Evaluating the Impact of Heteroscedasticity on the Predictive Ability of Modern Regression Techniques

by

Nathaniel Payne

B.B.A, Simon Fraser University, 2008

A Thesis Submitted In Partial Fulfillment of the
Requirements for the Degree of
Master of Science

in the
Department of Statistics and Actuarial Science
Faculty of Science

SIMON FRASER UNIVERSITY

Summer 2014

Approval

Name: **Nathaniel Payne**

Degree: **Master of Science**

Title: ***Evaluating the Impact of Heteroscedasticity on the Predictive Ability of Modern Regression Techniques***

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

Thomas M. Loughin
Senior Supervisor
Professor _____

Richard Lockhart
Supervisor
Professor _____

Derek Bingham
Supervisor
Professor _____

Robert Krider
Internal Examiner
Professor _____

Date Defended/Approved: August 22, 2014

Partial Copyright Licence

Abstract

Over the last decade, the number and sophistication of methods used to do regression on complex datasets have increased substantially. Despite this, our literature review found that research that explores the impact of heteroscedasticity on many widely used modern regression methods appears to be sparse. Thus, our research seeks to clarify the impact that heteroscedasticity has on the predictive effectiveness of modern regression methods.

In order to achieve this objective, we begin by analyzing the ability of ten different modern regression methods to predict outcomes for three medium-sized data sets that each feature heteroscedasticity. We then use insights provided from this work to develop a simulation model and design an experiment that explores the impact that various factors have on prediction accuracy of our ten different regression methods. These factors include linearity, sparsity, the signal to noise ratio, the number of explanatory variables, and the use of a variance stabilizing transformation.

Keywords: Regression; heteroscedasticity; random forest; multivariate adaptive regression splines; LASSO; BART; regression tree; data mining; machine learning;

Acknowledgements

There are so many people to whom I owe a debt of gratitude towards. Firstly, I would like to sincerely thank my senior supervisor, Dr. Tom Loughin, whose incredible patience, persistence, inspiration, hard work, and support, helped me complete not only this thesis, but also the entire statistics program. Tom helped mould me into the statistician that I am today, as well as the one that I will become, and is someone who I look up to immensely! His patience and leadership will never be forgotten.

Secondly, I would like to thank the incredible review committee, as well as the larger Faculty group in the Department of Statistics who have mentored me and who inspire me every day with their work and many unique worldly contributions. As reviewers, Dr. Richard Lockhart, Dr. Derek Binghman, and Dr. Robert Krider are leaders in their respective fields, and I aspire one day to mirror their leadership and contributions in my own way within both academia and industry! In addition, I cannot stop without mentioning directly so many of the incredible faculty who helped me survive and thrive in the incredible program at Simon Fraser University. Dr. Carl Schwarz, Dr. Jiguo Cao, Dr. Rachel Altman, Dr. Jinko Graham, Dr. Brad McNeney, Dr. Rick Routledge, Dr. Tim Swartz, Dr. David Campbell, and Dr. Steve Thompson have all made a significant contributions to my development, and I am grateful for their support.

Thirdly, but perhaps the most important of all, I would like to thank my family, specifically my beautiful wife Samareh, my son Cameron, and my brother-in-law Ali. These people give me a reason every day to push myself to the limit, as well as the time and support to accomplish the impossible. Without them, none of this journey would be worthwhile. I also want to deeply thank my parents, Barry & Cathy Payne, who have provided so much mental support, encouragement, and inspiration every day, and who have helped give me the tools that made persevering through the program possible! I love you all so much.

Finally, I want to close by acknowledging the many amazing peers who have helped make my journey possible and inspired me daily with their accomplishments and effort, including Andrew Henrey, Jack Davis, Abdollah Safari, Harsha Perera, Kasra Yousefi, and Biljana Stoijkova.

Table of Contents

List of Tables

List of Figures

1. Introduction

1.1. Understanding the Challenges of Applied Data Mining

Over the last decade, the size and complexity of data have continued to increase. At the same time, the number and sophistication of methods used to operate on these complex data sources have also expanded. Today, for example, modern regression methods can deal effectively with unknown, potentially nonlinear relationships between the response and explanatory variables. Some of these methods also deal with unknown or unexpected interactions among explanatory variables. Separately, progress has also been made developing new methods for basic linear regression models that deal with problems such as heteroscedasticity, non-normality and outliers.

While research progress has been immense in both of those aforementioned areas, our literature review found that research on the impact of heteroscedasticity on modern regression methods is not well developed. This is surprising, particularly considering the prevalence of heteroscedasticity in historical data sets. In general, we suspect that many practitioners use modern regression methods, including tree-based methods and other ensemble learning approaches, without considering the assumptions that underlie these models. For example, tree-based ensembles such as random forests are reportedly excellent omnibus predictors that can adapt too many regression shapes and interaction structures (Hastie, Tibshirani, & Friedman, 2009). Moreover, from conversations and informal reports, we also know that many analysts use random forests as a first choice for regression prediction problems. However, the standard tree models upon which random forests are built use algorithms based on an ordinary least squares (OLS) criterion for partitioning and estimation. This is problematic, particularly because OLS is known to have problems dealing with heteroscedastic data (Carroll & Ruppert, 1988).

1.2. Objective

This thesis seeks to clarify the impact that heteroscedasticity has on the predictive effectiveness of modern regression methods. In order to achieve this objective, this thesis utilizes both simulation and applied analysis. In particular, we begin by analyzing the ability of ten different regression methods to predict outcomes for three medium-sized data sets that feature heteroscedasticity. During this analysis, we attempt to understand the nature of the heteroscedasticity present, consider possible models for it, and, where appropriate, apply transformations to reduce its apparent magnitude. Following this, we use insights provided from this work to develop simulation experiments that explore the impact that various factors have on the prediction accuracy of our ten different regression methods. To do this, we create data from models that are both linear and nonlinear, and that have both homoscedastic and heteroscedastic errors. The other factors included in the simulation are the number of explanatory variables, the fraction of explanatory variables with nonzero coefficients, and the signal to noise ratio. We also consider the effect of using of a variance-stabilizing transformation.

As we move through the project, we have a number of initial hypotheses that support the different research questions asked. First, from our literature review, it is clear that linear regression methods that use ordinary least squares perform best under conditions of linearity (i.e., a linear relationship between the response and explanatory variables) and homoscedasticity. Since linearity is so critical, we generally expect that, when a transformation can both linearize and stabilize the variance of a sample, a linear method that relies on ordinary least squares should do well on the transformed data set. On the other hand, we hypothesize that when a variance-stabilizing transformation ruins linearity, we expect that these same methods will be adversely affected by the transformation.

Secondly, we hypothesize that, because methods such as regression trees have no reason to require linearity, they should not be adversely affected by nonlinearity, but *may* be affected by heteroscedasticity. We are unsure about the degree to which heteroscedasticity will affect such methods. Nonetheless, since these methods rely on unweighted sums of squared errors, we hypothesize that these methods will perform better after a variance-stabilizing transformation, regardless of the linearity of the situation.

1.3. Outline

This thesis is organized as follows. In Chapter 2, we describe heteroscedasticity, its causes, and the manner in which it manifests itself within data. We also review a number of traditional solutions that have been proposed to deal with heteroscedasticity, including transformations and weighting. In Chapter 3, we briefly summarize each of the different regression techniques that were chosen for our research. In particular, we focus on modern regression methods that are commonly used within industry, including LASSO, multivariate adaptive regression splines (MARS), regression trees, random forests, neural networks, and Bayesian Additive Regression Trees (BART). In Chapter 4, we apply these techniques to three real data sets in order to develop a practical understanding of the impact that heteroscedasticity can have on each method's ability to produce low out-of-sample prediction error. In Chapter 5, we use the insights generated from these analyses to create a simulation model and identify a total of 32 different combinations of factors that may affect prediction outcomes. For each of these combinations, we compute the methods' mean squared errors (MSEs) and median absolute deviations (MADs) and analyze the results graphically. Finally, in Chapter 6, we discuss our findings, review limitations, and state conclusions.

2. Heteroscedasticity

The assumption of equal variance, or heteroscedasticity, is fundamental in classical regression models (Carroll & Ruppert, 1988). Unfortunately, while this assumption is theoretically convenient, the findings of our work from Chapter 4 suggest that it is often not satisfied within practice (Wilcox & Keselman, 2012). In the sections below, we present a quick review of heteroscedasticity and the impact that heteroscedasticity has on the linear model. Following this, we discuss potential methods that are used to resolve heteroscedasticity when it is encountered within regression analysis.

2.1. Introducing Heteroscedasticity

Non-constant variability, or heteroscedasticity, is a phenomenon that is frequently encountered in nearly all fields, including genetics (Daye, Chen & Li, 2012; Brem & Krugylak, 2005), toxicology (Lim, Sen & Peddada, 2010), fisheries research (Ruppert & Carroll, 1988), experimental design (Box & Meyer, 1985; Box, 1987), and econometrics. Many different types of heteroscedasticity exist, including variability that depends on the mean of the data, variability that depends on one or more explanatory variables, as well as other structures (Carroll & Ruppert, 1988; Grissom, 2000; Moore & McCabe, 2005). For our research, we have chosen to focus only on variability that depends on the mean of the data.

2.2. Heteroscedasticity & Linear Regression

Heteroscedasticity and its impact on linear regression have been extensively studied. Based on this research, it has been shown that using linear regression methods to perform prediction with heteroscedastic data can fail dramatically (Carroll & Ruppert, 1988). This is because regions of low variability end up having significantly less influence setting parameters and making predictions than regions containing high variability. This divergence can result in predictions that significantly misrepresent the true mean of the data, especially in regions of low variability.

In addition to this, other research has shown that linear regression in the presence of heteroscedasticity can prevent the type 1 errors and coverage probabilities of confidence intervals (CIs) for model-based predictions from attaining the nominal level (Carroll and Ruppert 1988; Lim, Sen, & Peddada, 2010; Visek, 2011). This failure can cause practitioners to declare an outcome statistically significant when in fact it is not.

2.3. Stabilizing Heteroscedasticity

In general, there are two main classes of approaches used to deal with data which is heteroscedastic: transformations and weighting. Since we use transformations of the response variable in Chapters 4 and 5, we introduce transformations below.

In linear regression, transformations of the response variable are widely used to deal with heteroscedasticity. Depending on the distribution of the data, if the variance can potentially be represented as a known function of the mean, a practitioner can resolve the heteroscedasticity by transforming the response variable. The transformation used is called a variance-stabilizing transformation (Carroll & Ruppert, 1988). For example, if it appears that the variance is proportional to the square of the mean, then the log transformation can be applied to the response variable Y. On the other hand, if the variance appears to be directly proportional to the mean, which is true for a Poisson distribution, then the square-root transformation can be applied to stabilize the variance and make the distribution of the transformed response homoscedastic (Carroll & Ruppert, 1988).

While a correct transformation of the response variable can effectively produce homoscedasticity (Fleiss, 1986; Luh, 1992; Rasmussen, 1989), the use of transformations in practice is not without risk. For example, if the underlying data are linear and heteroscedastic, the application of an inappropriate transformation can destroy the linearity and reduce the accuracy of predictions generated by regression models that assume linearity, such as those used in simple linear regression, stepwise regression, ridge regression, and the LASSO.

Another challenge that often arises relates to the selection of the appropriate transformation. As noted in Carroll & Ruppert (1988), one of the most common ways to identify heteroscedasticity that depends on the mean of the response is to plot the squared residuals against the predicted values from a regression. A version of this process was followed for the examples that we explored in Chapter 4. Unfortunately, while this process is simple to implement, identifying an appropriate transformation is often very difficult by inspection. This is because the data can contain many features outside of heteroscedasticity that confound the visual inspection (Wilcox, 1998). This conclusion was supported by our own experience in Chapter 4, where the presence of outliers and nonlinearity made it difficult for us to identify the appropriate variance-stabilizing transformation to be applied.

2.4. Research Gaps

Despite the volume of research that exists exploring the impact of heteroscedasticity on classical linear regression, our review found that literature discussing the impact of heteroscedasticity on many modern regression methods is sparse (Daye, Chen & Li, 2012; Woolridge, 2009; Visek, 2011). This is surprising, particularly considering the prevalence of heteroscedasticity in the modern data sets which we reviewed. Thus, while significant room exists for follow up research, we hope that the insights gained from this thesis can begin to fill the gap in current literature discussing the impact of heteroscedasticity on modern regression methods. We also hope that this work will provide guidance to industry analysts and practitioners who routinely use modern regression methods to make predictions.

