

Table E.16: (Relative) MSE of Standard Errors of $\hat{\beta}_{\text{OLS}}$ (with Estimated Standard Error) for One-Covariate Linear Regression Model with Non-Normal Errors

	$\mathcal{H} = \emptyset$		$\mathcal{H} = \{2\}$			
	Homoskedastic		Additive Het.		Mult. Het.	
Model	Laplace	Uniform	Laplace	Uniform	Laplace	Uniform
HC3	1.98 (2.99×10^{-5})	2.15 (4.55×10^{-6})	1.44 (1.94×10^{-4})	1.55 (3.24×10^{-5})	1.23 (2.58×10^{-4})	1.18 (3.56×10^{-5})
HC4	1.91 (2.70×10^{-5})	1.96 (4.21×10^{-6})	1.37 (1.60×10^{-4})	1.43 (3.23×10^{-5})	1.15 (2.07×10^{-4})	1.23 (4.03×10^{-5})
HC6	24 (7.98×10^{-5})	177 (2.76×10^{-5})	12.6 (4.65×10^{-4})	80 (1.91×10^{-4})	8.18 (4.99×10^{-4})	45.6 (2.18×10^{-4})
Homoskedastic	1 (1.36×10^{-5})	1 (2.27×10^{-6})	4.15 (2.82×10^{-4})	17 (9.32×10^{-5})	3.14 (2.90×10^{-4})	11.4 (7.83×10^{-5})
Basic ALVM	1.81 (2.36×10^{-5})	2.04 (4.52×10^{-6})	1.31 (1.55×10^{-4})	1.45 (2.83×10^{-5})	1.23 (2.10×10^{-4})	1.26 (3.98×10^{-5})
Clustering ALVM	1.16 (1.81×10^{-5})	1.36 (3.28×10^{-6})	1.4 (1.85×10^{-4})	1.18 (2.57×10^{-5})	1.1 (1.84×10^{-4})	1 (3.17×10^{-5})
Linear ALVM	1.18 (1.56×10^{-5})	1.33 (2.99×10^{-6})	1 (1.38×10^{-4})	1 (1.94×10^{-5})	1 (1.41×10^{-4})	1.44 (3.87×10^{-5})
L_2-Norm Pen. Poly. ALVM	1.27 (1.78×10^{-5})	1.38 (3.31×10^{-6})	1.33 (1.58×10^{-4})	1.55 (2.67×10^{-5})	1.03 (1.76×10^{-4})	1.01 (2.53×10^{-5})
L_1-Norm Pen. Poly. ALVM	1.43 (1.89×10^{-5})	1.44 (3.22×10^{-6})	1.11 (1.20×10^{-4})	1.56 (2.93×10^{-5})	1.07 (1.67×10^{-4})	1.13 (3.13×10^{-5})
Thin-Plate spline ALVM	2.21 (2.72×10^{-5})	2.46 (5.40×10^{-6})	1.59 (1.43×10^{-4})	2 (4.47×10^{-5})	1.59 (1.76×10^{-4})	2.98 (7.67×10^{-5})
Miller-Startz SVR	21.9 (5.70×10^{-5})	31.2 (3.00×10^{-5})	13.1 (3.02×10^{-4})	20.8 (1.67×10^{-4})	10.2 (3.29×10^{-4})	16.1 (1.91×10^{-4})

E.4.2 Auxiliary Nonlinear Variance Model Results

Table E.17: Relative Performance Metrics (with Estimated SE) for ANLVMs Fit to One-Covariate Linear Regression Model with Non-Normal Errors

		$\boldsymbol{\mathcal{H}} = \emptyset$		$\boldsymbol{\mathcal{H}} = \{2\}$			
		Homoskedastic		Additive Het.		Multiplicative Het.	
Metric	Model	Laplace	Uniform	Laplace	Uniform	Laplace	Uniform
$\overline{\text{MSE}}_{\text{ust}}(\hat{\boldsymbol{\omega}})$	Quadratic	1.42 (3.86×10^{-3})	1.47 (5.50×10^{-4})	0.995 (7.13×10^{-1})	0.591 (4.15×10^{-2})	0.679 (3.27×10^{-1})	1.2 (7.05×10^{-2})
	Exponential	1.47 (4.14×10^{-3})	1.58 (5.89×10^{-4})	1.81 (1.06×10^{0})	1.94 (1.24×10^{-1})	0.979 (1.19×10^{0})	0.58 (7.44×10^{-2})
	Clustering	1.83 (5.54×10^{-3})	2.01 (8.13×10^{-4})	2.12 (1.23×10^{0})	1.93 (7.80×10^{-2})	1.46 (7.72×10^{-1})	1.67 (8.87×10^{-2})
$\overline{\text{MSE}}_{\text{std}}(\hat{\boldsymbol{\omega}})$	Quadratic	1.42 (3.86×10^{-3})	1.47 (5.50×10^{-4})	0.884 (2.44×10^{-2})	0.309 (5.55×10^{-4})	0.537 (1.94×10^{-2})	0.539 (6.31×10^{-4})
	Exponential	1.47 (4.14×10^{-3})	1.58 (5.89×10^{-4})	0.889 (1.74×10^{-2})	0.743 (8.05×10^{-4})	0.39 (1.22×10^{-2})	0.241 (6.10×10^{-4})
	Clustering	1.83 (5.54×10^{-3})	2.01 (8.13×10^{-4})	1.42 (2.07×10^{-2})	1.29 (1.02×10^{-3})	0.925 (1.09×10^{-2})	1.02 (1.01×10^{-3})
$\text{MSE}(\hat{\boldsymbol{\beta}}_{\text{FWLS}})$	Quadratic	1.14 (1.03×10^{-3})	1.06 (1.02×10^{-3})	0.966 (3.17×10^{-3})	0.92 (2.96×10^{-3})	1 (2.55×10^{-3})	1 (2.94×10^{-3})
	Exponential	1.24 (1.11×10^{-3})	1.08 (1.07×10^{-3})	1.02 (3.37×10^{-3})	0.917 (3.11×10^{-3})	1.07 (2.63×10^{-3})	0.945 (2.73×10^{-3})
	Clustering	1.18 (1.06×10^{-3})	1.08 (1.05×10^{-3})	0.953 (3.08×10^{-3})	1.01 (3.14×10^{-3})	0.983 (2.43×10^{-3})	1.06 (2.94×10^{-3})
$\text{MSE}(\text{SE}(\hat{\boldsymbol{\beta}}))$	Quadratic	1.2 (1.75×10^{-5})	1.16 (2.67×10^{-6})	0.961 (1.51×10^{-4})	0.65 (1.15×10^{-5})	0.815 (1.22×10^{-4})	1.49 (3.13×10^{-5})
	Exponential	1.17 (1.66×10^{-5})	1.24 (2.80×10^{-6})	1.42 (1.98×10^{-4})	1.99 (3.56×10^{-5})	0.85 (1.82×10^{-4})	0.679 (2.12×10^{-5})
	Clustering	1.21 (1.79×10^{-5})	1.21 (2.98×10^{-6})	1.35 (1.94×10^{-4})	1.2 (2.42×10^{-5})	1.08 (1.56×10^{-4})	1.03 (3.30×10^{-5})

E.5 Linear Regression with Two Correlated Normal Covariates

Table E.18: (Relative) Unstandardised MSE-of-Variances Estimate (with Estimated Standard Error) for Two-Covariate Linear Regression Model with Multicollinearity

