Understanding the Impact of Heteroscedasticity on the Predictive Ability of Modern Regression Methods

by

Sharla Jaclyn Gelfand

B.Sc., University of Calgary, 2013

Dissertation Submitted in Partial Fulfillment
of the Requirements for the Degree of
Master of Science

in the
Department of Statistical and Actuarial Science
Faculty of Science

© Sharla Jaclyn Gelfand 2015

SIMON FRASER UNIVERSITY
Summer 2015

All rights reserved.
However, in accordance with the *Copyright Act of Canada*, this work may be reproduced without authorization under the conditions for "Fair Dealing." Therefore, limited reproduction of this work for the purposes of private study, research, criticism, review and news reporting is likely to be in accordance with the law, particularly if cited appropriately.

Approval

Name: Sharla Jaclyn Gelfand

Degree: Master of Science

Title: *Understanding the Impact of Heteroscedasticity on the Predictive Ability of Modern Regression Methods*

Examining Committee: **Tim Swartz** (Chair)
Professor

Thomas Loughin
Senior Supervisor
Professor

Richard Lockhart
Supervisor
Professor

Hugh Chipman
Internal Examiner
Adjunct Faculty
Department of Statistics and Actuarial Science
Professor
Department of Mathematics and Statistics
Acadia University

Date Defended: 17 August 2015

Abstract

As the size and complexity of modern data sets grows, more and more prediction methods are developed. Despite the growing sophistication of methods, there is not a well-developed literature on how heteroscedasticity affects modern regression methods. We aim to understand the impact of heteroscedasticity on the predictive ability of modern regression methods. We accomplish this by reviewing the visualization and diagnosis of heteroscedasticity, as well as developing a measure for quantifying it. These methods are used on 42 real data sets in order to understand the prevalence and magnitude “typical” to data. We use the knowledge from this analysis to develop a simulation study that explores the predictive ability of nine regression methods. We vary a number of factors to determine how they influence prediction accuracy in conjunction with, and separately from, heteroscedasticity. These factors include data linearity, the number of explanatory variables, the proportion of unimportant explanatory variables, and the signal-to-noise ratio. We compare prediction accuracy with and without a variance-stabilizing log-transformation. The predictive ability of each method is compared by using the mean squared error, which is a popular measure of regression accuracy, and the median absolute standardized deviation, a measure that accounts for the potential of heteroscedasticity.

Keywords: Heteroscedasticity; regression; regression trees; random forests; Bayesian adaptive regression trees; artificial neural networks; multivariate adaptive regression splines

Acknowledgements

I would first like to thank my supervisor Dr. Tom Loughin for challenging me and for being patient with me. Tom, I really appreciate you taking me on as a student. You foster such a positive research environment that was an absolute pleasure to be a part of. I would also like to thank Dr. Richard Lockhart and Dr. Hugh Chipman for their helpful comments on my project. I would probably not be at Simon Fraser University if it weren't for Dr. Carl Schwarz, and I am thankful for the opportunity and for his guidance. Thank you as well to Charlene Bradbury, Sadika Jungic, and Kelly Jay for keeping the department running so smoothly.

Thank you to Thilini Surendra, Will Ruth, and Andrew Henrey for giving helpful insights on my project, challenging me, and even for the roastings! It was endlessly helpful to see how my work looked from another angle and to find different ways to examine things. Thank you to other fellow students, including Christina Nieuwouldt, Bingying Li, Jack Davis, Rachel Lipson, and Mike Grosskopf, for making graduate school so enjoyable. Special thanks to Elena Szefer for weekly hangouts and for providing so much understanding and support. I don't know if I could have done it without you.

Finally, thank you to my friends and family, and especially to my parents, Lewis and Stella Gelfand. I am eternally grateful for how much you support and love me, and how often you express that.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Modern electronics and computers permit the collection of data from an ever-increasing variety of sources. As more and more data become available in many different fields, the size and complexity of typical data sets grows as well. With this growth comes the potential for more precise and accurate predictions. Linear regression is a classic, commonly used prediction tool. However, it requires data that satisfy certain conditions, including linearity, additivity, and homoscedasticity, that may not be met in complex data sets. Modern regression techniques have been developed to handle many challenges prevalent in modern data, including non-linearity and non-additivity, but reviews of literature show that there is sparse research on how heteroscedasticity of data impacts the predictive ability of these methods. Our aim is to study heteroscedasticity in combination with, and separately from, some of the other challenges typically prevalent in modern data.

This thesis expands on previous work (Payne, 2014) by evaluating the impact of heteroscedasticity on the predictive ability of modern regression methods. We accomplish this by first reviewing how to recognize and diagnose heteroscedasticity, building a “mental library” of how it appears in residual plots, and discussing measures for quantifying its magnitude. We apply these measures to 42 data sets used previously by Chipman et al. (2010) for other purposes without regard to their potential for heteroscedasticity. We show that heteroscedasticity is widespread in data. With the knowledge gained from this analysis, we develop a simulation study comparing the predictive ability of nine modern regression methods under “typical” amounts of heteroscedasticity. The nine methods are: linear regression, stepwise linear regression, the least absolute shrinkage and selection operator (LASSO), regression trees (both full and pruned), random forests, boosted random forests (boosting), multivariate adaptive regression splines (MARS), artificial neural networks (ANNs), and Bayesian adaptive regression trees (BART). We generate linear and nonlinear heteroscedastic data, varying the number of explanatory variables, the proportion of unimportant explanatory variables, and the signal-to-noise ratio. We also analyze the responses with and without a variance-stabilizing log transformation, resulting in four distinct groups of data: linear and heteroscedastic, nonlinear and heteroscedastic, linear and homoscedastic, and nonlinear and homoscedastic. In each simulation scenario, we summarize the quality of the predictions using measures that focus on different aspects of the model fit.

We have a number of hypotheses relating to how the predictive methods will perform on each group of data. First, we expect that the methods based on a linear model (linear regression, stepwise linear regression, and the LASSO) will perform well on the linear, homoscedastic data, as these conditions satisfy the assumptions of these methods. However, we do not expect these methods to perform well when nonlinearity is present. Additionally, these methods rely on the unweighted sum of squares criterion, which implicitly assumes homoscedastic errors. Homoscedasticity is required to ensure that the regression coefficient estimates from linear regression have the smallest standard errors when compared to other linear, unbiased estimators. When heteroscedasticity is present and this condition is violated, the standard errors are biased, leading to incorrect inferences and conclusions. In addition, areas of high variability contribute more to minimizing the unweighted sum of squares criterion, and are favoured when making predictions for the mean. The resulting predictions for low-variance areas can be “off” by a relatively larger amount, even though they contain more precise information on the mean, in favour of less meaningful predictions for the higher-variance data. Therefore, we expect that these linear methods will be adversely affected by heteroscedasticity, particularly when performance is measured by a metric that weights errors according to the local variance.

On the other hand, the other methods we will compare (including regression trees and their ensembles, ANNs, and MARS) do not make assumptions about linearity of the data. Therefore we expect that they should perform better than the linear regression methods under nonlinearity and homoscedasticity. These methods all use the residual sum of squares as a loss function. Again, when there is heteroscedasticity, data with higher variability are more influential in minimizing this function and in making predictions. This also results in predictions that are further off in the low-variance area, even though there is more information there. We expect that these methods will also degrade under heteroscedasticity.

The outline of this thesis is as follows. In Chapter 2, we introduce heteroscedasticity and discuss methods for recognizing and quantifying it. We use these methods on 42 data sets in order to develop knowledge on the prevalence and severity of heteroscedasticity in “typical” data. In Chapter 3 we review previous work on heteroscedasticity in modern regression methods and discuss its limitations in context of our previous findings. In Chapter 4, we introduce and summarize the prediction methods that will be used and compared in the simulation study. In Chapter 5 we develop and execute a simulation study with 48 scenarios to challenge the prediction methods. The results are presented in terms of mean squared error (MSE) and median absolute standardized deviation (MASD). Finally, in Chapter 6 we discuss the results and limitations, and provide suggestions for future work.

Chapter 2

Heteroscedasticity

Homoscedasticity, or constant variance in the response, is an explicit assumption made when using linear regression and an implicit one with many other prediction tools. Heteroscedasticity is the violation of this assumption. In this chapter, we introduce heteroscedasticity, review methods for visualizing and identifying it through residual plots, introduce a measure for quantifying it, and analyze 42 data sets from Chipman et al. (2010) in order to assess the prevalence and magnitude of heteroscedasticity in "typical" data. We will show that heteroscedasticity is widespread and therefore there is the need to evaluate which prediction tools are robust against it.

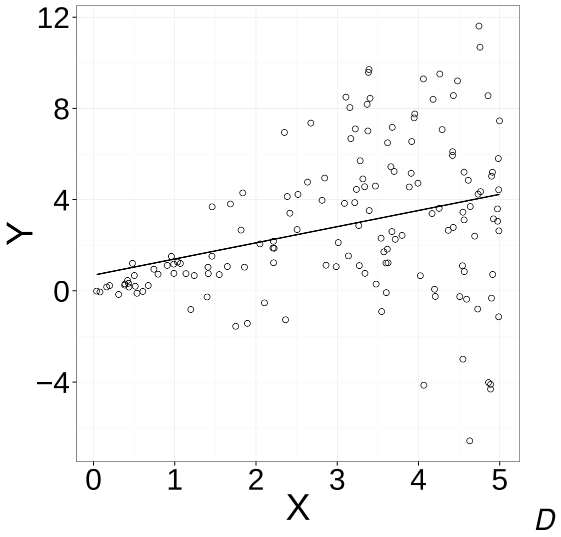
2.1 Introducing Heteroscedasticity

Linear regression allows us to study the relationship between p explanatory variables, $X_1, \ldots, X_p$, and a continuous response variable, Y. This model takes the form $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta}+\boldsymbol{\epsilon}$, where $\mathbf{X}$ is an $n \times (p+1)$ matrix of explanatory variables, $\mathbf{Y}$ is an $n \times 1$ vector of responses, $\boldsymbol{\beta}$ is a $(p+1) \times 1$ vector of unknown regression coefficients, and $\boldsymbol{\epsilon}$ is an $n \times 1$ vector of unobservable errors with a Normal$(\mathbf{0}, \sigma^2\mathbf{I})$ distribution (Cook and Weisberg, 1982). Homoscedasticity is the assumption that the error variance is equal to σ^2 for all n observations. This assumption is used for estimating the regression coefficients $\boldsymbol{\beta}$ through ordinary least squares (OLS), leading to the solution $\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}$. By the Gauss-Markov theorem, $\hat{\boldsymbol{\beta}}$ is the linear, unbiased estimator with the smallest variance.

Heteroscedasticity is the violation of the homoscedasticity assumption. When it occurs, the OLS estimates $\hat{\boldsymbol{\beta}}$ are still unbiased, but become inefficient. The regular standard errors of these estimates are wrong, leading to incorrect inferences, although White's heteroscedastic corrected standard errors (White, 1980) can be used instead. When heteroscedasticity is present, data from highly variable areas have a larger effect on minimizing the unweighted least squares criterion in OLS and contribute more to predictions.

Consider the case with one predictor, X, where the error variance is proportional to the mean. As we can see in the scatterplot of Y versus X in Figure 2.1, fitting the regression line using OLS results in over-prediction in the low-variance area, on the left of the plot. This is the part of the data that contains the most information about the local mean, and the mean should be fit most accurately here. However, predictions are relatively far from their apparent true means here, while relatively meaningless improvements are made to the predictions in the high-variance area.

Figure 2.1: Scatter plot of Y versus X, when the error variance is proportional to the mean, along with a regression line which overpredicts in the low variance area.



Heteroscedasticity comes in many forms, including as a function of the explanatory variables or of the mean of the data. In this thesis, we focus on the situation where the variance increases as the mean increases. This is a common feature of many distributional models other than the normal – e.g., Poisson, gamma, exponential – so it might be expected to occur in many regression problems where a normal distribution is incorrectly used to model data from another distribution. In particular, we focus on variance that increases as a power of the mean (Carroll and Ruppert, 1988). Transformations of the response can be used to correct for heteroscedasticity of this form. For example, if the variance is proportional to the mean, a square root transformation of the response can correct the heteroscedasticity. When the variance is proportional to the square of the mean, a log transformation is used to correct the heteroscedasticity (Carroll and Ruppert, 1988). These are examples of Box and Cox (1964) power transformations, and other transformations can be used, depending on the power of the mean. While transformations of data are possible, they are often difficult to choose and may lead to undesirable properties, such as nonlinearity when the original relationship is linear. It is therefore useful to identify methods that are robust against heteroscedasticity.

2.2 Identifying Heteroscedasticity

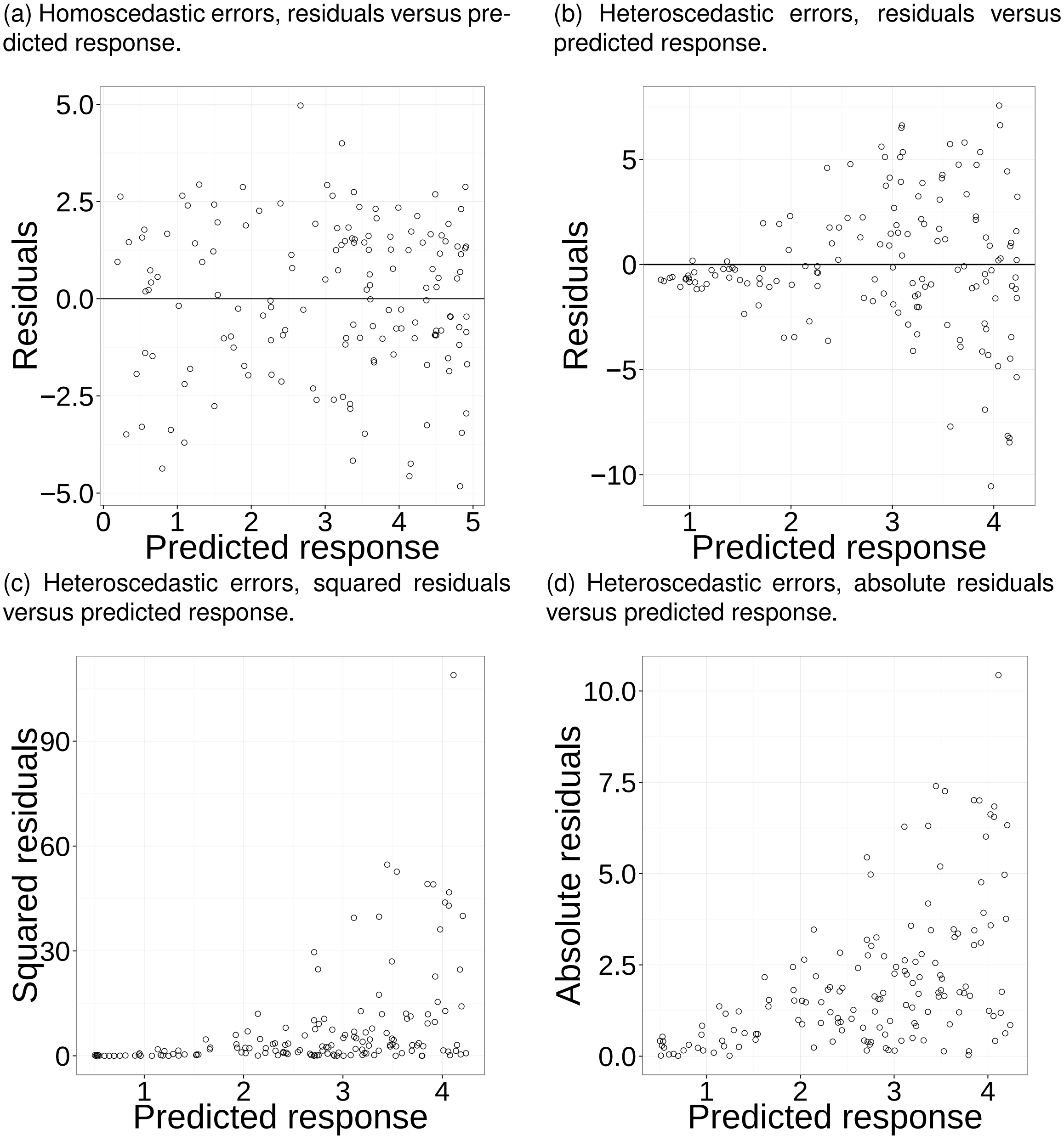
The residuals from linear regression are $\mathbf{e} = \mathbf{Y} - \hat{\mathbf{Y}} = \mathbf{Y} - \mathbf{X}\hat{\beta}$, and are used a proxy for the unobservable errors $\boldsymbol{\epsilon}$ (Cook and Weisberg, 1982). The residuals can be used to diagnose the behaviour of the variance within a data set. Residual plots, where residuals e_i are plotted on the y-axis, versus the predicted responses $\hat{y}_i$ on the x axis, are the most commonly used tool (Carroll and Ruppert, 1988). When the condition of homoscedasticity is satisfied, the residuals should be randomly and uniformly scattered around the horizontal line at 0, as seen in Figure 2.2a. When heteroscedasticity is present, particularly when the variance is proportional a power of the mean, there is a fan shape to the residuals, as in Figure 2.2b.

