Ian Dimapasok, Ben Jewell
STA 206
11 December 2023

Abalone Age: Rings

Abstract:

In this study, we aimed to predict the age of abalones using linear regression models, analyzing a dataset of 4,177 samples with their physical measurements. Our analysis began with some exploratory data analysis followed by fitting a first order regression model to the data. We addressed the challenge of multicollinearity, notably by removing the "Whole Weight" variable and applied forward stepwise regression based on the AIC and BIC criteria to develop the two models. This study demonstrates the potential of using linear regression for efficient prediction in abalones, highlighting the importance of careful variable selection and model refinement.

Introduction:

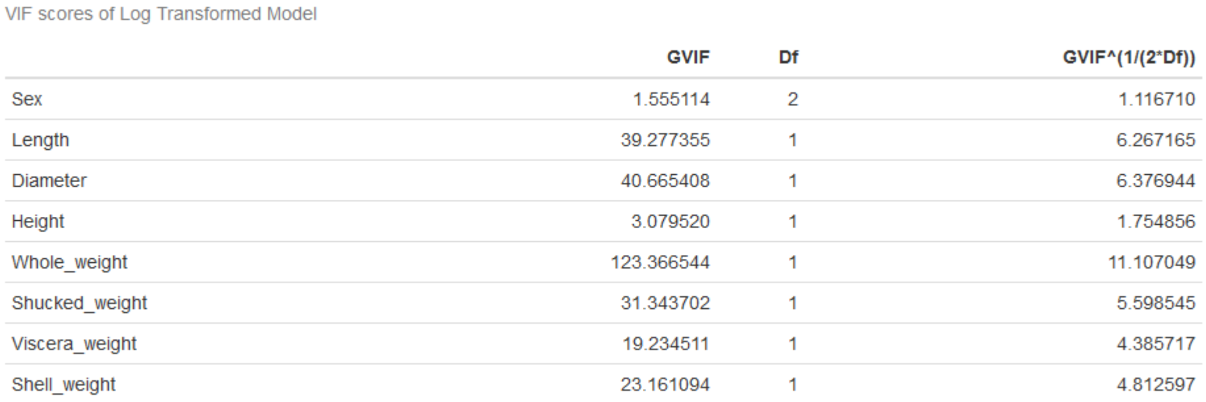
In the world of marine biology, abalones hold a unique place. These creatures, known for their shells and savory meat, also present a curious challenge in aging. Determining the age of an abalone consists of a slow and meticulous process, cutting its shell through the cone, staining it, and counting its rings through a microscope. Recognizing a need for a more efficient approach, our project focuses on using the Abalone dataset from the UCI machine learning repository and using linear regression to predict an abalone's age. This dataset is a compilation of physical measurements from 4,177 abalones and includes variables such as sex (male, female or infant), length, diameter, height, and weights in grams (whole weight, shucked weight, viscera weight, and shell weight). Notably, the 'rings' on the shell of an abalone, when incremented by 1.5, determine the age of an abalone.

Our project is driven by several questions: Can linear regression models accurately predict the age of an abalone based on the physical measurements given? Are there any significant interaction effects between an abalone's physical characteristics? And most importantly, which physical characteristics are most significant in predicting an abalone's age? The motivation of our project extends beyond academic interest. Specifically, the accurate prediction in age of an abalone can aid in the conservation efforts of an abalone. By being able to accurately predict the age of an abalone, conservationists and marine biologists can make more informed decisions regarding sustainable harvesting practices and habitat management.

Method & Results:

Given the dataset of abalone traits and rings, we first visualized the characteristics of our dataset using Exploratory Data Analysis plots such as histograms, scatterplots, and bar charts. (**Figures 1 through 10).** We see that the histograms for the predictor and response variables are slightly skewed, indicating a possible need for transformation of the data. Furthermore, based on our initial scatterplots, we see that there were some linear trends between predictor variables, possibly indicating some multicollinearity in our dataset. The initial models were then made using the AIC and BIC criterion for the forward stepwise selection process. Based on the forward stepwise selection process, a full first-order regression model was made to explore the relationship present between the age and other abalone data. From this process it was determined that the response variable needed transformation due to a violation in the normality assumption of our model (**Figures 12, 13**). Because of this, we applied the Box-Cox method to our data, and it was further seen that the response variable needed to be log-transformed. A single outlier was also noted (case 2502) (**Figures 17, 18**) and was thus not included in our further analysis.

In this early stage it was also noted that there was a high amount of multicollinearity among the variables in our first order model. Because all measurements were physical characteristics of the size and weight of the creatures they are not as uncorrelated as one might like. In investigating this it was found there were numerous variables that had a variance inflation factor (VIF) higher than 10 (**Table 1**). Most obvious of these was the `Whole Weight` variable, which is a direct sum of the other weight based variables. As such this variable was dropped, lowering many of the variance inflation factors among the renaming variables. However, some degree of multicollinearity persisted among the variables. To further address this issue, the data was also recentered and rescaled to aid with this multicollinearity, as some was still present among the data. Recentering involved adjusting the mean to zero and rescaling involved standardizing the range in variables. These steps were essential to mitigate the remaining multicollinearity and to enhance the robustness of our model.



VIF scores of Log Transformed Model

	GVIF	Df	GVIF^(1/(2*Df))
Sex	1.555114	2	1.116710
Length	39.277355	1	6.267165
Diameter	40.665408	1	6.376944
Height	3.079520	1	1.754856
Whole_weight	123.366544	1	11.107049
Shucked_weight	31.343702	1	5.598545
Viscera_weight	19.234511	1	4.385717
Shell_weight	23.161094	1	4.812597

Table 1: Initial Variance Inflation Factor (VIF) for all variables

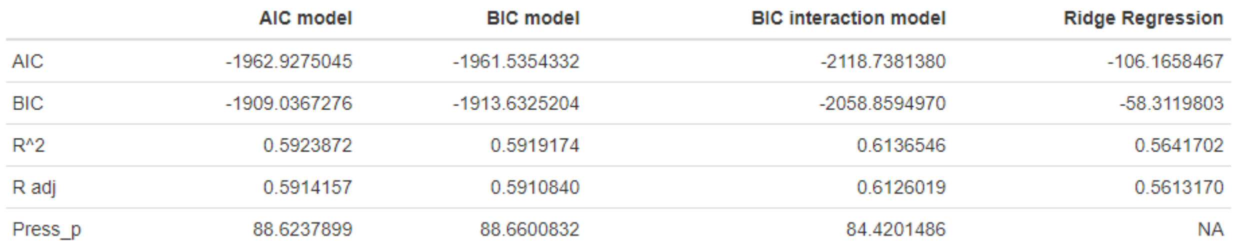
After these initial adjustments had been made the process of selecting our final model began. Again, both AIC and BIC were used as the criteria for two separate forward stepwise procedures to attempt to select the best model. The AIC based model selected a model dependent on shell weight, shucked weight, diameter, height and sex. The BIC based model selected the same predictors as the AIC model, except the model did not include the variable length.

Due to the previous encounters with high multicollinearity we decided to explore alternative modeling techniques as well. Specifically, we decided to implement a ridge regression process, a method renowned for its effectiveness in handling multicollinearity issues. We determined an optimal lambda value of 0.0181 and used the value to fit a ridge regression model. As stated previously, ridge regression is known to be useful in cases of multicollinearity. Therefore, it was hoped that this model would perform better than our previous linear regression models.

In exploring the interaction between the various physical characteristics of the abalone a model was fitted using a two factor interaction model. By fitting a model with all two factor interaction effects it was noted that shell weight and diameter along with shucked weight and diameter were significant when interacting together (**Table 2**). Combining this approach with the

BIC model, these two interaction terms were added with the goal of possibly improving the model.

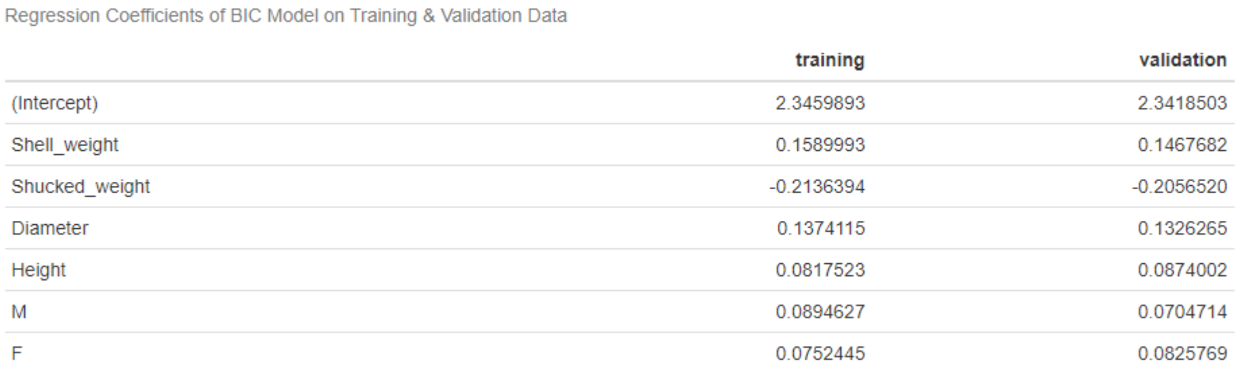
These four models discussed were compared using various model validation techniques. For each model the AIC, BIC, R squared, adjusted R squared, Mallow's Cp, Press p and RMSPE were calculated and compared (**Table 3, 4, 5**). While it was hoped that the ridge regression model would perform the best due to the multicollinearity previously found in the data, however in various regards this model is the worst of the four. In none of the validation techniques did the ridge regression model out compete the other models, and this may be due to the multicollinearity being removed from data when fitting the other models. Meanwhile the interaction model performed the best in all metrics. However, compared to the other three models the Cp score is terrible, indicating a strong model bias present. As such neither the ridge regression nor the interaction model were selected as our final model.



	AIC model	BIC model	BIC interaction model	Ridge Regression
AIC	-1962.9275045	-1961.5354332	-2118.7381380	-106.1658467
BIC	-1909.0367276	-1913.6325204	-2058.8594970	-58.3119803
R^2	0.5923872	0.5919174	0.6136546	0.5641702
R adj	0.5914157	0.5910840	0.6126019	0.5613170
Press_p	88.6237899	88.6600832	84.4201486	NA

Table 3: The AIC, BIC, R squared, adjusted R squared and Press p for the four final candidate models.

In many regards the AIC and BIC models are very similar. In four of the seven metrics the AIC model is slightly better than the BIC values. Further investigations were conducted due to how narrow the margins between the two model validation scores are, with each model being refitted with the validation data to ensure the scale and sign of the coefficients remained the same. As neither changed (**Table 6, 7**), the variance inflation factor scores were compared (**Table 8**). In the VIF scores of the AIC model both diameter and length variables were found to have a VIF score indicating multicollinearity, while none were found in the VIF scores of the BIC model. The AIC model had a high VIF score in Length and Diameter and because the BIC model did not include the Length measurements this removed much of the remaining multicollinearity. As the two models were so similar in other metrics and multicollinearity was a constant problem throughout this analysis the BIC model was chosen over the AIC model for our final linear regression model. See Table 7 for final model coefficients chosen.