	Homosked.	Add. Het.		Mult. Het.	
Model	$\mathcal{H} = \emptyset$	$\mathcal{H} = \{2\}$	$\mathcal{H} = \{2,3\}$	$\mathcal{H} = \{2\}$	$\mathcal{H} = \{2,3\}$
HC3	106 (8.15×10^{-3})	31.5 (3.75×10^{0})	26.5 (3.32×10^{1})	9.31 (4.47×10^{1})	4.83 (2.94×10^{5})
HC4	95.5 (7.18×10^{-3})	27.8 (3.12×10^{0})	23.5 (2.77×10^{1})	7.86 (3.47×10^{1})	4.04 (2.31×10^{5})
HC6	41.9 (6.36×10^{-4})	12.8 (8.16×10^{-1})	10.7 (6.07×10^{0})	4.63 (2.89×10^{1})	3.36 (2.66×10^{5})
Homoskedastic	1 (2.90×10^{-4})	2.93 (1.03×10^{-1})	2.66 (9.84×10^{-1})	2.49 (6.75×10^{-1})	1.92 (3.22×10^{3})
Basic ALVM	106 (7.99×10^{-3})	31.8 (3.87×10^{0})	26.6 (3.33×10^{1})	9.27 (4.32×10^{1})	4.84 (3.17×10^{5})
Clustering ALVM	3.92 (1.27×10^{-3})	2.88 (6.72×10^{-1})	2.88 (5.95×10^{0})	1.2 (7.20×10^{0})	1.37 (4.96×10^{4})
Linear ALVM	2.17 (5.47×10^{-4})	1 (2.37×10^{-1})	1 (1.69×10^{0})	1.45 (1.59×10^{0})	1.54 (7.77×10^{3})
L_2-Norm Pen. Poly. ALVM	2.74 (7.45×10^{-4})	1.47 (3.20×10^{-1})	1.07 (3.08×10^{0})	1 (4.55×10^{0})	1 (1.85×10^{4})
Thin-Plate spline ALVM	5.46 (9.44×10^{-4})	1.86 (5.80×10^{-1})	1.5 (4.98×10^{0})	1.15 (6.05×10^{0})	1.2 (8.40×10^{4})
Miller-Startz SVR	19.5 (9.22×10^{-4})	6.41 (3.57×10^{-1})	5.43 (3.23×10^{0})	2.34 (3.20×10^{0})	1.5 (1.37×10^{4})

Table E.19: (Relative) Standardised MSE-of-Variances Estimate (with Estimated Standard Error) for Two-Covariate Linear Regression Model with Multicollinearity

	Homosked.	Add. Het.		Mult. Het.	
Model	$\mathcal{H} = \emptyset$	$\mathcal{H} = \{2\}$	$\mathcal{H} = \{2,3\}$	$\mathcal{H} = \{2\}$	$\mathcal{H} = \{2,3\}$
HC3	106 (8.15×10^{-3})	12.9 (8.77×10^{-3})	11.4 (8.73×10^{-3})	7.26 (1.62×10^{-2})	37.4 (1.00×10^{0})
HC4	95.5 (7.18×10^{-3})	11.6 (7.65×10^{-3})	10.1 (7.33×10^{-3})	6.35 (1.32×10^{-2})	29.5 (7.39×10^{-1})
HC6	41.9 (6.36×10^{-4})	4.98 (8.60×10^{-4})	4.38 (8.06×10^{-4})	2.4 (1.38×10^{-3})	1 (7.74×10^{-3})
Homoskedastic	1 (2.90×10^{-4})	8.92 (6.13×10^{-3})	9.34 (7.46×10^{-3})	36.2 (6.57×10^{-2})	2190 (1.62×10^{1})
Basic ALVM	106 (7.99×10^{-3})	12.9 (8.72×10^{-3})	11.3 (8.44×10^{-3})	6.72 (1.33×10^{-2})	18.3 (5.74×10^{-1})
Clustering ALVM	3.92 (1.27×10^{-3})	1.66 (3.02×10^{-3})	1.67 (2.27×10^{-3})	1 (5.33×10^{-3})	15.4 (4.79×10^{-1})
Linear ALVM	2.17 (5.47×10^{-4})	1 (2.73×10^{-3})	1.11 (2.02×10^{-3})	5.51 (2.01×10^{-2})	299 (2.90×10^{0})
L_2-Norm Pen. Poly. ALVM	2.74 (7.45×10^{-4})	2.09 (4.17×10^{-3})	1.56 (3.65×10^{-3})	14.4 (1.67×10^{-1})	2940 (7.19×10^{1})
Thin-Plate spline ALVM	5.46 (9.44×10^{-4})	1.05 (1.74×10^{-3})	1 (2.18×10^{-3})	4.65 (1.92×10^{-2})	1420 (2.16×10^{1})
Miller-Startz SVR	19.5 (9.22×10^{-4})	2.34 (8.81×10^{-4})	2.05 (8.93×10^{-4})	1.18 (1.77×10^{-3})	9.9 (3.98×10^{-1})

Table E.20: (Relative) MSE of FWLS Estimate of $\boldsymbol{\beta}$ (with Estimated Standard Error) for Two-Covariate Linear Regression Model with Multicollinearity

	Homosked.	Add. Het.		Mult. Het.	
Model	$\boldsymbol{\mathcal{H}} = \emptyset$	$\boldsymbol{\mathcal{H}} = \{2\}$	$\boldsymbol{\mathcal{H}} = \{2,3\}$	$\boldsymbol{\mathcal{H}} = \{2\}$	$\boldsymbol{\mathcal{H}} = \{2,3\}$
OLS	1 (5.67×10^{-4})	1.25 (7.44×10^{-3})	1.33 (2.11×10^{-2})	2.38 (1.78×10^{-2})	9.6 (9.45×10^{-1})
HC3	1.05 (6.08×10^{-4})	1.25 (7.58×10^{-3})	1.29 (2.04×10^{-2})	2.3 (1.73×10^{-2})	8.52 (8.76×10^{-1})
HC4	1.03 (5.98×10^{-4})	1.29 (7.91×10^{-3})	1.32 (2.13×10^{-2})	2.25 (1.72×10^{-2})	8.59 (8.62×10^{-1})
HC6	1.06 (6.15×10^{-4})	1.29 (7.81×10^{-3})	1.34 (2.13×10^{-2})	2.37 (1.78×10^{-2})	9.54 (9.29×10^{-1})
Homoskedastic	1 (5.67×10^{-4})	1.25 (7.44×10^{-3})	1.33 (2.11×10^{-2})	2.38 (1.78×10^{-2})	9.6 (9.45×10^{-1})
Basic ALVM	1.14 (6.53×10^{-4})	1.33 (8.34×10^{-3})	1.43 (2.36×10^{-2})	2.19 (1.71×10^{-2})	6.71 (8.61×10^{-1})
Clustering ALVM	1.34 (6.43×10^{-3})	3.81 (2.56×10^{-1})	2.3 (1.21×10^{-1})	8.7 (3.88×10^{-1})	6.86 (2.27×10^{0})
Linear ALVM	1.29 (4.33×10^{-3})	1.63 (1.47×10^{-2})	2.47 (4.76×10^{-2})	1.35 (1.56×10^{-2})	1 (1.04×10^{-1})
L_2-Norm Pen. Poly. ALVM	1.36 (2.09×10^{-3})	2.1 (1.48×10^{-1})	153 (1.48×10^{2})	5630 (2.56×10^{3})	117000 (3.80×10^{5})
Thin-Plate spline ALVM	5.33 (1.14×10^{-1})	11.3 (4.91×10^{-1})	9.76 (7.96×10^{-1})	60.9 (1.58×10^{0})	87.5 (1.65×10^{1})
Miller-Startz SVR	1.11 (6.28×10^{-4})	1 (6.04×10^{-3})	1 (1.61×10^{-2})	1 (7.93×10^{-3})	1.17 (1.77×10^{-1})