This may not be the best method for detecting heteroscedasticity, as it is “difficult to interpret, particularly when the positive and negative residuals do not exhibit the same general pattern” (Cook and Weisberg, 1982). Cook and Weisberg suggest plotting the squared residuals, e_i^2, to account for this. Then, a wedge shape bounded below by 0 would indicate heteroscedasticity. However, as Carroll and Ruppert (1988) point out, squaring residuals that are large in magnitude creates scaling problems, resulting in a plot where patterns in the rest of the residuals are difficult to see, such as in Figure 2.2c. They instead advocate for plotting the absolute residuals, as in Figure 2.2d. This way, we do not need to identify positive and negative patterns, and do not need to worry about scaling issues. A wedge shape of the absolute residuals also indicates heteroscedasticity where the variance increases with the mean. This is the plotting method that we use for identifying heteroscedasticity moving forward, and will be used later on in this Chapter.

2.3 Quantifying Heteroscedasticity

Absolute residual plots are helpful in identifying if a data set is heteroscedastic, but do not provide a way to quantify the “amount” of heteroscedasticity in the data set. We will use two measures, described as follows, to measure heteroscedasticity. For the first measure, compute the average of the absolute residuals for the largest 10 percent of the predicted responses. Next, compute the average of the absolute residuals for the smallest 10 percent. Define the “Standard Deviation (SD) Ratio,” computed as the ratio of the average for the largest 10 percent to the average for the smallest 10 percent. For homoscedastic data,

Figure 2.2: Residual plots of homoscedastic and heteroscedastic errors.



(a) Homoscedastic errors, residuals versus predicted response.

(b) Heteroscedastic errors, residuals versus predicted response.

(c) Heteroscedastic errors, squared residuals versus predicted response.

(d) Heteroscedastic errors, absolute residuals versus predicted response.

this ratio is close to 1. For heteroscedastic data where the variance increases with the mean, the ratio is greater than 1. When the variance decreases with the mean, the ratio is less than 1.

The rationale for this measure is as follows. In linear regression, we make the assumption that $Var(\epsilon_i) = \sigma^2$ for all n observations. Since $Var(\epsilon_i) = E(\epsilon_i^2) - (E(\epsilon_i))^2$ and $E(\epsilon_i) = 0$, it follows that $E(\epsilon_i^2) = \sigma^2$, and $E(|\epsilon_i|)$ is proportional to σ. Therefore, if it were possible to plot the absolute values of the errors, they would be representative of the standard deviations in the data. Since the residuals e_i are used as a proxy for ϵ_i, we can instead plot the absolute values of the residuals, $|e_i|$, to represent the standard deviation. The SD Ratio therefore measures approximately the ratio of the standard deviation of Y in areas of high variability to the standard deviation of Y in areas of low variability.

The second measure we will use is the estimate of the slope from fitting a linear model of the absolute residuals versus the predicted responses. A slope of 0 indicates that the variance does not change with the mean, a slope greater than 0 indicates that the variance increases with the mean, and a slope less than 0 indicates that the variance decreases with the mean. We will not use this slope measure to quantify the amount of heteroscedasticity, but rather to "test" for its presence, as White's heteroscedasticity-corrected standard errors can be used to formally test if the slope is 0. This test is similar to White's test for heteroscedasticity (White, 1980), but uses the absolute residuals to test for heteroscedasticity instead of the squared residuals.

2.4 Example Data Sets

The methods for identifying and quantifying heteroscedasticity discussed above are used in this section to evaluate the prevalence and severity of heteroscedasticity in "typical" data. We looked at a collection of 42 data sets used for a "bake-off" competition by Chipman et al. (2010), who compared the predictive ability of Bayesian additive regression trees (BART) to the predictive ability of other modern regression methods. These 42 data sets are a subset of 52 data sets initially collected by Kim et al. (2007). Although the methods used to select data sets for inclusion into the group is not given, there is no indication that heteroscedasticity played any role in the process. We therefore consider these data sets to be representative of "typical" data from a variety of fields. We fit an optimal random forest (discussed in Chapter 4) to each data set to remove the mean trend, which allows for the possibility that the mean is nonlinearly and non-additively related to the explanatory variables. We then analyzed the absolute residuals, computing the SD Ratio and slope measures. Table 2.1 displays this information, along with the name, sample size, and number of explanatory variables for each data set.

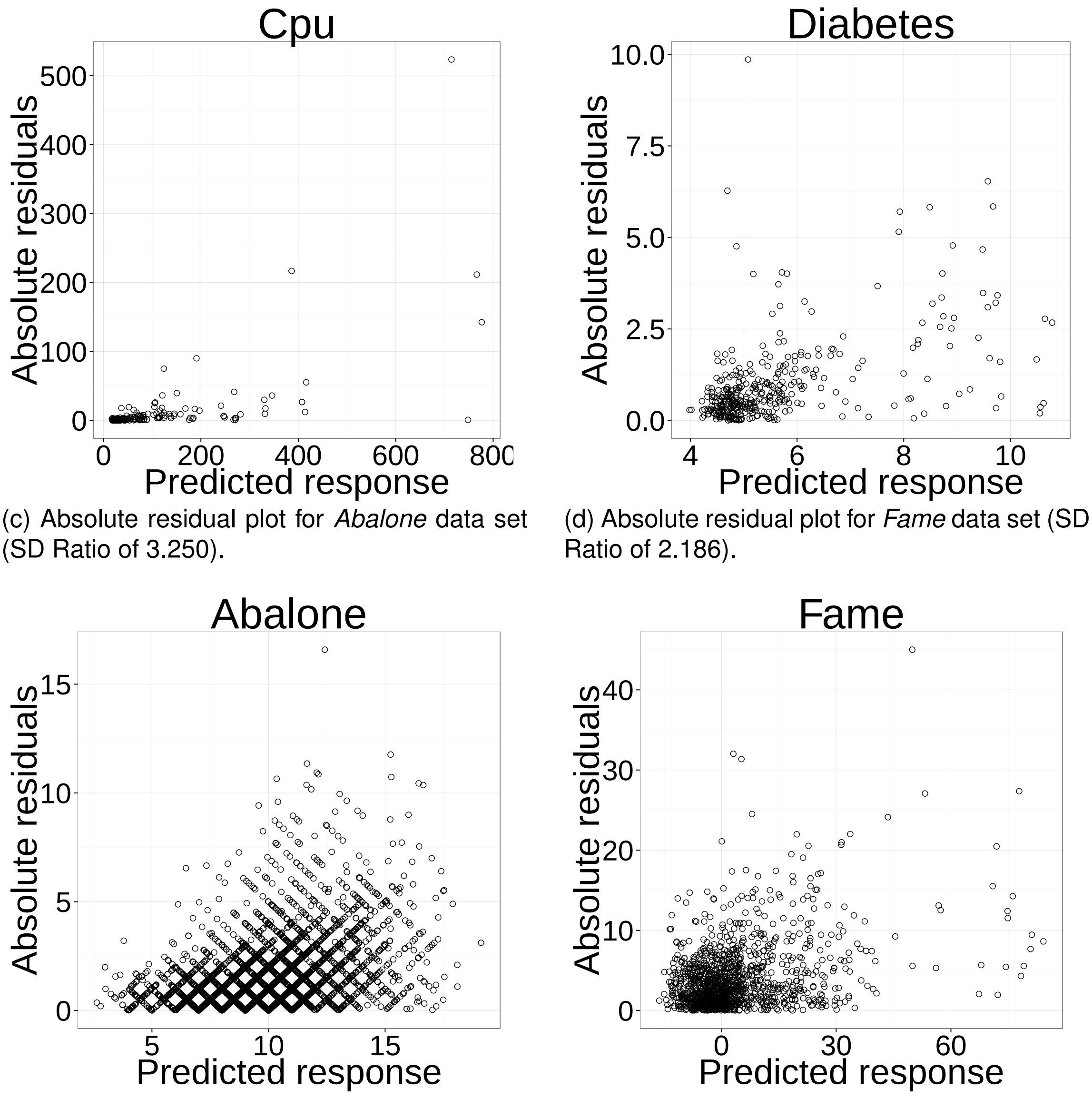
The SD Ratio, slope (along with its standard error), and a visual examination of the absolute residual plots were used to determine which data sets are heteroscedastic, focusing only on data sets where the heteroscedasticity manifests as the variance increasing with the mean. Those data sets are highlighted in Table 2.1. We detected heteroscedasticity, where the variability increases with the mean, in 25 of the 42 data sets. None of these data sets were selected with heteroscedasticity in mind; they were only used to compare the predictive ability of modern regression techniques. Nonetheless, heteroscedasticity was present in over half of the bake-off data sets, showing that heteroscedasticity is widespread in data and that there is a need to examine which regression methods are robust to heteroscedasticity. Of the 25 heteroscedastic data sets, the mean SD Ratio is 6.945, with a median of 3.079. The minimum SD Ratio is 1.142 and the maximum is 62.798.

Unfortunately, the SD Ratio measure is not robust to outliers. The SD Ratio for the *Cpu* data set is 62.798. However, as the absolute residual plot in Figure 2.3a shows, this is due to a residual that is more than two times as large as than the remaining largest residuals. When this point is removed, the SD ratio of this data set is 41.200. However, the measure is useful for identifying the amount of heteroscedasticity in data sets without outliers, such as for the *Diabetes* data set (SD Ratio of 5.313, absolute residual plot in Figure 2.3b), the *Abalone* data set (SD Ratio of 3.250, absolute residual plot in Figure 2.3c), and the *Fame* data set (SD Ratio of 2.186, absolute residual plot in Figure 2.3d). This gives us an idea of how severe heteroscedasticity typically is, and will be helpful when it comes time to develop a simulation study.

Table 2.1: Table showing data set name, sample size, number of explanatory variables (p), SD Ratio, and slope (with standard error) of the 42 bake-off data sets from Chipman et al. (2010). Highlighted data sets contain significant heteroscedasticity, according to the test described.

Data Set	Sample Size	p	SD Ratio	Slope (SE)
Abalone	4177	8	3.250	0.232 (0.011)
Ais	202	12	0.906	-0.001 (0.013)
Alcohol	2462	18	0.950	0.033 (0.054)
Amenity	3044	21	1.883	0.091 (0.010)
Attend	838	9	1.698	0.129 (0.020)
Baseball	263	20	2.214	0.076 (0.015)
Baskball	96	4	1.119	0.019 (0.136)
Boston	506	13	0.511	-0.073 (0.016)
Budget	1729	10	4.860	0.036 (0.003)
Cane	3775	9	1.142	0.091 (0.024)
Cardio	375	9	5.509	0.548 (0.123)
College	694	24	1.150	0.033 (0.012)
Cps	534	10	1.116	0.049 (0.048)
Cpu	209	7	62.798	0.200 (0.086)
Deer	654	13	0.511	-0.047 (0.019)
Diabetes	375	15	5.313	0.388 (0.059)
Diamond	308	4	0.888	-0.029 (0.014)
Edu	1400	5	1.992	0.139 (0.025)
Enroll	258	6	1.855	0.081 (0.022)
Fame	1318	22	2.186	0.115 (0.014)
Fat	252	14	0.936	-0.003 (0.029)
Fishery	6806	14	4.014	0.379 (0.020)
Hatco	100	13	0.863	-0.038 (0.031)
Insur	2182	6	0.161	-0.133 (0.006)
Labor	1189	18	2.041	0.115 (0.016)
Laheart	200	16	1.462	0.152 (0.064)
Medicare	4406	21	1.156	0.027 (0.021)
Mpg	392	7	3.079	0.084 (0.014)
Mumps	1523	3	0.606	-0.053 (0.009)
Mussels	201	4	4.272	0.155 (0.029)
Ozone	330	8	5.210	0.191 (0.023)
Price	159	15	6.118	0.171 (0.033)
Rate	144	9	2.192	0.300 (0.096)
Rice	171	15	0.671	0.006 (0.039)
Scenic	113	10	0.793	-0.152 (0.067)
Servo	167	4	6.581	0.196 (0.040)
Smsa	141	10	23.016	0.363 (0.200)
Strike	625	5	17.780	0.488 (0.063)
Tecator	215	10	2.002	0.045 (0.017)
Tree	100	8	0.561	-0.128 (0.069)
Triazine	186	28	0.279	-0.387 (0.083)
Wage	3380	13	1.057	0.017 (0.014)

Figure 2.3: Absolute residual plots of four bake-off data sets from Chipman et al. (2010).



(a) Absolute residual plot for *Cpu* data set (SD Ratio of 62.798).

(b) Absolute residual plot for *Diabetes* data set (SD Ratio of 5.313).

(c) Absolute residual plot for *Abalone* data set (SD Ratio of 3.250).

(d) Absolute residual plot for *Fame* data set (SD Ratio of 2.186).

Chapter 3

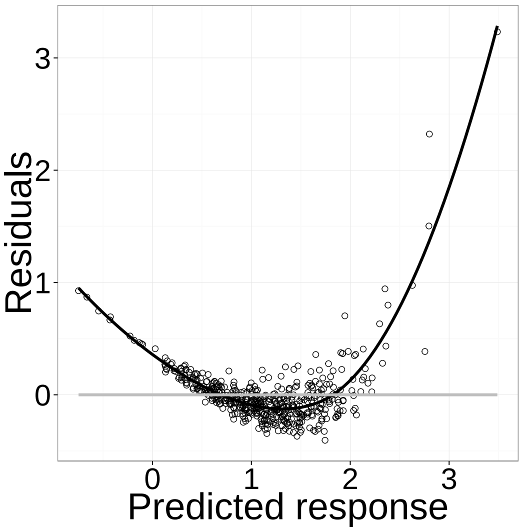
Previous Work

In this chapter we review previous work on evaluating the impact of heteroscedasticity on the predictive ability of modern regression methods. We outline the work done and discuss some limitations.