Regression Coefficients of BIC Model on Training & Validation Data

	training	validation
(Intercept)	2.3459893	2.3418503
Shell_weight	0.1589993	0.1467682
Shucked_weight	-0.2136394	-0.2056520
Diameter	0.1374115	0.1326265
Height	0.0817523	0.0874002
M	0.0894627	0.0704714
F	0.0752445	0.0825769

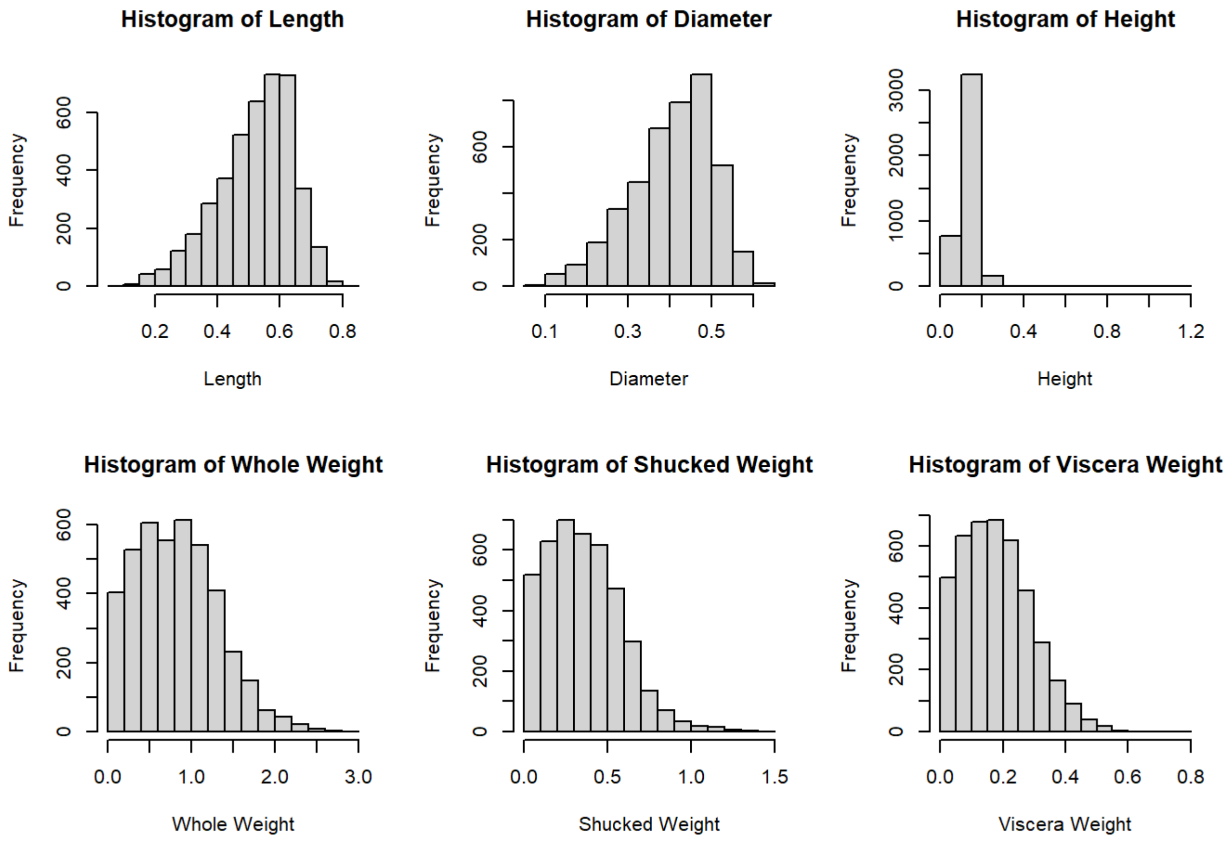
Table 7: Regression coefficients compared between the final BIC model on both the training data and validation data.

Conclusion & Discussion:

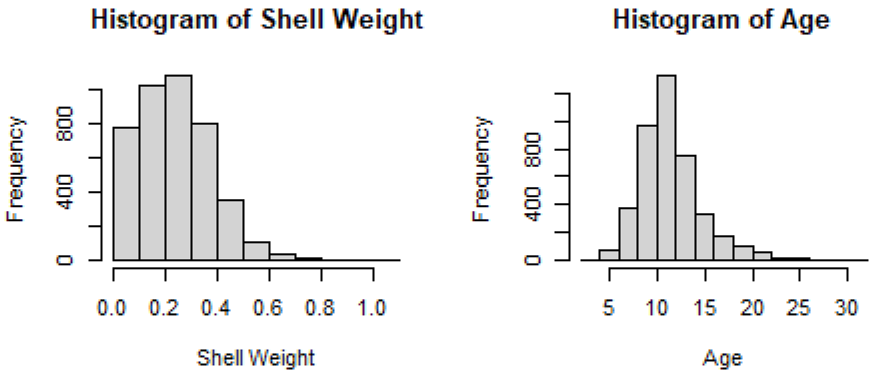
Our investigation into predicting the age of abalones using linear regression models yielded a multitude of significant results. Initially, we applied forward stepwise selection based on AIC and BIC criterion to develop our first-order model. A major transgression encountered was significant multicollinearity among most of the predictor variables. To address this problem, we removed Whole Weight from our model as well as a rescaling and recentering of the data. The refined approach resulted in two models, with the second model based on the BIC criterion proving to be the most effective in age prediction of an abalone. Attempts to include interaction effects in our model lead to an significant increase in bias in our model, leading us to conclude that the interaction-free second BIC model was our best fit. In addition, we found that the most significant predictors for the age of an abalone was the shell weight, shucked weight, diameter, sex and height.

While our results are promising for the prediction of an abalone's age, they are not without limitations. More specifically, the generalizability of our findings may be limited to the specific dataset we used. Furthermore, the exclusion of certain variables to reduce multicollinearity, while necessary, could have omitted some potentially important information. To address this, future analysis could involve testing our model on a diverse group of abalone datasets from various geographic regions and environments. This could help in assessing the robustness of our model across different populations. In addition, more rigorous pre-processing and thoughtful selection of our predictor variables by the data providers and gatherers could significantly enhance the quality and usability of our dataset while also minimizing multicollinearity. With these changes a more meaningful and thorough analysis could be made into predicting the age of abalone while preserving their shells.

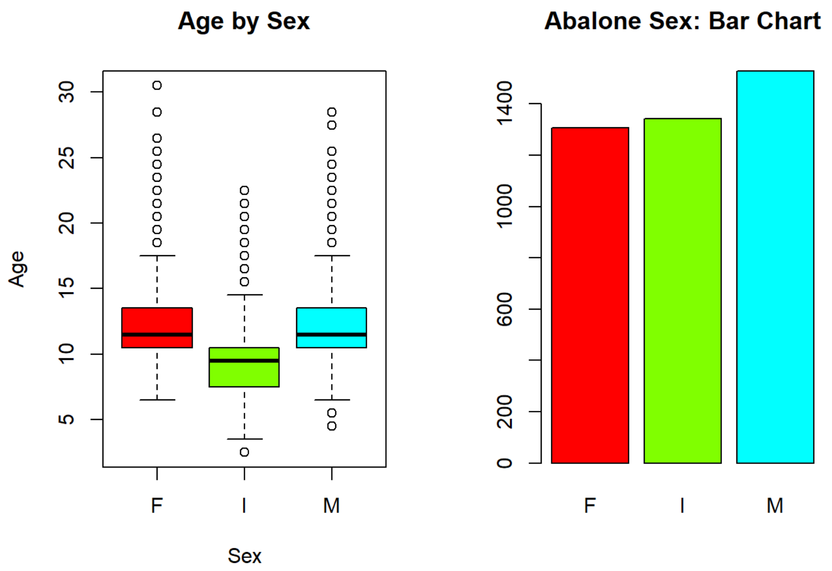
Appendices:



Figures 1 through 6: Histograms of variable distributions for Length, Diameter, Height, Whole Weight, Shucked Weight and Viscera Weight.



Figures 7, 8: Histograms of variable distributions for Shell Weight and Age



Figures 9, 10: Boxplot of Age distributed by Sex, Barchart of Sex distribution

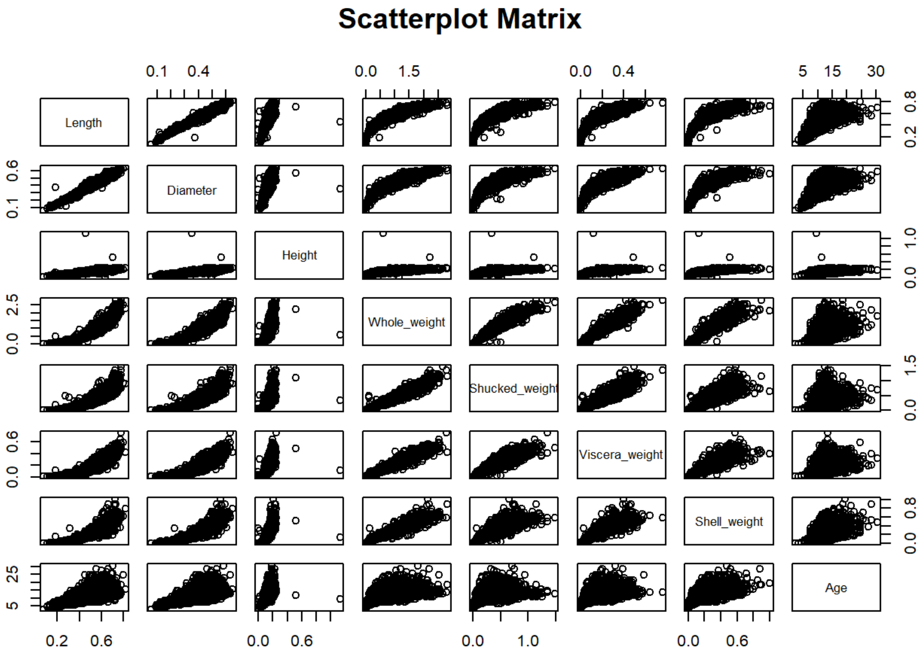


Figure 11: Scatterplot Matrix of original variables.