Table E.21: (Relative) MSE of Standard Errors of $\hat{\boldsymbol{\beta}}_{\mathrm{OLS}}$ (with Estimated Standard Error) for Two-Covariate Linear Regression Model with Multicollinearity

	Homosked.	Add. Het.		Mult. Het.	
Model	$\boldsymbol{\mathcal{H}} = \emptyset$	$\boldsymbol{\mathcal{H}} = \{2\}$	$\boldsymbol{\mathcal{H}} = \{2,3\}$	$\boldsymbol{\mathcal{H}} = \{2\}$	$\boldsymbol{\mathcal{H}} = \{2,3\}$
HC3	3.35 (1.05×10^{-5})	2.31 (1.63×10^{-4})	2.31 (4.77×10^{-4})	1.64 (7.59×10^{-4})	1.81 (8.64×10^{-2})
HC4	3.01 (8.67×10^{-6})	2.09 (1.27×10^{-4})	2.06 (3.64×10^{-4})	1.51 (6.02×10^{-4})	1.72 (7.23×10^{-2})
HC6	67.4 (3.76×10^{-5})	37.1 (6.06×10^{-4})	38.8 (1.73×10^{-3})	10.3 (1.85×10^{-3})	5.26 (1.07×10^{-1})
Homoskedastic	1 (3.35×10^{-6})	3.19 (1.37×10^{-4})	4.06 (5.26×10^{-4})	1.38 (2.27×10^{-4})	1 (2.87×10^{-2})
Basic ALVM	3.2 (9.74×10^{-6})	2.28 (1.52×10^{-4})	2.23 (4.15×10^{-4})	1.55 (6.62×10^{-4})	1.86 (8.50×10^{-2})
Clustering ALVM	1.61 (6.43×10^{-6})	1.57 (1.12×10^{-4})	1.53 (3.02×10^{-4})	1.15 (5.09×10^{-4})	1.68 (4.24×10^{-2})
Linear ALVM	1.42 (4.64×10^{-6})	1 (7.75×10^{-5})	1 (2.02×10^{-4})	1 (3.35×10^{-4})	1.37 (3.54×10^{-2})
L_2-Norm Pen. Poly. ALVM	1.56 (5.65×10^{-6})	1.33 (8.87×10^{-5})	1.31 (2.85×10^{-4})	1.41 (7.20×10^{-4})	1.7 (6.13×10^{-2})
Thin-Plate spline ALVM	3.45 (9.61×10^{-6})	1.56 (1.10×10^{-4})	1.5 (3.14×10^{-4})	1.45 (5.35×10^{-4})	1.56 (6.22×10^{-2})
Miller-Startz SVR	29.3 (2.99×10^{-5})	18.6 (4.08×10^{-4})	19.1 (1.12×10^{-3})	9.37 (1.31×10^{-3})	7.49 (9.30×10^{-2})

E.6 Linear Regression with Eight Correlated Normal Covariates

Table E.22: (Relative) Unstandardised MSE-of-Variances Estimate (with Estimated Standard Error) for Eight-Covariate Linear Regression Model with Multicollinearity

Model	Homosked.	Add. Het.		Mult. Het.	
	$\mathcal{H} = \emptyset$	$\mathcal{H} = \{2\}$	$\mathcal{H} = \{2,3,4,5\}$	$\mathcal{H} = \{2\}$	$\mathcal{H} = \{2,3,4,5\}$
HC3	115 (9.75×10^{-3})	21.1 (4.50×10^{0})	18.3 (4.98×10^{2})	6.65 (7.02×10^{1})	2.72 $(2.24 \times 10^{+13})$
HC4	81.7 (6.47×10^{-3})	15.1 (2.86×10^{0})	13 (3.26×10^{2})	4.88 (4.90×10^{1})	1.88 $(1.46 \times 10^{+13})$
HC6	39.1 (5.46×10^{-4})	7.43 (4.28×10^{-1})	6.36 (4.05×10^{1})	2.7 (2.37×10^{1})	1.35 $(9.20 \times 10^{+12})$
Homoskedastic	1 (3.10×10^{-4})	1.7 (1.17×10^{-1})	1.35 (1.38×10^{1})	1.82 (1.07×10^{0})	1.22 $(2.72 \times 10^{+11})$
Basic ALVM	112 (9.26×10^{-3})	20.9 (4.43×10^{0})	18 (4.84×10^{2})	6.78 (7.65×10^{1})	2.59 $(2.18 \times 10^{+13})$
Clustering ALVM	9.53 (2.01×10^{-3})	2.49 (9.05×10^{-1})	2.43 (1.22×10^{2})	1.39 (1.66×10^{1})	1.4 $(7.19 \times 10^{+12})$
Linear ALVM	5.42 (9.38×10^{-4})	1 (3.45×10^{-1})	1 (3.44×10^{1})	1 (3.20×10^{0})	1.12 $(6.63 \times 10^{+11})$
L_2-Norm Pen. Poly. ALVM	24.6 (5.75×10^{-3})	5.69 (2.12×10^{0})	4.43 (2.01×10^{2})	2.26 (2.91×10^{1})	1.07 $(1.06 \times 10^{+12})$
Miller-Startz SVR	22.4 (1.01×10^{-3})	4.48 (3.79×10^{-1})	3.72 (4.55×10^{1})	1.93 (3.25×10^{0})	1 $(8.57 \times 10^{+11})$

Table E.23: (Relative) Standardised MSE-of-Variances Estimate (with Estimated Standard Error) for Eight-Covariate Linear Regression Model with Multicollinearity

Model	Homosked.	Add. Het.		Mult. Het.	
	$\mathcal{H} = \emptyset$	$\mathcal{H} = \{2\}$	$\mathcal{H} = \{2,3,4,5\}$	$\mathcal{H} = \{2\}$	$\mathcal{H} = \{2,3,4,5\}$
HC3	115 (9.75×10^{-3})	11.5 (1.05×10^{-2})	8.82 (1.08×10^{-2})	7.68 (1.77×10^{-2})	209 (2.30×10^{5})
HC4	81.7 (6.47×10^{-3})	8.09 (6.84×10^{-3})	6.19 (6.94×10^{-3})	5.21 (1.08×10^{-2})	122 (1.30×10^{5})
HC6	39.1 (5.46×10^{-4})	3.73 (5.40×10^{-4})	2.83 (5.60×10^{-4})	1.95 (7.03×10^{-4})	1 (1.56×10^{3})
Homoskedastic	1 (3.10×10^{-4})	2.71 (2.93×10^{-3})	3.36 (4.29×10^{-3})	17.3 (4.30×10^{-2})	2620 (1.12×10^{6})
Basic ALVM	112 (9.26×10^{-3})	11.2 (1.02×10^{-2})	8.53 (1.01×10^{-2})	6.91 (1.55×10^{-2})	84.9 (1.28×10^{5})
Clustering ALVM	9.53 (2.01×10^{-3})	1.62 (3.16×10^{-3})	1.47 (2.79×10^{-3})	1.31 (6.91×10^{-3})	46.7 (2.45×10^{5})
Linear ALVM	5.42 (9.38×10^{-4})	1 (2.19×10^{-3})	1 (2.19×10^{-3})	2.97 (1.29×10^{-2})	482 (3.58×10^{5})
L_2-Norm Pen. Poly. ALVM	24.6 (5.75×10^{-3})	4.46 (8.86×10^{-3})	3.67 (9.70×10^{-3})	18.9 (1.69×10^{-1})	248 (9.67×10^{5})
Miller-Startz SVR	22.4 (1.01×10^{-3})	2.03 (1.13×10^{-3})	1.54 (1.18×10^{-3})	1 (1.60×10^{-3})	39.2 (4.69×10^{4})