3. Modern Regression Techniques

3.1. Introduction & Modelling Focus

In this thesis, ten different regression methods are explored. For each of these methods, our focus is on prediction rather than parameter estimation. To provide context, this Chapter briefly reviews each of the ten methods that we have explored, and states hypotheses relating to the performance of each method on both linear and non-linear heteroscedastic data.

3.2. Regression Methods

3.2.1. Linear Regression

The initial foundation for linear regression was laid in 1894 by Sir Francis Galton (Galton, 1894; Stanton, 2001). In general, the linear regression model can be represented using the following form (Hastie, Tibshirani & Friedman, 2009):

$$f(X) = \beta_0 + \sum_{j=1}^{p} X_j \beta_j$$

In this model, β_0 is the regression model intercept, β_j is the j^{th} regression coefficient, X_j is the j^{th} explanatory variable, and p refers to the number of explanatory variables. As discussed in Chapter 2, classical linear regression models are very sensitive to heteroscedasticity (Carroll & Ruppert, 1988; Wilcox & Keselman, 2012). Based on this, we expect that classical linear regression will do a poor job predicting untransformed, non-linear, heteroscedastic data. On the other hand, we expect that this method will do a better job than other methods predicting linear, homoscedastic data that results after a variance stabilizing transformation is applied. Finally, when the data in question is linear and heteroscedastic, we expect that classical linear regression will struggle on both the transformed and untransformed cases. This is because the presence of heteroscedasticity in the linear-heteroscedastic case, or the presence of the non-linearity in the transformed case, will decrease the accuracy of predictions generated by the classical linear regression model.

3.2.2. Stepwise Regression

Stepwise regression, as opposed to classical linear regression, considers variable importance when determining the final predictors that remain within a model. For our work, we use the version of stepwise regression that is described by Venables and Ripley (2002) and that is implemented in the stepAIC() function in R (Venables & Ripley, 2002). For all our methods, we use the backwards option. This option starts with a "full" model which includes all explanatory regression variables, considers the removal of each variable from this model, and chooses to remove the variable that results in the smallest increase to the error sum of squares. This process is repeated on the resulting model and iterated until a stopping criterion is reached.

While stepwise regression attempts to decrease estimation variance by removing unnecessary variables from the regression model, the underlying parameter estimation algorithm used in stepwise regression is ordinary least squares. Because of this, we expect that stepwise regression will perform similarly to the classical linear regression model for both non-linear and linear heteroscedastic data. However, since stepwise regression helps a model take into account variable importance, we expect that it will improve over the full linear regression when the regression model contains many variables with zero coefficients.

3.2.3. Ridge Regression

Ridge regression was developed with the goal of improving parameter estimates in the presence of significant correlation between explanatory variables. In general, ridge regression is very similar to linear regression, except that it penalizes the size of the regression coefficients resulting in a modified version of the least squares criterion (Hoerl & Kennard, 1970). The formula for ridge regression parameter estimates (or estimated regression coefficients) can be expressed as:

$$\hat{\beta}^{ridge} = \ argmin_{\beta} \left\{ \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \gamma \sum_{j=1}^{p} \beta_j^2 \right\}$$

In this model, x_{ij} is the i^{th} measurement of explanatory variable X and y_i is the i^{th} observation of the response variable. Also, γ represents a tuning parameter which controls the strength of the penalty term $\sum_{j=1}^{p} \beta_j^2$ (Hastie, Tibshirani & Friedman, 2009).

The penalty that is applied to the formula is a shrinkage penalty, the main goal of which is to reduce the variance of OLS parameter estimates arising from highly correlated explanatory variables. This shrinkage reduces variance but introduces bias into the estimates. When gamma is equal to zero, an OLS estimate is returned. On the other hand, when gamma goes to infinity, the ridge regression estimates go to zero. Between these two extremes, the estimates balance between fitting a linear model of y on X more closely and producing parameter estimates with lower variance (Hastie, Tibshirani & Friedman, 2009).

While ridge regression is able to deal more effectively with highly correlated explanatory variables, this method, fundamentally, still relies on the use of sums of squared errors. Because of this, we expect that ridge regression will perform similarly to classical linear regression for both linear and non-linear heteroscedastic data. Indeed, we expect that this model will outperform classical linear regression and stepwise regression in cases where there is a high degree of multi-collinearity existing within the sample data.

3.2.4. Least Absolute Shrinkage & Selection Operator (LASSO)

The LASSO (Least absolute shrinkage and selection operator), is a popular modern regression method with many similarities to ridge regression. This method is designed to select variables and subsequently estimate their coefficients with less bias than occurs following other variable selection methods (Tibshirani, 1996). The LASSO performs well when the number of explanatory variables is large. As a result, the LASSO is of keen interest in the data mining and big data communities, where the number of explanatory variables within a data set can be exceptionally high. The LASSO uses an absolute value on the penalty term rather than a squared penalty term (Hastie, Tibshirani & Friedman, 2009). The formula for the LASSO parameter estimation can be expressed as follows:

$$\hat{\beta}^{lasso} = \ argmin_{\beta} \left\{ \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij} \beta_j \right)^2 + \gamma \sum_{j=1}^{p} |\beta_j| \right\}$$

In this model, γ represents a tuning parameter which controls the strength of the penalty term $\sum_{j=1}^{p}|\beta_j|$. It is important to recognize here that the use of the absolute value within the penalty term makes the LASSO perform very differently from ridge regression, despite the similarity of the objective functions (Tibshirani, 1996). In particular, the LASSO penalty causes some coefficients to be shrunk exactly to zero. This occurs with greater frequency as γ increases. (Tibshirani, 2013)

The LASSO parameter estimation formula is solved using an objective function that minimizes the sums of squared errors using a penalty term. Because of this, the model is arguably potentially sensitive to heteroscedasticity. Thus, we expect that the LASSO generated regression models will perform similar to classical linear regression on both linear and non-linear heteroscedastic data except when the number of explanatory variables in the regression model is large. In this specific case, we expect that the LASSO will perform better and produce predictions with lower overall prediction error.

3.2.5. Regression Trees

Regression trees come from a class of modern regression models that continue to be very popular within industry (Rokach & Maimon, 2008). A regression tree recursively partitions data into smaller and smaller groups and uses the sample mean within each group as the predicted value for various combinations and levels of the explanatory variables (Breiman, Friedman, Olshen & Stone, 1984; Quinlan, 1986; Strobl, Malley & Tutz, 2009).

In order to determine how to split the data, squared error loss is used. In particular, at each step of the tree growing process, the regression tree algorithm seeks to optimally split the data in a way that the residual sum of squares (RSS) after the split are minimized. Thus, if the RSS of the full data can be represented as:

$$RSS(Full\ Data) = \sum_{i=1}^{n}(y_i - \bar{y})^2$$

the minimized RSS value of the split can be calculated as:

$$RSS(Split) = \sum_{i \in R_1}^{n}(y_i - \bar{y}_1)^2 + \sum_{i \in R_2}^{n}(y_i - \bar{y}_2)^2$$

In this split model, R_1 and R_2 are the two subsets of data formed by any one split, y_i is the i^{th}observation of the response variable, $\bar{y}_1$ is the mean of the data in R_1, and $\bar{y}_2$ is the mean of the data in R_2. During the model fitting process, the regression tree is typically grown out so that it over-fits the data. Following this, the tree is pruned using cross-validation (Friedman, 1999; Barros, Basgalupp, Carvalho & Freitas, 2011).

In general, regression trees have the capacity to deal effectively any form of relationship between response and explanatory variables (Breiman, Friedman, Olshen & Stone, 1984). This means that we can expect the regression tree model to perform similarly regardless of whether the data it is modelling are linear or non-linear. Unfortunately, trees generally suffer from high variance because of the fact that a small change in the observed data can lead to a dramatically different sequence of splits and predictions. This instability occurs because of the way the tree is constructed hierarchically. In particular, once a split is made, the split cannot be reversed further down the tree. Because of this, we expect the regression tree predictions to produce mean squared prediction errors and median absolute deviations that have variances than the predictions generated from other models, regardless of the data that they are modelling. We also expect that the method itself will be sensitive to the presence of heteroscedasticity because regression trees minimize the sum of squared errors when determining the optimal way to split the tree.

3.2.6. Boosted Decision Trees (Boosting)

Unlike standard regression trees, boosting seeks to improve model fit by fitting a sequence of single trees which explicitly consider the amount of variation not explained by earlier trees in the sequence (Hastie, Tibshirani & Friedman, 2009). In particular, the boosting algorithm starts by using the average of all response values as the first guess for prediction and computes the residuals from the fit. An optimal regression tree is fit to these residuals, the predictions are updated, and new residuals are computed. This process repeats itself again and again until some stopping rule is reached. A typical stopping rule is the number of trees in the sequence (Breiman, 1996; Breiman, 2001; Bulhmann, 2010; Elith, Leathwick, Hastie, 2008, Freund & Schapire, 1997; Friedman, 1999; Friedman, 2001; Friedman, 2002). Cross-validation is then often applied in order to ensure that the optimal number of final trees is used. This prevents the model from over-fitting.

In general, the development of a boosted regression tree has 2 main steps: 1) creating residuals 2) building trees. While there are many important parameters that one can tweak to control this process, including the increment rate, two highly relevant tuning parameters which can be tweaked include the number of splits to be made or nodes allowed on the tree, as well as the number of trees to be fit (Elith, Leathwick, Hastie, 2008). One important item to recognize is that the boosted regression tree still relies on squared error loss. This is because the selection of splits is based on minimizing the residual sum of squares (RSS), just like in the simple regression tree case (Loughin, 2012). Since this is true, while we expect that the regression model developed using a boosted regression tree will perform better than the regression tree models for both linear and non-linear heteroscedastic data due to their use of a sequential, weighted, model building process, we do expect that boosted trees will still be sensitive to the presence of heteroscedasticity.

3.2.7. Multivariate Adaptive Regression Splines (MARS)

In general, regression trees are incredibly flexible but suffer from some serious draw backs. In particular, regression trees cannot produce smoothed regression functions. In an attempt to improve upon this, Multivariate Adaptive Regression Trees (MARS) were developed. MARS attempts to improve regression tree models by using hinge functions which allow it to produce smoothed regression prediction surfaces.

During model fitting process, the MARS algorithm begins with a forward pass through the data. Within the forward pass, MARS starts with a model that simply represents the mean of the response values. MARS then repeatedly adds basis functions to the model in a stepwise manner. During this process, MARS splits the basis function such that the final split produces the largest drop in the overall sum of squared error. Then, during the backwards pass, MARS uses cross-validation in order to choose the best final model that has the lowest overall cross-validated residual sum of squares (Friedman, 1991).

While the MARS algorithm proceeds differently than the classical regression model, the final MARS regression model still relies on the minimization of the sum of squared errors (Loughin, 2012). Thus, we expect that the MARS models will perform better on non-linear data than other models which require linearity, but will struggle overall when left to model

heteroscedastic data. We also note that the MARS model does attempt to do variable selection. Thus, we expect to see the MARS model improve the accuracy of its predictions as the number of explanatory variables with zero coefficients increases.

3.2.8. Random Forests

The creation of random forests was motivated by the desire to substantially improve upon the predictive accuracy of basic tree models (Breiman, 1996). In particular, for each tree, a random forest model chooses m variables from all available explanatory variables in the regression tree randomly and then builds a regression tree using the randomly selected explanatory variables. (Loughin, 2012). To complete this process, within each regression tree, the random forest model also randomly splits variables, with the initial splitting process starting with a resample. After building a single regression tree, the random forest algorithm then repeats the process n times, building many more regression trees using the same criteria for error minimization as the regression tree model. The final random forest then averages over the predicted values generated from the many regression trees that have been produced (Hastie, Tibshirani, & Friedman, 2009).

Random forests represent a vast improvement over individual regression tree models and are “considered by many to be a primary tool for prediction (Loughin, 2012).” These models have also shown tremendous promise dealing with nonlinear data (Prinzie & Van Den Poel, 2008). Moreover, as Buhlmann (2010) points out, “there are virtually no competing methods which can so easily deal with high-dimensional continuous data yielding powerful predictions as well as information about variable importance.” Nonetheless, we do recognize that the random forest model relies on minimizing the sum of squared errors when deciding how to split the generated regression trees at each node and implicitly for estimating the mean within each terminal node of each tree. . Thus, we expect that the random forest model will perform similarly to the other classical regression techniques and struggle when modelling heteroscedastic data. However, we expect that its predictions will be better than those of other methods under conditions of nonlinearity.