We are aware of no other studies examining the effect of heteroscedasticity on modern regression methods prior to Payne (2014). Payne (2014) explores the topic via a simulation study, similar to the one we discuss in Chapter 5. He generates linear and nonlinear data, and varies the number of explanatory variables, the proportion of unimportant explanatory variables, and the signal-to-noise ratio. The responses are analyzed with and without a variance-stabilizing log transformation, for a total of 32 simulation scenarios. Each scenario consists of 50 simulations, each with $n = 1000$. On each data set, the following methods are used to make predictions: linear regression, stepwise linear regression, ridge regression, the LASSO, regression trees, boosted regression trees, random forests, MARS, ANNs, and BART.

There are a few limitations of the work done in Payne (2014). The first is that the data generated do not possess all of the properties desired in order to challenge the regression methods. In particular, the data are not "heteroscedastic enough" and the nonlinear data are not "nonlinear enough." We compute the SD Ratio, as described in Chapter 3, with the absolute residuals found by fitting either a random forest (in nonlinear cases) or a linear model (in linear cases) to remove the mean trend. This is repeated on the first 10 simulated data sets for each of the eight nonlinear-and-heteroscedastic, and eight linear-and-heteroscedastic scenarios. We look at the mean SD Ratio for each scenario.

Figure 3.1: Scatter plot of residuals from artificial nonlinear data fitted with a linear model.

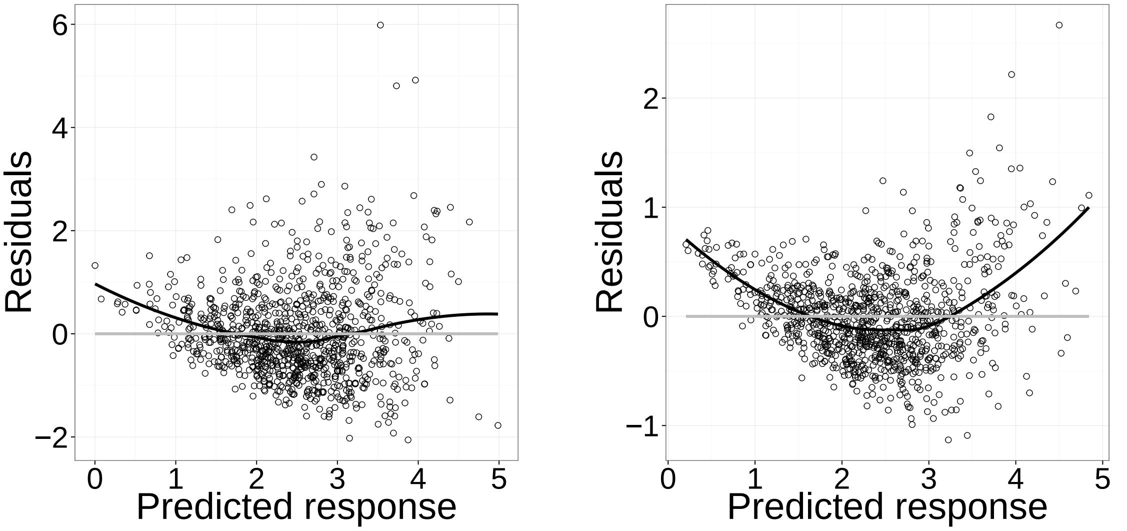


We compare the SD Ratios of the data from Payne (2014) to the SD Ratios of the bake-off data from Chipman et al. (2010). The SD Ratios of the data sets from Payne (2014) range from 1.610 to 3.358, with a mean of 2.347 and a median of 2.289. Meanwhile, the median SD Ratio of the bake-off data is 3.079, which is almost as large as the largest SD Ratio from Payne (2014). Compared to the typical amount of heteroscedasticity we found in the bake-off data, the data sets from Payne (2014) are not very heteroscedastic. In our simulation study, we aim to simulate data with SD Ratios consistently around 3, the typical amount of heteroscedasticity in the bake-off data.

The data in the nonlinear heteroscedastic case are not nonlinear "enough" to challenge the linear-based methods (linear regression, stepwise linear regression, ridge regression, and the LASSO). When a linear model is fit to nonlinear data, the residuals should convey considerable nonlinearity, often identified by curvature in a residual plot. See Figure 3.1 for an exaggerated example.

To measure the nonlinearity in a data set, consider fitting a LOESS curve with degree 2 to the residuals. This is indicated by the curved black line in Figure 3.1. Ideally, the LOESS curve should "wobble" randomly around zero, explaining very little apparent trend in the residuals. When it does not, this is an indication that the linear regression is a poor fit, and that curvature may be present.

Figure 3.2: Residual plots of nonlinear data from Payne (2014).



(a) Residual plot of data with LOESS R^2 of 0.036.

(b) Residual plot of data with LOESS R^2 of 0.195.

We use the coefficient of determination (R^2) of the LOESS curve on the residual plot as a rough measure of curvature in our simulated nonlinear data sets. The farther from zero R^2 is, the more nonlinear the data. For the residuals in Figure 3.1, the R^2 is 0.76, because the original data from which the residual plot was made are quite nonlinear.

We repeat this same procedure on the nonlinear heteroscedastic data from Payne (2014), again looking at the first ten simulated data sets from the eight nonlinear heteroscedastic scenarios. The LOESS R^2 ranges from 0.018 to 0.24. Figure 3.2 shows two examples of residuals from this nonlinear data, specifically one where the LOESS R^2 is 0.036 (Figure 3.2a) and one where it is 0.195 (Figure 3.2b). This exploratory analysis gives us a way to quantify nonlinearity, and is useful for developing our own simulation study. In our work, we develop simulation settings that achieve greater levels of nonlinearity and therefore provide an opportunity for modern methods to provide better results than linear methods.

Another limitation of Payne (2014) is that the methods are used at their default settings. As we discuss in Chapter 4, the “modern” methods (such as random forests, boosting, and ANNs) have tuning parameters that affect their performance. While they sometimes perform adequately at their default settings, these methods are not always “one size fits all,” and should be tailored to individual data sets to obtain optimal performance. This tuning can be done through K-fold cross-validation, where the data are split randomly into K distinct groups of roughly equal size. For k in $1, \ldots, K$, we fit a model at specific tuning parameter settings on all of the data except the k^{th} group, then make predictions for the data in the k^{th} group. Thus, observations do not influence their own predicted values. The cross-validation error then summarizes the differences between the responses and their predictions. This process can be repeated for various combinations of the tuning parameters, and the combination that minimizes the cross-validation error is chosen as the best. The resulting tuning parameter values may be quite different from the default settings, and the resulting prediction error can be considerably smaller with the selected values.

In addition, Payne (2014) uses two measures to evaluate the performance of each method. The first is the mean squared error (MSE), defined as $MSE = \frac{1}{n}\sum_{i=1}^{n}(\mu_i - \hat{y}_i)^2$. The second measure is the median absolute deviation, $MAD = median\,(|\mu_i - \hat{y}_i|)$. Because these measures treat each residual equally, they do not take into account any heteroscedasticity that might be present in a data set. In Chapter 5, we propose an alternative method for quantifying model performance.

Finally, the study in Payne (2014) was only repeated on 50 simulated data sets for each scenario. Considering the variability that may be present among data sets, this may not be enough simulations to distinguish the relative performance of different regression methods. It would be useful to have more simulations so that we can have better precision in our comparisons between methods.

The work done by Payne (2014) serves well as a pilot study for understanding the impact of heteroscedasticity on the predictive ability of modern regression methods. In particular, the models used for producing nonlinear data and heteroscedastic data are sensible and convenient. Payne (2014) varies the number of explanatory variables, the proportion of unimportant explanatory variables, and the signal-to-noise ratio, which are all factors that we investigate in our own study. By adding additional features, such as tuned methods, more heteroscedastic data, and greater number of simulated data sets, we hope to develop a simulation study that further investigates this topic.

Chapter 4

Methods

In Chapter 5 we compare the predictive ability of nine regression methods under heteroscedasticity. In this chapter, we introduce each method and present hypotheses on how they will perform on four distinct groups of data, described in further detail in Chapter 5: nonlinear heteroscedastic data, linear heteroscedastic data, nonlinear homoscedastic data, and linear homoscedastic data.

4.1 Linear Regression

Linear regression is a method used to study the relationship between p explanatory variables $X_1, \ldots, X_p$, and a continuous response variable Y. The linear regression model takes the form $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$, where $\mathbf{X}$ is an $n \times (p+1)$ matrix of explanatory variables, $\mathbf{Y}$ is an $n \times 1$ vector of responses, $\boldsymbol{\beta}$ is a $(p+1) \times 1$ vector of unknown regression coefficients, and $\boldsymbol{\epsilon}$ is an $n \times 1$ vector of unobservable errors with a Normal($\mathbf{0}, \sigma^2\mathbf{I}$) distribution (Cook and Weisberg, 1982). The estimated regression coefficients are found through ordinary least squares (OLS), by minimizing the unweighted sum of squares $(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})$, leading to the OLS estimates $\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}\mathbf{Y}$.

As explained in Chapter 2, the homoscedasticity assumption is integral to linear regression. In addition, linear regression cannot model nonlinearity in data. Therefore, we expect linear regression to do poorly on the nonlinear heteroscedastic data. When the log transformation is used, these data become linear and homoscedastic. We expect linear regression to do well in this case, since it is ideal for the model assumptions. On the other hand, we expect it to do poorly on both the untransformed linear heteroscedastic data, and on its log-transformed version, the nonlinear homoscedastic data. Each of these data structures has properties (heteroscedasticity and nonlinearity, respectively), that challenge linear regression.

4.2 Stepwise Linear Regression

In our linear regression setting, we use all available variables. This means estimating a coefficient for each explanatory variable, regardless of whether or not the variable might be "important", resulting in $p + 1$ estimated coefficients. Each additional parameter estimate adds to the variability of predictions.

It is therefore desirable to find a model that contains only the important variables, which would potentially provide much less variable predictions. Many techniques exist that attempt to identify which variables are important. Stepwise linear regression is a classic method used to reduce the number of variables in a model.

The method is as follows. We construct a null model that includes only the intercept, and a full model that includes all of the explanatory variables. In forward stepwise regression, we start by fitting the null model and add one explanatory variable at a time. As discussed by Hastie et al. (2009), there are several criteria that can be used to decide which explanatory variable is added at each step. However, they are all based on the variable's partial sum of squares, given the previous model, so they all create the same "path" from the empty model to the full model. A popular, older method is to add the variable that has the strongest relationship with the response, according to a Z- test. We instead use the smallest value of the Bayesian Information Criterion (BIC, Schwarz, 1978). The BIC is a measure that penalizes against adding too many parameters to a model and therefore prevents overfitting. The criterion is defined as $BIC = -2loglik + k\log(n)$, where $loglik$ is the log likelihood for the model, given the data, and k is the total number of parameters currently in the model. We use a form of the stepwise algorithm where one variable is added at a time until no variable further reduces the BIC, with the BIC computed at each step. The model that minimizes the BIC is chosen.

Analogously, in backward elimination, we begin by fitting the full model and remove the explanatory variable that results a reduced model with the smallest value of BIC among all possible deletions. This process is carried out one variable at a time until there are no variables whose deletion could not further reduce the BIC. Then, the model with the lowest BIC is chosen. The models selected by forward selection and backward elimination do not always coincide. We therefore use a combination of both methods that involves performing both forward stepwise regression and backward elimination, and choosing whichever final model has smaller BIC. Performing both algorithms and selecting the better model among the two chosen offers the potential for improvement over either direction alone.

Since stepwise regression is a linear regression method and the final model is fit using OLS, it implicitly involves the same assumptions as full linear regression. Therefore, our hypotheses are similar for this method. We expect it to do well on linear homoscedastic data, but poorly on nonlinear homoscedastic, nonlinear heteroscedastic, and linear heteroscedastic data. However, when the data are linear and homoscedastic and the number of "worthless" explanatory variables is large, we expect stepwise linear regression to do better than linear regression because it focuses on those explanatory variables with a larger impact on the response, rather than estimating the coefficients for all variables.

4.3 Least Absolute Selection and Shrinkage Operator

The least absolute selection and shrinkage operator (LASSO) is another method used to obtain sparse models. The estimate $\hat{\boldsymbol{\beta}}$ is found by minimizing the $L1$-penalized least squares criterion, $(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}) + \lambda \sum_{j=1}^{p} |\beta_j|$ for a given value of λ (Hastie et al., 2009). The parameter λ is a penalty which prevents the coefficient estimates from growing to their full OLS estimates and is chosen through cross-validation. Using this criterion results in parameter estimates that are shrunk relative to the OLS solutions. Some estimates are shrunk to 0, resulting in a sparser model. It is therefore used both as a shrinkage estimator and as a variable selection technique (Izenman, 2013).

Since the LASSO coefficients are found by minimizing a criterion that involves the unweighted sum of squares, we expect the LASSO to also be adversely affected by heteroscedasticity and nonlinearity, similar to linear regression. Again, when the number of unimportant explanatory variables is large, we expect the LASSO to perform better than linear regression.

4.4 Regression Trees

Regression trees partition data into groups that are as different as possible and fit the mean response for each group as its prediction (Hastie et al., 2009). The partitioning is done recursively, as follows. The data are split into two subgroups, based on one explanatory variable. Only the splits $\{i : X_{ij} > c\}, \{i : X_{ij} < c\}$ are considered for all appropriate values of c for each explanatory variable X_j. The variable and splitting point are chosen to reduce the residual sum of squares (RSS) as much as possible after the split as compared to before the split. The RSS of the full data, before any splits, is $RSS_{Full} = \sum_{i=1}^{n}(y_i - \bar{y})^2$. Once the data are split into two subgroups, R_1 and R_2, the mean response of the data in R_1, $\bar{y}_1$, is fit to that data, and the mean of the data in R_2, $\bar{y}_2$, is fit to that data. The resulting RSS is $RSS_{Split} = \sum_{y_i \in R_1}(y_i - \bar{y}_1)^2 + \sum_{y_i \in R_2}(y_i - \bar{y}_2)^2$. After the first split, the same process is repeated in each subgroup. A new split is chosen, either on the same variable at a different splitting point, or on an entirely new variable. This process is repeated until some stopping rule is reached, which is typically a minimum number of observations in a final group (called a terminal node).

Regression trees that are too large overfit the data, but trees that are too small may miss important information. Full trees can be pruned according to a cost-complexity criterion, which takes into account the amount of squared error explained by each subtree plus a penalty for the number of terminal nodes in the subtree. The penalty parameter, α, balances between tree size and overfitting. According to Hastie et al. (2009), for each α there is a unique subtree that minimizes the cost-complexity criterion. The subtree, T_α, is found by collapsing internal nodes of the full tree one at a time, with the collapsed node being the one that produces the smallest increase in the residual sum of squares per added node. This sequence of trees will contain the tree T_α that minimizes the cost-complexity criterion (Hastie et al., 2009). The optimal value of α is chosen through cross-validation.

Regression trees use the residual sum of squares to decide what splits to make, so we expect that they will do poorly under heteroscedasticity, both in the linear and nonlinear case. However, regression trees do not make any assumption about the shape of the relationship between the response and explanatory variables, so we expect them to perform well under nonlinearity. We use both full and pruned regression trees, and expect the pruned trees to outperform the full trees, since they focus on important trends in the data and do not overfit.

4.5 Random Forests

Regression trees are highly variable, and small changes in data can lead to very different splits and results. Despite their popularity, individual regression trees are not powerful predictors. However, they have properties that make them good candidates for combining via ensemble methods to obtain less variable predictions. Ensemble methods combine multiple predictors that have low bias but high variability. In doing so, the variability of the averaged predictor is reduced (Hastie et al., 2009).