Table E.24: (Relative) MSE of FWLS Estimate of $\boldsymbol{\beta}$ (with Estimated Standard Error) for Eight-Covariate Linear Regression Model with Multicollinearity

Model	Homosked.	Add. Het.		Mult. Het.	
	$\mathcal{H} = \emptyset$	$\mathcal{H} = \{2\}$	$\mathcal{H} = \{2,3,4,5\}$	$\mathcal{H} = \{2\}$	$\mathcal{H} = \{2,3,4,5\}$
OLS	1 (2.36×10^{-3})	1.02 (4.00×10^{-2})	1.09 (4.61×10^{-1})	1.62 (8.72×10^{-2})	10 (1.47×10^{4})
HC3	1.07 (2.58×10^{-3})	1.08 (4.35×10^{-2})	1.1 (4.58×10^{-1})	1.5 (8.17×10^{-2})	6.78 (1.00×10^{4})
HC4	1.04 (2.43×10^{-3})	1.09 (4.21×10^{-2})	1.09 (4.60×10^{-1})	1.49 (8.21×10^{-2})	6.88 (1.00×10^{4})
HC6	1.07 (2.49×10^{-3})	1.13 (4.43×10^{-2})	1.13 (4.69×10^{-1})	1.55 (8.44×10^{-2})	8.45 (1.25×10^{4})
Homoskedastic	1 (2.36×10^{-3})	1.02 (4.00×10^{-2})	1.09 (4.61×10^{-1})	1.62 (8.72×10^{-2})	10 (1.47×10^{4})
Basic ALVM	1.18 (2.74×10^{-3})	1.23 (5.07×10^{-2})	1.25 (5.12×10^{-1})	1.63 (9.06×10^{-2})	4.92 (7.83×10^{3})
Clustering ALVM	1.13 (3.00×10^{-3})	1.13 (5.73×10^{-2})	1.22 (9.79×10^{-1})	1.43 (4.14×10^{-1})	2.69 (1.72×10^{4})
Linear ALVM	1.27 (3.39×10^{-3})	1.36 (6.12×10^{-2})	1.36 (6.43×10^{-1})	1.32 (8.65×10^{-2})	2.9 (4.07×10^{3})
L_2-Norm Pen. Poly. ALVM	439 (3.78×10^{1})	604 (1.37×10^{3})	85600 (2.71×10^{6})	705 (2.02×10^{3})	2.12 (3.42×10^{3})
Miller-Startz SVR	1.04 (2.45×10^{-3})	1 (3.96×10^{-2})	1 (4.26×10^{-1})	1 (5.48×10^{-2})	1 (2.10×10^{3})

Table E.25: (Relative) MSE of Standard Errors of $\hat{\boldsymbol{\beta}}_{\text{OLS}}$ (with Estimated Standard Error) for Eight-Covariate Linear Regression Model with Multicollinearity

Model	Homosked.	Add. Het.		Mult. Het.	
	$\mathcal{H} = \emptyset$	$\mathcal{H} = \{2\}$	$\mathcal{H} = \{2,3,4,5\}$	$\mathcal{H} = \{2\}$	$\mathcal{H} = \{2,3,4,5\}$
HC3	3.53 (5.14×10^{-5})	3.1 (1.01×10^{-3})	3.21 (1.04×10^{-2})	2.35 (4.21×10^{-3})	1.46 (1.77×10^{3})
HC4	2.73 (3.23×10^{-5})	2.59 (6.69×10^{-4})	2.6 (6.85×10^{-3})	2.05 (2.67×10^{-3})	1.25 (1.16×10^{3})
HC6	70.7 (1.31×10^{-4})	54.1 (2.69×10^{-3})	56.8 (2.72×10^{-2})	21.4 (8.02×10^{-3})	4.11 (1.71×10^{3})
Homoskedastic	1 (1.34×10^{-5})	1 (2.89×10^{-4})	1 (3.09×10^{-3})	1 (1.14×10^{-3})	1 (1.14×10^{3})
Basic ALVM	2.95 (4.04×10^{-5})	2.75 (8.30×10^{-4})	2.78 (8.13×10^{-3})	2.2 (3.34×10^{-3})	1.91 (1.90×10^{3})
Clustering ALVM	1.16 (1.56×10^{-5})	1.25 (3.84×10^{-4})	1.23 (3.81×10^{-3})	1.47 (2.54×10^{-3})	1.68 (2.17×10^{3})
Linear ALVM	1.07 (1.43×10^{-5})	1.14 (3.60×10^{-4})	1.26 (3.93×10^{-3})	1.41 (2.43×10^{-3})	1.82 (2.29×10^{3})
L_2-Norm Pen. Poly. ALVM	2.28 (5.69×10^{-5})	2.48 (1.20×10^{-3})	2 (9.97×10^{-3})	3.24 (7.52×10^{-3})	2.78 (3.30×10^{3})
Miller-Startz SVR	28.9 (1.07×10^{-4})	24.7 (1.97×10^{-3})	25.7 (2.08×10^{-2})	14.5 (4.75×10^{-3})	4.74 (1.44×10^{3})

Bibliography

Adamec, V. (2017), 'Power of Heteroskedasticity Tests in Presence of Various Types of Skedastic Function and Sample Size', *AIP Conference Proceedings* **1863**(1).

Aftab, N. and Chand, S. (2016), 'A New Heteroskedastic Consistent Covariance Matrix Estimator Using Deviance Measure', *Pakistan Journal of Statistics and Operations Research* **12**(2), 235–244.

Aftab, N. and Chand, S. (2018), 'A Simulation-Based Evidence on the Improved Performance of a New Modified Leverage Adjusted Heteroskedastic Consistent Covariance Matrix Estimator in the Linear Regression Model', *Kuwait Journal of Science* **45**(3), 29–38.

Anscombe, F. (1961), Examination of Residuals, *in* J. Neyman, ed., 'Fourth Berkeley Symposium on Mathematical Statistics and Probability June 20-July 30, 1960', Berkeley: University of California Press, pp. 1–36.

Anselin, L. and Bera, A. K. (1998), Spatial Dependence in Linear Regression Models with an Introduction to Spatial Econometrics, *in* A. Ullah and D. E. A. Giles, eds, 'Handbook of Applied Economic Statistics', Marcel Dekker, New York, pp. 237–290.

Bai, Z., Pan, G. and Yin, Y. (2016), 'Homoscedasticity Tests for Both Low and High-Dimensional Fixed Design Regressions'. 1603.03830.

Balakrishnan, N. and Lai, C.-D. (2009), *Continuous Bivariate Distributions*, 2nd edn, Springer, New York.

Bartlett, M. S. (1937), 'Properties of Sufficiency and Statistical Tests', *Proceedings of the Royal Society of London Series A* **260**, 268–282.

Berry, W. D. (1993), *Understanding Regression Assumptions*, Sage, Newbury Park, CA.

Best, M. J. (2017), *Quadratic Programming with Computer Programs*, CRC, Boca Raton, FL.

Bickel, P. (1978), 'Using Residuals Robustly I: Tests for Heteroscedasticity, Nonlinearity', *The Annals of Statistics* **6**(2), 266–291.