3.2.9. Bayesian Additive Regression Trees (BART)

Bayesian Additive Regression Trees (BART) represent a new tree model that was proposed by Chipman, George, and McCulloch (2010). This approach remains under active development, particularly the computational components. BART is an ensemble method that creates a linear combination of trees using a Bayesian approach. In particular, the BART algorithm specifies a prior for each tree, defines a likelihood using the sample data, and then uses Markov Chain Monte Carlo (MCMC) to make multiple draws from the posterior of the distribution and form a single regression tree. This process repeats itself many times depending on the number of trees that are specified in the model building step, with each new tree explaining slightly more of the total error in the model. This results in a final model which includes many trees, each which explain a small amount of the total overall error (Abu-Nimeh et al., 2008, Chipman, George, and McCulloch, 2010).

BART has several interesting features which have led us to believe that it might perform well within our research. In particular, we note that BART conducts automatic variable selection while searching for models with highest posterior probabilities. Because of this, we hypothesize that BART may do well in situations where the number of explanatory variables is large and there are many zero coefficients (Chipman, George & McCulloch; 1998, Chipman, George, Lemp & McCulloch, 2010; Wu, Tjelmeland, & West, 2007). Furthermore, since BART is a tree-based model, we expect that it may do well in situations where non-linearity is present. However, we are unsure of the impact that heteroscedasticity will have on the BART method and look forward to evaluating this within the context of our study.

3.2.10. Neural Network

Neural networks are considered to be one of “the premiere tools for automated prediction (Loughin, 2012).” In a neural network, simple nodes, called "neurons", are connected together to form a network which mimics a biological neural network. (Ripley, 1996). While neural networks have traditionally been used for classification tasks, they do have a clear application to regression problems (Duda, Hart & Stork, 2001; Secomandi, 2000; Damas et al, 2000). In particular, in regression, the neural network algorithm starts by randomly selecting weights and applying those weights to all explanatory variables. The weighted

explanatory variables then produce many different hidden nodes within a specified number of hidden layers. These nodes then are weighted using secondary weights to produce the final predictions. Once this process is completed, the neural network algorithm makes repeated passes between the original explanatory variables and the predicted values over the m pre-defined hidden layers. After many iterations, the weights within the model converge using methods like gradient descent to settle on final optimized weights. The overall optimization goal in this context is to have the weights converge to values that minimize the overall sum of squared errors for the predicted values (Deng & Ferris, 2008; Rosenblatt, 1958).

While neural networks are very popular, it is important to recognize that they have a number of drawbacks which temper their usefulness. For example, as Loughin points out, neural networks can be unstable, particularly when the data set is small (Loughin, 2012). This is because the use of a small dataset can deprive the model from having enough data to effectively tune the model weights. Furthermore, neural networks also require a fair amount of tuning in order to generate accurate results (Loughin, 2012). Because of these issues, we are unsure how neural networks will perform overall when used within the thesis. Nonetheless, since the process relies on minimization of the residual sum of squares, we expect that neural networks may perform similarly to other modern regression methods when faced with the task of predicting heteroscedastic data.

4. Exploring the Impact of Heteroscedasticity on Three Applied Regression Problems

4.1. Analysis Goals

Prior to the completion of a simulation, we analyzed three heteroscedastic datasets. This initial exploration had a number of goals. The first goal was to understand the ways in which heteroscedasticity appears and is discovered in real data, while also observing how successfully transformations could be used to stabilize the variance. This would provide observational insights into the general trends observed both within and between regression methods. Insights gleaned from this work would also help provide guidance into the factors that should be varied within our proposed final simulation, which is designed to measure the impact of heteroscedasticity on the ten different modern regression methods described in the previous chapter.

In addition to this, the other goal of the analysis was to understand the potential impact that an appropriate transformation would have on various performance measures. Conveniently, in all three cases, after visually inspecting the plots of the squared errors vs the predicted values generated from a random forest model that included all explanatory variables, the log transformation appeared to best resolve the heteroscedasticity.

4.2. Data Description

All three of the datasets used in Chapter 4 were found on the UCI Machine Learning Repository (Bache & Lichman, 2013). They were selected based on their containing apparent heteroscedasticity, as well as other features such as sample size and number of explanatory variables. The data sets are described below.

4.2.1. Abalone Data

The abalone dataset is a well-known machine learning dataset that challenges the user to predict the ages of abalone using physical measurements taken from a large sample of abalone shells (Nash, Sellers, Talbot, Cawthorn & Ford, 1994; Waugh, 1995). This dataset

was relatively clean and contained no missing values. Prior to analysis, the categorical variable SEX was removed. This was done because the LASSO cannot not explicitly deal with categorical predictors. Binary variables can be analyzed simply as an indicator variable, but SEX includes three levels of gender (male, female, and immature). It is unclear how this variable should be recoded as indicators. As such, we omitted gender from the analysis.

Below is a full description of the variables that were used within the initial applied study:

Table 4.1 Description of Abalone data

Variable Name	Type (levels)	Description
SEX	Categorical (3)	M, F, and I (infant)
LENGTH	Continuous	Length in mm = the longest shell measurement
DIAMETER	Continuous	Diameter in mm perpendicular to length
HEIGHT	Continuous	Height in mm with meat in shell
WHOLE_WEIGHT	Continuous	Weight of whole abalone in Grams
SHUCKED_WEIGHT	Continuous	Weight of meat in Grams
VISCERA_WEIGHT	Continuous	Gut weight in Grams (after bleeding)
SHELL_WEIGHT	Continuous	Grams after being dried
RINGS	Integer	+/- 1.5 gives the age in years

4.2.2. Airfoil Data

The airfoil data set is a unique data set that was processed originally by NASA in 1989. The dataset contains 1503 entries, 1 response variable (SCALED_SOUND), and 5 explanatory variables (Brooks, Pope & Marcolini, 1989). All variables are shown in the chart below.

Table 4.2 Description of Airfoil Data

Variable Name	Type (levels)	Description
FREQUENCY	Numeric	Frequency (In Hertz)
ANGLE_OF_ATTACK	Numeric	Angle of Attack (in Degrees)
CHORD_LENGTH	Numeric	In Meters
FREE_STREAM_VELOCITY	Numeric	In Meters / Second
SUCTION_SIDE	Numeric	Suction Side Displacement Thickness (in Meters)
SCALED_SOUND	Numeric	Scaled Sound Pressure Level, $SPL_{1/3}$ (in Decibels)

The data for this experiment was gathered from various tests done on two and three-dimensional airfoil blades within a large wind tunnel. The full description of these experiments can be found in the original work & online (Brooks, Pope & Marcolini, 1989; Lopez, 2014).

4.2.3. Bike Rentals Data

The bike rentals data set gives the daily count of bikes that were rented in 2011 and 2012 in the Capitalbikeshare System. This data also includes information such as weather and other seasonal factors. The original source of the data was the District Department of Transportation (DDOT) in Arlington County, Virginia. (Fanaee-T & Gama, 2013).

Many potentially important explanatory variables in the data set were categorical. Since removal of those variables would materially impact the regression outcomes, we decided instead to convert SEASON, YEAR, MNTH, HOLIDAY, WORKINGDAY, and WEATHERSIT into indicator variables. The baseline for each of these was chosen as the level which would most explain the variation of the response variable. For example, within SEASON, summer was chosen as the baseline since it was hypothesized that rentals of bikes would be significantly greater in summer than in other seasons.

In addition to the use of indicators, a number of other changes were made. Prior to initial analysis, the variables INSTANT, as well as DTEDAY, were removed, as these were case-identification variables and not considered relevant for the analysis. Additionally, working day was also removed, since this variable was perfectly collinear with WEEKDAY. Furthermore, since the REGISTERED and CASUAL explanatory variables summed to

give the total bike rental count, these two variables were removed from the experimental dataset and converted into a proportion showing the relative percentage of registered casual users who rented. Finally, within the WEATHERSIT category, category 4 was not observed in the dataset, and was thus removed. This made the WEATHERSIT variable a 3-category variable, with level 1, or CLEAR, being chosen as the baseline. For reference, below is the full description of the bike rentals dataset (Fanaee-T & Gama, 2013):

Table 4.3 Description of Bike Rentals Data

Variable Name	Type (levels)	Description
INSTANT	Numeric	Record Instant
DTEDAY	Numeric	Date
SEASON	Categorical (4)	Season (1:springer, 2:summer, 3:fall, 4:winter)
YR	Categorical (2)	Year (0: 2011, 1:2012)
MNTH	Categorical (12)	mnth : month (1 to 12)
HOLIDAY	Categorical (2)	holiday : weather day is holiday or not (extracted from [Web Link])
WEEKDAY	Categorical (7)	weekday : day of the week
WORKINGDAY	Categorical (2)	workingday : if day is neither weekend nor holiday is 1, otherwise is 0
WEATHERSIT	Categorical (4)	- 1: Clear, Few clouds, Partly cloudy, Partly cloudy - 2: Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist - 3: Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered clouds - 4: Heavy Rain + Ice Pallets + Thunderstorm + Mist, Snow + Fog
TEMP	Continuous	temp : Normalized temperature in Celsius. The values are divided to 41 (max)
ATEMP	Continuous	atemp: Normalized feeling temperature in Celsius. The values are divided to 50 (max)
HUM	Continuous	hum: Normalized humidity. The values are divided to 100 (max)
WINDSPEED	Continuous	windspeed: Normalized wind speed. The values are divided by 67 (max)
CASUAL	Continuous	casual: count of casual users
REGISTERED	Continuous	registered: count of registered users
CNT	Continuous	cnt: count of total rental bikes including both casual and registered

4.3. Exploring Heteroscedasticity

Prior to model fitting, the squared residuals were plotted against the predicted values generated from a random forest model. The random forest model was chosen because of its ability to cope with non-linearity and thus more effectively isolate heteroscedasticity from other potential model flaws. In general, all three plots showed an increasing trend, indicating that the error variance was increasing with the mean in each case.

Figure 4.1 shows the output of our initial data exploration, in particular the squared errors versus the predicted values generated from a random forest model using all explanatory variables. As can be observed in Figure 4.1, each of the three sample data sets seemed to show different forms of heteroscedasticity.

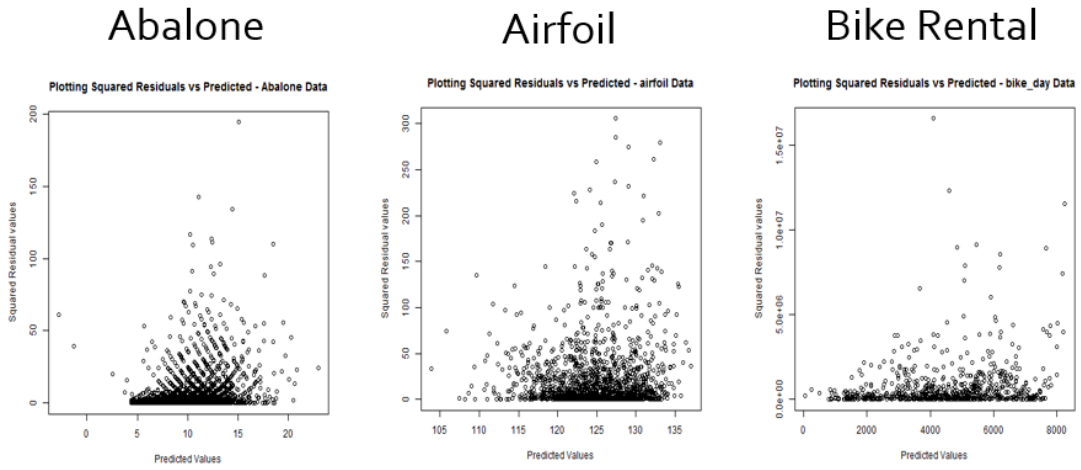

Figure 4.1 Plots Showing the Squared Errors versus the Predicted Values Generated From a Random Forest Model Pre Transformation

After plotting the data, five possible variance stabilizing transformations were considered: log, inverse, square root, quarter root, and squared. Note that the squared transformation was not fully considered because, when the variance increases with the mean, the use of the square transformation increases the heteroscedasticity that is present. After visually inspecting the squared error vs predicted plots for all possible variance stabilizing transformations, we produced an initial diagnosis on the appropriate variance stabilizing transformation to be applied to each data set. We also determined that the log transformation most effectively resolved the heteroscedasticity present in each of the three different data sets. The post-transformation residual plots, generated after a log

transformation was applied to the response variable Y, are shown in Figure 4.2. The red loess curve was used as a visual check to help determine which variance stabilizing transformation most effectively resolved the heteroscedasticity present.