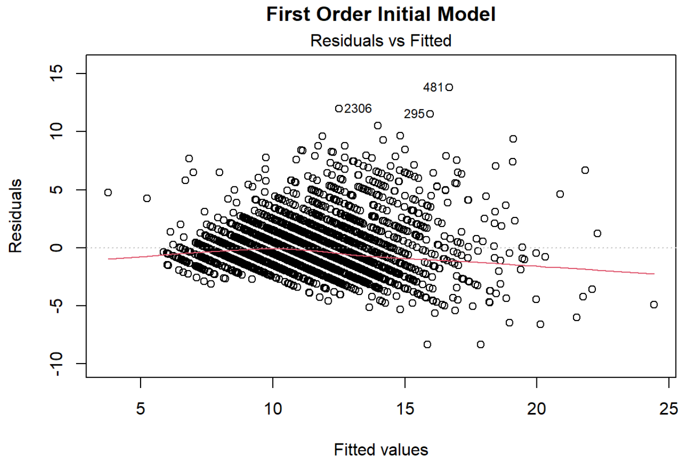


Figure 12: Residuals vs Fitted Values plot for initial fit model

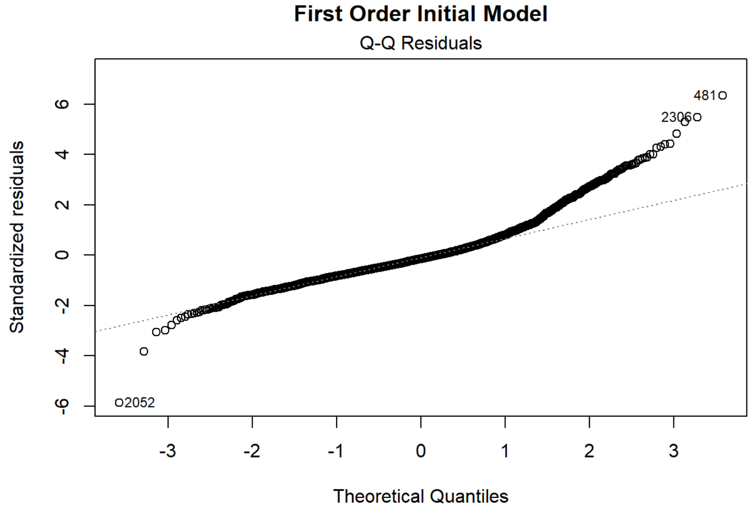


Figure 13: QQ-Plot for initial fit model

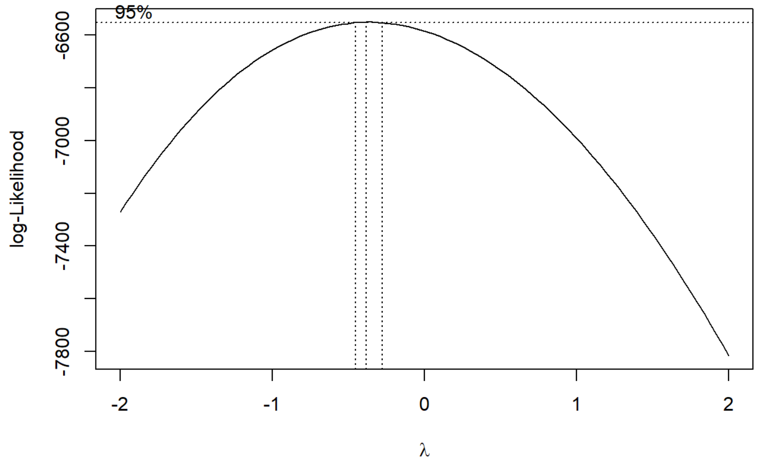


Figure 14: Box-Cox Transformation plot for lambda

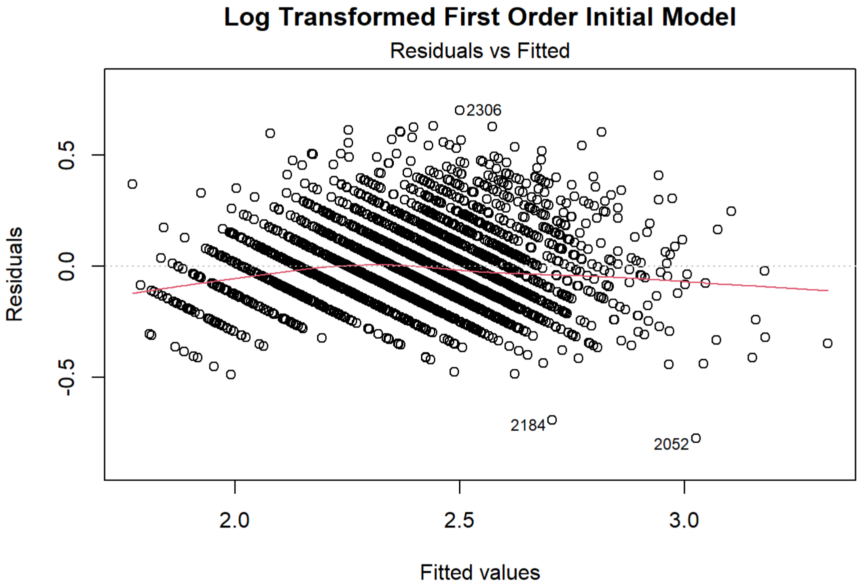


Figure 15: Residuals vs Fitted Values plot for log transformed initial model

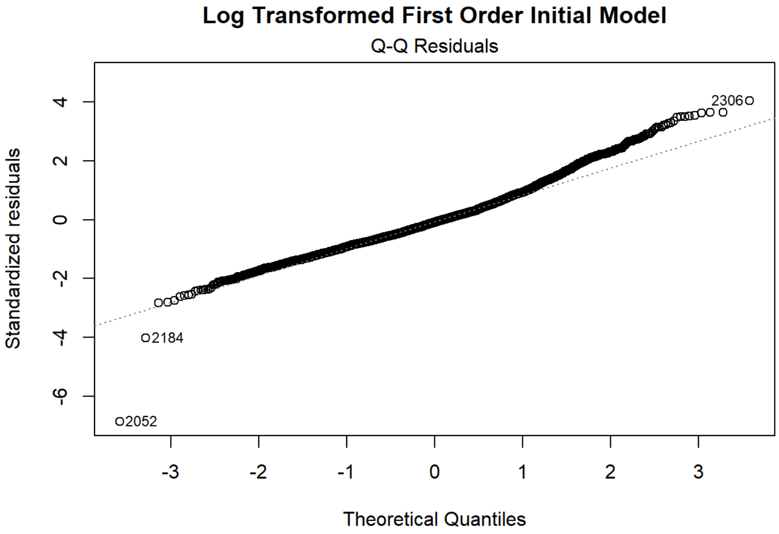


Figure 16: QQ-plot for log transformed initial model

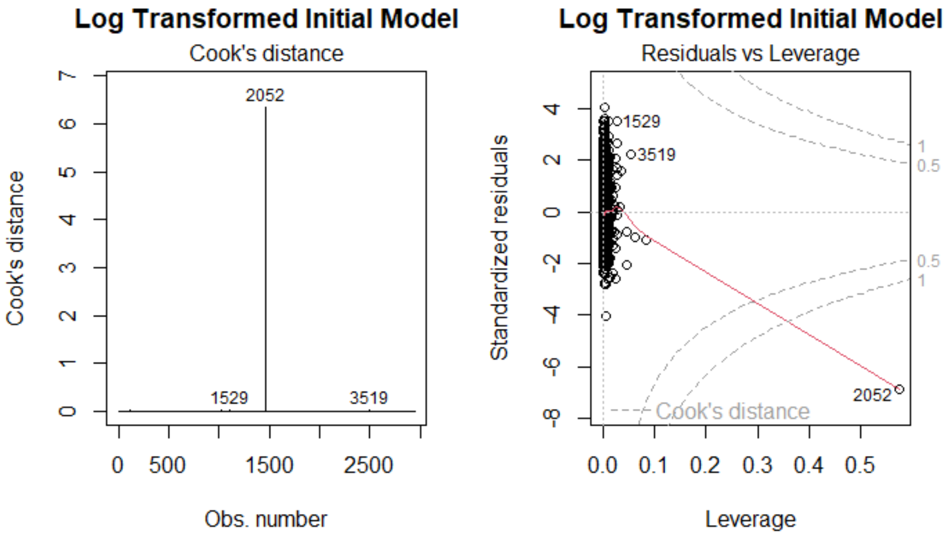


Figure 17, 18: Cook's distance for log transformed model, alongside Residuals vs Leverage plot.