Borchers, H. W. (2022), *pracma: Practical Numerical Math Functions*. R package version 2.3.8.
URL: *https://CRAN.R-project.org/package=pracma*

Boyd, S. and Vandenberghe, L. (2004), *Convex Optimization*, Cambridge University Press, Cambridge.

Breusch, T. and Pagan, A. (1979), 'A Simple Test for Heteroscedasticity and Random Coefficient Variation', *Econometrica* **47**(5), 1287–1294.

Bruce, P. and Bruce, A. (2017), *Practical Statistics for Data Scientists: 50 Essential Concepts*, O'Reilly, Sebastopol, CA.

Carapeto, M. and Holt, W. (2003), 'Testing for Heteroscedasticity in Regression Models', *Journal of Applied Statistics* **30**(1), 13–20.

Carroll, R. J. and Ruppert, D. (1981), 'On Robust Tests for Heteroscedasticity', *The Annals of Statistics* **9**(1), 206–210.

Çelik, R. (2017), 'A New Test to Detect Monotonic and Non-Monotonic Types of Heteroscedasticity', *Journal of Applied Statistics* **44**(2), 342–361.

Çelik, R. (2018), 'RCEV Heteroscedasticity Test Based on the Studentized Residuals', *Communications in Statistics - Theory and Methods* **48**(13), 3258–3268.

Charrad, M., Ghazzali, N., Boiteau, V. and Niknafs, A. (2014), 'NbClust: An R Package for Determining the Relevant Number of Clusters in a Data Set', *Journal of Statistical Software* **61**(6), 1–36.
URL: *http://www.jstatsoft.org/v61/i06/*

Chen, J. M. (2021), 'An Introduction to Machine Learning for Panel Data', *International Advances in Economic Research* **27**, 1–16.

Cheng, T.-C. (2012), 'On Simultaneously Identifying Outliers and Heteroscedasticity without Specific Form', *Computational Statistics & Data Analysis* **56**(7), 2258–2272.

Chernick, M. R. (2008), *Bootstrap Methods: A Guide for Practitioners and Researchers*, 2nd edn, Wiley, New York.

Cho, H. and Fryzlewicz, P. (2012), 'High Dimensional Variable Selection via Tilting', *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* **74**(3), 593–622.

Christopoulos, D. T. (2019), *inflection: Finds the Inflection Point of a Curve*. R package version 1.3.5.
URL: *https://CRAN.R-project.org/package=inflection*

Churchill, W. S. (1943), *The End of the Beginning*, Cassell, London.

Cook, R. D. (1977), 'Detection of Influential Observation in Linear Regression', *Technometrics* **19**(1), 15–18.

Cook, R. D. (1979), 'Influential Observations in Linear Regression', *Journal of the American Statistical Association* **74**(365), 169–174.

Cook, R. D. and Weisberg, S. (1982), *Residuals and Influence in Regression*, Chapman & Hall, New York.

Cook, R. D. and Weisberg, S. (1983), 'Diagnostics for Heteroscedasticity in Regression', *Biometrika* **70**(1), 1–10.

Cribari-Neto, F. (2004), 'Asymptotic Inference under Heteroskedasticity of Unknown Form', *Computational Statistics & Data Analysis* **45**, 215–233.

Cribari-Neto, F. and da Silva, W. B. (2011), 'A New Heteroskedasticity-Consistent Covariance Matrix Estimator for the Linear Regression Model', *Advances in Statistical Analysis* **95**(2), 129–146.

Cribari-Neto, F., Souza, T. C. and Vasconcellos, K. L. P. (2007), 'Inference under Heteroskedasticity and Leveraged Data', *Communications in Statistics - Theory and Methods* **36**(10), 1877–1888.

Cribari-Neto, F. and Zarkos, S. G. (1999), 'Bootstrap Methods for Heteroskedastic Regression Models: Evidence on Estimation and Testing', *Econometric Reviews* **18**(2), 211–228.

Davidson, R. and Flachaire, E. (2008), 'The Wild Bootstrap, Tamed at Last', *Journal of Econometrics* **146**, 162–169.

Davidson, R. and MacKinnon, J. G. (2004), *Econometric Theory and Methods*, Oxford University Press, New York.

Davies, R. (1980), 'Algorithm AS 155: The Distribution of a Linear Combination of χ^2 Random Variables', *Journal of the Royal Statistical Society. Series C (Applied Statistics)* **29**, 323–333.

Diblasi, A. and Bowman, A. (1997), 'Testing for Constant Variance in a Linear Model', *Statistics & Probability Letters* **33**, 95–103.

Duchesne, P. and de Micheaux, P. L. (2010), 'Computing the Distribution of Quadratic Forms: Further Comparisons between the Liu-Tang-Zhang Approximation and Exact Methods', *Computational Statistics and Data Analysis* **54**, 858–862.

Dufour, J.-M., Khalaf, L., Bernard, J.-T. and Genest, I. (2004), 'Simulation-Based Finite-Sample Tests for Heteroskedasticity and ARCH Effects', *Journal of Econometrics* **122**, 317–347.

Efron, B., Hastie, T., Johnstone, I. and Tibshirani, R. (2004), 'Least Angle Regression', *The Annals of Statistics* **32**, 407–451.

Efron, B. and Tibshirani, R. (1997), 'Improvements on Cross-Validation: The .632+ Bootstrap Method', *Journal of the American Statistical Association* **92**(438), 548–560.

Efron, B. and Tibshirani, R. J. (1993), *An Introduction to the Bootstrap*, Springer Science+Business Media, Dordrecht.

Einstein, A. ([1936] 2003), 'Physics & Reality', *Daedalus* **132**(4), 22–25.

Escobar, L. A. and Skarpness, B. (1984), 'A Closed Form Solution for the Least Squares Regression Problem with Linear Inequality Constraints', *Communications in Statistics - Theory and Methods* **13**, 1127–1134.

Evans, M. (1992), 'Robustness of Size of Tests of Autocorrelation and Heteroscedasticity to Nonnormality', *Journal of Econometrics* **51**(1–2), 7–24.

Evans, M. A. and King, M. L. (1988), 'A Further Class of Tests for Heteroscedasticity', *Journal of Econometrics* **37**, 265–276.

Evans, M. and King, M. L. (1985), 'A Point Optimal Test for Heteroscedastic Disturbances', *Journal of Econometrics* **27**(2), 163–178.

Ferrari, S. L. and Cribari-Neto, F. (2002), 'Corrected Modified Profile Likelihood Heteroskedasticity Tests', *Statistics & Probability Letters* **57**, 353–361.

Ferrari, S. L., Cysneiros, A. H. and Cribari-Neto, F. (2004), 'An Improved Test for Heteroskedasticity Using Adjusted Modified Profile Likelihood Inference', *Journal of Statistical Planning and Inference* **124**, 423–437.

Fidell, L. S. and Tabachnick, B. G. (2003), Preparatory Data Analysis, *in* I. B. Weiner, J. A. Schinka and W. F. Velicer, eds, 'Handbook of Psychology: Research Methods in Psychology', Vol. 2, Wiley, Hoboken, NJ, pp. 115–142.

Fox, J. and Weisberg, S. (2019), *An R Companion to Applied Regression*, 3rd edn, Sage, Thousand Oaks, CA.
URL: *https://socialsciences.mcmaster.ca/jfox/Books/Companion/*

Fox, W. P. (2021), *Nonlinear Optimization: Models and Applications*, 1st edn, Chapman and Hall/CRC, Boca Raton, FL.