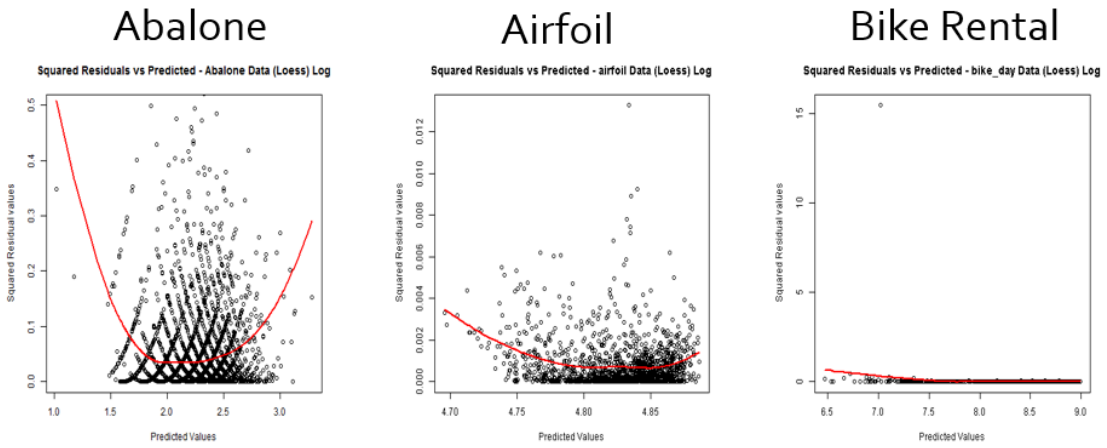

Figure 4.2 Plots Showing the Squared Errors versus the Predicted Values Generated From a Random Forest Model Post Log Transformation

Based on this work, the log transformation was chosen to be used as the focal transformation within the final research simulation.

4.4. Analysis Results

After exploring heteroscedasticity, each data set was analyzed using the ten different regression methods discussed in Chapter 3. During this analysis, we had two main goals: (1) compare the predictive abilities of the ten selected regression methods, and (2) assess whether the use of a variance stabilizing transformations might improve the predictive ability of various methods. To complete the analysis, we measured predictive ability by splitting the data set into a training set and a test set, each containing 50% of the total sample. Each method was applied to the training set in order to generate an estimate of the model parameters. The test data were then predicted using the estimated model. Importantly, for every data set, we re-randomized the split of the training and test set 100 times in order to reduce the effects of the random split. We also compared the predictions with and without a variance stabilizing method. Thus, for every regression method, 200 performance measures were obtained for the method on each data set, 100 generated

using the non-transformed data and 100 generated using the transform variance stabilized data.

During these experiments, we used two main performance measures: mean squared prediction error and median absolute deviation (MAD). The MSPE was chosen based on its pervasiveness in industry as a measurement standard. On the other hand, the MAD was chosen based on its resistance to extreme prediction errors which have a more significant influence on the MSPE than MAD.

4.4.1. Mean Squared Prediction Error

The Mean Squared Prediction Error (MSPE) measures the squared distance between the predicted value generated from a particular regression method and an actual value. The formula for the MSPE is listed below:

$$MSPE = \left[\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}\right]$$

In this model, y_i is the actual value of the i^{th} observation and $\hat{y}_i$ is the predicted value generated by the modern regression method under consideration.

4.4.2. Median Absolute Deviation

As we noted previously, the mean is potentially sensitive to skewness and may not be robust to outliers. Because of this, another more effective performance measure may need to be used. A measure that we selected is the median absolute deviation, or MAD. In general, the MAD can be expressed using the following notation:

$$MAD = median[|y_i - \hat{y}_i|]$$

In this model, y_i is the actual value of the i^{th} observation and $\hat{y}_i$ is the predicted value generated by the modern regression method under consideration.

4.4.3. Abalone Dataset

MSPE Analysis

Figure 4.3 shows a boxplot of the MSPE that was generated after each of the 10 different methods were applied to the transformed and untransformed abalone data. After running 100 randomized data splits for each method, both BART and the random forest model were most effective at generating small prediction errors, with others being close behind. The mean MSPE for untransformed data was similar for the four methods based on a linear regression model. Interestingly, the mean MSPEs computed using the transformed data were noticeably greater for these same models, and the distribution of MSPE values had a much longer right tail than with original data. This observation provided initial support for the argument that, when the MSPE is used as the performance measure, the log transformation does not efficiently transform away the heteroscedasticity in a manner that allows these methods to perform better. This finding will be explored further within the simulation study.

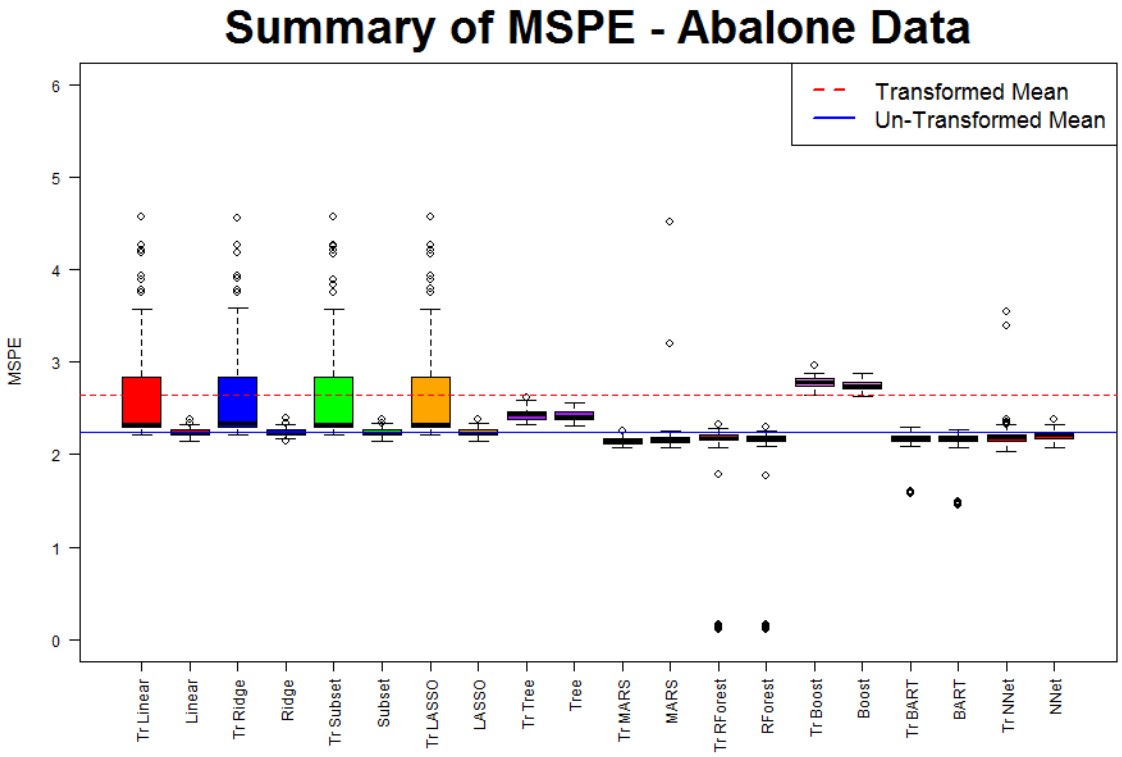

Figure 4.3 Summary of MSPE – Abalone Data

One interesting point of concern relates to the few extremely small MSPE values that were obtained using the random forest output. These values were not explained as of this writing, and will be included in future research. Our initial hypotheses was that these values were caused by a computing error. Unfortunately, this hypotheses was checked exhaustively and no immediately recognizable computational errors were found.

Overall, as is shown in Figure 4.3, the use of a variance stabilizing transformation seemed to have little impact on the predictive accuracy of the modern regression methods that we reviewed, including BART, NNET, Random Forest, Boosting, and MARS. This provides some initial support for the argument that these more modern methods may perhaps be able to make accurate predictions in the presence of heteroscedasticity. This also provides support for the argument that these methods may perform well regardless of the shape of the $X - Y$ relationship.

MAD Analysis

Figure 4.4 shows the results from the MAD analysis of the abalone data. When using the MAD, no method seemed to outperform other methods for this particular dataset. Moreover, in every case, we note that the transformed datasets had a slightly lower MAD than the non-transformed dataset. This observation was the opposite of what was observed in the case of the MSPE, and may indicate that there were a few poorly-predicted values in the transformed data that resulted in the generation of very large squared errors.

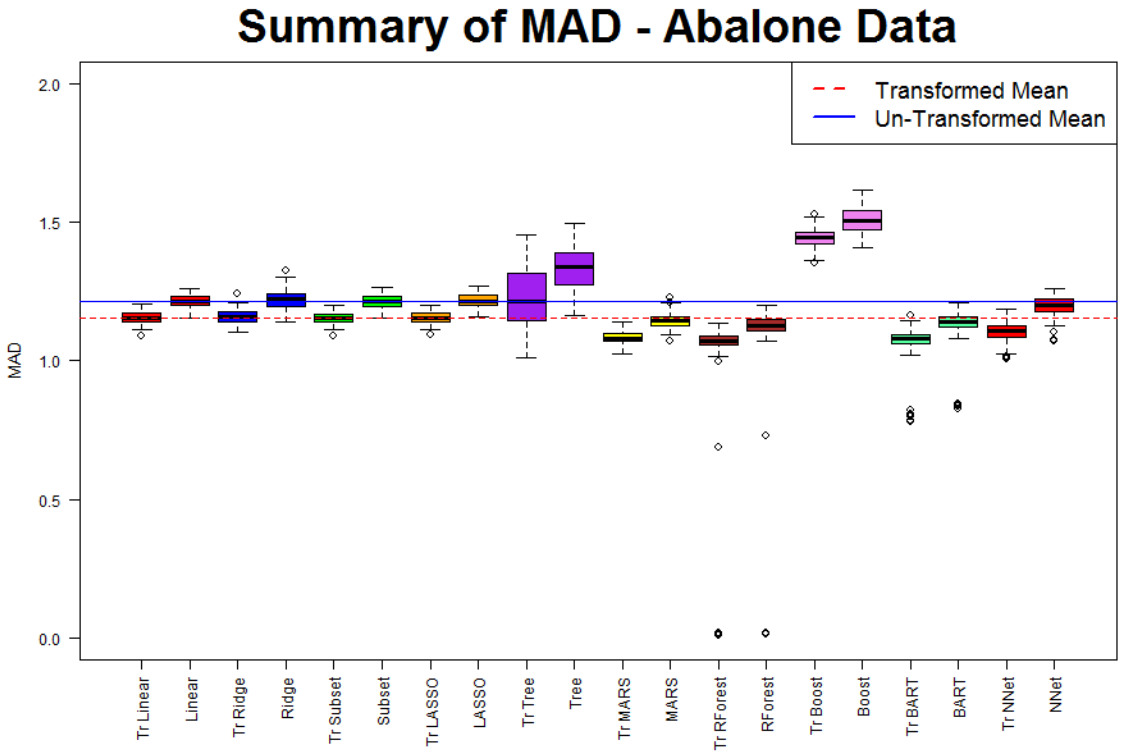

Figure 4.4 Summary of MAD – Abalone Data

Based on this analysis, a few conclusions can be drawn. Firstly, it is interesting to see that, for some methods, both the MAD and MSPE give directionally opposite conclusions regarding the impact that a variance stabilizing transformation has on the performance measure. This may support our hypotheses from the MSPE section that the MSPE is simply not robust to heteroscedasticity. More broadly, this conclusion also suggests that care must be taken around the selection of a performance measure prior to analysis, particularly if one does not have the luxury of testing or comparing different statistical methods during analysis.

4.4.4. Airfoil Dataset

MSPE Analysis

Analysis of the MSPE output from the airfoil dataset yielded interesting results. As is shown in Figure 4.5, there were identified differences in the predictive effectiveness between the ten different methods evaluated. In particular, for both the transformed (variance stabilized) and untransformed data, BART outperformed the other methods of analysis. Furthermore, random forests also outperformed the remaining methods, despite returning a MSPE that was on average double the MSPE generated through the BART models. In fact, a close inspection of the MSPE statistics shows that in every case other than Boosting, methods that relied specifically on linearity seemed to significantly underperform the other methods which did not. Moreover, methods like the LASSO, ridge regression, stepwise regression, and classical linear regression, performed nearly identically for all cases, and showed no noticeable difference between the transformed and untransformed cases.