Random forests are one example of an ensemble method. They are constructed by averaging the predictions from B trees. Each tree is based on a bootstrapped resampling of the data, and at each split, $m \leq p$ explanatory variables are randomly chosen as candidates to split on. This results in lower correlation between individual trees, because they are not all based on the same data or on the same variables (Izenman, 2013). The variance of the average of B identically distributed random variables, each with variance σ^2

and pairwise correlation ρ is $\rho\sigma^2 + \frac{1-\rho}{B}\sigma^2$ (Hastie et al., 2009). Increasing B reduces the variance of this average, as does reducing ρ. Applying this to an ensemble suggests that averaging trees that are less correlated, but perhaps slightly more variable than a typical pruned tree, results in lower variance of the ensemble.

Since an ensemble method reduces variability by design, pruning individual trees is not of primary importance. However, in order to prevent excess variability that averaging cannot fix, we set a maximum number of terminal nodes, k, that each tree can grow to. In a random forest, the prediction for y_i is made by averaging the predictions from the trees that do not have y_i as part of their sample; i.e., it is part of the "out-of-bag" sample (Izenman, 2013). The tuning parameters $B, m,$ and k are chosen to minimize out-of-bag error. This process allows a way to select tuning parameters without the additional step of cross-validation.

Since random forests are built from multiple regression trees, they also do not use any assumption about a linear relationship between the explanatory variables and the response, so we do not expect them to be adversely affected by nonlinearity. However, the individual trees are constructed based on the minimization of the residual sum of squares. We therefore expect that random forests will also be sensitive to heteroscedasticity. We do anticipate that they will perform much better than an individual regression tree.

4.6 Boosted Regression Trees

Boosted regression trees, also known as boosting, is another ensemble method that combines regression trees. It differs from random forests in a number of ways. First, the ensemble building process is sequential. We start by taking the mean response as the prediction for all observations. Then a regression tree is fit to the residuals from the initial prediction. The new predictions are multiplied to by a shrinkage parameter, ν, and added to the previous predictions. This process is repeated M times, and each new regression tree is fit to the residuals of the predictions from the weighted sum of the previous trees. The shrinkage parameter, ν, is typically small, which allows the ensemble to adapt to the data slowly before it overfits. The number of trees M needed is therefore related inversely to ν. This process is called Gradient Boosting, with details available in Izenman (2013).

Another key difference is that the trees are typically small. The maximum depth of variable interactions, J, is not usually greater than 4 or 5. Keeping J small ensures that each individual tree does not explain too much, and allows the new trees in the sequence to "catch" the patterns that the previous ones missed. According to Hastie et al. (2009), combining many “weak” predictors in this way results in a powerful ensemble.

Boosting can also use a bagging fraction η, which determines the proportion of data that each tree is based on. Having each tree based on a fraction of the data reduces the correlation between trees and improves performance over using $\eta = 1$ (Hastie et al., 2009). A typical choice is $\eta = 0.5$. The parameters ν and J are chosen through cross-validation. Boosting is prone to overfitting if too many trees are used, so M is chosen through cross-validation separately for each combination of ν and J.

Similarly to random forests, boosting is based on multiple regression trees and no assumptions are made about data linearity. We expect boosting to do well in both linear and nonlinear cases. Again, each tree is built by minimizing the residual sum of squares, so boosted regression trees may also be adversely affected by heteroscedasticity. We anticipate that boosting will perform better than individual regression trees, given that they are an ensemble of “weak” predictors and are therefore more powerful. They may also perform better than random forests.

4.7 Multivariate Adaptive Regression Splines

Multivariate adaptive regression splines (MARS) is a flexible regression method that produces continuous models Friedman (1991). In MARS, the response $\mathbf{Y}$ is related to the explanatory variables through the model $\mathbf{Y} = f(\mathbf{X}) + \boldsymbol{\epsilon}$, where ϵ has mean $\mathbf{0}$ and $f(\mathbf{X})$ is a weighted sum of M basis functions, $f(\mathbf{X}) = \beta_0 + \sum_{m=1}^{M} \beta_m h_m(\mathbf{X})$ (Hastie et al., 2009). The m^{th} basis function, $h_m(\mathbf{X})$, is either a spline function or the product of two or more spline functions (Izenman, 2013). In particular, the splines that are used by most MARS implementations are “hinge” functions that are flat for $X_{ij} < c$ and linear for $X_{ij} > c$ or vice versa. Selecting j and c uses a process similar to regression trees, in the sense that the chosen hinge function is the one that reduces the residual sum of squares the most.

The model is built by first fitting the mean of the responses as predictions. At each step, basis functions are added into the model, in terms of which gives the largest reduction in the residual sum of squares (Hastie et al., 2009). The final model is too large and overfits the data, so a backward deletion process is used to obtain a smaller model. The maximum degree, d, of interactions allowed (i.e., the number of products in the basis functions) is usually constrained to be small, with $d = 1, 2,$ or 3 as common choices. The value of d can be chosen through cross-validation.

We anticipate that MARS will be an excellent predictor for the nonlinear data, since the basis functions model nonlinearity through interactions. Given that the model is grown by minimizing the residual sum of squares, there is again the potential that MARS will be adversely affected by heteroscedasticity.

4.8 Artificial Neural Nets

Artificial neural networks (ANNs) constitute a two-stage modelling process. The explanatory variables are combined into a layer of "hidden nodes," and the results from the nodes are combined to make predictions. More specifically, the M hidden nodes are formed as a linear combination of the explanatory variables, with $Z_m = \sigma(\alpha_{0m} + \mathbf{X}\boldsymbol{\alpha_m})$, where $\sigma(v)$ is usually the sigmoidal function, $\sigma(v) = \frac{1}{1+e^{-v}}$ (Hastie et al., 2009). In regression, the response is then formed as a linear combination of the hidden nodes, as $\mathbf{Y} = \beta_0 + \sum_{m=1}^{M} \beta_m Z_m$.

The weights in ANNs consist of $\alpha_{0m}, \beta_0, \beta_m$, and all the terms in $\boldsymbol{\alpha_m}$ for $m = 1, \ldots, M$, for a total of $M(p+1) + (M+1)$ parameters (Hastie et al., 2009). These are estimated by minimizing the residual sum of squares, through an approach called back-propagation. ANNs are prone to overfitting, so the parameters are found by minimizing a regularized criterion, $RSS + \lambda\{||\boldsymbol{\beta}||^2 + ||\boldsymbol{\alpha}||^2\}$, where $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_M)$, and $\boldsymbol{\alpha}$ contains all the α parameters. The penalty term, λ, is called the decay parameter. We find the optimal value of λ in combination with the optimal number of hidden nodes, M, through cross-validation. ANNs are a random process, because the back-propagation algorithm is initialized with random weights. They should be run several times with predictions averaged as final results. The back-propagation algorithm is initialized with random weights

Artificial neural networks do not make any assumptions about the relationship between the explanatory and response variables. We do not expect them to be adversely affected by nonlinearity. However, the weights are estimated by minimization of the residual sum of squares, so they may be adversely affected by heteroscedasticity.

4.9 Bayesian Additive Regression Trees

Bayesian additive regression trees (BART) are the most novel of the modern methods we compare. Developed by Chipman et al. (2010), BART is also a tree based ensemble. The relationship between the explanatory variables and the response is modelled as $\mathbf{Y} = f(\mathbf{X}) + \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim Normal(\mathbf{0}, \sigma^2\mathbf{I})$, and $f(\mathbf{X})$ is the sum of regression trees. This differs from the other ensemble methods we discussed (random forests and boosting) because the errors are *explicitly* assumed to be homoscedastic. The model for $\mathbf{X}$ is the *sum* of regression trees, similarly to boosting (which uses a weighted sum), but differently from random forests (which use an average).

Each tree is built to explain parts of the relationship that the other trees do not. As the name implies, the model building process is Bayesian. The prior distribution on the structure of individual trees affects their size. In particular, changing the hyperparameters results in different probabilities on the number of terminal nodes for each tree, allowing them to become generally larger or smaller. By default, they are set to ensure that the individual trees remain small and only capture main effects or a small number of interaction effects.

There are two hyperparameters, k and M, that control the prior distribution of the values assigned to terminal nodes. The first, k, controls the shrinkage of these values, while M is the total number of regression trees in the ensemble. Greater values of k and M result in terminal node values that are closer to 0, ensuring the trees do not overfit too quickly. This process is a scaled version of assigning high probability to the terminal node values being between y_{min} and y_{max}, with k resulting in shrinkage (Chipman et al., 2010). The values of k and M are chosen through cross-validation.

The prior on σ^2 is taken to be an inverse chi-square distribution, with hyperparameters ν and λ. They are chosen to give a distribution of “reasonable” values of σ, relative to $\hat{\sigma}$, a data-based estimate of σ. This can be taken to be the sample standard deviation of the responses or the standard deviation of the residuals from a linear model (Chipman et al., 2010). The parameter ν controls the shape of the distribution, while λ is chosen such that the q^{th} quantile of the prior on σ is located at $\hat{\sigma}$. That is, λ is chosen so that $P(\sigma < \hat{\sigma}) = q$. We choose (ν, q) in combination, and λ is adjusted accordingly. Again, (ν, q) are chosen through cross-validation, *in combination* with values of k and M.

BART uses a Bayesian backfitting MCMC algorithm to fit the trees. This is a cyclical process, with later trees influencing the formation of earlier trees. It results in samples that constitute the approximate posterior distribution of the true regression function $f(\mathbf{X})$. Predictions of $\mathbf{Y}$ are found by taking means from this distribution.

To reiterate, BART is different from other ensemble methods in that it *explicitly* assumes homoscedasticity. Therefore, we expect it to do poorly on heteroscedastic data. No assumptions are made about the form of the relationship between the explanatory and response variables for the individual trees that BART is based on. We anticipate that it will not be adversely affected by nonlinearity.

Chapter 5

Simulation Study

In this chapter we develop a simulation study to compare the predictive ability of the nine regression methods outlined in Chapter 4. We examine how each method performs when faced with heteroscedasticity in conjunction with, and separately from, other challenges present in modern data.

5.1 Simulation Data Models

We consider two heteroscedastic models for our simulation study, each with a different form for the relationship between the explanatory and response variables. The first case we consider is linear heteroscedastic data, arising from the model

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim Normal\left(\mathbf{0}, \sigma^2 diag\left((\mathbf{X}\boldsymbol{\beta})^2\right)\right),$$

where here squaring a vector is done elementwise. For this model, $\boldsymbol{\mu} = E(\mathbf{Y}) = \mathbf{X}\boldsymbol{\beta}$ and $Var(\mathbf{Y}) = \sigma^2 diag\left((\mathbf{X}\boldsymbol{\beta})^2\right)$. We use these properties to evaluate the performance of each prediction method, as we discuss in Section 5.4.

We also use these data with a log transformation applied to the response. Using a log transformation when the variance is proportional to the square of the mean stabilizes the variance (Carroll and Ruppert, 1988). However, this transformation destroys the linear relationship, resulting in nonlinear homoscedastic data with $\boldsymbol{\mu} = E(\mathbf{Y}) = \log(\mathbf{X}\boldsymbol{\beta})$ (again, with the log taken elementwise) and $Var(\mathbf{Y}) = \tau^2\mathbf{I}$.

The second case is nonlinear heteroscedastic data, generated from a log-normal model as

$$\mathbf{Y} = \exp\left\{\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}\right\}, \quad \boldsymbol{\epsilon} \sim Normal\left(\mathbf{0}, \sigma^2\mathbf{I}\right),$$

where, $\boldsymbol{\mu} = E(\mathbf{Y}) = \exp\left\{\mathbf{X}\boldsymbol{\beta} + \frac{\sigma^2}{2}\right\}$ and $Var(\mathbf{Y}) = diag\left(\exp\left\{2\mathbf{X}\boldsymbol{\beta} + \sigma^2\right\}\right)\left(\exp\left\{\sigma^2\right\} - 1\right)$. We use the log-transformed version of these data as well. The result is linear homoscedastic data with $\boldsymbol{\mu} = E(\mathbf{Y}) = \mathbf{X}\boldsymbol{\beta}$ and $Var(\mathbf{Y}) = \sigma^2\mathbf{I}$.

While we aim to understand how well the prediction methods work on these four distinct groups of data, our primary goal in using the log transformation is to make predictions on the original heteroscedastic data. That is, we would like to see how transforming heteroscedastic data to homoscedastic data, using the methods on the homoscedastic data, then transforming the predictions back to their original form, compares to using the methods on the original form of the data.

5.2 Simulation Factors

Based on the simulation study in Payne (2014), we vary three factors for each of the model cases, resulting in a total of 24 simulation scenarios. The first factor is the number of explanatory variables, p. We use $p = 10$ and $p = 100$ in our simulations. The median number of explanatory variables in the bake-off data from Chipman et al. (2010) is 10, so using $p = 10$ represents a typical number of explanatory variables, while $p = 100$ represents a "large" number of variables. Of course, taking high-dimensional data (e.g., genetic data) into account, $p = 100$ is not that large.

The second factor we vary is the sparsity, which is the proportion of unimportant explanatory variables. Varying the sparsity allows us to examine how the methods' performances degrade when faced with uninformative noise variables. We use sparsity of 0%, 50%, and 80%. Sparsity of 0% indicates that all explanatory variables are important in the model-building process. Sparsity of 50% indicates that the last 50% of the explanatory variables are unimportant in the model building process, and their coefficients are zero. Similarly, sparsity of 80% indicates that last 80% of the explanatory variables are unimportant in the model building process. In each case, we set all non-zero regression parameters to be equal, with values for different scenarios as given in Tables 5.1 and 5.2.

Finally, we vary the signal-to-noise ratio (SNR) to be 1 and 5. The SNR measures how strong the relationship is between the explanatory and response variables, compared to the amount of error noise in the data. It is the variance of the means divided by the variance of the data around their respective means. For linear heteroscedastic data, it is

$$SNR = \frac{E_{\mathbf{X}}[(\mathbf{x}'\boldsymbol{\beta} - E_{\mathbf{X}}(\mathbf{x}'\boldsymbol{\beta}))'(\mathbf{x}'\boldsymbol{\beta} - E_{\mathbf{X}}(\mathbf{x}'\boldsymbol{\beta}))]}{\sigma^2[E_{\mathbf{X}}(\mathbf{x}'\boldsymbol{\beta})]^2}.$$

For linear homoscedastic data, it is

$$SNR = \frac{1}{\sigma^2}E_{\mathbf{X}}[(\mathbf{x}'\boldsymbol{\beta} - E_{\mathbf{X}}(\mathbf{x}'\boldsymbol{\beta}))'(\mathbf{x}'\boldsymbol{\beta} - E_{\mathbf{X}}(\mathbf{x}'\boldsymbol{\beta}))], 6$$

as in Dicker (2012). The linear homoscedastic data are then exponentiated into data for the nonlinear heteroscedastic case. In both cases, the expectations are approximated by simulation for given values of $\beta_0, \beta_{j|j>0}$, and σ, taking into consideration the number of explanatory variables and sparsity in each scenario.

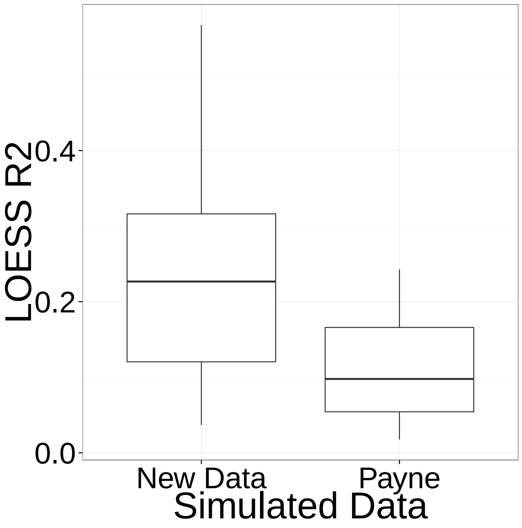
A SNR of 5 indicates the amount of “signal” is five than the amount of noise, while $SNR = 1$ indicates that they are present in equal amounts. We anticipate that all of the methods will degrade when the relationship is unclear and conflated with noise.