Fuller, W. A. and Rao, J. N. K. (1978), 'Estimation for a Linear Regression Model with Unknown Diagonal Covariance Matrix', *The Annals of Statistics* **6**(5), 1149–1158.

Gaines, B. R., Kim, J. and Zhou, H. (2018), 'Algorithms for Fitting the Constrained LASSO', *Journal of Computational and Graphical Statistics* **27**(4), 861–871.

Gerhard, D. and Kuiper, R. M. (2021), *goric: Generalized Order-Restricted Information Criterion*. R package version 1.1-2.
URL: *https://CRAN.R-project.org/package=goric*

Geweke, J. (1986), 'Inference in the Inequality Constrained Normal Linear Regression Model', *Journal of Applied Econometrics* **1**(2), 127–141.

Gilley, O. W. and Pace, R. K. (1996), 'On the Harrison and Rubinfeld Data', *Journal of Environmental Economics and Management* **31**, 403–405.

Glejser, H. (1969), 'A New Test for Heteroskedasticity', *Journal of the American Statistical Association* **64**(325), 316–323.

Godfrey, L. and Orme, C. (1999), 'The Robustness, Reliability and Power of Heteroskedasticity Tests', *Econometric Reviews* **18**(2), 169–194.

Godfrey, L., Orme, C. and Silva, J. S. (2006), 'Simulation-Based Tests for Heteroskedasticity in Linear Regression Models: Some Further Results', *Econometrics Journal* **9**, 76–97.

Goldfeld, S. M. and Quandt, R. E. (1965), 'Some Tests for Homoscedasticity', *Journal of the American Statistical Association* **60**(310), 539–547.

Greenbaum, A. (1997), *Iterative Methods for Solving Linear Systems*, Society for Industrial Applied Mathematics, Philadelphia.

Greene, W. H. (2012), *Econometric Analysis*, 7th edn, Pearson, Essex.

Griffiths, W. and Surekha, K. (1986), 'A Monte Carlo Evaluation of the Power of Some Tests for Heteroscedasticity', *Journal of Econometrics* **31**(1), 219–231.

Gujarati, D. N. (2018), *Linear Regression: A Mathematical Introduction*, Sage, London.

Gut, A. (2005), *Probability: A Graduate Course*, Springer, New York.

Gut, A. (2009), *The Multivariate Normal Distribution*, Springer, New York, chapter 5, pp. 117–145.

Hampel, F. R. (1974), 'The Influence Curve and Its Role in Robust Estimation', *Journal of the American Statistical Association* **69**(346), 383–393.

Harlow, B. (2020), *quadprogXT: Quadratic Programming with Absolute Value Constraints*. R package version 0.0.5.
URL: *https://CRAN.R-project.org/package=quadprogXT*

Harrison, D. and Rubinfeld, D. (1978), 'Hedonic Prices and the Demand for Clean Air', *Journal of Environmental Economics and Management* **5**, 81–102.

Harrison, M. and McCabe, B. (1979), 'A Test for Heteroscedasticity Based on Ordinary Least Squares Residuals', *Journal of the American Statistical Association* **74**(366), 494–499.

Harvey, A. C. (1976), 'Estimating Regression Models with Multiplicative Heteroscedasticity', *Econometrica* **44**(3), 461–465.

Hastie, T., Tibshirani, R. and Friedman, J. H. (2009), *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd edn, Springer, New York.

Hastie, T., Tibshirani, R. and Tibshirani, R. (2020), 'Best Subset, Forward Stepwise or Lasso? Analysis and Recommendations Based on Extensive Comparisons', *Statistical Science* **35**(4), 579 – 592.

Heij, C., de Boer, P., Franses, P. H., Kloek, T. and van Dijk, H. K. (2004), *Econometric Methods with Applications in Business and Economics*, Oxford University Press, Oxford.

Henderson, C. R. (1975), 'Best Linear Unbiased Estimation and Prediction under a Selection Model', *Biometrics* **31**(2), 423–447.

Hesterberg, T. (1999), Bootstrap tilting confidence intervals, Technical Report 84, MathSoft, Inc.

Hesterberg, T. (2011), 'Bootstrap', *Wiley Interdisciplinary Reviews: Computational Statistics* **3**(6), 497–526.

Hesterberg, T. (2015), 'What Teachers Should Know about the Bootstrap: Resampling in the Undergraduate Statistics Curriculum'.

Hinkley, D. V. (1977), 'Jackknifing in Unbalanced Situations', *Technometrics* **19**(3), 285–292.

Honda, Y. (1989), 'On the Optimality of Some Tests of the Error Covariance Matrix in the Linear Regression Model', *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* **51**(1), 71–79.

Hooper, P. M. (1993), 'Iterative Weighted Least Squares Estimation in Heteroscedastic Linear Models', *Journal of the American Statistical Association* **88**(421), 179–184.

Horn, P. (1981), 'Heteroscedasticity of Residuals: A Non-Parametric Alternative to the Goldfeld-Quandt Peak Test', *Communications in Statistics - Theory and Methods* **10**(8), 795–808.

Horn, S. D., Horn, R. A. and Duncan, D. B. (1975), 'Estimating Heteroscedastic Variances in Linear Models', *Journal of the American Statistical Association* **70**, 380–385.

Huang, C. J. and Bolch, B. W. (1974), 'On the Testing of Regression Disturbances for Normality', *Journal of the American Statistical Association* **69**(346), 330–335.

Imhof, J. (1961), 'Computing the Distribution of Quadratic Forms in Normal Variables', *Biometrika* **48**(3/4), 419–426.

James, G. M., Paulson, C. and Rusmevichientong, P. (2020), 'Penalized and Constrained Optimization: An Application to High-Dimensional Website Advertising', *Journal of the American Statistical Association* **115**, 107–122.

James, G., Witten, D., Hastie, T. and Tibshirani, R. (2013), *An Introduction to Statistical Learning: with Applications in R*, Springer, New York.

Kalirajan, K. (1989), 'A Test for Heteroscedasticity and Non-Normality of Regression Residuals: A Practical Approach', *Economics Letters* **30**(2), 133–136.

Kalirajan, K. and Jayasuriya, S. (1991), 'Simultaneous Testing of Regression Disturbances for Heteroscedasticity and Non-Normality', *Journal of Applied Statistics* **18**(3), 307–312.

Kauermann, G., Claeskens, G. and Opsomer, J. D. (2009), 'Bootstrapping for Penalized Spline Regression', *Journal of Computational and Graphical Statistics* **18**(1), 126–146.

Kibble, W. F. (1941), 'A Two-Variate Gamma Type Distribution', *Sankhyā: The Indian Journal of Statistics* **5**(2), 137–150.

Knottnerus, P. (2016), 'On New Variance Approximations for Linear Models with Inequality Constraints', *Statistica Neerlandica* **70**(1), 26–46.

Koenker, R. (1981), ‘A Note on Studentizing a Test for Heteroscedasticity’, *Journal of Econometrics* **17**, 107–112.

Koenker, R. (2020), *quantreg: Quantile Regression.* R package version 5.61.
URL: *https://CRAN.R-project.org/package=quantreg*

Kotz, S., Balakrishnan, N. and Johnson, N. L. (2000), *Continuous Multivariate Distributions: Volume 1: Models and Applications*, 2nd edn, Wiley, New York.

Krishnamoorthy, A. S. and Parthasarathy, M. (1951), ‘A Multivariate Gamma-Type Distribution’, *The Annals of Mathematical Statistics* **22**(4), 549–557.

Kulinskaya, E. (2008), ‘On Two-Sided p-Values for Non-Symmetric Distributions’. 0810.2124.