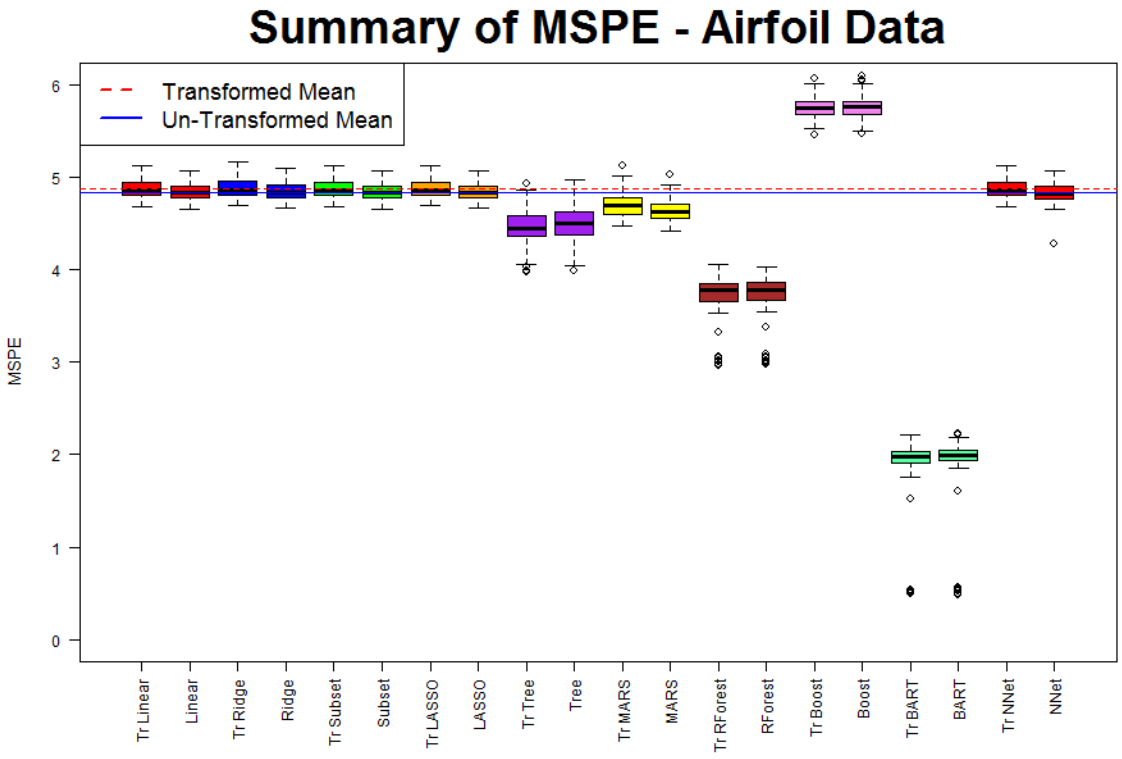

Figure 4.5 Summary of MSPE – Airfoil

The other major observation that came out of the initial analysis dataset relates to the issue of transformations and proper transformation selection. As is noted in Figure 4.5, for this dataset, both the transformed and non-transformed datasets showed no visually obvious differences in their performance. This suggests that potentially, for this dataset, the transformation itself that we determined to be optimal has in fact no ability to resolve the apparent heteroscedasticity.

MAD Analysis

Results from the analysis of the median absolute deviation study are shown in Figure 4.6. These results mirror the conclusions obtained from the MSPE analysis. For this case, the BART models produced lower MAD values than all other methods that that were considered. This performance is consistent between the variance stabilized data and non-transformed dataset.

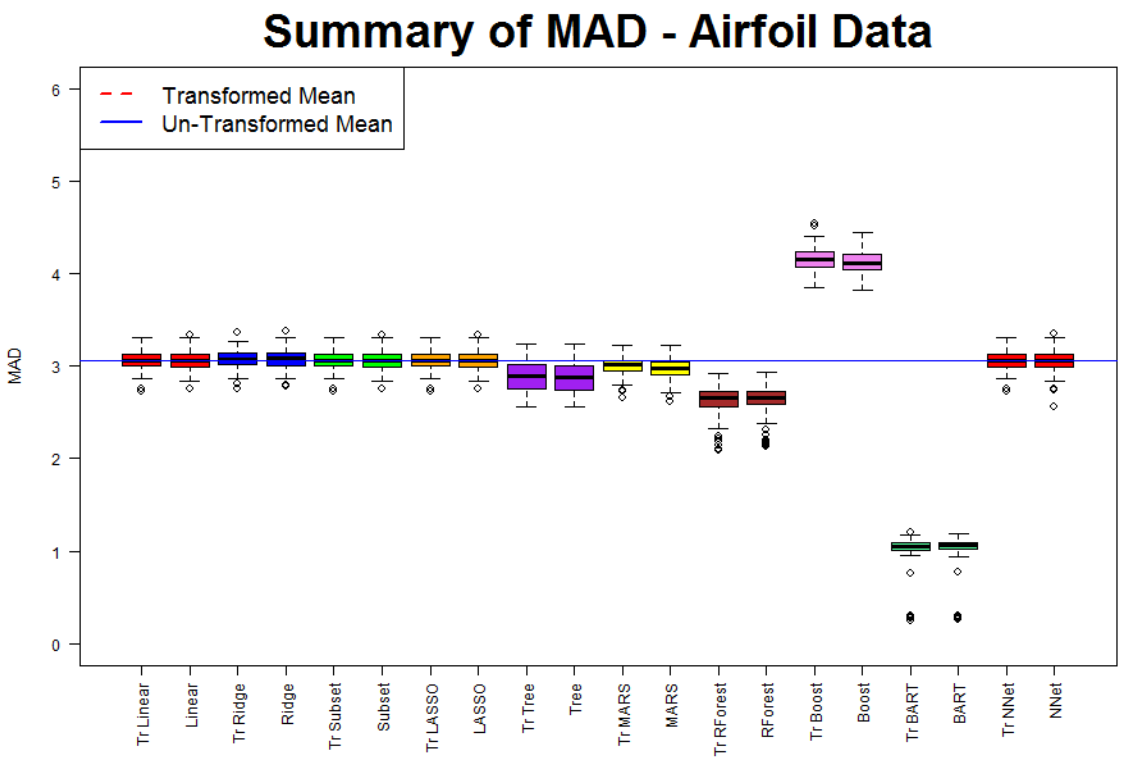

Figure 4.6 Summary of MAD – Airfoil

4.4.5. Daily Bike Rental Dataset

MSPE Analysis

The final dataset that we considered was the bike rental dataset. In general, the MSPE analysis of the bike rental dataset produced findings that were consistent with the MSPE analysis of the Abalone data. As is shown in the output, large differences were observed between the output derived from using variance stabilized data and the non-transformed dataset. Interestingly, the transformation actually increased the total mean squared predictive error when compared with the non-transformed dataset for linear regression, ridge, subset, and the LASSO. This suggests that, particularly when a linear model is used, the use of a variance stabilizing transformation actually made the model fit worse.

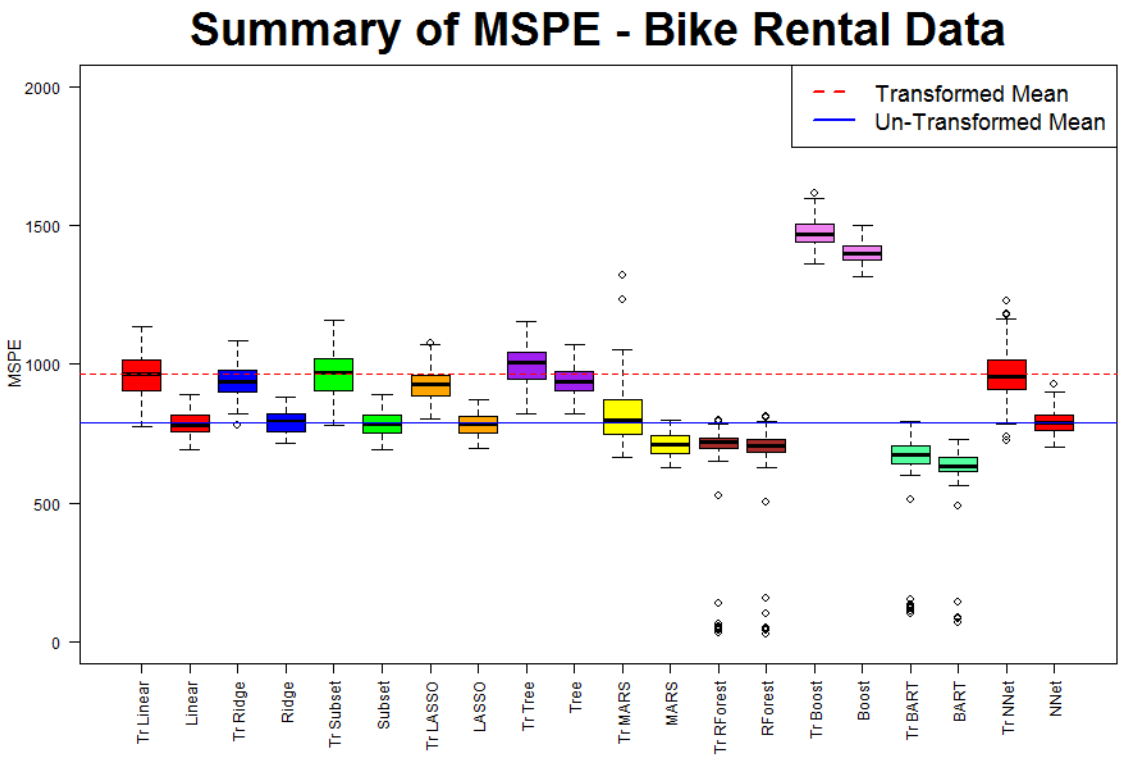

Figure 4.7 Summary of MSPE – Daily Bike Rentals Data

Finally, while BART and the random forest models seemed to perform somewhat better than most other methods, the differences were not as stark as in the other data sets.

MAD Analysis

Figure 4.8 shows the summary of the analysis generated by using the MAD. As is evident from visual inspection, the MAD analysis followed a similar pattern as the MSPE analysis for this particular dataset.

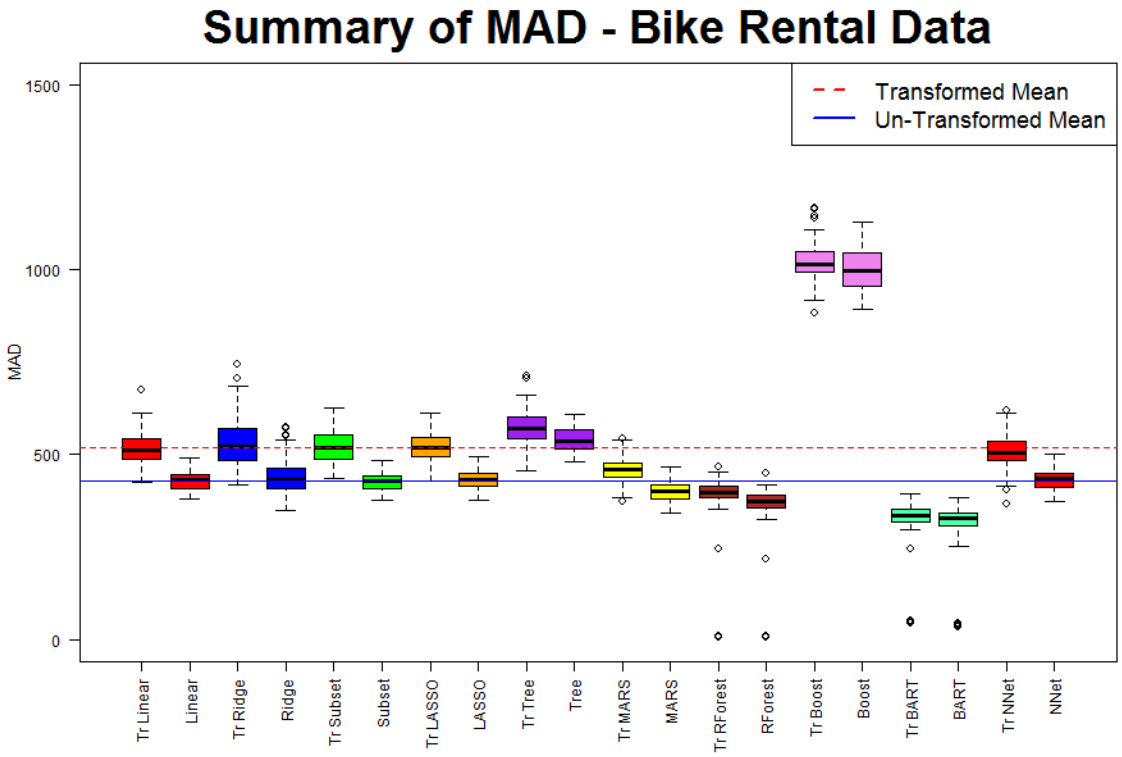

Figure 4.8 Summary of MAD – Daily Bike Rentals Data

4.4.6. Overall Conclusions

The significant time spent working on these three individual datasets was extremely beneficial both in terms of an exercise in regression, and as a precursor to the simulation. We observed that modern regression methods that rely on linearity within the model seemed to perform more poorly than the tree based methods and other modern regression methods. In fact, in all cases, BART seemed to outperform other methods, with the random forest model coming in second. This result held on both the MAD and MSPE analysis across the three data sets, suggesting that fundamentally, these two methods may be able to deal most effectively with data that is heteroscedastic and / or non-linear.