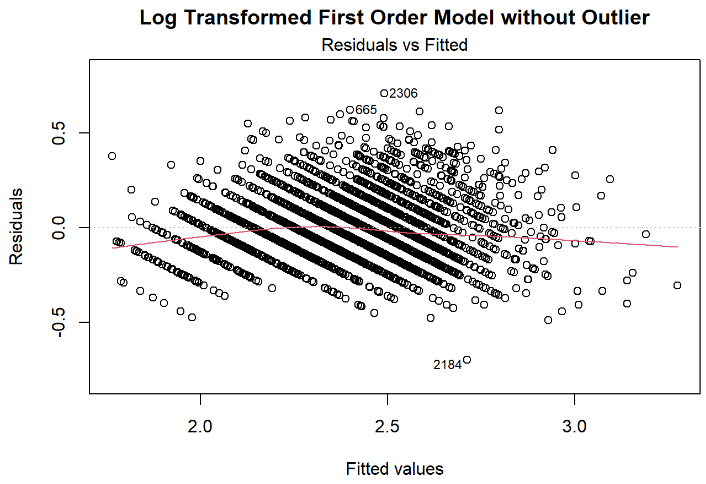


Figure 19: Residuals vs Fitted values without outlier

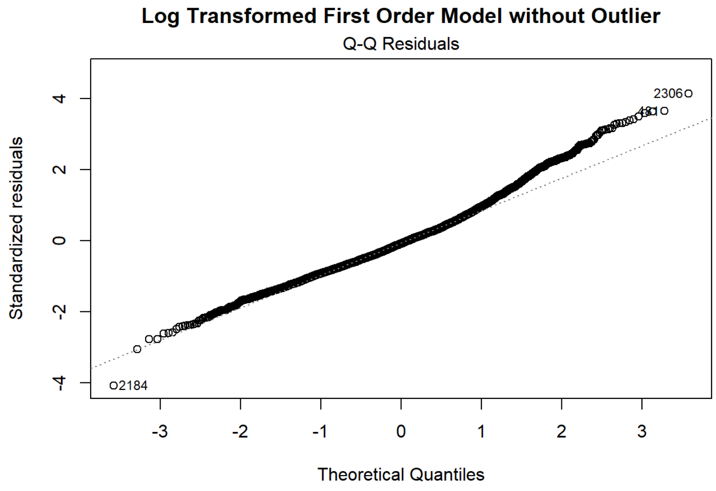


Figure 20: QQ-Plot for log transformed model without outlier

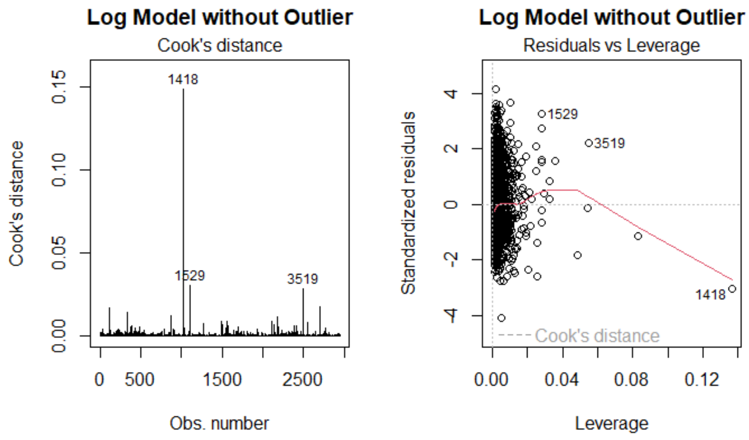
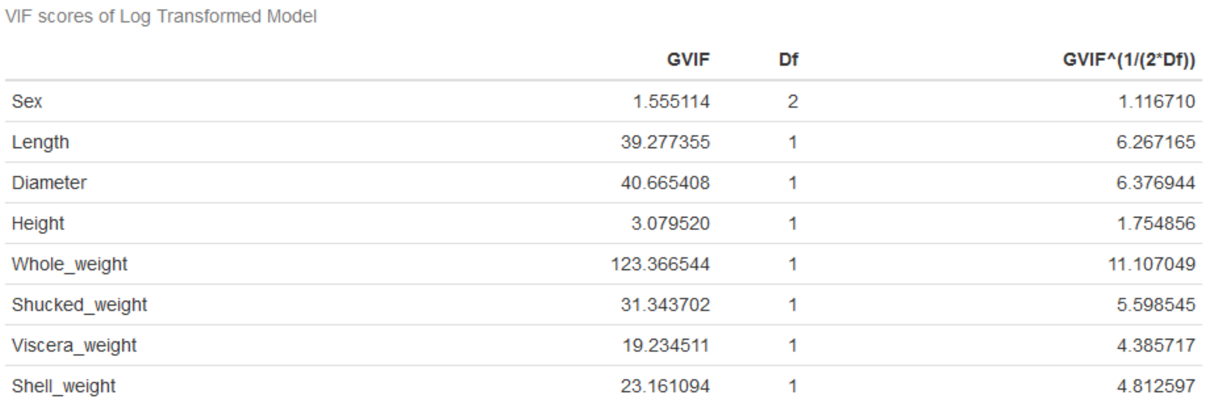


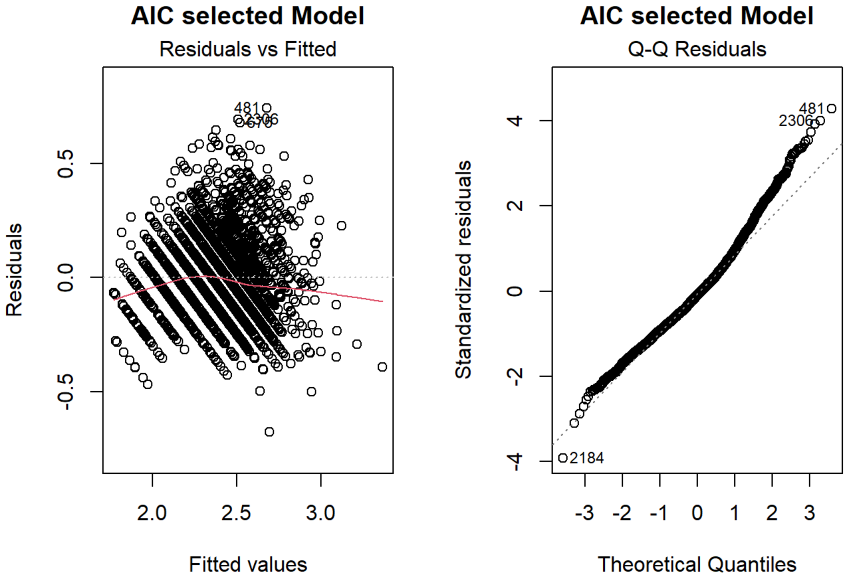
Figure 21, 22: Cook’s Distance without outlier and Residuals vs Leverage without outlier.



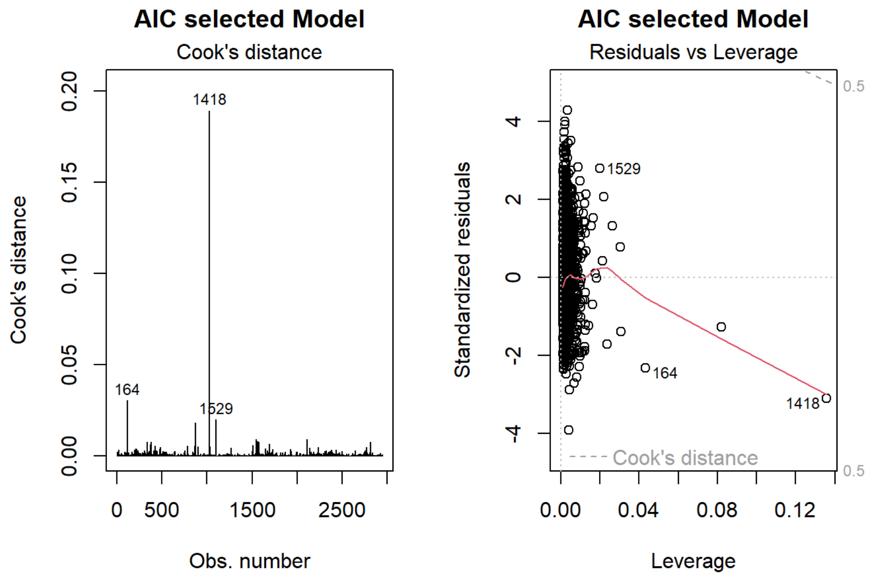
VIF scores of Log Transformed Model

	GVIF	Df	GVIF^(1/(2*Df))
Sex	1.555114	2	1.116710
Length	39.277355	1	6.267165
Diameter	40.665408	1	6.376944
Height	3.079520	1	1.754856
Whole_weight	123.366544	1	11.107049
Shucked_weight	31.343702	1	5.598545
Viscera_weight	19.234511	1	4.385717
Shell_weight	23.161094	1	4.812597

Table 1: Variance Inflation Factor (VIF) for all variables



Figures 23, 24: Residuals vs Fitted values and QQ-plot for the AIC selected model



Figures 25, 26: Cook’s Distance and Residuals vs Leverage values for the AIC selected model

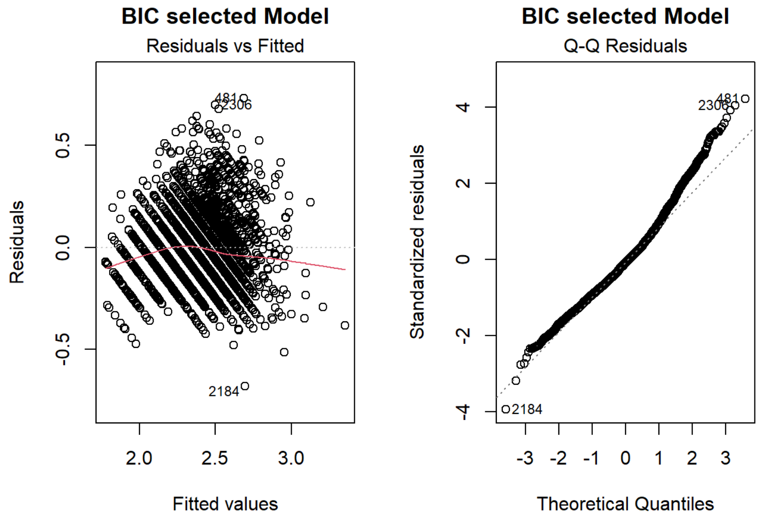
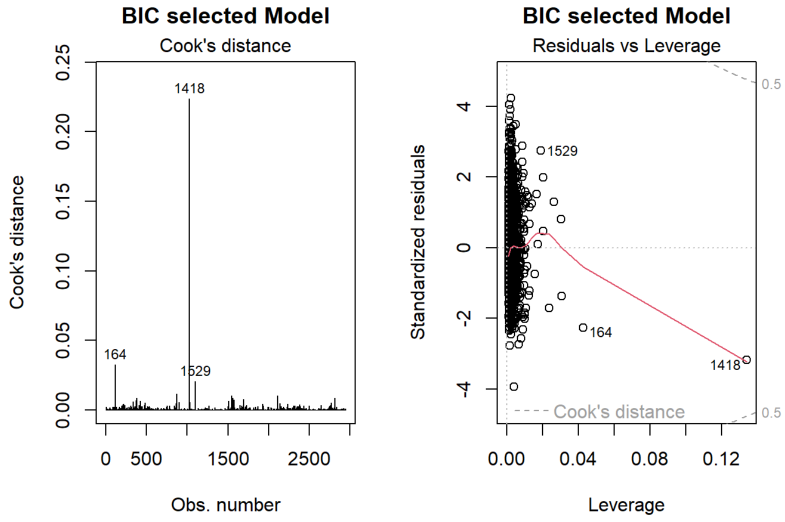
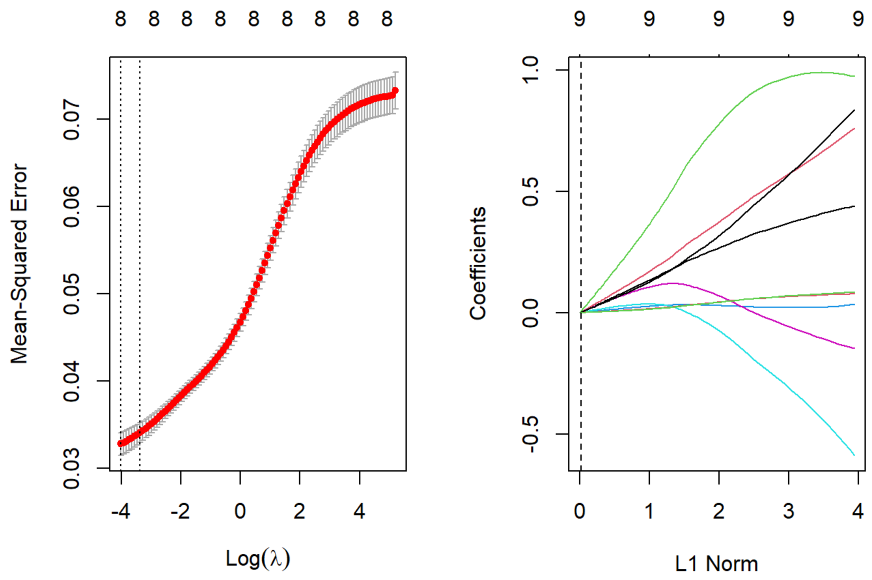


Figure 27, 28: Residuals vs Fitted values and QQ-plot for the BIC selected model



Figures 29, 30: Cook's Distance and Residuals vs Leverage values for the BIC selected model



Figures 31, 32: Lambda tuning graph and parameter tuning graph for ridge regression.