5.3 Simulation Details

We generate data for the simulation scenarios as follows. The explanatory variables, X_j, are generated as standard normal random variables with AR-1 correlation such that $Corr(X_l, X_k) = 0.8^{|l-k|}$ for all l, k. We simulate 500 data sets for each scenario, each with $n = 1000$ different observations. The values of the simulation parameters, $\beta_0, \beta_{j|j>0}$, and σ, vary in combination with the simulation factors in order to obtain $SNR = 1$ and $SNR = 5$. They are also chosen to achieve data that is “heteroscedastic enough,” as discussed in Chapter 4. We aim to generate data that contain a magnitude of heteroscedasticity equivalent to the “typical” amount in the bake-off data from Chipman et al. (2010), where the median SD Ratio is 3.079. We therefore aim for data in each scenario with an SD Ratio around 3, computed as in Chapters 3 and 4.

We also aim to generate nonlinear data that are “more nonlinear” than the data in Payne (2014), as described in Chapter 3. Figure 5.1 shows box plots of the LOESS R^2 (computed as in Chapter 3) of our nonlinear heteroscedastic data, versus the LOESS R^2 of the nonlinear heteroscedastic data in Payne (2014). Each box is based on the first ten simulated data sets for each nonlinear heteroscedastic scenario. Our data achieve higher LOESS R^2 and therefore convey more nonlinearity than the data from Payne (2014).

Figure 5.1: Box plots of the LOESS R^2 of our nonlinear heteroscedastic data and the nonlinear heteroscedastic data in Payne (2014).



Tables 5.1 and 5.2 show the simulation scenarios, parameter settings, and SD Ratios for the linear heteroscedastic data and nonlinear heteroscedastic data, respectively. The simulation scenarios are named to include information on the data linearity, the sparsity, the number of explanatory variables, and the signal-to-noise ratio. The names are formatted as “linearity.sparsity.p.SNR,” where linearity is NL (nonlinear) or L (linear); sparsity is 00 (0% sparsity), 50 (50%), or 80 (80%); p is 010 or 100; and SNR is 1 or 5.

Another consideration is that we require positive values of Y in the linear heteroscedastic case, since we also use the log-transformed version of these data. Given randomness in the data generation process, the parameter settings do not guarantee strictly positive responses. In the linear heteroscedastic case, we kept only those observations with Y greater than 0. In no case were more than 1% of observations discarded, and the SNR and SD Ratio were unaffected.

Table 5.1: Table showing simulation parameter settings for linear heteroscedastic data.

Scenario	Linearity	Sparsity	p	SNR	σ	$\beta_{j\|j>0}$	β_0	SD Ratio
L.00.010.1	Linear	0%	10	1	0.303	2.505	61.017	3.21
L.00.010.5	Linear	0%	10	5	0.135	1.120	27.288	3.21
L.00.100.1	Linear	0%	100	1	0.303	0.629	61.017	2.47
L.00.100.5	Linear	0%	100	5	0.135	0.281	27.288	2.56
L.50.010.1	Linear	50%	10	1	0.303	4.338	61.017	3.30
L.50.010.5	Linear	50%	10	5	0.135	1.940	27.288	3.32
L.50.100.1	Linear	50%	100	1	0.303	0.912	61.017	2.59
L.50.100.5	Linear	50%	100	5	0.135	0.408	27.288	2.59
L.80.010.1	Linear	80%	10	1	0.303	9.729	61.017	3.30
L.80.010.5	Linear	80%	10	5	0.135	4.351	27.288	3.35
L.80.100.1	Linear	80%	100	1	0.303	1.557	61.017	2.60
L.80.100.5	Linear	80%	100	5	0.135	0.697	27.288	2.73

Table 5.2: Table showing simulation parameter settings for nonlinear heteroscedastic data.

Scenario	Linearity	Sparsity	p	SNR	σ	$\beta_{j\|j>0}$	β_0	SD Ratio
NL.00.010.1	Nonlinear	0%	10	1	0.360	0.050	0.100	3.71
NL.00.010.5	Nonlinear	0%	10	5	0.162	0.050	0.100	3.49
NL.00.100.1	Nonlinear	0%	100	1	0.500	0.018	0.100	4.06
NL.00.100.5	Nonlinear	0%	100	5	0.234	0.018	0.100	3.26
NL.50.010.1	Nonlinear	50%	10	1	0.370	0.090	0.100	4.15
NL.50.010.5	Nonlinear	50%	10	5	0.170	0.090	0.100	3.79
NL.50.100.1	Nonlinear	50%	100	1	0.500	0.025	0.100	4.30
NL.50.100.5	Nonlinear	50%	100	5	0.225	0.025	0.100	3.35
NL.80.010.1	Nonlinear	80%	10	1	0.370	0.200	0.100	3.93
NL.80.010.5	Nonlinear	80%	10	5	0.170	0.200	0.100	3.55
NL.80.100.1	Nonlinear	80%	100	1	0.470	0.040	0.100	4.42
NL.80.100.5	Nonlinear	80%	100	5	0.210	0.040	0.100	3.82

5.4 Performance Measures

We use two measures to evaluate the performance of each regression method. The first measure used is the mean squared error (MSE), defined as

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\mu_i - \hat{y}_i)^2$$

It measures the average squared distance between the expected value of the response, μ_i, and its prediction, $\hat{y}_i$. The MSE is popular, but is sensitive to large prediction errors (i.e., large values of $\mu_i - \hat{y}_i$), since they are squared. It also does not account for heteroscedasticity in data, since all errors are treated equally in the sum, regardless of the local variance.

Table 5.3: Table showing R function and package used, and levels of tuning parameters, for all methods used in the simulation study.

Method	Function (Package)	Tuning Parameters
Linear Regression	lm (stats)	—
Stepwise Linear Regression	step (stats)	—
LASSO	cv.glmnet (glmnet)	—
Regression Trees	rpart (rpart)	—
Random Forests	randomForest (randomForest)	$B = 250, 500, \ldots, 1500$
		$m = (1/4, 1/3, 1/2) \times p$
		$k = 10, 25, 40, 50, 75, 100, 125, 150, 200, 300$
Boosting	gbm (gbm)	$J = 1, 2, 3, 4$
		$\nu = 0.01, 0.05, 0.1, 0.025$
		$\eta = 0.5$
MARS	earth (earth)	$d = 1, 2, 3$
ANNs	nnet (nnet)	$M = 1, 2, 3, 4$
		$\lambda = 0.001, 0.01, 0.1$
BART	bart (dbarts)	$k = 2, 3, 5$
		$M = 50, 200$
		$(\nu, q) = (10, 0.75), (3, 0.90), (3, 0.90)$

The second measure used is the median absolute standardized deviation (MASD), defined as

$$MASD = median\left[\frac{|\mu_i - \hat{y}_i|}{\sigma_i}\right],$$

where σ_i is the standard deviation of Y at $\mathbf{x}_i'$. The MASD measures the median number of standard deviations, σ_i, the predictions $\hat{y}_i$ are from the means, μ_i. The median is used instead of the mean so that the measure is not excessively influenced by occasional large prediction errors in low-variance data.

5.5 Computational Details

All simulations are run using R (R Core Team, 2015). Table 5.3 shows the R function and package used, along with the levels of tuning parameters considered, for each method.

For the LASSO, the best value of the penalty parameter λ is chosen separately for each simulated data set as the value that minimizes five-fold cross-validation error.

The tuning process for random forests, boosting, ANNs, and BART is as follows. We fit a model with each of the combinations of tuning parameters to the first three simulated data sets for each scenario. We then choose a combination that workw “well” for all three data sets and use that combination for all 500 simulated data sets for that scenario. This is done because it would not be feasible, in terms of computation time, to tune these four methods on all 24,000 simulated data sets. Given that each data set is simulated from the same distribution, we expect that a combination of tuning parameters that works well on the first three should work reasonably well on the rest.

For random forests, the best combination of tuning parameters is chosen in terms of minimizing the mean out-of-bag error. For boosting, the best combination of the tuning parameters ν and J is chosen in terms of minimizing five-fold cross-validation error. Given this combination, the number of trees, M, is chosen for each simulated data set through another round of five-fold cross-validation. For ANNs, the best combination of tuning parameters is chosen in terms of minimizing mean five-fold cross-validation error, from ten ANNs fit each time. The average from ten neural networks is used to make the predictions because of the randomness of this method. For BART, the best combination of tuning parameters is chosen in terms of minimizing five-fold cross-validation error. The tuning parameters used for each scenario are given in Tables A.1 and A.2.

We attempted the same tuning process for MARS. However, there was no consensus best choice of d among the first three data sets for scenarios for $p = 10$. Therefore the best value of the degree d is chosen from the values in Table 5.3 separately for each simulated data set, again as the value that minimizes five-fold cross-validation error. While different values of d did not result in large differences in error, the speed of fitting MARS for scenarios with $p = 10$ made it fast and plausible to choose d separately for each data set. However, for scenarios with $p = 100$, degree $d = 3$ was consistently the best in the first three data sets, so this value was used for all 500 simulated data sets with $p = 100$.

5.6 Results

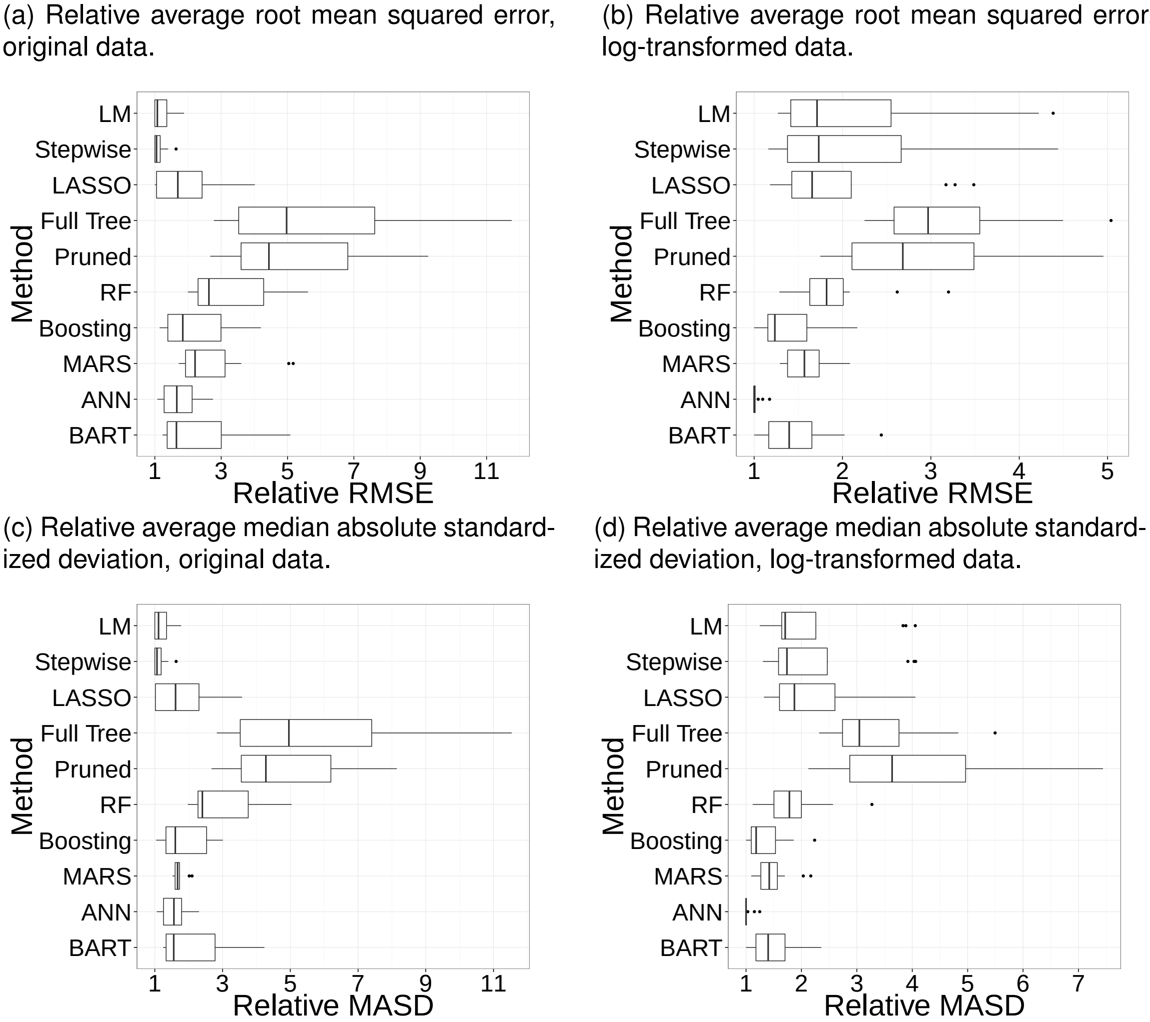
Tables B.1, B.2, B.3, and B.4 show the average mean squared error and average median absolute standardized deviation, along with their standard errors, for the linear heteroscedastic and nonlinear heteroscedastic data, both from the data's original form and from the variance-stabilizing log transformation. For each scenario, the method that minimizes the average error is highlighted.

5.6.1 Linear Heteroscedastic Data Results

From the average mean squared errors for the linear heteroscedastic data in Table B.1, the linear methods (linear regression, stepwise linear regression, and the LASSO) consistently perform best on the untransformed linear data, despite the heteroscedasticity. When the log transformation is used, the data are nonlinear and homoscedastic. In this form, ANNs perform very well, though they are sometimes surpassed in performance by boosting and BART. We generally see a degradation of the methods, in terms of average MSE, when the data are log transformed.

These results are consistent when looking at the average MASD, in Table B.2. The linear-regression-based methods outperform the more "modern" methods on the linear heteroscedastic data, while ANNs, boosting, and BART do better on the log-transformed data. Most of the methods' overall predictive abilities degrade in terms of MASD when the transformation is used, though the degradation is not nearly as significant as when looking at the MSE.

Figure 5.2: Box plots showing relative average RMSE and MASD for the linear heteroscedastic data, in its original form and with the use of a variance-stabilizing log transformation.


(a) Relative average root mean squared error, original data.

(b) Relative average root mean squared error, log-transformed data.

(c) Relative average median absolute standardized deviation, original data.

(d) Relative average median absolute standardized deviation, log-transformed data.

Figure 5.2 shows the relative average performance of the methods, in terms of relative average root mean squared error (RMSE) and relative average MASD, on the original and log-transformed data. These values are found by dividing the values in each row of Tables B.1 or B.2 by the minimum error for that scenario. Each box is therefore based on 12 points. The results are consistent with the preliminary look at the errors in terms of which methods perform best. The figures also provide an easier way to compare how the methods perform. For all cases and error measures, the performance of trees (both full and pruned) is poor. Pruned regression trees outperform full regression trees, except when looking at the relative average MASD on the log-transformed data. Boosting and BART seem to perform overall similarly, slightly better than random forests, and worse than ANNs.