5. Simulation Study

5.1. Simulation Objective

In Chapter 4, our analysis of three different data sets revealed some interesting patterns. Unfortunately, none of the work in that section allowed us to generalize or control for any of the various factors that might have been confounding the success of the ten different regression methods that were evaluated. Thus, in order to address this gap, a simulation has been designed that examines the performance of each of the ten different modern regression techniques.

5.2. Cases Selected & Simulation Models

During the simulation, two distinct cases were tested.

5.2.1. Case 1: Heteroscedastic, Non-Linear Simulation Data

Case 1 represents data that was heteroscedastic, non-linear, and which was generated using the log normal distribution.

$$Y = exp(X\beta + \varepsilon), \qquad \varepsilon \sim N(0, \sigma^2)$$

The expected value is the value against which all predicted values are compared using the mean squared error and the median absolute deviation. The model for the expected value of case one is listed below:

$$E(Y) = exp\left(X\beta + \left(\frac{\sigma^2}{2}\right)\right)$$

5.2.2. Case 2: Heteroscedastic, Linear Simulation Data

Case 2 represents data that was originally generated as heteroscedastic and linear. This is a particularly interesting case for our simulation, because the use of a log transformation as a variance stabilizing transformation results in the destruction of linearity. Below is the model used to represent the heteroscedastic, nonlinear simulation data:

$$Y = X\beta + \varepsilon, \qquad \varepsilon \sim N(0, \sigma^2(X\beta)^2)$$

The variance $\sigma^2(X\beta)^2$ was selected because, when the variance is proportional to the square of the mean, the log transformation can be used to stabilize the variance. The expected value of this simulation model can be stated as:

$$E(Y) = X\beta$$

5.3. Simulation Factors

During the simulation, for each of the two cases, we varied 3 distinct factors. These factors were determined after conducting an extensive literature review of published work and reviewing the output from the three data sets that we analyzed in Chapter 4. For our simulations, we varied the number of explanatory variables, the sparsity of the explanatory variables (i.e., the fraction with non-zero coefficients), and the signal-to-noise ratio within the simulated data. Scenarios which were sparse had the last 80% of their regression coefficients β_j set to 0. On the other hand, scenarios that were not sparse had all of their regression coefficients set to the β value indicated in Tables 5.1 and 5.2.

For each simulation, the number of explanatory variables was set at either 10 or 100. The level 10 was selected because it was similar to the number of explanatory variables found in the three data sets that we explored in Chapter 4. The level 100 was selected based on our desire to approximate a medium- to large-dimension data set. For all simulation cases, the correlation between any pair of explanatory variables X_j and X_k was $\rho^{|j-k|}$ with $\rho = 0.8$. The signal-to-noise ratio (SNR) within the simulation had two levels, 1 or 5. In order to generate data with an accurate signal to noise ratio, the intercept of the simulated data, β_0, the common value for all non-zero regression coefficients, β, and the value of σ were varied for each of the 16 experimental scenarios. Tables 5.1 and 5.2 provide the simulation settings that were used for case 1 and case 2. For each simulation setting, 50 data sets were generated, each containing 1000 observations. Only 50 simulations were run due to the fact that that computational time required to run some of the more advanced methods was significant.

Table 5.1 Table Illustrating Simulation Settings for Case 1 Simulations

Scenario	Linearity	P	SNR	Sparsity	σ	β	β_0
1	Non-Linear	10	1	Not	0.3	0.04	0.8
2	Non-Linear	10	5	Not	0.13	0.04	0.8
3	Non-Linear	10	1	Sparse	0.32	0.18	0.8
4	Non-Linear	10	5	Sparse	0.15	0.18	0.8
5	Non-Linear	100	1	Not	0.3	0.01	0.8
6	Non-Linear	100	5	Not	0.13	0.01	0.8
7	Non-Linear	100	1	Sparse	0.245	0.02	0.8
8	Non-Linear	100	5	Sparse	0.11	0.02	0.8

Table 5.2 Table Illustrating Simulation Settings for Case 2 Simulations

Scenario	Linearity	P	SNR	Sparsity	σ	β	β_0
9	Linear	10	1	Not	0.23	0.1	3
10	Linear	10	5	Not	0.1	0.1	3
11	Linear	10	1	Sparse	0.25	0.4	3
12	Linear	10	5	Sparse	0.11	0.4	3
13	Linear	100	1	Not	0.21	0.02	3
14	Linear	100	5	Not	0.09	0.02	3
15	Linear	100	1	Sparse	0.27	0.065	3
16	Linear	100	5	Sparse	0.12	0.065	3

5.4. Performance Measures

The two performance measures of test error were the mean squared error (MSE) and the median absolute deviation (MAD). These measures calculated the difference between the predicted values generated by the various methods, and the expected value that was calculated using the mean of the distributions as noted in Section 5.2.

5.5. Computational Details

The simulations were programmed using R, a statistical programming language. Table 5.3 shows the statistical packages and functions that were used during the simulation. The settings described were the same settings used in Chapter 4 for each of the three applied machine learning data sets. With all functions, the default settings were used, except as listed in Table 5.3 under the column titled “Customized Parameter Settings”.

Table 5.3 Table Illustrating R Computational Settings for Each of the 16 Different Simulations That Were Run

Model	R Function Used	Customized Parameter Settings
Linear Regression	stats::lm()	None
Ridge Regression	MASS::lm.ridge()	lambda = seq(0,50,0.01)
Subset Regression	MASS::stepAIC()	direction = "backward"
LASSO	lars::lars()	cv.lars() was used for cross-validation
Regression Tree	rpart::rpart()	method = “anova”
MARS	earth::earth()	nfold = 10, pmethod = "backward"
Random Forest	randomForest::randomForest()	ntree = 500
Boosted Trees	gbm::gbm()	distribution="gaussian", n.trees = 5000, interaction.depth = 5, shrinkage = 0.001, bag.fraction = 0.5, cv.folds=10
Neural Net	nnet::nnet()	preProcess="range"; trace=FALSE; tuneGrid=expand.grid(.size=c(1,2,5,10),.decay=c(0,0.001,0.1,1)
BART	bartMachine::bartMachine()	bartMachineCV() was not used due to computational intensity but is recommended

5.6. Overall Simulation Results

Below is the summary of overall simulation results. The red, bolded text highlights the cells which had the best performance measure for a given scenario.

5.6.1. *Tabular View of Overall Simulation Results*

Table 5.4 Table Illustrating The Average Mean Squared Error Results For The Heteroscedastic Non-Linear Simulations (Scenarios 1 through 8).

		Scenario							
Method	**Status**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Linear Regression	Transformed	**0.13**	0.04	**0.15**	0.05	0.26	0.10	0.20	0.08
	Untransformed	0.17	0.15	0.22	0.20	0.29	0.17	0.21	0.12
Ridge Regression	Transformed	**0.13**	**0.03**	0.16	0.05	**0.22**	**0.08**	0.17	0.07
	Untransformed	0.17	0.15	0.22	0.20	0.25	0.16	0.18	0.11
Subset Regression	Transformed	0.14	0.04	**0.15**	**0.04**	0.25	0.11	0.18	0.07
	Untransformed	0.18	0.15	0.22	0.20	0.28	0.18	0.19	0.12
LASSO	Transformed	**0.13**	0.04	**0.15**	**0.04**	**0.22**	0.09	**0.13**	**0.05**
	Untransformed	0.17	0.15	0.22	0.20	0.23	0.16	0.14	0.10
Regression Tree	Transformed	0.40	0.33	0.35	0.28	0.67	0.54	0.37	0.30
	Untransformed	0.40	0.32	0.35	0.26	0.69	0.55	0.38	0.30
MARS	Transformed	0.20	0.07	0.19	0.07	0.43	0.18	0.3	0.11
	Untransformed	0.21	0.10	0.20	0.09	0.47	0.21	0.33	0.12
Random Forest	Transformed	0.44	0.19	0.49	0.23	0.48	0.25	0.36	0.17
	Untransformed	0.45	0.19	0.50	0.23	0.49	0.24	0.37	0.17
Boosted Tree	Transformed	0.21	0.10	0.24	0.12	0.30	0.23	0.18	0.10
	Untransformed	0.19	0.11	0.21	0.12	0.31	0.22	0.19	0.10
Neural Net	Transformed	0.15	0.06	0.17	0.08	0.29	0.12	0.22	0.09
	Untransformed	0.18	0.13	0.23	0.17	0.37	0.19	0.28	0.14
BART	Transformed	0.21	0.10	0.24	0.12	0.29	0.16	0.20	0.10
	Untransformed	0.23	0.12	0.27	0.15	0.31	0.18	0.20	0.11

As noted in Table 5.4, for all cases where the original data was heteroscedastic and non-linear, using the mean squared error (MSE) as a performance measure, the log

transformed LASSO consistently outperformed nearly all other methods & was the best method on average. The MSE generated by predictions from the linear regression model, ridge regression, and subset regression, all seemed to perform similarly to the LASSO. One troubling observation, which was identified in case 1, was the weak performance of the random forest model. This model was expected to perform well, particularly when required to model and predict non-linear data. An investigation into the causes of this poor performance is set for future work.

Table 5.5 Table Illustrating Median Absolute Deviation Results for the Heteroscedastic Non-Linear Simulations (Scenarios 1 through 8).

		Scenario							
Method	**Status**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Linear Regression	Transformed	0.10	**0.02**	0.11	**0.03**	0.16	0.06	0.13	0.05
	Untransformed	**0.09**	0.08	0.12	0.11	0.19	0.10	0.14	0.07
Ridge Regression	Transformed	0.10	**0.02**	0.11	**0.03**	0.14	0.05	0.11	0.04
	Untransformed	**0.09**	0.08	0.12	0.11	0.16	0.09	0.12	0.07
Subset Regression	Transformed	0.10	**0.02**	0.11	**0.03**	0.16	0.07	0.12	0.05
	Untransformed	0.10	0.08	0.12	0.11	0.18	0.10	0.13	0.07
LASSO	Transformed	0.10	**0.02**	0.11	**0.03**	**0.13**	**0.06**	**0.08**	**0.03**
	Untransformed	**0.09**	0.08	0.12	0.11	0.14	0.09	0.09	0.05
Regression Tree	Transformed	0.24	0.19	0.19	0.15	0.40	0.33	0.23	0.19
	Untransformed	0.25	0.20	0.21	0.16	0.44	0.36	0.24	0.19
MARS	Transformed	0.12	0.03	0.11	**0.03**	0.25	0.10	0.18	0.06
	Untransformed	0.11	0.05	**0.07**	0.04	0.29	0.13	0.20	0.07
Random Forest	Transformed	0.28	0.11	0.30	0.13	0.30	0.15	0.24	0.11
	Untransformed	0.27	0.11	0.29	0.13	0.28	0.15	0.23	0.11
Boosted Tree	Transformed	0.12	0.05	0.13	0.05	0.17	0.12	0.11	0.06
	Untransformed	0.10	0.05	0.09	0.05	0.17	0.14	0.11	0.06
Neural Net	Transformed	0.10	0.03	0.11	0.04	0.18	0.07	0.14	0.05

	Untransformed	0.10	0.03	0,13	0.09	0.24	0.11	0.19	0.08
BART	Transformed	0.13	0.06	0.14	0.06	0.17	0.10	0.12	0.06
	Untransformed	0.11	0.06	0.13	0.06	0.18	0.11	0.12	0.07

As noted in Table 5.5, for all cases where the original data was heteroscedastic and non-linear, using the median absolute deviation (MAD), the log transformed LASSO seemed to produce the lowest MAD on average. For each linear-regression-based methods, the log transformation resulted in a lower mean MSE in all but one scenario. Transformation also seemed to help the neural nets (six out of eight scenarios) and the regression tree (all eight scenarios).

Table 5.6 Table Illustrating Average Mean Squared Error Results for the Heteroscedastic Linear Simulations (Scenarios 9 through 16).