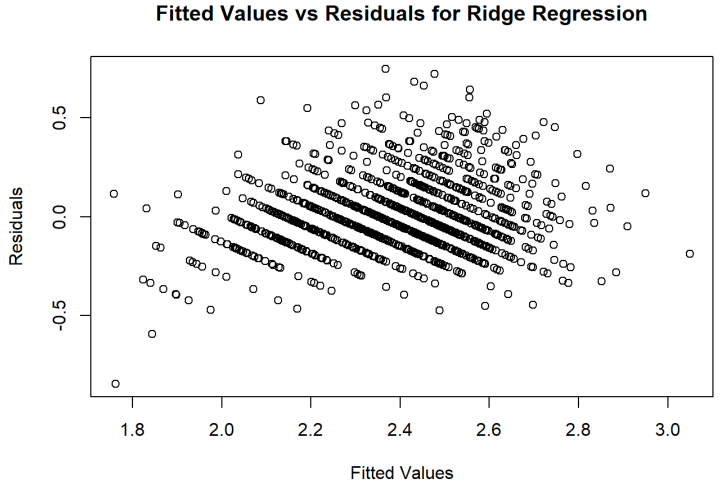
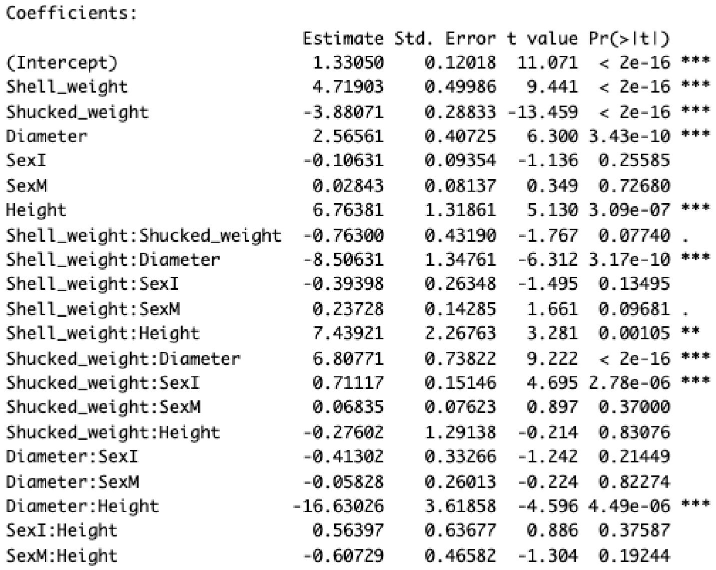
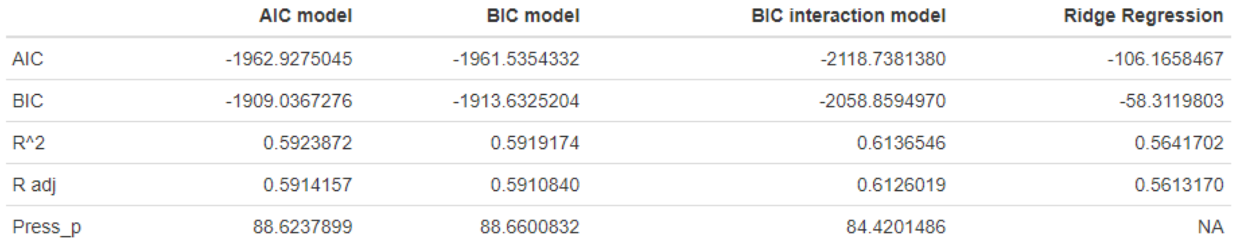


Figure 33: Residuals vs Fitted Values for ridge regression.



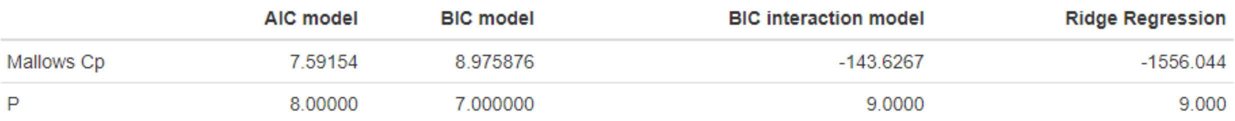
```
Coefficients:
                             Estimate Std. Error t value Pr(>|t|)    
(Intercept)                   1.33050    0.12018  11.071  < 2e-16 ***
Shell_weight                  4.71903    0.49986   9.441  < 2e-16 ***
Shucked_weight               -3.88071    0.28833 -13.459  < 2e-16 ***
Diameter                      2.56561    0.40725   6.300 3.43e-10 ***
SexI                         -0.10631    0.09354  -1.136  0.25585    
SexM                          0.02843    0.08137   0.349  0.72680    
Height                        6.76381    1.31861   5.130 3.09e-07 ***
Shell_weight:Shucked_weight  -0.76300    0.43190  -1.767  0.07740 .  
Shell_weight:Diameter        -8.50631    1.34761  -6.312 3.17e-10 ***
Shell_weight:SexI            -0.39398    0.26348  -1.495  0.13495    
Shell_weight:SexM             0.23728    0.14285   1.661  0.09681 .  
Shell_weight:Height           7.43921    2.26763   3.281  0.00105 ** 
Shucked_weight:Diameter       6.80771    0.73822   9.222  < 2e-16 ***
Shucked_weight:SexI           0.71117    0.15146   4.695 2.78e-06 ***
Shucked_weight:SexM           0.06835    0.07623   0.897  0.37000    
Shucked_weight:Height        -0.27602    1.29138  -0.214  0.83076    
Diameter:SexI                -0.41302    0.33266  -1.242  0.21449    
Diameter:SexM                -0.05828    0.26013  -0.224  0.82274    
Diameter:Height             -16.63026    3.61858  -4.596 4.49e-06 ***
SexI:Height                   0.56397    0.63677   0.886  0.37587    
SexM:Height                  -0.60729    0.46582  -1.304  0.19244    
```

Table 2: Coefficients for every possible interaction effects



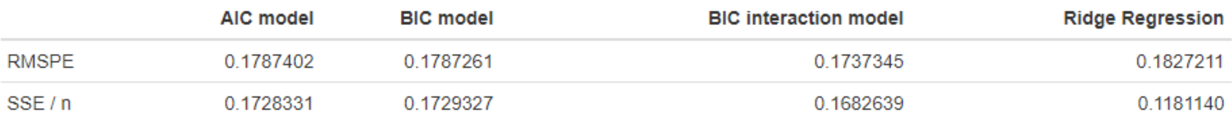
	AIC model	BIC model	BIC interaction model	Ridge Regression
AIC	-1962.9275045	-1961.5354332	-2118.7381380	-106.1658467
BIC	-1909.0367276	-1913.6325204	-2058.8594970	-58.3119803
R^2	0.5923872	0.5919174	0.6136546	0.5641702
R adj	0.5914157	0.5910840	0.6126019	0.5613170
Press_p	88.6237899	88.6600832	84.4201486	NA

Table 3: The AIC, BIC, R squared, adjusted R squared and Press p for the four final candidate models.



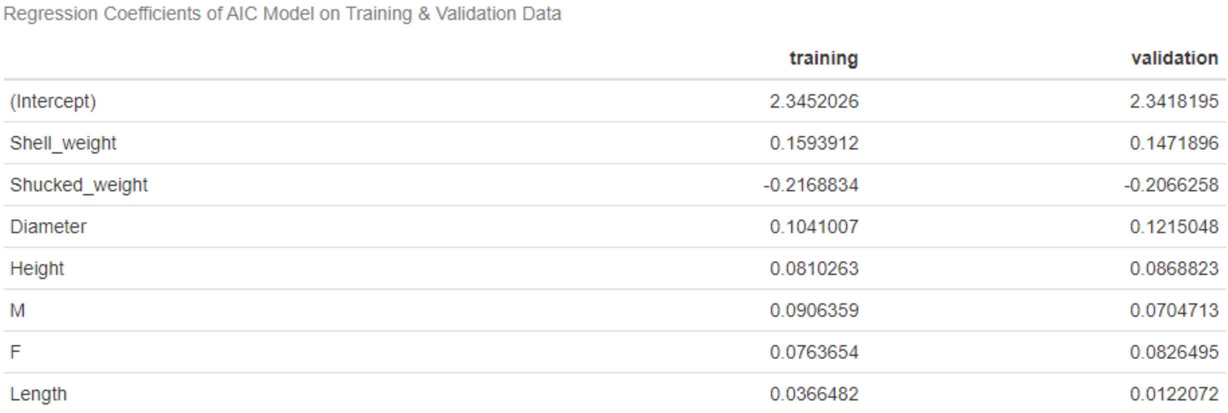
	AIC model	BIC model	BIC interaction model	Ridge Regression
Mallows Cp	7.59154	8.975876	-143.6267	-1556.044
P	8.00000	7.000000	9.0000	9.000

Table 4: The Mallow's Cp score vs number of coefficients in each of the final four candidate models.



	AIC model	BIC model	BIC interaction model	Ridge Regression
RMSPE	0.1787402	0.1787261	0.1737345	0.1827211
SSE / n	0.1728331	0.1729327	0.1682639	0.1181140

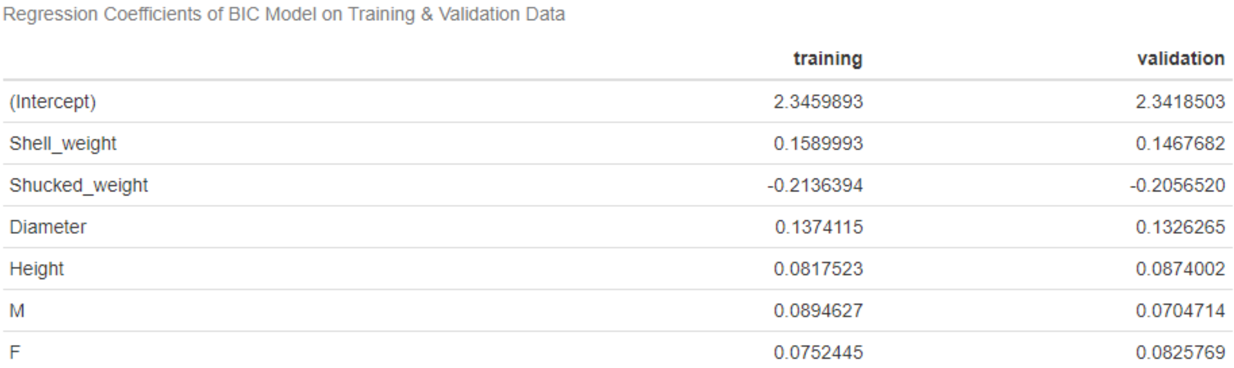
Table 5: Comparison of the RMSPE value vs root Sum of residuals squared divided by number of data points.