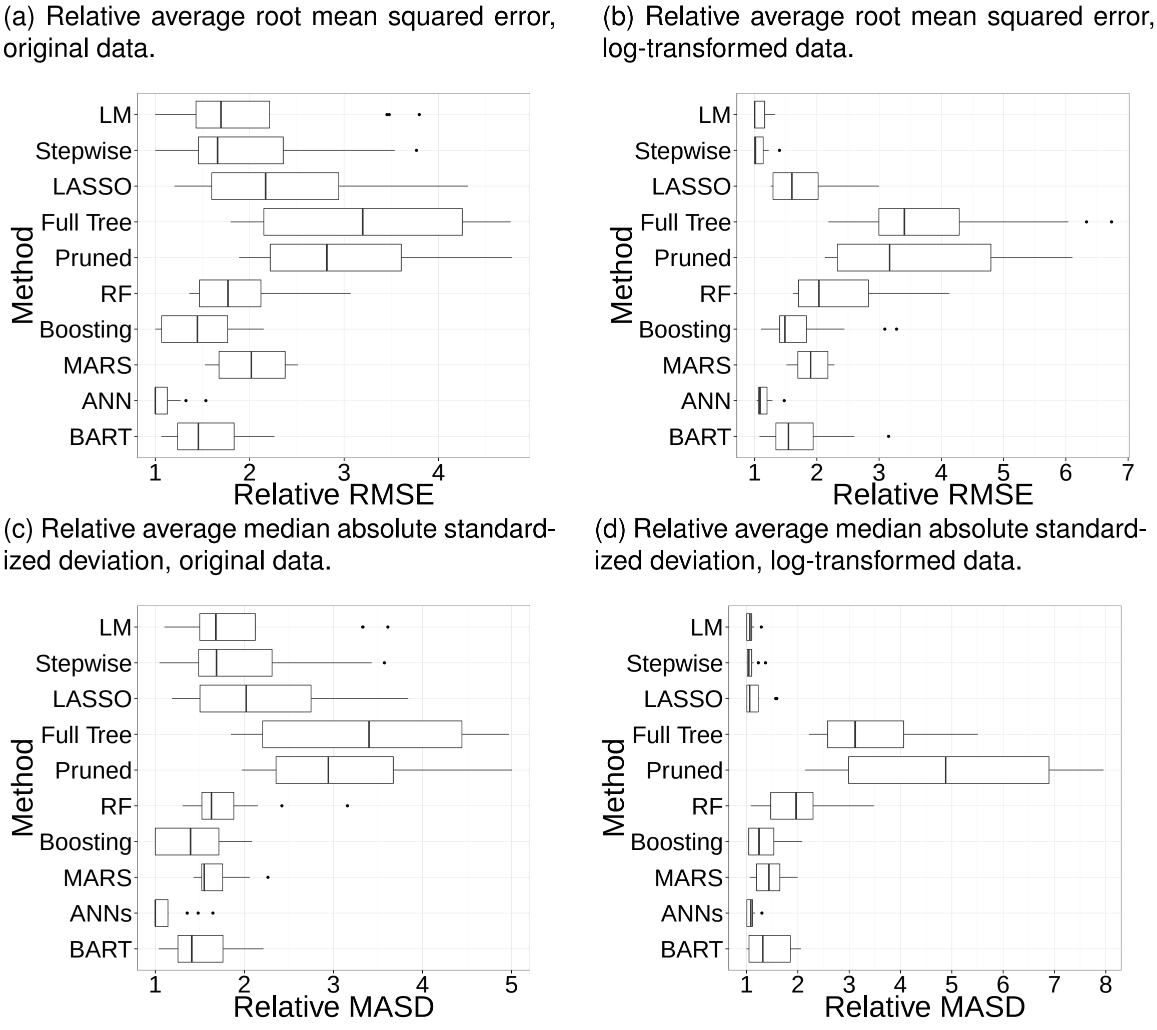
The relative performances are very similar when looking at either the relative RMSE or the relative MASD. One exception is the relative performance of MARS on the original data. Figure 5.2c shows the relative average MASD on the original data. In this case, MARS' relative performance is consistent across all scenarios. This is in contrast to its relative average RMSE in Figure 5.2a, where its performance is much more variable. Taking heteroscedasticity into account in this case shows the stability of MARS' predictive ability on linear heteroscedastic data.

5.6.2 Nonlinear Heteroscedastic Data Results

Looking at the average MSE in Table B.3, ANNs are fairly consistent in terms of having smallest MSE on the original data. When the log transformation is used, the linear models work best. In particular, linear regression has the smallest MSE for scenarios with no sparsity, while stepwise linear regression does when there is considerable sparsity. The use of the log transformation results in an improvement of almost all of the methods, since it produces linear homoscedastic data. This result is not consistent for regression trees (full or pruned) or random forests, whose performances degrade when the transformation is used. This may be because regression tree fit constants, rather than slopes, and have a difficult time handling linearity. We see the same patterns of best methods when looking at the average MASD, in Table B.4. However, when the MASD is used instead of the MSE, regression trees' and random forests' performances improve after the log transformation.

Figure 5.3 shows the relative average RMSE and the relative MASD for the nonlinear heteroscedastic data. These values are found by dividing the values in each row of Tables B.3 or B.4 by the minimum error for that scenario. For the nonlinear heteroscedastic data in its original form, as shown in Figure 5.3a, ANNs have the best performance in terms of average RMSE in more than half of the scenarios. In some scenarios, linear regression and stepwise linear regression do best, though for the most part their RMSEs are, at best, almost twice as large as ANNs'. The performance of boosting and BART are quite close, though boosting achieves the smallest average RMSE in one scenario while BART never does. Regression trees (both full and pruned) perform poorly. Again, we see that pruned regression trees perform similarly to (or better than) full regression trees, except when looking at the MASD of the log-transformed data.

Figure 5.3: Box plots showing relative average RMSE and MASD for the nonlinear heteroscedastic data, in its original form and with the use of a variance-stabilizing log transformation.



(a) Relative average root mean squared error, original data.

(b) Relative average root mean squared error, log-transformed data.

(c) Relative average median absolute standardized deviation, original data.

(d) Relative average median absolute standardized deviation, log-transformed data.

For the nonlinear heteroscedastic data after a log transformation, linear regression and stepwise linear regression have the best performance in terms of average RMSE, as shown in Figure 5.3b. This is to be expected, since the transformed data are linear and homoscedastic, which are ideal conditions for linear methods. Interestingly, ANNs perform almost as well as the linear regression methods. Regression trees again perform poorly, and boosting's and BART's performances are close.

The results are fairly consistent across the two measures. One exception is the LASSO – we see considerable improvement in its performance when looking at the MASD instead of the MSE on the log-transformed data, as in Figure 5.3d.

5.6.3 Simulation Factor Results

In our simulation study, we vary the number of explanatory variables (p), the proportion of unimportant explanatory variables (sparsity), and the signal-to-noise ratio. We look at how each of these factors impacts the performance of the methods.

The number of explanatory variables has a consistent effect on each method's performance. Increasing from $p = 10$ to $p = 100$ results in worse mean estimation with all of the methods, regardless of sparsity and signal-to-noise ratio. This effect is consistent across nonlinear and linear heteroscedastic data, and the use of the log transformation. It appears when looking at either the MSE and MASD.

When the sparsity changes, linear regression performs the same. This is consistent in both linear- and nonlinear heteroscedastic data, and when we look at the MSE or the MASD. This is because the SNR remains the same, so linear regression has the same potential. Stepwise linear regression and the LASSO both improve in terms of error as the sparsity increases, especially in scenarios where $p = 100$. These methods are intended to produce sparse solutions, so we expected that they would do well when faced with a large number of unimportant variables. We see some improvement in the performance of regression trees and MARS as the sparsity increases. Meanwhile, boosting, ANNs, and BART are not consistently affected by changes in sparsity.

The last factor we vary is the signal-to-noise ratio. For the most part, the errors are much lower when $SNR = 5$ than when $SNR = 1$. This is to be expected, since the relationship between the explanatory and response variables is clearer in the former case. We do not see this same effect when looking at the MASD of the nonlinear heteroscedastic data in its original form, specifically when looking at linear regression. Linear regression performs better on low SNR data in this case. The same pattern continues when looking at the MASD of nonlinear data. When $SNR = 1$ instead of $SNR = 5$, the curvature in the data is likely not as prominent. All of the other methods perform better with high signal-to-noise ratio.

We used a variance-stabilizing log transformation with the goal of making predictions on the original data. When the transformation is used on linear heteroscedastic data, the data become homoscedastic, but the linearity is destroyed. In this case, the methods mostly performed worse on the log-transformed data. One exception is regression trees, which are not as affected by this transformation as the other methods.

Using the log transformation on the nonlinear heteroscedastic data produces linear heteroscedastic data. As we expected, this improved the performance of almost all methods, especially the methods that explicitly assume linearity. Improvements are not consistent when looking at the MSE of regression trees or random forests, but are when the MASD is used to quantify error.

Chapter 6

Conclusions and Future Work

In this thesis we aimed to develop a better understanding of the impact of heteroscedasticity on the predictive ability of modern regression methods. We first discussed methods for recognizing and quantifying heteroscedasticity. We then implemented them on 42 real data sets to develop knowledge on the "typical" prevalence and magnitude of heteroscedasticity. This knowledge was used to develop a simulation study, where we varied data linearity with heteroscedasticity in addition to other simulation factors. In this chapter, we summarize conclusions from our work and discuss some limitations and opportunities for further research.

We have a number of findings from our simulation study, some of which are consistent with the hypotheses we made about each method in Chapter 4. A surprising result is that the linear methods (linear regression, stepwise linear regression, and the LASSO) outperformed many of "modern" methods on linear heteroscedastic data, despite the fact that homoscedasticity is an explicit assumption in the linear methods. It is possible that these data still do not contain "enough heteroscedasticity," despite our best efforts. It was a challenge to generate linear heteroscedastic data with a high SD Ratio, while ensuring that most of the responses were positive.

Despite their popularity, regression trees performed worst for all cases and combinations of simulation factors. However, they were least affected by the variance-stabilizing log transformation, likely because no assumptions are made about the model between the explanatory and response variables. The ensemble methods based on regression trees had considerably better performance. In particular, boosting and BART had similar performances.

While their performance was not always the "best," artificial neural nets performed very well in almost all cases, regardless of data linearity or the form of the error variance. ANNs make no assumptions about the data model or about heteroscedasticity, and proved to be robust predictors.

We achieved a number of goals set forth in writing this thesis. In particular, we developed an understanding of how to identify and quantify heteroscedasticity, and we attempted to simulate data that were “heteroscedastic enough” to challenge our prediction methods. We varied a number of simulation factors to evaluate how they affected the methods. However, our findings are limited to the constraints of the models we chose, which still have limitations in their challenges (e.g., no interactions between explanatory variables). We also ran a large enough number of simulations to get a precise look at how the methods performed, and tuned the more modern methods in order to optimize their performance.

For future work, we recommend a reconsideration of the data generation process. In particular, more research needs to be done on achieving “high enough” SD Ratios in conjunction with the other factors (signal-to-noise ratio and “enough nonlinearity,” in particular). For the linear heteroscedastic case, our data generation process was constrained by requiring almost all positive response values, which impacted the highest achievable SD Ratio. It would be useful to consider other models for generating our data, in addition to looking into theoretical methods for finding simulation parameter settings.

Bibliography

G.E.P. Box and D.R. Cox. An analysis of transformations. *Journal of the Royal Statistical Society. Series B (Methodological)*, 26(2):211–252, 1964.

Raymond J. Carroll and David Ruppert. *Transformation and weighting in regression*. Chapman and Hall, 1988.

Hugh A. Chipman, Edward I. George, and Robert E. McCulloch. BART: Bayesian Additive Regression Trees. *Annals of Applied Statistics*, 4(1):266–298, 2010.

R. Dennis Cook and Sanford Weisberg. *Residuals and influence in regression*. New York Chapman and Hall, New York, 1982.

M Dowle, T Short, S Lianoglou, R Srinivasan, A with contributions from Saporta, and E Antonyan. *data.table: Extension of data.frame*, 2014. URL `http://CRAN.R-project.org/package=data.table`. R package version 1.9.4.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010. URL `http://www.jstatsoft.org/v33/i01/`.

Jerome H. Friedman. Multivariate Adaptive Regression Splines. *The Annals of Statistics*, 19(1):1–67, 1991.

Trevor Hastie, Jerome Friedman, and Robert Tibshirani. *The elements of statistical learning: data mining, inference, and prediction*. Springer-Verlag New York, second edition, 2009.

Alan J. Izenman. *Modern multivariate statistical techniques: regression, classification, and manifold learning*. Springer, second edition, 2013.

Hyunjoong Kim, Wei-Yin Loh, Yu-Shan Shih, and Probal Chaudhuri. Visualizable and interpretable regression models with good prediction power. *IIE Transactions*, 39:565–579, 2007.

Andy Liaw and Matthew Wiener. Classification and regression by randomForest. *R News*, 2(3):18–22, 2002. URL `http://CRAN.R-project.org/doc/Rnews`.

Stephen Milborrow. *earth: Multivariate Adaptive Regression Splines*, 2015. URL `http://CRAN.R-project.org/package=earth`. R package version 4.3.0.

Nathaniel Payne. Evaluating the impact of heteroscedasticity on the predictive ability of modern regression techniques. Master's thesis, Simon Fraser University, 2014.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2015. URL `http://www.R-project.org/`.

Gideon Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6(2):461–464, March 1978.

Terry Therneau, Beth Atkinson, and Brian Ripley. *rpart: Recursive Partitioning and Regression Trees*, 2015. URL `http://CRAN.R-project.org/package=rpart`. R package version 4.1-9.

W.N. Venables and B.D. Ripley. *Modern applied statistics with S*. Springer, fourth edition, 2002.

Halbert White. A heteroskedasticity-consistent covariance matrix estimator and a direct test for heteroskedasticity. *Econometrica*, 48(4):817–838, 1980.

Greg Ridgeway with contributions from others. *gbm: Generalized Boosted Regression Models*, 2015. URL `http://CRAN.R-project.org/package=gbm`. R package version 2.1.1.

Appendix A

Tuning Parameters

Table A.1: Table showing final tuning parameters used on linear heteroscedastic data.

		Random Forests			**Boosting**				**ANNs**		**BART**		
Scenario	Form	B	m	k	Average M	J	ν	η	M	λ	k	M	(ν, d)
L.00.010.1	Orig.	1000	2	75	376	3	0.01	0.5	1	0.010	3	200	(3, 0.99)
	Log	250	2	100	406	3	0.01	0.5	1	0.010	3	200	(3, 0.90)
L.00.010.5	Orig.	1250	2	150	875	2	0.01	0.5	1	0.001	5	50	(3, 0.99)
	Log	1000	2	200	215	3	0.05	0.5	2	0.001	5	200	(10, 0.75)
L.00.100.1	Orig.	1250	33	300	803	4	0.01	0.5	1	0.100	3	50	(3, 0.99)
	Log	1500	25	300	236	2	0.05	0.5	1	0.100	5	200	(3, 0.90)
L.00.100.5	Orig.	1500	50	300	2801	2	0.01	0.5	1	0.100	5	200	(3, 0.99)
	Log	1250	25	300	4028	2	0.01	0.5	1	0.001	5	200	(10, 0.75)
L.50.010.1	Orig.	750	2	150	347	3	0.01	0.5	2	0.100	5	200	(3, 0.90)
	Log	500	3	50	369	3	0.01	0.5	1	0.010	2	200	(3, 0.90)
L.50.010.5	Orig.	1000	3	150	478	4	0.01	0.5	1	0.000	5	50	(3, 0.90)
	Log	1000	3	125	149	3	0.05	0.5	2	0.010	2	200	(10, 0.75)
L.50.100.1	Orig.	1250	33	200	701	3	0.01	0.5	1	0.010	2	200	(3, 0.90)
	Log	1000	25	300	703	3	0.01	0.5	1	0.100	2	50	(3, 0.99)
L.50.100.5	Orig.	1000	50	300	1171	2	0.01	0.5	1	0.100	2	200	(3, 0.90)
	Log	1000	50	300	1692	3	0.01	0.5	1	0.001	3	50	(10, 0.75)
L.80.010.1	Orig.	1250	3	50	122	1	0.05	0.5	1	0.100	3	200	(3, 0.90)
	Log	750	5	25	40	2	0.10	0.5	1	0.001	5	50	(3, 0.90)
L.80.010.5	Orig.	1000	5	50	901	1	0.01	0.5	1	0.001	3	50	(3, 0.90)
	Log	750	5	50	223	1	0.05	0.5	2	0.010	3	50	(10, 0.75)
L.80.100.1	Orig.	750	25	200	883	1	0.01	0.5	1	0.010	3	200	(3, 0.90)
	Log	1000	25	200	497	3	0.01	0.5	1	0.100	5	200	(10, 0.75)
L.80.100.5	Orig.	1500	33	300	849	3	0.01	0.5	1	0.100	5	200	(3, 0.90)
	Log	1000	33	300	890	4	0.01	0.5	1	0.001	3	200	(10, 0.75)

Table A.2: Table showing final tuning parameters used on nonlinear heteroscedastic data.