		Scenario							
Method	**Status**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
Linear Regression	Transformed	0.19	0.16	0.20	0.17	0.24	0.12	0.35	0.21
	Untransformed	0.07	**0.03**	0.08	0.04	0.20	0.09	0.26	0.12
Ridge Regression	Transformed	0.18	0.16	0.20	0.17	0.20	0.11	0.31	0.19
	Untransformed	**0.06**	**0.03**	0.07	0.04	0.17	**0.07**	0.22	0.10
Subset Regression	Transformed	0.20	0.16	0.20	0.17	0.23	0.13	0.32	0.20
	Untransformed	0.09	**0.03**	0.06	**0.03**	0.19	0.10	0.23	0.10
LASSO	Transformed	0.19	0.16	0.19	0.16	0.18	0.11	0.23	0.17
	Untransformed	0.07	**0.03**	**0.05**	**0.03**	**0.14**	0.08	**0.14**	**0.07**
Regression Tree	Transformed	0.40	0.34	0.30	0.24	0.55	0.46	0.52	0.42
	Untransformed	0.38	0.33	0.28	0.23	0.54	0.45	0.51	0.42
MARS	Transformed	0.19	0.09	0.17	0.06	0.37	0.17	0.48	0.18
	Untransformed	0.18	0.07	0.14	0.05	0.36	0.15	0.44	0.16
Random Forest	Transformed	0.42	0.18	0.46	0.20	0.41	0.20	0.53	0.24
	Untransformed	0.42	0.18	0.45	0.20	0.40	0.20	0.51	0.24

Boosted Tree	Transformed	0.18	0.09	0.20	0.09	0.23	0.18	0.28	0.14
	Untransformed	0.17	0.10	0.16	0.09	0.22	0.18	0.25	0.14
Neural Net	Transformed	0.13	0.07	0.14	0.08	0.33	0.15	0.45	0.15
	Untransformed	0.08	0.03	0.10	0.04	0.28	0.11	0.39	0.16
BART	Transformed	0.20	0.10	0.23	0.10	0.24	0.14	0.30	0.15
	Untransformed	0.19	0.11	0.20	0.11	0.23	0.14	0,26	0.15

As observed in Table 5.6, when the original data is heteroscedastic and linear, and the MSE is used as a performance measure, the LASSO applied to the untransformed data delivered the lowest MSE on average across all cases. This is because the use of a log transformation on the heteroscedastic, linear data would have destroyed linearity in the sample. This conclusion fits with our initial hypotheses that the untransformed data would most effectively be modelled using linear modelling methods. Interestingly, the more flexible modern methods generally had numerically lower MSE with the untransformed data, contrary to our initial beliefs. This is especially evident for the neural net.

Table 5.7 Table Illustrating Median Absolute Deviation Results for the Heteroscedastic Linear Simulations (Scenarios 9 through 16).

		Scenario							
Method	Status	9	10	11	12	13	14	15	16
Linear Regression	Transformed	0.14	0.09	0.16	0.10	0.16	0.08	0.23	0.12
	Untransformed	0.05	**0.02**	0.05	**0.02**	0.14	0.06	0.18	0.08
Ridge Regression	Transformed	0.14	0.09	0.16	0.10	0.14	0.07	0.2	0.11
	Untransformed	**0.04**	**0.02**	0.05	**0.02**	0.11	0.05	0.15	0.06
Subset Regression	Transformed	0.14	0.09	0.16	0.10	0.15	0.08	0.21	0.11
	Untransformed	0.06	**0.02**	0.04	**0.02**	0.13	0.07	0.16	0.07
LASSO	Transformed	0.14	0.09	0.17	0.10	0.12	0.07	0.18	0.10
	Untransformed	0.05	**0.02**	0.04	**0.02**	**0.10**	**0.05**	**0.09**	**0.04**
Regression Tree	Transformed	0.26	0.22	0.20	0.16	0.37	0.31	0.35	0.28
	Untransformed	0.25	0.22	0.19	0.15	0.36	0.30	0.34	0.28

MARS	Transformed	0.11	0.05	0.11	0.04	0.23	0.11	0.29	0.11
	Untransformed	0.09	0.03	0.05	**0.02**	0.23	0.10	0.27	0.10
Random Forest	Transformed	0.26	0.12	0.28	0.13	0.26	0.13	0.32	0.15
	Untransformed	0.27	0.12	0.29	0.13	0.27	0.14	0.33	0.16
Boosted Tree	Transformed	0.11	0.06	0.12	0.05	0.14	0.12	0.16	0.08
	Untransformed	0.10	0.06	0.09	0.05	0.15	0.12	0.15	0.09
Neural Net	Transformed	0.09	0.04	0.11	0.04	0.21	0.09	0.27	0.09
	Untransformed	0.05	**0.02**	0.06	**0.02**	0.18	0.08	0.27	0.11
BART	Transformed	0.13	0.06	0.14	0.06	0.16	0.09	0.19	0.09
	Untransformed	0.11	0.06	0.12	0.06	0.15	0.09	0.17	0.09

As noted in Table 5.7, for most scenarios where the original data was heteroscedastic and linear, the untransformed LASSO seemed to produce the lower MAD on average. The transformation had a less clear effect on the mean MAD values for the more flexible methods than it had on the MSE. .

5.7. Specific Simulation Results

5.7.1 Heteroscedastic, Non-Linear Data

When the sample data was heteroscedastic and non-linear, and the number of explanatory variables was 10, the use of a variance stabilizing transform resulted in the linear, ridge, subset, and LASSO models outperforming the other models. The only other modern regression method that was comparable in terms of minimizing the squared error was the BART model applied to the transformed data. In general, these trends held constant in cases 1 through 4, regardless of the value of the signal to noise ratio that was used or the sparsity.

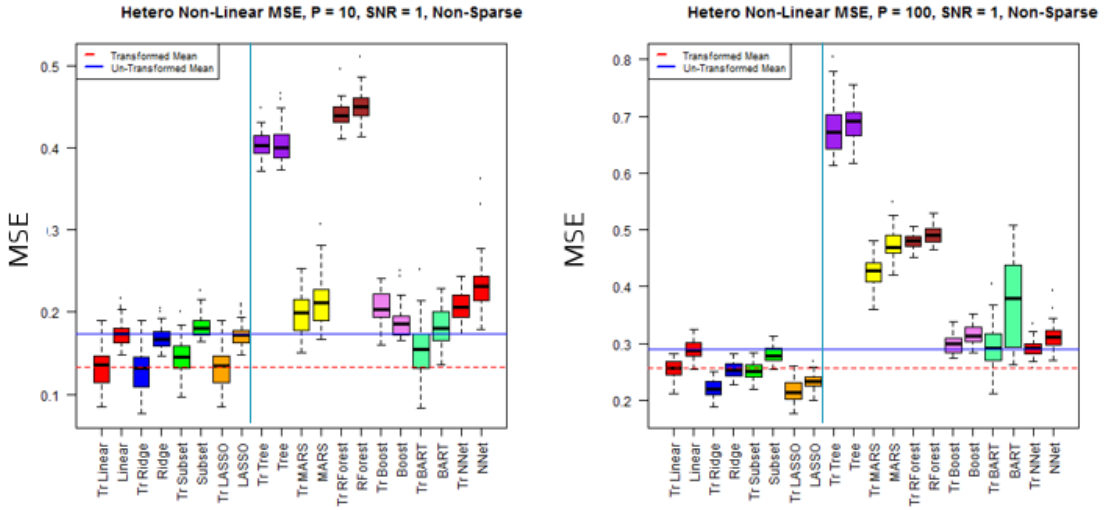

Figure 5.1 Summary of Heteroscedastic, Non-Linear Simulation Output

On the other hand, when the number of explanatory variables was 100 in the same data structure, the use of a variance stabilizing transform resulted in ridge regression and the LASSO models outperforming the other methods that were evaluated. In general, these trends held constant in cases 5 through 8 when the number of explanatory variables was set to 100. This trend also held regardless of the signal to noise ratio that was used or whether sparsity that was applied. These trends help reinforce the general conclusion that a positive effect was observed from the use of a transformation in this case. In fact, all methods, seem to prefer both linearity and equal variance when the number of explanatory variables is large.

5.7.1 Heteroscedastic, Linear Data

When the data was heteroscedastic and linear, and the number of explanatory variables was held at 10, the linear, ridge, subset, and LASSO models outperformed the other models when applied to the non-transformed data. This is illustrated generally in Figure 5.2. The only other modern regression technique that comparable in terms of minimizing the squared error was the BART model. In general, these trends held constant in cases 9 through 12, regardless of the value of the signal to noise ratio that was used or the sparsity level.

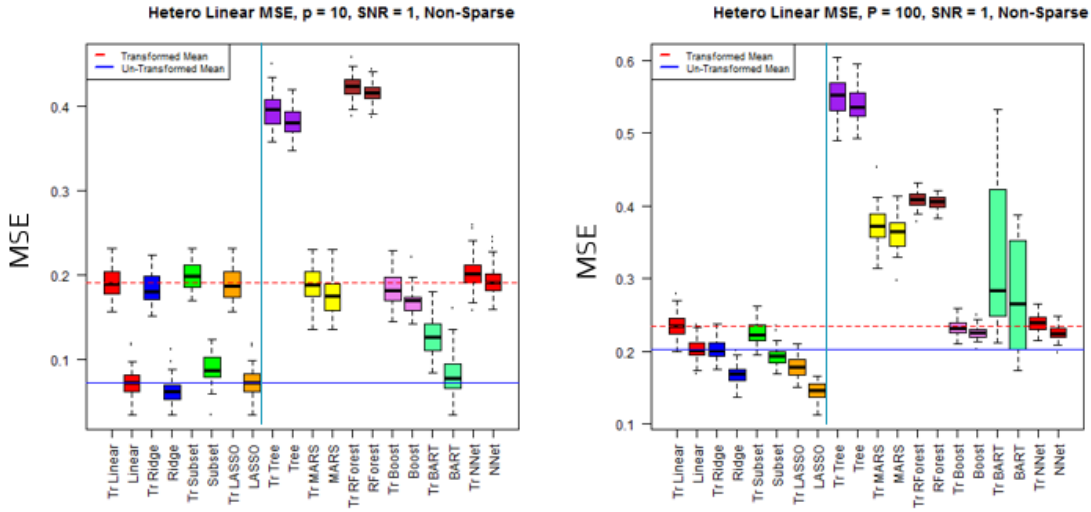

Figure 5.2 Summary of Heteroscedastic, Linear Simulation Output

When the number of explanatory variables was held at 100, the LASSO model, when applied to the non-transformed data generally outperformed all other models. This trend held in cases 13 through 16, regardless of the value of the signal to noise ratio that was used or the sparsity level applied. All linear methods performed better on the untransformed data, although the differences were smaller than in the scenarios with fewer variables. None of the modern methods are competitive with the linear-based methods, even when all are applied to transformed data that is nonlinear. This result was unexpected.

5.7.2 Comparison of Number of Explanatory Variables

The change in the number of explanatory variables produced differing results. For example, when the number of explanatory variables in the model was set at 100, the LASSO, which uses variable selection, consistently outperformed the other methods. This trend held consistent in all cases where the same settings were used and the only differences were the number of explanatory variables.

5.7.3 Comparison of the Impact of Differing SNR Ratios

When the signal to noise ratio was increased, little change was observed in terms of the ranking of the most successful methods. That said, when the signal to noise ratio was 5,

the simulation outputs for each of the methods had overall lower MSE and MAD's. This trend held true across all cases where the model factors were held constant and the only differences between the experimental set-up was the signal to noise ratio.

5.7.4 Comparison of MAD vs MSE

During each of the simulations, two different performance methods were used – the MAD and the MSE. When the number of explanatory variables was 10, the use of the MAD provided less clarity on which method produced the lowest overall error. In fact, for the linear, ridge, subset, and LASSO methods, the performance of these measures on both the transformed and untransformed data was similar. On the contrary, the MSE analysis suggested that a variance stabilizing transformation produced the best result for the linear, ridge, subset, and LASSO models. The performance of MARS was also different when the MAD versus the MSE was used. These observations highlight the importance of selecting the correct performance measure before analysis, and were observations that were similarly observed in scenarios 1 through 4 during the simulations.

6 Conclusion and Future Work

This thesis sought to clarify the impact that heteroscedasticity had on the predictive effectiveness of modern regression methods. In order to achieve this objective, we utilized both simulation and applied analysis. In particular, we began by analyzing the ability of ten different modern regression methods to predict outcomes for three medium-sized data sets that each featured heteroscedasticity. During this analysis, we attempted to understand the nature of the heteroscedasticity present, consider possible models for it, and, where appropriate, apply transformations to reduce its apparent magnitude. Following this, we used insights provided from this work to develop simulation experiments that explore the impact that various factors have on the prediction accuracy of our ten different regression methods. To close the thesis, in Section 6.1, we summarize the main difficulties and challenges we faced and discuss conclusions gleaned from both the exploratory data studies and the simulation methods. Section 6.2 discusses some project limitations and future research opportunities which are planned.