Regression Coefficients of AIC Model on Training & Validation Data

	training	validation
(Intercept)	2.3452026	2.3418195
Shell_weight	0.1593912	0.1471896
Shucked_weight	-0.2168834	-0.2066258
Diameter	0.1041007	0.1215048
Height	0.0810263	0.0868823
M	0.0906359	0.0704713
F	0.0763654	0.0826495
Length	0.0366482	0.0122072

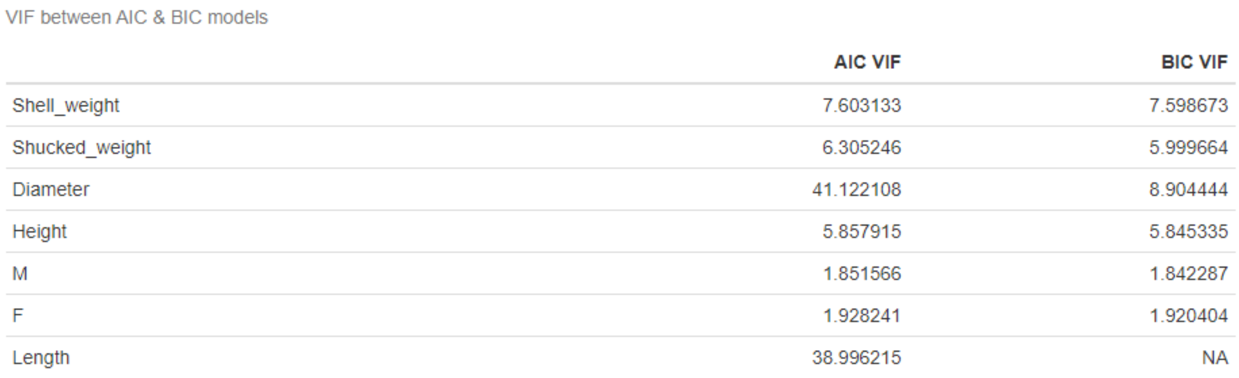
Table 6: Regression coefficients compared between the final AIC model on both the training data and validation data.



Regression Coefficients of BIC Model on Training & Validation Data

	training	validation
(Intercept)	2.3459893	2.3418503
Shell_weight	0.1589993	0.1467682
Shucked_weight	-0.2136394	-0.2056520
Diameter	0.1374115	0.1326265
Height	0.0817523	0.0874002
M	0.0894627	0.0704714
F	0.0752445	0.0825769

Table 7: Regression coefficients compared between the final BIC model on both the training data and validation data.



VIF between AIC & BIC models

	AIC VIF	BIC VIF
Shell_weight	7.603133	7.598673
Shucked_weight	6.305246	5.999664
Diameter	41.122108	8.904444
Height	5.857915	5.845335
M	1.851566	1.842287
F	1.928241	1.920404
Length	38.996215	NA

Table 8: Comparison of variance inflation factor scores between the AIC and BIC models.

STA 206 Project

Ian Dimapasok & Ben Jewell

2023-12-11

```r
# Load in the abalone dataset
# Loading the necessary libraries
library(knitr)
library(ggplot2)
library(MASS)
library(faraway)
library(pls)
```

```
##
## Attaching package: 'pls'

## The following object is masked from 'package:stats':
##
##     loadings
```

```r
library(car)
```

```
## Loading required package: carData

##
## Attaching package: 'car'

## The following objects are masked from 'package:faraway':
##
##     logit, vif
```

```r
library(glmnet)
```

```
## Loading required package: Matrix

## Loaded glmnet 4.1-8
```

```r
library(olsrr)
```

```
##
## Attaching package: 'olsrr'

## The following object is masked from 'package:faraway':
##
##     hsb

## The following object is masked from 'package:MASS':
##
##     cement

## The following object is masked from 'package:datasets':
##
##     rivers
```

```
rmspe<-function(y, yh) sqrt(mean((y-yh)^2)) #As provided by TA

# Reading the dataset
abalone = read.table('/Users/iandimapasok/Desktop/UC_Davis_Courses/STA206/Project/abalone.txt', header=

# Rename the columns of the dataset
colnames(abalone) = c('Sex', 'Length', 'Diameter', 'Height', 'Whole_weight', 'Shucked_weight', 'Viscera
head(abalone)
```

```
##   Sex Length Diameter Height Whole_weight Shucked_weight Viscera_weight
## 1   M  0.455    0.365  0.095       0.5140         0.2245         0.1010
## 2   M  0.350    0.265  0.090       0.2255         0.0995         0.0485
## 3   F  0.530    0.420  0.135       0.6770         0.2565         0.1415
## 4   M  0.440    0.365  0.125       0.5160         0.2155         0.1140
## 5   I  0.330    0.255  0.080       0.2050         0.0895         0.0395
## 6   I  0.425    0.300  0.095       0.3515         0.1410         0.0775
##   Shell_weight Rings
## 1        0.150    15
## 2        0.070     7
## 3        0.210     9
## 4        0.155    10
## 5        0.055     7
## 6        0.120     8
```

```
# Add an age column to get the age of all the abalones
abalone$Age = abalone$Rings + 1.5
abalone$Rings = NULL

# Converting X2 into a factor
abalone$Sex = as.factor(abalone$Sex)

# Check how many M, F, I are there
obs_level = table(abalone$Sex)
obs_level
```

```
##
##    F    I    M
## 1307 1342 1528
```

```
# Summary of the dataset
summary(abalone)
```

```
##  Sex          Length          Diameter          Height        Whole_weight
##  F:1307   Min.   :0.075   Min.   :0.0550   Min.   :0.0000   Min.   :0.0020
##  I:1342   1st Qu.:0.450   1st Qu.:0.3500   1st Qu.:0.1150   1st Qu.:0.4415
##  M:1528   Median :0.545   Median :0.4250   Median :0.1400   Median :0.7995
##           Mean   :0.524   Mean   :0.4079   Mean   :0.1395   Mean   :0.8287
##           3rd Qu.:0.615   3rd Qu.:0.4800   3rd Qu.:0.1650   3rd Qu.:1.1530
##           Max.   :0.815   Max.   :0.6500   Max.   :1.1300   Max.   :2.8255
##  Shucked_weight   Viscera_weight    Shell_weight          Age
##  Min.   :0.0010   Min.   :0.0005   Min.   :0.0015   Min.   : 2.50
##  1st Qu.:0.1860   1st Qu.:0.0935   1st Qu.:0.1300   1st Qu.: 9.50
##  Median :0.3360   Median :0.1710   Median :0.2340   Median :10.50
##  Mean   :0.3594   Mean   :0.1806   Mean   :0.2388   Mean   :11.43
##  3rd Qu.:0.5020   3rd Qu.:0.2530   3rd Qu.:0.3290   3rd Qu.:12.50
##  Max.   :1.4880   Max.   :0.7600   Max.   :1.0050   Max.   :30.50
```

```r
head(abalone)
```

```
##   Sex Length Diameter Height Whole_weight Shucked_weight Viscera_weight
## 1   M  0.455    0.365  0.095       0.5140         0.2245         0.1010
## 2   M  0.350    0.265  0.090       0.2255         0.0995         0.0485
## 3   F  0.530    0.420  0.135       0.6770         0.2565         0.1415
## 4   M  0.440    0.365  0.125       0.5160         0.2155         0.1140
## 5   I  0.330    0.255  0.080       0.2050         0.0895         0.0395
## 6   I  0.425    0.300  0.095       0.3515         0.1410         0.0775
##   Shell_weight  Age
## 1        0.150 16.5
## 2        0.070  8.5
## 3        0.210 10.5
## 4        0.155 11.5
## 5        0.055  8.5
## 6        0.120  9.5
```

Exploratory Data Analysis

```r
# Histograms for continuous variables
par(mfrow=c(2, 3))
hist(abalone$Length, main='Histogram of Length', xlab='Length')
hist(abalone$Diameter, main='Histogram of Diameter', xlab='Diameter')
hist(abalone$Height, main='Histogram of Height', xlab='Height')
hist(abalone$Whole_weight, main='Histogram of Whole Weight', xlab='Whole Weight')
hist(abalone$Shucked_weight, main='Histogram of Shucked Weight', xlab='Shucked Weight')
hist(abalone$Viscera_weight, main='Histogram of Viscera Weight', xlab='Viscera Weight')
```

```r
hist(abalone$Shell_weight, main='Histogram of Shell Weight', xlab='Shell Weight')
hist(abalone$Age, main='Histogram of Age', xlab='Shell Weight')
```

```r
par(mfrow = c(1,2))
# Boxplot for categorical variable 'Sex'
boxplot(Age ~ Sex, data=abalone, main='Age by Sex', xlab='Sex', ylab='Age', col=rainbow(4))

# Bar Chart
barplot(table(abalone$Sex),col=rainbow(4),main='Abalone Sex: Bar Chart')
```

```r
# Scatterplots for relationships
pairs(~Length + Diameter + Height + Whole_weight + Shucked_weight + Viscera_weight + Shell_weight + Age
```

Data Splitting: Training & Validation

```r
#Set seed to ensure our data is split the same each time
set.seed(206)

#Split the data into train and test data sets
tv_split = sample(c(TRUE, FALSE), nrow(abalone), replace = TRUE, prob = c(0.7, 0.3))

#Train Data
abalone_t = abalone[tv_split, ]
#Validation Data
abalone_v = abalone[!tv_split, ]
```

```r
#unbind seed for future random procedures
set.seed(NULL)
```

Preliminary Model Fitting

```r
#Fitting first order model as starting point
first_order = lm(Age ~ ., data = abalone_t)


#Summary plots
plot(first_order, which =1, sub.caption = '', main = 'First Order Initial Model')
```

```r
plot(first_order, which = 2, sub.caption = '', main = 'First Order Initial Model')
```

```r
# Based on qq-plot and residual vs fitted, it seems to violate our normality assumptions
MASS::boxcox(first_order)
```

```r
# Use log transformation on response variable to see if it helps with our first-order model assumptions
first_order_log = lm(log(Age) ~ ., data = abalone_t)
first_order_log
```

```
##
## Call:
## lm(formula = log(Age) ~ ., data = abalone_t)
##
## Coefficients:
##    (Intercept)             SexI             SexM           Length         Diameter
##        1.68145         -0.07739          0.01463          0.38136          1.16525
##         Height     Whole_weight   Shucked_weight  Viscera_weight     Shell_weight
##        0.81110          0.60598         -1.52959         -0.69231          0.50746
```

```r
#Plotting diagnostic plots
plot(first_order_log, which = 1:2, sub.caption = '', main = 'Log Transformed First Order Initial Model'
```

```r
par(mfrow=c(1,2))
plot(first_order_log, which = 4:5, sub.caption = '', main = 'Log Transformed First Order Initial Model'
```

```r
# Run the first order model again w/ transformed and subsetting case '2052'
first_order_sub =lm(log(Age) ~ ., data = abalone_t, subset=setdiff(rownames(abalone), "2052"))
first_order_sub
```

```
##
## Call:
## lm(formula = log(Age) ~ ., data = abalone_t, subset = setdiff(rownames(abalone),
##     "2052"))
##
## Coefficients:
##    (Intercept)             SexI             SexM           Length         Diameter
##        1.65092         -0.07513          0.01408          0.34411          1.00747
##         Height     Whole_weight   Shucked_weight  Viscera_weight     Shell_weight
##        1.83597          0.59317         -1.49943         -0.73479          0.41945
```

```r
plot(first_order_sub, which = c(1, 2), sub.caption = '', main = 'Log Transformed First Order Model with
```

```r
par(mfrow=c(1,2))
plot(first_order_sub, which = c(4, 5), sub.caption = '', main = 'Log Transformed First Order Model with
library(car)
# Calculate VIF for the model
vif_scores = vif(first_order_log)

# Display the VIF scores
kable(vif_scores, caption = 'VIF scores of Log Transformed Model')
```