Lange, K. (2010), *Numerical Analysis for Statisticians*, 2nd edn, Springer, New York.

Laurin, C., Boomsma, D. and Lubke, G. (2016), ‘The Use of Vector Bootstrapping to Improve Variable Selection Precision in Lasso Models’, *Statistical Applications in Genetics and Molecular Biology* **15**(4), 305–320.

Lehmann, E. L. (1975), *Nonparametrics: Statistical Methods Based on Ranks*, Holden-Day, San Francisco.

Leisch, F. and Dimitriadou, E. (2010), *mlbench: Machine Learning Benchmark Problems.* R package version 2.1-1.
URL: *https://cran.r-project.org/package=mlbench*

Li, S., Zhang, N., Zhang, X. and Wang, G. (2017), ‘A New Heteroskedasticity-Consistent Covariance Matrix Estimator and Inference under Heteroskedasticity’, *Journal of Statistical Computation and Simulation* **87**, 198–210.

Li, Z. and Yao, J. (2019), ‘Testing for Heteroscedasticity in High-Dimensional Regressions’, *Econometrics and Statistics* **9**, 122–139.

Liew, C. K. (1976), ‘Inequality Constrained Least-Squares Estimation’, *Journal of the American Statistical Association* **71**(355), 746–751.

Liu, S. (2016), *Computational and Statistical Methods for Analysing Big Data with Applications*, Elsevier, Amsterdam.

Lloyd, C. J. (2005), ‘Estimating Test Power Adjusted for Size’, *Journal of Statistical Computation and Simulation* **75**(11), 921–934.

Long, J. S. and Ervin, L. H. (2000), ‘Using Heteroscedasticity Consistent Standard Errors in the Linear Regression Model’, *The American Statistician* **54**, 217–224.

Lucas, A., Scholz, I., Boehme, R., Jasson, S. and Maechler, M. (2020), *gmp: Multiple Precision Arithmetic.* R package version 0.6-0.
URL: *https://CRAN.R-project.org/package=gmp*

Luger, R. (2010), ‘An Omnibus Test for Heteroskedasticity’, *Economics Letters* **106**, 22–24.

Lyon, J. D. and Tsai, C.-L. (1996), ‘A Comparison of Tests for Heteroscedasticity’, *Journal of the Royal Statistical Society. Series D (The Statistician)* **45**(3), 337–349.

MacKinnon, J. G. (2013), Thirty Years of Heteroskedasticity-Robust Inference, *in* X. Chen and N. R. Swanson, eds, ‘Recent Advances and Future Directions in Causality, Prediction, and Specification Analysis: Essays in Honor of Halbert L. White Jr’, Springer, New York, pp. 437–462.

MacKinnon, J. G. and White, H. (1985), ‘Some Heteroskedasticity-Consistent Covariance Matrix Estimators with Improved Finite Sample Properties’, *Journal of Econometrics* **29**(3), 305–325.

Maechler, M. (2020), *Rmpfr: R MPFR - Multiple Precision Floating-Point Reliable.* R package version 0.8-1.
URL: *https://CRAN.R-project.org/package=Rmpfr*

Magnus, J. R. and Sinha, A. K. (2005), ‘On Theil’s Errors’, *Econometrics Journal* **8**, 39–54.

Mahalanobis, P. C. (1936), On the Generalized Distance in Statistics, *in* ‘Proceedings of the National Institute of Science of India’.

Mann, H. B. and Whitney, D. R. (1947), ‘On a Test of Whether One of Two Random Variables is Stochastically Larger than the Other’, *The Annals of Mathematical Statistics* **18**(1), 50–60.

McCullagh, P. (1983), 'Quasi-Likelihood Functions', *Annals of Statistics* **11**(1), 59–67.

Miller, I. and Miller, M. (2019), *John E. Freund's Mathematical Statistics with Applications*, 8th edn, Pearson, London.

Miller, S. and Startz, R. (2019), 'Feasible Generalized Least Squares Using Support Vector Regression', *Economics Letters* **175**, 28–31.

Mittelhammer, R. C., Judge, G. G. and Miller, D. J. (2000), *Econometric Foundations*, Cambridge University Press, Cambridge.

Murteira, J. M., Ramalho, E. A. and Ramalho, J. J. (2013), 'Heteroskedasticity Testing through a Comparison of Wald Statistics', *Portuguese Economic Journal* **12**(2), 131–160.

Narasimhan, B., Johnson, S. G., Hahn, T., Bouvier, A. and Kiêu, K. (2020), *cubature: Adaptive Multivariate Integration over Hypercubes.* R package version 2.0.4.1.
URL: *https://CRAN.R-project.org/package=cubature*

Ng, M. and Wilcox, R. R. (2011), 'A Comparison of Two-Stage Procedures for Testing Least-Squares Coefficients under Heteroscedasticity', *British Journal of Mathematical and Statistical Psychology* **64**, 244–258.

Noorian, F. (2015), *quadprogpp: Quick Quadratic Programming Solver in C++*. https://github.com/fnoorian/quadprogpp, http://www.diegm.uniud.it/digaspero/index.php?page=software, http://www.labri.fr/perso/guenneba/code/QuadProg/.

Olive, D. J. (2018), 'Applications of Hyperellipsoidal Prediction Regions', *Statistical Papers* **59**, 913–931.

Paloyo, A. R. (2011), 'When Did We Begin to Spell 'Heteros*edasticity' Correctly?', *Ruhr Economic Papers* **300**, 1–24.

Paula, G. A. (1993), 'Assessing Local Influence in Restricted Regression Models', *Computational Statistics & Data Analysis* **16**, 63–79.

Paula, G. A. (1999), 'Leverage in Inequality-Constrained Regression Models', *Journal of the Royal Statistical Society. Series D (The Statistician)* **48**, 529–538.

Paulson, C. (2019), *PACLasso: Penalized and Constrained Lasso Optimization.* R package version 1.0.0.
URL: *https://CRAN.R-project.org/package=PACLasso*

Perperoglou, A., Sauerbrei, W. and Abrahamowicz, M. (2019), 'A Review of Spline Function Procedures in R', *BMC Medical Research Methodology* **19**.

Petersen, K. B. and Pedersen, M. S. (2012), 'The Matrix Cookbook', Available at http://www2.imm.dtu.dk/pubdb/edoc/imm3274.pdf.

R Core Team (2022), *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria.
URL: *https://www.R-project.org/*

Račkauskas, A. and Zuokas, D. (2007), 'New Tests of Heteroskedasticity in Linear Regression Model', *Lithuanian Mathematical Journal* **47**(3), 248–265.

Radchenko, P. and James, G. M. (2011), 'Improved Variable Selection with Forward-LASSO adaptive shrinkage', *The Annals of Applied Statistics* **5**(1), 427–448.

Ramsey, J. (1969), 'Tests for Specification Errors in Classical Linear Least-Squares Regression Analysis', *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* **31**(2), 350–371.

Rencher, A. C. and Schaalje, G. B. (2008), *Linear Models in Statistics*, 2nd edn, Wiley, Hoboken, NJ.

Robinson, P. M. (1987), 'Asymptotically Efficient Estimation in the Presence of Heteroskedasticity of Unknown Form', *Econometrica* **55**(4), 875–891.

Romano, J. P. and Wolf, M. (2017), 'Resurrecting Weighted Least Squares', *Journal of Econometrics* **197**, 1–19.

Rosopa, P. J., Brawley, A. M., Atkinson, T. P. and Robertson, S. A. (2018), 'On the Conditional and Unconditional Type I Error Rates and Power of Tests in Linear Models with Heteroscedastic Errors', *Journal of Modern Applied Statistical Methods* **17**, 2–26.