		Random Forests			**Boosting**				**ANNs**		**BART**		
Scenario	Form	B	m	k	Average M	J	ν	η	M	λ	k	M	(ν, d)
NL.00.010.1	Orig.	500	2	50	93	2	0.05	0.5	1	0.010	3	200	(3, 0.90)
	Log	1000	2	50	347	4	0.01	0.5	1	0.010	5	50	(3, 0.99)
NL.00.010.5	Orig.	750	2	300	560	4	0.01	0.5	1	0.010	3	200	(3, 0.90)
	Log	1250	2	200	939	2	0.01	0.5	1	0.000	3	200	(10, 0.75)
NL.00.100.1	Orig.	1500	25	200	871	3	0.01	0.5	1	0.001	3	200	(3, 0.90)
	Log	1500	50	300	1262	2	0.01	0.5	1	0.100	5	200	(3, 0.99)
NL.00.100.5	Orig.	1500	25	300	2749	4	0.01	0.5	1	0.001	2	50	(3, 0.90)
	Log	1500	33	300	4302	1	0.01	0.5	1	0.001	3	200	(10, 0.75)
NL.50.010.1	Orig.	750	3	40	305	4	0.01	0.5	1	0.010	5	50	(10, 0.75)
	Log	750	3	50	437	2	0.01	0.5	1	0.010	5	200	(10, 0.75)
NL.50.010.5	Orig.	1250	3	100	551	3	0.01	0.5	1	0.010	3	200	(3, 0.99)
	Log	1000	3	150	98	4	0.05	0.5	1	0.000	5	200	(3, 0.90)
NL.50.100.1	Orig.	1500	33	300	679	3	0.01	0.5	1	0.010	2	50	(3, 0.90)
	Log	1000	50	300	723	3	0.01	0.5	1	0.100	3	200	(3, 0.90)
NL.50.100.5	Orig.	1250	33	300	414	2	0.05	0.5	1	0.000	3	200	(3, 0.90)
	Log	1250	50	300	1445	3	0.01	0.5	1	0.001	3	200	(10, 0.75)
NL.80.010.1	Orig.	1000	3	50	58	1	0.10	0.5	1	0.000	3	50	(10, 0.75)
	Log	1250	5	25	78	2	0.05	0.5	1	0.001	3	200	(10, 0.75)
NL.80.010.5	Orig.	750	5	50	432	4	0.01	0.5	1	0.000	5	50	(10, 0.75)
	Log	1500	5	50	582	2	0.01	0.5	1	0.000	5	50	(3, 0.99)
NL.80.100.1	Orig.	750	25	300	77	4	0.05	0.5	1	0.010	3	50	(3, 0.90)
	Log	1000	33	150	121	2	0.05	0.5	1	0.100	3	200	(3, 0.99)
NL.80.100.5	Orig.	1250	33	200	1089	2	0.01	0.5	1	0.000	2	200	(3, 0.99)
	Log	1000	33	200	803	4	0.01	0.5	1	0.000	2	200	(10, 0.75)

Appendix B

Simulation Results

Table B.1: Table of average mean squared error, with standard errors, for linear heteroscedastic data. Highlighted cells indicate which method minimizes the error for each scenario.

Scenario	Form	Linear Reg.		Stepwise		LASSO		Full Tree		Pruned Tree		RF		Boosting		MARS		ANNs		BART	
L.00.010.1	Orig.	4.16	(1.91)	10.83	(2.81)	21.01	(8.60)	157.99	(12.53)	95.99	(10.12)	30.12	(3.04)	18.79	(3.18)	35.17	(9.25)	18.76	(8.40)	15.58	(3.63)
	Log	40.09	(8.47)	46.74	(8.60)	63.15	(27.92)	173.42	(14.55)	100.78	(12.10)	37.21	(7.47)	29.73	(7.54)	39.75	(11.52)	20.86	(9.49)	35.80	(8.81)
L.00.010.5	Orig.	0.17	(0.08)	0.26	(0.18)	0.85	(0.34)	8.32	(0.55)	8.35	(0.56)	2.72	(0.19)	1.30	(0.19)	0.89	(0.43)	0.22	(0.10)	1.29	(0.19)
	Log	6.45	(1.59)	6.61	(1.68)	4.20	(2.68)	8.41	(00.56)	8.12	(0.55)	3.39	(0.26)	1.56	(0.22)	1.45	(0.30)	0.34	(0.14)	1.37	(0.28)
L.00.100.1	Orig.	38.11	(5.40)	46.20	(5.56)	35.25	(8.66)	268.54	(14.43)	288.77	(26.77)	147.34	(7.52)	59.31	(7.25)	105.46	(11.9)	99.00	(6.68)	52.88	(5.93)
	Log	82.98	(11.78)	86.33	(10.43)	78.33	(25.56)	276.90	(15.23)	325.93	(26.77)	154.23	(10.23)	63.63	(8.64)	145.89	(17.83)	56.53	(16.09)	46.65	(9.30)
L.00.100.5	Orig.	1.53	(0.22)	3.01	(0.31)	1.80	(0.34)	20.85	(1.01)	33.20	(9.67)	8.28	(0.42)	4.13	(0.46)	6.50	(0.59)	1.77	(0.23)	2.98	(0.25)
	Log	7.59	(1.58)	8.95	(1.61)	4.89	(1.56)	21.18	(1.04)	33.25	(12.74)	8.29	(0.51)	4.64	(0.75)	7.63	(0.64)	1.96	(0.66)	3.65	(0.87)
L.50.010.1	Orig.	4.17	(1.92)	5.39	(3.24)	19.97	(8.42)	154.96	(12.39)	72.27	(9.52)	58.65	(4.42)	16.65	(2.92)	33.23	(8.72)	10.19	(3.30)	10.57	(2.28)
	Log	39.87	(8.27)	41.11	(8.60)	60.96	(25.43)	171.20	(13.67)	79.06	(11.05)	38.25	(7.38)	28.99	(7.57)	35.51	(11.71)	20.44	(8.31)	45.30	(10.12)
L.50.010.5	Orig.	0.17	(0.08)	0.10	(0.07)	0.80	(0.36)	7.06	(0.55)	6.53	(0.53)	2.83	(0.21)	1.19	(0.16)	1.19	(0.35)	0.26	(0.19)	1.11	(0.18)
	Log	6.40	(1.51)	6.33	(1.50)	3.93	(1.75)	7.19	(0.55)	6.08	(0.51)	2.44	(0.23)	1.26	(0.19)	0.85	(0.29)	0.37	(0.13)	2.13	(0.34)
L.50.100.1	Orig.	38.03	(5.32)	31.00	(4.81)	32.15	(9.38)	249.78	(14.59)	220.00	(17.10)	122.86	(6.43)	40.54	(5.15)	91.65	(11.5)	95.28	(6.41)	48.35	(7.11)
	Log	83.19	(12.40)	70.13	(10.29)	73.43	(24.83)	260.41	(16.11)	237.46	(35.20)	150.49	(10.04)	51.78	(8.65)	128.82	(18.54)	57.28	(16.74)	57.07	(12.46)
L.50.100.5	Orig.	1.53	(0.22)	1.93	(0.24)	1.48	(0.37)	17.28	(0.85)	21.68	(3.81)	7.22	(0.39)	2.96	(0.31)	4.75	(0.50)	1.76	(0.23)	4.13	(0.43)
	Log	7.63	(1.69)	8.08	(1.64)	4.71	(2.11)	17.45	(0.84)	20.96	(3.43)	7.22	(0.46)	3.31	(0.44)	5.76	(0.66)	1.98	(0.71)	5.08	(1.20)
L.80.010.1	Orig.	4.15	(1.92)	1.41	(1.49)	18.96	(8.66)	153.53	(12.39)	50.10	(9.22)	25.14	(3.20)	13.31	(2.94)	30.38	(8.38)	8.92	(4.46)	16.55	(3.39)
	Log	40.06	(8.07)	36.54	(7.80)	59.26	(25.95)	169.37	(14.57)	58.31	(10.61)	31.69	(7.48)	30.63	(8.82)	32.74	(12.80)	19.63	(7.87)	27.88	(7.11)
L.80.010.5	Orig.	0.17	(0.08)	0.06	(0.06)	0.75	(0.35)	6.24	(0.50)	3.86	(0.48)	1.35	(0.17)	0.80	(0.14)	1.19	(0.41)	0.21	(0.09)	1.18	(0.23)
	Log	6.43	(1.55)	6.33	(1.54)	3.67	(1.52)	6.36	(0.54)	3.45	(0.40)	1.41	(0.21)	1.08	(0.34)	0.64	(0.35)	0.37	(0.13)	1.13	(0.22)
L.80.100.1	Orig.	37.98	(5.34)	18.52	(4.16)	26.84	(9.24)	231.83	(14.35)	237.63	(13.41)	147.34	(12.34)	24.38	(3.42)	82.91	(12.23)	97.63	(6.64)	31.15	(4.92)
	Log	82.84	(12.55)	55.32	(9.75)	68.25	(25.11)	243.93	(16.05)	155.09	(19.12)	122.29	(8.99)	43.48	(9.02)	118.99	(20.21)	57.25	(17.54)	41.20	(9.09)
L.80.100.5	Orig.	1.53	(0.22)	1.01	(0.19)	1.16	(0.38)	13.03	(0.71)	14.83	(1.35)	6.23	(0.34)	2.17	(0.24)	3.88	(0.48)	1.77	(0.23)	2.02	(0.22)
	Log	7.60	(1.61)	7.20	(1.58)	4.21	(1.73)	13.09	(0.71)	13.98	(1.32)	6.23	(0.38)	2.34	(0.30)	4.62	(0.68)	1.93	(0.63)	3.81	(0.47)

Table B.2: Table of average median absolute standardized error, with standard errors, for linear heteroscedastic data. Actual values are times 10^{-1}. Highlighted cells indicate which method minimizes the error for each scenario.

Scenario	Form	Linear Reg.		Stepwise		LASSO		Full Tree		Pruned Tree		RF		Boosting		MARS		ANNs		BART	
L.00.010.1	Orig.	0.75	(0.18)	1.22	(0.16)	1.61	(0.34)	4.74	(0.26)	3.54	(0.19)	1.79	(0.10)	1.36	(0.13)	1.57	(0.22)	1.31	(0.28)	1.35	(0.14)
	Log	2.77	(0.33)	2.76	(0.30)	3.40	(0.60)	4.93	(0.25)	4.81	(1.83)	2.09	(0.17)	1.94	(0.28)	2.14	(0.28)	1.63	(0.36)	2.10	(0.25)
L.00.010.5	Orig.	0.75	(0.18)	0.90	(0.31)	1.63	(0.34)	5.29	(0.22)	5.30	(0.24)	2.88	(0.12)	1.90	(0.15)	1.31	(0.35)	0.81	(0.18)	2.00	(0.16)
	Log	3.93	(0.45)	3.91	(0.44)	3.89	(0.88)	5.33	(0.24)	7.21	(4.32)	3.17	(0.13)	2.17	(0.16)	2.10	(0.25)	0.97	(0.20)	2.00	(0.20)
L.00.100.1	Orig.	2.31	(0.18)	2.53	(0.17)	2.14	(0.27)	6.06	(0.25)	6.12	(0.30)	4.45	(0.16)	2.67	(0.15)	3.56	(0.22)	3.45	(0.19)	2.69	(0.17)
	Log	3.33	(0.25)	3.48	(0.24)	3.53	(0.54)	6.19	(0.25)	9.44	(2.65)	4.21	(0.16)	2.79	(0.19)	4.06	(0.23)	2.75	(0.35)	2.67	(0.32)
L.00.100.5	Orig.	2.32	(0.18)	3.24	(0.20)	2.45	(0.24)	8.32	(0.32)	10.38	(1.47)	5.29	(0.20)	3.62	(0.20)	4.67	(0.24)	2.45	(0.19)	3.17	(0.17)
	Log	4.16	(0.39)	4.80	(0.39)	4.02	(0.68)	8.37	(0.32)	14.29	(4.67)	5.11	(0.19)	3.82	(0.30)	4.92	(0.23)	2.42	(0.33)	3.36	(0.33)
L.50.010.1	Orig.	0.75	(0.17)	0.83	(0.28)	1.56	(0.35)	4.78	(0.27)	3.06	(0.21)	2.51	(0.11)	1.24	(0.14)	1.19	(0.31)	1.00	(0.20)	1.08	(0.13)
	Log	2.76	(0.33)	2.76	(0.31)	3.44	(0.58)	4.95	(0.27)	4.11	(0.43)	1.87	(0.24)	1.92	(0.31)	1.92	(0.32)	1.63	(0.35)	2.30	(0.23)
L.50.010.5	Orig.	0.75	(0.18)	0.57	(0.19)	1.56	(0.36)	4.88	(0.23)	4.65	(0.24)	2.89	(0.12)	1.70	(0.14)	0.96	(0.29)	0.87	(0.25)	1.81	(0.16)
	Log	3.91	(0.43)	4.00	(0.43)	3.93	(0.74)	4.93	(0.21)	5.98	(0.94)	2.62	(0.13)	1.90	(0.16)	1.32	(0.23)	1.02	(0.20)	2.41	(0.18)
L.50.100.1	Orig.	2.31	(0.17)	2.08	(0.17)	2.02	(0.30)	5.89	(0.24)	5.41	(0.24)	4.00	(0.14)	2.22	(0.15)	3.08	(0.23)	3.25	(0.20)	2.52	(0.19)
	Log	3.35	(0.26)	3.16	(0.26)	3.52	(0.55)	6.03	(0.25)	7.57	(1.95)	4.16	(0.16)	2.41	(0.22)	3.60	(0.28)	2.77	(0.36)	2.85	(0.33)
L.50.100.5	Orig.	2.31	(0.18)	2.59	(0.18)	2.20	(0.28)	7.61	(0.28)	8.55	(0.73)	4.93	(0.18)	3.08	(0.18)	3.77	(0.23)	2.45	(0.19)	3.67	(0.20)
	Log	4.18	(0.40)	4.38	(0.39)	4.01	(0.82)	7.68	(0.29)	11.33	(4.38)	4.77	(0.18)	3.24	(0.21)	4.13	(0.25)	2.43	(0.35)	4.05	(0.34)
L.80.010.1	Orig.	0.75	(0.18)	0.42	(0.22)	1.50	(0.37)	4.85	(0.27)	2.49	(0.24)	1.57	(0.12)	1.07	(0.15)	0.73	(0.32)	0.97	(0.22)	1.40	(0.15)
	Log	2.76	(0.32)	2.92	(0.34)	3.59	(0.56)	5.00	(0.29)	3.48	(0.73)	1.83	(0.28)	1.92	(0.31)	1.82	(0.37)	1.64	(0.35)	1.91	(0.28)
L.80.010.5	Orig.	0.75	(0.18)	0.42	(0.22)	1.50	(0.37)	4.66	(0.25)	3.45	(0.23)	1.90	(0.14)	1.27	(0.16)	0.71	(0.32)	0.80	(0.18)	1.79	(0.17)
	Log	3.91	(0.43)	4.09	(0.44)	4.09	(0.71)	4.70	(0.25)	4.45	(0.94)	1.90	(0.15)	1.53	(0.16)	1.10	(0.31)	1.01	(0.20)	1.82	(0.19)
L.80.100.1	Orig.	2.31	(0.17)	1.60	(0.18)	1.83	(0.33)	5.73	(0.25)	4.43	(0.23)	3.79	(0.14)	1.66	(0.15)	2.56	(0.26)	3.32	(0.20)	2.03	(0.17)
	Log	3.34	(0.27)	2.90	(0.27)	3.48	(0.56)	5.90	(0.26)	5.87	(0.65)	3.55	(0.14)	2.22	(0.26)	3.17	(0.33)	2.76	(0.37)	2.36	(0.31)
L.80.100.5	Orig.	2.31	(0.17)	1.87	(0.18)	1.92	(0.32)	6.64	(0.26)	7.10	(0.38)	4.53	(0.17)	2.54	(0.17)	2.98	(0.24)	2.44	(0.18)	2.54	(0.16)
	Log	4.17	(0.39)	4.03	(0.38)	3.88	(0.69)	6.65	(0.25)	8.99	(1.48)	4.42	(0.17)	2.67	(0.17)	3.38	(0.28)	2.40	(0.31)	3.48	(0.22)

Table B.3: Table of average mean squared error, with standard errors, for nonlinear heteroscedastic data. Actual values are times 10^{-2}. Highlighted cells indicate which method minimizes the error for each scenario.