Table 1: VIF scores of Log Transformed Model

	GVIF	Df	GVIF^(1/(2*Df))
Sex	1.555114	2	1.116710
Length	39.277355	1	6.267165
Diameter	40.665408	1	6.376944
Height	3.079520	1	1.754856
Whole_weight	123.366544	1	11.107049
Shucked_weight	31.343702	1	5.598545
Viscera_weight	19.234511	1	4.385717
Shell_weight	23.161094	1	4.812597

```r
# You can also check for VIF scores greater than a certain threshold, say 5 or 10
kable(vif_scores[vif_scores > 9], col.names = 'VIF scores', caption = 'VIF scores > 10')
```

Table 2: VIF scores > 10

VIF scores
39.27736
40.66541
123.36654
31.34370
19.23451
23.16109
11.10705

```r
#Define the model with only the intercept, subsetting the influential case
none_mod = lm(log(Age) ~ 1, data = abalone_t, subset=setdiff(rownames(abalone_t), "2052"))
#Define the full model, subsetting the influential case
full_mod = lm(log(Age) ~., data = abalone_t, subset=setdiff(rownames(abalone_t), "2052"))

#Forward stepwise based on AIC
stepAIC(none_mod, scope=list(upper=full_mod, lower = ~1), direction="forward", k=2, trace = FALSE)
```

```
##
## Call:
## lm(formula = log(Age) ~ Shell_weight + Shucked_weight + Diameter +
##     Sex + Height + Whole_weight + Viscera_weight + Length, data = abalone_t,
##     subset = setdiff(rownames(abalone_t), "2052"))
##
## Coefficients:
##    (Intercept)    Shell_weight  Shucked_weight        Diameter            SexI
##        1.65092         0.41945        -1.49943         1.00747        -0.07513
##           SexM          Height    Whole_weight  Viscera_weight          Length
##        0.01408         1.83597         0.59317        -0.73479         0.34411
```

```
#Forward stepwise based on BIC
stepAIC(none_mod, scope=list(upper=full_mod, lower = ~1), direction="forward", k=log(nrow(abalone_t)),
```

```
##
## Call:
## lm(formula = log(Age) ~ Shell_weight + Shucked_weight + Diameter +
##     Sex + Height + Whole_weight + Viscera_weight, data = abalone_t,
##     subset = setdiff(rownames(abalone_t), "2052"))
##
## Coefficients:
##    (Intercept)    Shell_weight  Shucked_weight        Diameter            SexI
##        1.66925         0.41657        -1.48509         1.38377        -0.07366
##           SexM          Height    Whole_weight  Viscera_weight
##        0.01413         1.85159         0.59009        -0.71367
```

Reattempting stepAIC removing whole weight from data set and recentering data

```
#Reprocessing data to use indicator functions for later ridge regression

#Training Data: Indicator functions & y variable log transformed
abalone_idc_t = abalone_t
abalone_idc_t$F = rep(0, nrow(abalone_idc_t))
abalone_idc_t$F[which(abalone_t$Sex == 'F')] = 1
abalone_idc_t$M = rep(0, nrow(abalone_idc_t))
abalone_idc_t$M[which(abalone_t$Sex == 'M')] = 1
abalone_idc_t$Sex = NULL
abalone_idc_t$Age = log(abalone_idc_t$Age)

#Validation Data: Indicator functions & y variable log transformed
abalone_idc_v = abalone_v
abalone_idc_v$F = rep(0, nrow(abalone_idc_v))
abalone_idc_v$F[which(abalone_v$Sex == 'F')] = 1
abalone_idc_v$M = rep(0, nrow(abalone_idc_v))
abalone_idc_v$M[which(abalone_v$Sex == 'M')] = 1
abalone_idc_v$Sex = NULL
abalone_idc_v$Age = log(abalone_idc_v$Age)

#Setting data as matrices for ridge regression later
x_t_data = as.matrix(abalone_idc_t[,-8])
y_t_data = as.matrix(abalone_idc_t[,8])
x_v_data = as.matrix(abalone_idc_v[,-8])
y_v_data = as.matrix(abalone_idc_v[,8])

#Centering & Rescaling the data

#Training Data: Rescaled, dropping `Whole_weight` from data
abalone_rs_t = as.data.frame(scale(abalone_t[, -c(1, 5, 9)]))
abalone_rs_t$M = abalone_idc_t$M
abalone_rs_t$F = abalone_idc_t$F
abalone_rs_t$Age = abalone_idc_t$Age

#Validation Data: Rescaled, dropping `Whole_weight` from data
abalone_rs_v = as.data.frame(scale(abalone_v[, -c(1, 5, 9)]))
abalone_rs_v$M = abalone_idc_v$M
```

```
abalone_rs_v$F = abalone_idc_v$F
abalone_rs_v$Age = abalone_idc_v$Age

#Setting data as matrices for ridge regression later
x_ts_data = as.matrix(abalone_rs_t[,-9])
y_ts_data = as.matrix(abalone_rs_t[,9])
x_vs_data = as.matrix(abalone_rs_v[,-9])
y_vs_data = as.matrix(abalone_rs_v[,9])
```

```
# Define the model with only the intercept, subsetting the influential case
none_mod = lm(y_ts_data ~ 1, data = as.data.frame(x_ts_data), subset=setdiff(rownames(abalone), "2052")

# Define the full model, subsetting the influential case
full_mod = lm(y_ts_data ~ ., data = as.data.frame(x_ts_data), subset=setdiff(rownames(abalone), "2052")

# Forward stepwise based on AIC
stepAIC(none_mod, scope=list(upper=full_mod, lower = ~1), direction="both", k = 2, trace = FALSE)
```

```
##
## Call:
## lm(formula = y_ts_data ~ Shell_weight + Shucked_weight + Diameter +
##     Height + M + F + Length, data = as.data.frame(x_ts_data),
##     subset = setdiff(rownames(abalone), "2052"))
##
## Coefficients:
##    (Intercept)    Shell_weight  Shucked_weight        Diameter          Height
##        2.34520         0.15939        -0.21688         0.10410         0.08103
##              M               F          Length
##        0.09064         0.07637         0.03665
```

```
# Forward stepwise based on BIC
stepAIC(none_mod, scope=list(upper=full_mod, lower = ~1), direction="both", k = log(nrow(abalone_t)), t
```

```
##
## Call:
## lm(formula = y_ts_data ~ Shell_weight + Shucked_weight + Diameter +
##     Height + M + F, data = as.data.frame(x_ts_data), subset = setdiff(rownames(abalone),
##     "2052"))
##
## Coefficients:
##    (Intercept)    Shell_weight  Shucked_weight        Diameter          Height
##        2.34599         0.15900        -0.21364         0.13741         0.08175
##              M               F
##        0.08946         0.07524
```

```
#Our two AIC/BIC models
aic_model = stepAIC(none_mod, scope=list(upper=full_mod, lower = ~1), direction="both", k = 2, trace = 
bic_model = stepAIC(none_mod, scope=list(upper=full_mod, lower = ~1), direction="both", k = log(nrow(aba

aic_model
```

```
##
## Call:
## lm(formula = y_ts_data ~ Shell_weight + Shucked_weight + Diameter +
##     Height + M + F + Length, data = as.data.frame(x_ts_data),
##     subset = setdiff(rownames(abalone), "2052"))
```

```
##
## Coefficients:
##    (Intercept)    Shell_weight  Shucked_weight        Diameter          Height
##        2.34520         0.15939        -0.21688         0.10410         0.08103
##              M               F          Length
##        0.09064         0.07637         0.03665
```

```r
anova(aic_model)
```

```
## Analysis of Variance Table
##
## Response: y_ts_data
##                  Df Sum Sq Mean Sq   F value    Pr(>F)
## Shell_weight      1 96.124  96.124 3208.1039 < 2.2e-16 ***
## Shucked_weight    1 10.637  10.637  355.0066 < 2.2e-16 ***
## Diameter          1 14.637  14.637  488.5089 < 2.2e-16 ***
## Height            1  3.313   3.313  110.5812 < 2.2e-16 ***
## M                 1  1.215   1.215   40.5375 2.231e-10 ***
## F                 1  1.865   1.865   62.2445 4.243e-15 ***
## Length            1  0.101   0.101    3.3848    0.0659 .
## Residuals      2937 88.001   0.030
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
bic_model
```

```
##
## Call:
## lm(formula = y_ts_data ~ Shell_weight + Shucked_weight + Diameter +
##     Height + M + F, data = as.data.frame(x_ts_data), subset = setdiff(rownames(abalone),
##     "2052"))
##
## Coefficients:
##    (Intercept)    Shell_weight  Shucked_weight        Diameter          Height
##        2.34599         0.15900        -0.21364         0.13741         0.08175
##              M               F
##        0.08946         0.07524
```

```r
anova(bic_model)
```

```
## Analysis of Variance Table
##
## Response: y_ts_data
##                  Df Sum Sq Mean Sq  F value    Pr(>F)
## Shell_weight      1 96.124  96.124 3205.502 < 2.2e-16 ***
## Shucked_weight    1 10.637  10.637  354.719 < 2.2e-16 ***
## Diameter          1 14.637  14.637  488.113 < 2.2e-16 ***
## Height            1  3.313   3.313  110.492 < 2.2e-16 ***
## M                 1  1.215   1.215   40.505 2.268e-10 ***
## F                 1  1.865   1.865   62.194 4.351e-15 ***
## Residuals      2938 88.102   0.030
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Testing for interactions for the other step BIC model
other_step_BIC_interact = lm(formula = log(Age) ~ (Shell_weight + Shucked_weight + Diameter + Sex + Hei
other_step_BIC_interact
```

```
##
## Call:
## lm(formula = log(Age) ~ (Shell_weight + Shucked_weight + Diameter +
##     Sex + Height)^2, data = abalone_t, subset = setdiff(rownames(abalone),
##     "2052"))
##
## Coefficients:
##                 (Intercept)                 Shell_weight
##                     1.33050                      4.71903
##              Shucked_weight                     Diameter
##                    -3.88071                      2.56561
##                        SexI                         SexM
##                    -0.10631                      0.02843
##                      Height  Shell_weight:Shucked_weight
##                     6.76381                     -0.76300
##       Shell_weight:Diameter            Shell_weight:SexI
##                    -8.50631                     -0.39398
##           Shell_weight:SexM          Shell_weight:Height
##                     0.23728                      7.43921
##     Shucked_weight:Diameter          Shucked_weight:SexI
##                     6.80771                      0.71117
##         Shucked_weight:SexM        Shucked_weight:Height
##                     0.06835                     -0.27602
##               Diameter:SexI                Diameter:SexM
##                    -0.41302                     -0.05828
##             Diameter:Height                  SexI:Height
##                   -16.63026                      0.56397
##                 SexM:Height
##                    -0.60729
```

```
#Interaction attempt
bic_intr_model = lm(formula = y_ts_data ~ . - Length - Viscera_weight + Shell_weight*Diameter + Shucked_

bic_intr_model
```

```
##
## Call:
## lm(formula = y_ts_data ~ . - Length - Viscera_weight + Shell_weight *
##     Diameter + Shucked_weight * Diameter, data = as.data.frame(x_ts_data),
##     subset = setdiff(rownames(abalone), "2052"))
##
## Coefficients:
##             (Intercept)                 Diameter                   Height
##                 2.38541                  0.06471                  0.06340
##          Shucked_weight             Shell_weight                        M
##                -0.25470                  0.28701                  0.07340
##                       F    Diameter:Shell_weight  Diameter:Shucked_weight
##                 0.06285                 -0.11625                  0.08443
```

```
anova(bic_intr_model)
```

```
## Analysis of Variance Table
##
## Response: y_ts_data
##                           Df Sum Sq Mean Sq  F value    Pr(>F)
## Diameter                   1 93.511  93.511 3291.580 < 2.2e-16 ***
## Height                        1  6.490   6.490  228.454 < 2.2e-16 ***
## Shucked_weight                1 14.283  14.283  502.766 < 2.2e-16 ***
## Shell_weight                  1 10.427  10.427  367.025 < 2.2e-16 ***
## M                             1  1.215   1.215   42.754 7.293e-11 ***
## F                             1  1.865   1.865   65.649 7.822e-16 ***
## Diameter:Shell_weight         1  2.521   2.521   88.744 < 2.2e-16 ***
## Diameter:Shucked_weight       1  2.172   2.172   76.446 < 2.2e-16 ***
## Residuals                  2936 83.409   0.028
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
par(mfrow=c(1,2))
plot(aic_model, which = c(1,2), sub.caption = '', main = 'AIC selected Model')
```

```r
plot(aic_model, which = c(4,5), sub.caption = '', main = 'AIC selected Model')
```

```r
plot(bic_model, which = c(1,2), sub.caption = '', main = 'BIC selected Model')
```

```r
plot(bic_model, which = c(4,5), sub.caption = '', main = 'BIC selected Model')
```