Royen, T. (2007), 'Integral Representations and Approximations for Multivariate Gamma Distributions', *Annals of the Institute of Statistical Mathematics* **59**(3), 499–513.

Salem, M., Fattah, A. A. and Rady, E. H. A. (2019), 'A New Heteroscedasticity Consistent Covariance Matrix Estimator and Inference Based on Robust Methods', *Journal of Computational and Theoretical Nanoscience* **16**, 2687–2694.

Sartori, S. (2010), Penalized Regression: Bootstrap Confidence Intervals and Variable Selection for High Dimensional Data Sets, PhD thesis, University of Milan.

Schwendinger, F. (2020), *ROI.plugin.qpoases: 'qpOASES' Plugin for the 'R' Optimization Infrastructure.* R package version 1.0-0.
URL: *https://CRAN.R-project.org/package=ROI.plugin.qpoases*

Seber, G. A. F. and Wild, C. J. (2003), *Nonlinear Regression*, Wiley, Hoboken, NJ.

Simlai, P. (2014), 'Estimation of Variance of Housing Prices Using Spatial Conditional Heteroskedasticity (SARCH) Model with an Application to Boston Housing Price Data', *The Quarterly Review of Economics and Finance* **54**(1), 17–30.

Simonoff, J. S. and Tsai, C.-L. (1994), 'Use of Modified Profile Likelihood for Improved Tests of Constancy of Variance in Regression', *Journal of the Royal Statistical Society. Series C (Applied Statistics)* **43**(2), 357–370.

Stellato, B., Banjac, G., Goulart, P., Bemporad, A. and Boyd, S. (2020), 'OSQP: An Operator Splitting Solver for Quadratic Programs', *Mathematical Programming Computation* **12**(4), 637–672.
URL: *https://doi.org/10.1007/s12532-020-00179-2*

Stellato, B., Banjac, G., Goulart, P. and Boyd, S. (2021), *osqp: Quadratic Programming Solver using the 'OSQP' Library.* R package version 0.6.0.5.
URL: *https://CRAN.R-project.org/package=osqp*

Szroeter, J. (1978), 'A Class of Parametric Tests for Heteroscedasticity in Linear Econometric Models', *Econometrica* **46**(6), 1311–1327.

Theil, H. (1965), 'The Analysis of Disturbances in Regression Analysis', *Journal of the American Statistical Association* **60**(312), 1067–1079.

Theil, H. (1968), 'A Simplification of the BLUS Procedure for Analyzing Regression Disturbances', *Journal of the American Statistical Association* **63**(321), 242–251.

Theußl, S., Schwendinger, F. and Hornik, K. (2020), 'ROI: An extensible R optimization infrastructure', *Journal of Statistical Software* **94**(15), 1–64.

Tibshirani, R. (1996), 'Regression Shrinkage and Selection via the Lasso', *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* **58**(1), 267–288.

Toker, S., Şiray, G. Ü. and Kaçıranlar, S. (2013), 'Inequality Constrained Ridge Regression Estimator', *Statistics & Probability Letters* **83**, 2391–2398.

Tukey, J. W. (1977), *Exploratory Data Analysis*, Addison-Wesley, Reading, MA.

Turlach, B. A., Weingessel, A. and Moler, C. (2019), *quadprog: Functions to Solve Quadratic Programming Problems.* R package version 1.5-8.
URL: *https://CRAN.R-project.org/package=quadprog*

United States Department of Commerce (1979), *Statistical Abstract of the United States, 1979*, US Government Printing Office, Washington, D.C.

Uyanto, S. S. (2019), 'Monte Carlo Power Comparison of Seven Most Commonly Used Heteroscedasticity Tests', *Communications in Statistics - Simulation and Computation* .

Venables, W. N. and Ripley, B. D. (2002), *Modern Applied Statistics with S*, 4th edn, Springer, New York. ISBN 0-387-95457-0.
URL: *http://www.stats.ox.ac.uk/pub/MASS4/*

Verbyla, A. (1993), 'Modelling Variance Heterogeneity: Residual Maximum Likelihood and Diagnostics', *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* **55**(2), 493–508.

Vinod, H. D. (2014), 'Theil's BLUS Residuals and R Tools for Testing and Removing Autocorrelation and Heteroscedasticity', Available at SSRN: http://dx.doi.org/10.2139/ssrn.2412740.

Wang, Y. and Wahba, G. (1994), Bootstrap Confidence Intervals for Smoothing Splines and their Comparison to Bayesian 'Confidence Intervals', Technical Report 913, Department of Statistics, University of Wisconsin.

Ward, Jr., J. H. (1963), 'Hierarchical Grouping to Optimize an Objective Function', *Journal of the American Statistical Association* **48**, 236–244.

Werner, H. J. (1990), 'On Inequality Constrained Generalized Least-Squares Estimation', *Linear Algebra and its Applications* **127**, 379–392.

White, H. (1980), 'A Heteroskedasticity-Consistent Covariance Matrix Estimator and a Direct Test for Heteroskedasticity', *Econometrica* **48**(4), 817–838.

Wickham, H., Hester, J., Chang, W. and Bryan, J. (2021), *devtools: Tools to Make Developing R Packages Easier*. R package version 2.4.3.
URL: *https://CRAN.R-project.org/package=devtools*

Wilcox, R. R. (2020), *Wilcox' Robust Statistics*.
URL: *https://github.com/nicebread/WRS*

Wilcox, R. R. and Keselman, H. J. (2006), 'Detecting Heteroscedasticity in a Simple Regression Model via Quantile Regression Slopes', *Journal of Statistical Computation and Simulation* **76**(8), 705–712.

Wood, S. N. (2003), 'Thin Plate Regression Splines', *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* **65**(1), 95–114.

Wood, S. N. (2017), *Generalized Additive Models: An Introduction with R*, 2nd edn, CRC, Boca Raton, FL.

Wooldridge, J. M. (2013), *Introductory Econometrics: A Modern Approach*, 5th edn, Cengage, Toronto.

Wu, T. T. and Lange, K. (2008), 'Coordinate Descent Algorithms for Lasso Penalized Regression', *The Annals of Applied Statistics* **2**(1), 224–244.

Yüce, M. (2008), 'An Asymptotic Test for the Detection of Heteroskedasticity', *Istanbul University Econometrics and Statistics e-Journal* **8**, 33–44.

Zeileis, A. (2004), 'Econometric Computing with HC and HAC Covariance Matrix Estimators', *Journal of Statistical Software* **11**, 1–17.

Zeileis, A. (2006), 'Object-oriented computation of sandwich estimators', *Journal of Statistical Software* **16**(9), 1–16.

Zeileis, A. and Hothorn, T. (2002), 'Diagnostic Checking in Regression Relationships', *R News* **2**(3), 7–10.
URL: *https://CRAN.R-project.org/doc/Rnews/*

Zeileis, A., Köll, S. and Graham, N. (2020), 'Various Versatile Variances: An Object-Oriented Implementation of Clustered Covariances in R', *Journal of Statistical Software* **95**, 1–36.

Zhou, Q. M., Song, P. X.-K. and Thompson, M. E. (2015), 'Profiling Heteroscedasticity in Linear Regression Models', *The Canadian Journal of Statistics* **43**(3), 358–377.

Zimmermann, G., Pauly, M. and Bathke, A. C. (2017), 'Can the Wild Bootstrap Be Tamed into a General Analysis of Covariance Model?'. 1709.08031.