6.1 Conclusion

In Chapter 4, we started by analyzing three individual data sets. When analyzing these data sets, we observed that, in most cases, modern regression methods which rely on linearity seemed to perform worse than the tree based ensembles and other regression methods. In fact, we found that in all three cases, BART seemed to outperform other methods, with the random forest model coming in second. From this conclusion, it seems that both BART and random forests were set up to most effectively deal with data that is heteroscedastic and non-linear. Unfortunately, as was previously noted, the random forest model did not perform well in our simulations. This is something that will definitely be investigated as part of future research.

Additionally, when analyzing the three data sets, we noted that differences in predictive accuracy did sometimes exist when methods were applied to data that was transformed using a best-guessed variance stabilization method versus untransformed data. In the airfoil and bike rentals data, we found that the use of a variance stabilizing transformation actually increased the overall error. This conclusion is important, and reinforces the point

that practitioners in industry and academia cannot always rely on a variance stabilizing transformation to resolve heteroscedasticity and improve prediction accuracy. Finally, for the three applied data sets, we observed that most of the tree models and other modern regression methods produced similar predictions regardless of whether a variance stabilizing transformation had been used. This suggests that predictions from those methods are not impacted by the use of a variance stabilizing transformation.

While the three data sets did not allow us to generalize our findings, the simulation output did produce some interesting results. For example, as we expected, methods that assume linearity perform best when the data in question is linear. Our simulations suggested that, if a variance stabilizing transformation can be identified that simultaneously linearizes the relationship between X and Y, then linear methods, including classical regression, ridge regression, stepwise regression, or the LASSO, return the best predictions. On the other hand, as expected, when a variance stabilizing transformation ruins linearity, regression methods that rely on the assumption of linearity fail. This suggests that analysts must remain concerned about understanding the linearity of their data sets and the potential impact that a variance stabilizing transformation can have on their data before a regression method is applied. This recommendation applies not only to traditional linear regression approaches, but also to other more modern methods including BART which performed similarly.

Overall, these results have some important implications for practitioners. Firstly, as we discussed in Chapter 3, most of the various newer modern regression techniques do not implicitly account for heteroscedasticity. However, we did not find clear evidence that heteroscedasticity has an impact on the predictive effectiveness of these methods. On the other hand, we found a slight suggestion that many of the modern methods we explored seem to prefer linearity rather than homoscedasticity. This was contrary to expectation.

6.2 Limitations and Future Work

Overall, there are a number of limitations that existed within our work. Firstly, the largest limitation relates to the type of model that we used. In our Chapter 5 simulations, we only used two regression models, linear and non-linear. This should be addressed in future work and expanded to generate a more encompassing set of models. Secondly, we only

used one model of heteroscedasticity, in which the variance was proportional to the squared mean. Thirdly, within the simulations themselves, we only used a limited subset of factors. For example, we only looked at a single case of sparsity, and only varied the number of explanatory variables using 2 levels – 10 and 100. It is very possible that our conclusions could change as the number of variables increase beyond 100, or sit between 10 and 100. In general, this is something that should be addressed in future.

Furthermore, in our simulations, we only generated sample data sets with a sample size of 1000. In future, expanding this research to including samples of differing size, including high dimensional case with large p and small$n$, could be very valuable. Other cases, including the case when the number of explanatory variables significantly exceeds the sample size, should be addressed.

Another key limitation relates to the number of simulations that were run. During the simulations, we only completed 50 simulation runs for each model within each of the 16 different cases. Much of this related to computational load, since each scenario within our simulation took about 10 hours to run. While this number was half of the total number of analysis runs completed in the 3 dataset analysis, the number of simulations could be massively expanded. To do this, computational speed and modelling inefficiency are barriers that need to be addressed. This could include, in cases like BART, faster versions of the method as well as alternative methodologies or extensions of the BART methodology that generate more accurate predictions in various situations.

Finally, in terms of the modelling limitations, it is important to note that we did not use cross-validation on the BART models, or on any of the other models that required tuning parameters. This was because the code underlying the models took an extraordinary length of time to run. This could be improved and is a critical focus point of future research. Moreover, we only looked at a single correlation structure when generating the data for our simulations. Since certain models deal with correlation differently, modelling different correlation values could have been incredibly useful. Finally, future work is needed to explore some of the peculiar observations that we observed. For example, exploring why the random forest prediction results within the simulations were so poor is a key focus. Furthermore, understanding and auditing the instances of very low prediction error within

each of the three practice data sets is also key. Both of these areas will be targeted in future research.

References

Abu-Nimeh, S. et al. (2008). Bayesian Additive Regression Trees-Based Spam Detection for Enhanced Email Privacy. *ARES 08. Third International Conference.* 1044–1051.

Bache, K. & Lichman, M. (2013). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

Barros, R., Basgalupp, M., Carvalho, A., & Freitas, A. (2011). A Survey of Evolutionary Algorithms for Decision-Tree Induction. *IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews*, 42(3), 291-312

Breiman, L. (1996) Bagging Predictors. *Machine Learning*, 26(2):123-140

Breiman, L. (2001). Random forests. Machine Learning 45 5–32.

Breiman, L.; Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and Regression trees*. Monterey, CA: Wadsworth & Brooks/Cole Advanced Books & Software

Brem, R., Kruglyak, L. (2005). The landscape of genetic complexity across 5,700 gene expression traits in yeast. Proceedings of the National Academy of Sciences of the United States of America. 102: 1572 – 1577.

Brooks, T., Pope, D., & Marcolini, A. (1989) Airfoil self-noise and prediction. Technical report, *NASA RP-1218*, July.

Buhlmann, P. (2010) Remembrance of Leo Breiman. *The Annals of Applied Statistics*, 4(4): 1638 – 1641.

Carroll, R., & Ruppert, D. (1988). *Transformation and weighting in regression.* New York: Chapman and Hall.

Chipman, H., George, E., & McCulloch, R. (1998). Bayesian CART Model Search (with discussion and a rejoinder by the authors). *Journal of the American Statistical Association*, 93: 935–960

Chipman, H., George, E., & McCulloch, R. (2010). BART: Bayesian Additive Regression Trees. *Annals of Applied Statistics*, 4(1):266-298.

Chipman, H., Georrge, E., Lemp, J., & McCulloch, R. (2010). Bayesian Flexible Modelling Of Trip Durations. *Transportation Research*, Part B. 44:686-698

Damas, M., Salmeron, M., Diaz, A., Ortega, J., Prieto, A., & Olivares, G. (2000). Genetic algorithms and neuro-dynamic programming: application to water supply

networks". *Proceedings of 2000 Congress on Evolutionary Computation. 2000 Congress on Evolutionary Computation.* La Jolla, California, USA: IEEE.

Daye, J., & Chen, J. (2012) High-Dimensional Heteroscedastic Regression with an Application to eQTL Data Analysis. *Biometrics*, 68, 316–326

Deng, G., & Ferris, M. (2008). Neuro-dynamic programming for fractionated radiotherapy planning. *Springer Optimization and Its Applications*, 12:47-53

Duda, R., Hart, P., & Stork, D. (2001) *Pattern classification (2nd edition),* Wiley

Elith, J., Leathwick, J. & Hastie, T. (2008). A working guide to boosted regression trees. *Journal of Animal Ecology*, 77:802-813.

Fanaee-T, H., & Gama, J. (2013). Event labeling combining ensemble detectors and background knowledge. *Progress in Artificial Intelligence.* 3:1-15

Fanaee-T, H. (2013). UCI Machine Learning Repository [https://archive.ics.uci.edu/ml/datasets/Bike+Sharing+Dataset]. Irvine, CA: University of California, School of Information and Computer Science.

Fleiss, J. (1986). *The design and analysis of clinical experiments*. New York: Wiley.

Freund, J., & Shaphire, J. (1997). A decision theoretical generalization of online learning and an application to boosting. *Journal of Computer Systems Science.* 55:119-131

Friedman, J. (1991). Multivariate adaptive regression splines (with discussion and a rejoinder by the author). *Annals of Statistics*, 19:1–67.

Friedman, J. (1999). *Stochastic gradient boosting.* Stanford University.

Friedman, J. (2001). Greedy function approximation: A gradient boosting machine. *Annals Of Statistics*, 29:1189–1232

Friedman, J. (2002). Stochastic Gradient Boosting. *Computational Statistics and Data Analysis*, 38(4):367-378.

Galton, F. (1894), *Natural Inheritance (5th ed.),* New York: Macmillan and Company

Grissom, R. (2000). Heterogeneity of Variance in Clinical Data. *Journal of Consulting and Clinical Psychology,* 68(1):155-165

Hastie, T., Tibshirani, R., & Freidman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction. (2nd ed.).* New York: Springer Science Business Media, LLC.

Heritier, S., Cantoni, E., Copt, S., Victoria-Feser, M. (2009). *Robust methods in biostatistics*. New York: Wiley.

Hoerl, A., Kennard , R. (1970). Ridge regression: Biased estimation for non-orthogonal problems . *Technometrics*, 12:55 – 67

Lim, C., Sen, T. & Peddada K. (2010) Statistical inference in nonlinear regression under heteroscedasticity. *Sankhya B.* November, 72(2): 202-218

Lopez, R. (2014). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml/datasets/Airfoil+Self-Noise]. Irvine, CA: University of California, School of Information and Computer Science.

Loughin, T. (2012) Machine Learning Teaching Notes. *Stat 890: Modern Applied Statistics*. Simon Fraser University.

Luh, W. (1992). Heterogeneous variances in one-way fixed model ANOVA: Variance-stabilizing transformations and other alternatives. *Dissertation Abstracts International*, 53, DA9301212

Moore, D., & McCabe, G. (2005). *Introduction to the Practice of Statistics, 5th ed.* New York: W. H. Freeman.

Nash, W., Sellers, T., Talbot, S., Cawthorn, A., & Ford, W. (1994) "The Population Biology of Abalone (_Haliotis_ species) in Tasmania. I. Blacklip Abalone (_H. rubra_) from the North Coast and Islands of Bass Strait", *Sea Fisheries Division, Technical Report No. 48*

Prinzie, A., & Van Den Poel, D. (2008). Random Forests for multiclass classification: Random MultiNomial Logit. *Expert Systems with Applications*, 34 (3):1721

Quinlan, J. (1986). Induction of Decision Trees. *Machine Learning*, 1: 81-106

Rasmussen, J. (1989). Data transformation, Type I error rate, and power. *British Journal of Mathematical and Statistical Psychology*, 42: 203- 213.

Ripley, B. (1996) *Pattern Recognition and Neural Networks*, Cambridge

Rokach, L., & Maimon, O. (2008). *Data mining with decision trees: theory and applications*. World Scientific Pub Co Inc.

Rosenblatt, F. (1958). The Perceptron: A Probalistic Model for Information Storage and Organization in the Brain. *Psychological Review*, 65(6): 386–408

Secomandi, N. (2000). Comparing neuro-dynamic programming algorithms for the vehicle routing problem with stochastic demands. *Computers & Operations Research*, 27(11–12): 1201–1225

Strobl, C., Malley, G. & Tutz, P. (2009). An Introduction to Recursive Partitioning: Rationale, Application and Characteristics of Classification and Regression Trees, Bagging and Random Forests. *Psychological Methods*, 14(4): 323–348

Tibshirani, R. (1996). Regression Shrinkage and Selection Via the Lasso. *Journal of the Royal Statistical Society B*., 58(1):267-288

Tibshirani, R. (2013) Data Mining. *Course Notes*. Carnegie Mellon University. http://www.stat.cmu.edu/~ryantibs/datamining/lectures/

Venables, W., & Ripley, B. (2002) *Modern Applied Statistics with S*. Fourth edition. Springer

Visek, J. (2011) Empirical distribution function under heteroscedasticity. *Statistics*, 45(5):497-523

Waugh, S. (1995). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml/datasets/Abalone]. Irvine, CA: University of California, School of Information and Computer Science.

Wilcox, R. (1998). The goals and strategies of robust methods. *British Journal of Mathematical and Statistical Psychology*, 51:1-39.

Wilcox, R., & Keselman, H. (2012) Modern Regression Methods that can Substantially Increase Power and Provide a more Accurate Understanding of Associations. *European Journal of Personality*, 26:165–174

Wooldridge, J. (2009). Regression Analysis with Cross Sectional Data. *Introductory Econometrics: A Modern Approach (4th ed.)*. Cengage Learning

Wu, Y., Tjelmeland, H., & West, M. (2007). Bayesian CART: Prior specification and posterior simulation. *Journal of Computational Graphics and Statistics*, 16: 44–66.