Scenario	Form	Linear Reg		Stepwise		LASSO		Full Tree		Pruned Tree		RF		boosting		MARS		ANNs		BART	
NL.00.010.1	Orig.	1.79	(0.27)	2.24	(0.30)	3.94	(1.15)	10.21	(1.11)	7.28	(1.06)	2.18	(0.40)	1.66	(0.39)	3.65	(1.19)	0.61	(0.42)	1.83	(0.80)
	Log	0.93	(0.32)	1.39	(0.37)	2.95	(0.85)	9.62	(0.91)	6.55	(0.75)	2.58	(0.52)	2.10	(0.49)	2.34	(0.79)	1.06	(0.36)	1.79	(0.43)
NL.00.010.5	Orig.	1.41	(0.20)	1.47	(0.22)	1.83	(0.33)	2.67	(0.27)	2.69	(0.27)	1.11	(0.09)	0.55	(0.10)	0.63	(0.17)	0.12	(0.05)	0.61	(0.15)
	Log	0.08	(0.03)	0.09	(0.05)	0.39	(0.14)	2.62	(0.24)	2.62	(0.24)	1.09	(0.10)	0.43	(0.09)	0.26	(0.09)	0.09	(0.04)	0.41	(0.07)
NL.00.100.1	Orig.	16.08	(2.60)	17.45	(2.48)	23.73	(8.66)	51.82	(5.87)	64.53	(6.96)	29.64	(3.77)	19.18	(4.06)	45.97	(12.29)	25.79	(3.40)	19.00	(10.72)
	Log	8.80	(1.33)	11.45	(1.66)	14.81	(3.06)	46.51	(4.31)	61.10	(8.49)	27.41	(2.22)	11.34	(1.63)	28.77	(6.60)	10.11	(1.87)	11.68	(2.01)
NL.00.100.5	Orig.	7.85	(1.42)	9.29	(1.58)	9.73	(2.42)	19.20	(2.24)	31.50	(12.63)	7.47	(0.75)	6.80	(1.26)	9.29	(1.23)	3.47	(1.04)	5.05	(0.78)
	Log	1.35	(0.22)	2.64	(0.33)	2.29	(0.54)	18.46	(2.19)	28.77	(8.61)	8.76	(1.02)	2.84	(0.34)	6.60	(0.99)	1.45	(0.27)	3.50	(0.40)
NL.50.010.1	Orig.	2.11	(0.33)	2.25	(0.38)	4.47	(1.34)	10.97	(1.20)	6.44	(1.06)	2.24	(0.48)	1.70	(0.36)	3.96	(1.31)	0.70	(0.46)	1.57	(0.66)
	Log	1.05	(0.35)	1.12	(0.41)	3.31	(0.95)	10.27	(1.06)	5.74	(0.82)	2.69	(0.46)	2.20	(0.55)	2.43	(0.96)	1.21	(0.42)	1.98	(0.52)
NL.50.010.5	Orig.	1.67	(0.26)	1.65	(0.26)	2.18	(0.43)	2.52	(0.29)	2.42	(0.30)	0.86	(0.10)	0.53	(0.12)	0.62	(0.17)	0.14	(0.06)	0.67	(0.17)
	Log	0.09	(0.04)	0.07	(0.04)	0.45	(0.17)	2.48	(0.28)	2.30	(0.27)	1.05	(0.11)	0.60	(0.17)	0.34	(0.18)	0.11	(0.05)	0.42	(0.10)
NL.50.100.1	Orig.	14.17	(2.07)	12.73	(1.96)	20.16	(7.37)	44.82	(4.94)	48.03	(7.13)	28.45	(3.23)	12.76	(2.52)	39.63	(9.20)	22.25	(2.92)	20.13	(7.67)
	Log	8.47	(1.27)	8.51	(1.49)	14.15	(3.19)	40.30	(3.74)	40.63	(5.29)	24.74	(1.96)	10.28	(1.72)	25.49	(5.97)	9.58	(1.79)	9.85	(1.59)
NL.50.100.5	Orig.	6.49	(1.16)	7.14	(1.28)	8.18	(2.07)	13.87	(1.52)	15.79	(2.91)	5.68	(0.59)	3.65	(0.55)	6.53	(0.89)	2.83	(0.73)	4.41	(0.83)
	Log	1.19	(0.19)	1.53	(0.23)	1.91	(0.53)	13.59	(1.45)	12.24	(0.97)	6.25	(0.74)	2.53	(0.38)	4.87	(0.76)	1.27	(0.25)	2.68	(0.36)
NL.80.010.1	Orig.	2.04	(0.33)	1.85	(0.31)	4.31	(1.28)	10.70	(1.14)	4.69	(1.08)	2.78	(0.52)	1.48	(0.42)	3.83	(1.28)	0.72	(0.50)	2.10	(0.94)
	Log	1.04	(0.35)	0.89	(0.35)	3.26	(0.94)	10.04	(0.98)	4.22	(0.77)	2.35	(0.51)	2.30	(0.61)	2.25	(0.94)	1.15	(0.40)	2.16	(0.48)
NL.80.010.5	Orig.	1.61	(0.25)	1.58	(0.25)	2.08	(0.40)	2.15	(0.25)	1.51	(0.25)	0.61	(0.10)	0.50	(0.15)	0.64	(0.17)	0.12	(0.06)	0.47	(0.12)
	Log	0.09	(0.04)	0.05	(0.04)	0.44	(0.17)	2.12	(0.26)	1.40	(0.23)	0.63	(0.13)	0.52	(0.19)	0.27	(0.18)	0.11	(0.05)	0.47	(0.13)
NL.80.100.1	Orig.	10.77	(1.62)	7.89	(1.29)	14.32	(5.05)	33.46	(3.93)	27.49	(4.23)	22.01	(2.60)	7.72	(1.39)	29.07	(7.64)	18.14	(2.35)	10.26	(5.89)
	Log	6.72	(0.99)	5.02	(1.07)	10.09	(2.34)	30.81	(2.90)	22.54	(2.64)	15.22	(1.45)	7.84	(1.61)	18.93	(5.15)	7.44	(1.28)	7.52	(1.28)
NL.80.100.5	Orig.	4.84	(0.81)	4.90	(0.89)	5.93	(1.41)	8.46	(0.87)	9.41	(1.26)	3.68	(0.39)	1.80	(0.33)	4.53	(0.76)	2.13	(0.62)	3.46	(0.69)
	Log	0.96	(0.15)	0.71	(0.16)	1.26	(0.42)	8.25	(0.82)	9.32	(1.20)	3.78	(0.44)	1.71	(0.33)	3.30	(0.64)	1.00	(0.17)	2.25	(0.32)

Table B.4: Table of average median absolute standardized error, with standard errors, for nonlinear heteroscedastic data. Actual values are times 10^{-1}. Highlighted cells indicate which method minimizes the error for each scenario.

Scenario	Form	Linear Reg		Stepwise		LASSO		Full Tree		Pruned Tree		RF		Boosting		MARS		ANNs		BART	
NL.00.010.1	Orig.	2.23	(0.22)	2.59	(0.20)	3.24	(0.54)	5.85	(0.25)	4.80	(0.28)	2.25	(0.12)	2.22	(0.20)	2.70	(0.30)	1.31	(0.34)	2.23	(0.30)
	Log	1.57	(0.29)	1.64	(0.23)	1.75	(0.25)	4.21	(0.20)	4.58	(0.21)	1.82	(0.19)	1.71	(0.22)	1.79	(0.24)	1.57	(0.29)	1.65	(0.23)
NL.00.010.5	Orig.	4.66	(0.31)	4.80	(0.33)	5.01	(0.37)	6.96	(0.25)	7.01	(0.26)	4.42	(0.14)	2.92	(0.18)	3.17	(0.35)	1.40	(0.28)	3.10	(0.28)
	Log	0.87	(0.21)	0.93	(0.25)	1.38	(0.25)	4.79	(0.19)	6.92	(0.25)	3.03	(0.12)	1.73	(0.13)	1.24	(0.24)	0.89	(0.21)	1.79	(0.14)
NL.00.100.1	Orig.	5.11	(0.44)	5.27	(0.37)	5.51	(1.08)	8.58	(0.31)	10.11	(0.51)	6.06	(0.21)	4.64	(0.33)	6.96	(0.38)	6.30	(0.33)	4.82	(0.62)
	Log	2.31	(0.16)	2.47	(0.16)	2.21	(0.19)	5.02	(0.18)	9.15	(0.52)	3.78	(0.13)	2.37	(0.15)	3.38	(0.17)	2.35	(0.16)	2.17	(0.18)
NL.00.100.5	Orig.	8.17	(0.56)	9.06	(0.58)	8.18	(0.74)	12.81	(0.46)	17.01	(3.72)	8.03	(0.26)	7.09	(0.56)	9.08	(0.44)	5.22	(0.52)	6.63	(0.36)
	Log	2.06	(0.16)	2.82	(0.16)	2.08	(0.17)	7.29	(0.28)	15.56	(2.26)	4.51	(0.16)	2.87	(0.15)	4.11	(0.20)	2.08	(0.16)	3.17	(0.16)
NL.50.010.1	Orig.	2.34	(0.22)	2.44	(0.24)	3.32	(0.56)	5.91	(0.27)	4.33	(0.32)	2.16	(0.14)	2.06	(0.17)	2.46	(0.34)	1.36	(0.34)	1.96	(0.24)
	Log	1.61	(0.29)	1.63	(0.29)	1.75	(0.25)	4.23	(0.21)	4.09	(0.23)	1.87	(0.20)	1.67	(0.25)	1.73	(0.28)	1.60	(0.29)	1.64	(0.25)
NL.50.010.5	Orig.	4.83	(0.33)	4.80	(0.33)	5.18	(0.39)	6.43	(0.27)	6.30	(0.32)	3.51	(0.15)	2.64	(0.18)	2.40	(0.31)	1.45	(0.29)	3.10	(0.28)
	Log	0.89	(0.21)	0.83	(0.25)	1.32	(0.25)	4.41	(0.20)	6.07	(0.27)	2.72	(0.12)	1.73	(0.14)	1.12	(0.28)	0.91	(0.21)	1.53	(0.14)
NL.50.100.1	Orig.	4.78	(0.36)	4.38	(0.33)	5.03	(0.96)	7.98	(0.30)	8.62	(0.71)	6.03	(0.20)	3.91	(0.28)	6.11	(0.38)	5.79	(0.30)	4.72	(0.37)
	Log	2.32	(0.16)	2.23	(0.17)	2.18	(0.21)	4.84	(0.18)	7.54	(0.36)	3.75	(0.14)	2.27	(0.17)	3.15	(0.20)	2.36	(0.16)	2.27	(0.17)
NL.50.100.5	Orig.	7.70	(0.51)	8.03	(0.52)	7.66	(0.67)	11.35	(0.38)	12.46	(1.29)	7.24	(0.23)	5.78	(0.31)	7.51	(0.36)	4.90	(0.46)	6.38	(0.40)
	Log	2.06	(0.15)	2.30	(0.15)	1.87	(0.19)	6.71	(0.25)	12.63	(2.92)	4.26	(0.16)	2.82	(0.17)	3.62	(0.20)	2.08	(0.15)	2.95	(0.16)
NL.80.010.1	Orig.	2.30	(0.22)	2.15	(0.23)	3.25	(0.54)	5.90	(0.24)	3.62	(0.38)	2.40	(0.15)	1.92	(0.23)	2.04	(0.31)	1.34	(0.36)	2.15	(0.28)
	Log	1.61	(0.29)	1.67	(0.31)	1.74	(0.25)	4.31	(0.22)	3.43	(0.25)	1.73	(0.23)	1.65	(0.26)	1.70	(0.32)	1.60	(0.29)	1.75	(0.21)
NL.80.010.5	Orig.	4.73	(0.33)	4.68	(0.33)	5.03	(0.37)	6.06	(0.26)	4.84	(0.30)	2.82	(0.16)	2.32	(0.18)	1.99	(0.35)	1.31	(0.29)	2.53	(0.24)
	Log	0.89	(0.21)	0.81	(0.29)	1.27	(0.26)	4.27	(0.22)	4.49	(0.25)	1.83	(0.13)	1.30	(0.16)	0.98	(0.32)	0.91	(0.21)	1.55	(0.15)
NL.80.100.1	Orig.	4.49	(0.36)	3.61	(0.30)	4.54	(0.86)	7.52	(0.29)	6.80	(0.53)	5.77	(0.19)	3.45	(0.22)	5.30	(0.39)	5.68	(0.32)	3.87	(0.50)
	Log	2.31	(0.16)	2.01	(0.20)	2.08	(0.22)	4.80	(0.19)	6.05	(0.53)	3.16	(0.13)	2.09	(0.19)	2.93	(0.22)	2.35	(0.16)	2.20	(0.17)
NL.80.100.5	Orig.	7.11	(0.47)	6.91	(0.47)	6.93	(0.60)	9.50	(0.33)	10.32	(0.71)	6.16	(0.20)	4.29	(0.25)	6.13	(0.36)	4.59	(0.44)	5.91	(0.35)
	Log	2.06	(0.16)	1.74	(0.17)	1.60	(0.23)	5.85	(0.22)	9.94	(0.51)	3.76	(0.14)	2.40	(0.17)	3.08	(0.23)	2.08	(0.16)	3.01	(0.18)