Ridge Regression GLMNET

```r
abalone.g = glmnet(x_t_data[-2502, ], y_t_data[-2502], alpha = 0)

#Finding optimal lambda value
crossv_model = cv.glmnet(x_ts_data[-2505, ], y_ts_data[-2505], alpha = 0)
lambda_min = crossv_model$lambda.min
lambda_min
```

```
## [1] 0.01805637
```

```r
par(mfrow=c(1,2))
plot(crossv_model)

#Using our best optimal values remake our model
abalone.g.best = glmnet(x_ts_data[-2505, ], y_ts_data[-2505], alpha = 0, lambda = lambda_min)
coef(abalone.g.best)
```

```
## 9 x 1 sparse Matrix of class "dgCMatrix"
##                         s0
## (Intercept)     2.34618945
## Length          0.05343607
## Diameter        0.07697804
## Height          0.04255503
## Shucked_weight -0.12527500
## Viscera_weight -0.01174074
## Shell_weight    0.12170395
## M               0.08500536
## F               0.07785860
```

```r
#Coefficients as lambda changes
plot(abalone.g)
abline(v = lambda_min, lty = 'dashed')
```

```r
#Using our ridge regression model predict our validation dataset
y.ridge_pred = predict(abalone.g, s = lambda_min, newx = x_v_data[-2502, ])
```

```
#Get coefficients of multiple determination for ridge regession
ssto = sum((y_v_data - mean(y_v_data))^2)
sse = sum((y.ridge_pred - y_v_data)^2)
c(ssto, sse)
```

```
## [1] 94.30146 41.09938
```

```
#Calculate R^2 and R_a^2 for ridge regression
r_squared = 1 - (sse / ssto)
r_adj_squared = 1 - ((nrow(x_v_data) - 1) / ((nrow(x_v_data) - length(coef(abalone.g.best))))) * (sse /
c(r_squared, r_adj_squared)
```

```
## [1] 0.5641702 0.5613170
```

```
#Ridge Regression RMSPE score vs SSE/n
rmspe(y_v_data, y.ridge_pred)^2
```

```
## [1] 0.03338699
```

```
sse / nrow(x_t_data)
```

```
## [1] 0.01395091
```

```
ridge_residuals = y_v_data - y.ridge_pred
plot(y.ridge_pred, ridge_residuals, xlab = 'Fitted Values', ylab = 'Residuals', main = 'Fitted Values v
```

Credit to Oliver & Johnnyheineken on Stat Exchange for providing AIC/BIC code for ridge regression: https://stackoverflow.com/questions/63171921/is-there-a-way-in-r-to-determine-aic-from-cv-glmnet https://stackoverflow.com/questions/40920051/r-getting-aic-bic-likelihood-from-glmnet

```
#Algorithm to calculate AIC & BIC for ridge regression
tLL = abalone.g.best$nulldev - deviance(abalone.g.best)

1 - abalone.g.best$dev.ratio
```

```
## [1] 0.4335771
```

```
k = abalone.g.best$df
n = abalone.g.best$nobs
aic.ridge = - tLL + 2 * k + 2 * k * (k + 1) / (n - k - 1)
bic.rdige = log(n) * k - tLL

aic.ridge
```

```
## [1] -106.1658
```

```
bic.rdige
```

```
## [1] -58.31198
```

###Dealing with Multi Collinearity

```
#Gathering the significant stats for all our models

press_aic = sum((aic_model$residuals/(1-influence(aic_model)$hat))^2)
press_bic = sum((bic_model$residuals/(1-influence(bic_model)$hat))^2)
press_intr_bic = sum((bic_intr_model$residuals/(1-influence(bic_intr_model)$hat))^2)

cp.ridge = ((sse * (nrow(x_t_data) - length(coef(abalone.g.best)))) / sum(full_mod$residuals^2)) - nrow
```

Note that BIC model has better VIF values

```r
#Comparison statistics for our three models

#AIC, BIC, R squared, R squared adj, Press p
abra_summary_df = data.frame(aic_model = c(AIC(aic_model), BIC(aic_model), summary(aic_model)$r.squared
                             bic_model = c(AIC(bic_model), BIC(bic_model), summary(bic_model)$r.squared
                             bic_intr_model = c(AIC(bic_intr_model), BIC(bic_intr_model), summary(bic_in
                             rgd_model = c(aic.ridge, bic.rdige, r_squared, r_adj_squared, NA)
                             )
rownames(abra_summary_df) = c('AIC', 'BIC', 'R^2', 'R adj', 'Press_p')
abra_summary_df
```

```
##              aic_model     bic_model bic_intr_model    rgd_model
## AIC     -1962.9275045 -1961.5354332  -2118.7381380 -106.1658467
## BIC     -1909.0367276 -1913.6325204  -2058.8594970  -58.3119803
## R^2         0.5923872     0.5919174      0.6136546    0.5641702
## R adj       0.5914157     0.5910840      0.6126019    0.5613170
## Press_p    88.6237899    88.6600832     84.4201486           NA
```

```r
#Mallows Cp vs model P
cp_summary_df = data.frame(aic_model = c(ols_mallows_cp(aic_model, full_mod), length(aic_model$coeffici
                           bic_model = c(ols_mallows_cp(bic_model, full_mod), length(bic_model$coeffici
                           bic_intr_model = c(ols_mallows_cp(bic_intr_model, full_mod), length(bic_intr_
                           rgd_model = c(cp.ridge, length(coef(abalone.g.best)))
                           )
rownames(cp_summary_df) = c('Mallows Cp', 'P')
cp_summary_df
```

```
##            aic_model bic_model bic_intr_model rgd_model
## Mallows Cp   7.59154  8.975876      -143.6267 -1556.044
## P            8.00000  7.000000         9.0000     9.000
```

```r
#RMSPE vs sqrt of SSE/n
rmspe_df = data.frame(aic_model = c(rmspe(y_vs_data, predict(aic_model, as.data.frame(x_vs_data))), sqr
                      bic_model = c(rmspe(y_vs_data, predict(bic_model, as.data.frame(x_vs_data))), sqr
                      bic_intr_model = c(rmspe(y_vs_data, predict(bic_intr_model, as.data.frame(x_vs_da
                      rgd_model = c(rmspe(y_v_data, y.ridge_pred), sqrt(sse / nrow(x_t_data)))
                      )
rownames(rmspe_df) = c('RMSPE', 'SSE / n')
rmspe_df
```

```
##         aic_model bic_model bic_intr_model rgd_model
## RMSPE   0.1787402 0.1787261      0.1737345 0.1827211
## SSE / n 0.1728331 0.1729327      0.1682639 0.1181140
```

###Comparing Re-fitting AIC & BIC

```r
#Refit our models on the validation data to see if any coefficients of regression change
aic_valid = lm(y_vs_data ~ Shell_weight + Shucked_weight + Diameter + Height + M + F + Length, data = as
bic_valid = lm(y_vs_data ~ Shell_weight + Shucked_weight + Diameter + Height + M + F, data = as.data.fra

#Summary table of model coefficients
kable(data.frame( training = coef(aic_model), validation = coef(aic_valid)), caption = 'Regression Coeff
```

Table 3: Regression Coefficients of AIC Model on Training & Validation Data

	training	validation
(Intercept)	2.3452026	2.3418195
Shell_weight	0.1593912	0.1471896
Shucked_weight	-0.2168834	-0.2066258
Diameter	0.1041007	0.1215048
Height	0.0810263	0.0868823
M	0.0906359	0.0704713
F	0.0763654	0.0826495
Length	0.0366482	0.0122072

```r
kable(data.frame( training = coef(bic_model), validation = coef(bic_valid)), caption = 'Regression Coef
```

Table 4: Regression Coefficients of BIC Model on Training & Validation Data

	training	validation
(Intercept)	2.3459893	2.3418503
Shell_weight	0.1589993	0.1467682
Shucked_weight	-0.2136394	-0.2056520
Diameter	0.1374115	0.1326265
Height	0.0817523	0.0874002
M	0.0894627	0.0704714
F	0.0752445	0.0825769

```r
#Summary table of model VIFs
kable(data.frame(aic_vif = vif(aic_model), c(bic_vif = vif(bic_model), NA)), col.names = c('AIC VIF', '
```

Table 5: VIF between AIC & BIC models

	AIC VIF	BIC VIF
Shell_weight	7.603133	7.598673
Shucked_weight	6.305246	5.999664
Diameter	41.122108	8.904444
Height	5.857915	5.845335
M	1.851566	1.842287
F	1.928241	1.920404
Length	38.996215	NA

https://statisticsbyjim.com/regression/multicollinearity-in-regression-analysis/

References:

Faller, et al. “R: Getting AIC/BIC/Likelihood from Glmnet.” *Stack Overflow*, 1 Dec. 2016, stackoverflow.com/questions/40920051/r-getting-aic-bic-likelihood-from-glmnet.

Frost, Jim. “Multicollinearity in Regression Analysis: Problems, Detection, and Solutions.” *Statistics By Jim*, 29 Jan. 2023, statisticsbyjim.com/regression/multicollinearity-in-regression-analysis/.

Thomas, and Oliver. “Is There a Way in R to Determine AIC from Cv.Glmnet?” *Stack Overflow*, 30 July 2020, stackoverflow.com/questions/63171921/is-there-a-way-in-r-to-determine-aic-from-cv-glmnet.

Zach. “Ridge Regression in R (Step-by-Step).” *Statology*, 13 Nov. 2020, www.statology.org/ridge-regression-in-r/.