Multiple Linear Regression Analysis: Predicting Age of Abalone

Shehran Syed - snsyed@ucdavis.edu

Lokesh Gorrela Krishna Reddy - lgorrelakrishnareddy@ucdavis.edu

Pamela Ruiz - pamruiz@ucdavis.edu

1. Abstract

We are using the data aggregated from Warwick J Nash, Tracy L Sellers, Simon R Talbot, Andrew J Cawthorn and Wes B Ford (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. We will use a multilinear regression model in order to predict the age of the abalone fish considering the physical measurements. The questions we hope to address are the following: Can measurements other than the number of rings be used to effectively predict the age of abalone? Which attributes are the most significant contributors in predicting the age of abalone? Can the age of abalone be predicted only using first-order predictors, or are higher order and/or interaction terms required for a better predictor? Can a good enough model for predicting age of abalone be designed to replace the need for manual counting of rings? In the end, we obtained a model with a standard error of prediction = 2.27 years, and a 95% prediction interval with an average width of 9.97 years.

2. Introduction

Abalone is a seafood traditionally caught in Oceania, the United States, Mexico and the Indo-Pacific region. The abalone live inside of a tough and heavy shell exterior on rocky outcrops in the ocean. The shellfish are camouflaged and difficult to access by hand which makes fishing abalone a timely and skillful task. Extracting the abalone meat is another challenge in itself, as this requires the skill and time which may result in only about 250 grams of meat for about a kilo of caught abalone including the shell. The economic value of the abalone is positively correlated with its age therefore in order to determine its price the age must be accurately estimated. Lastly though the abalone is especially valued in Southeast Asian cuisine, its popularity is growing quickly and as a result the abalone run a high risk of extinction. This in turn has led to a cap on the number of fish captured in some countries.

We will use our models to try to address one of the fisherman's challenging tasks: to determine the age of the abalone considering its physical characteristics. This modeling can provide a possible solution to a time consuming and skillful task, as well as provide more information regarding the maturity of an abalone upon capture using the physical characteristics. This could in turn impact

the fishing process as they will be able to use physical characteristics to predict age and capture abalone with particular physical characteristics that will lead to more economic value and possibly better/different/more sustainable fishing practices.

Our goal is to first conduct exploratory data analysis to assess linear assumptions between our predictor variables and the abalone age. After performing possible transformations for necessary variables, we will then split the data into two sets in order to assess precision of the model. Next, we will perform model fitting with second order and interaction terms to estimate abalone age considering the optimal model based on model criterion namely, AIC, BIC, Cp, R^2. Finally, we will use Pressp to assess the effectiveness of the model considering training and test data.

2.1 Data

The data was provided by Warwick J Nash, Tracy L Sellers, Simon R Talbot, Andrew J Cawthorn and Wes B Ford (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. The original database was aggregated by the Marine Resources Division in the Department of Primary Industry and FIsheries, Tasmania, Australia. The original dataset contained missing values, which were removed, and the ranges of the continuous values were scaled according to ANN (artificial neural network with a factor of 1/200). It was noted that additional information regarding the weather and location would be beneficial variables to consider in addition to the original data. This in turn could provide a more comprehensive understanding of how these variables impact the abalone age prediction when selecting a particular model. Table 1 explains the types of variables used in this paper.

3. Methods and Results

3.1 EDA

For our exploratory data analysis, we began by creating a series of plots to understand the relationships present in abalone data. First, we created a pairs plot that shows the distribution of each variable, bivariate scatter plots and correlation between the variables (fig1). We can clearly see linear and non-linear relationships between predictor variables. There is a very high positive correlation between several predictor variables as well. This positive high correlation causes problems in model fitting like: (a) regression coefficient estimates can swing wildly based on other independent variables, (b) multicollinearity reduces the precision of the estimated coefficients, which weakens the statistical power of our regression.

From the pairs plot, there seems to be some outliers in height, so we plotted a box plot and histograms shows the distribution of height. Without the outlier's height seems to be normally

distributed and is clearly seen in the histogram plot. Different sex/gender of abalone have similar distribution in the dataset (fig2), and sex - Male and Female are similarly distributed (fig3). So, while fitting a model, we could consider Infants as a class and the remaining genders a single class, since their distributions are similar. As such, we include a new column for the dummy variable Infant, in place of sex. We then randomly split our dataset of 4177 observation into a training data of 2088 observations and a testing data of 2089 observations.

3.2 Model Fitting and Model Selection

We started out fitting the basic model with all the first order terms.

Age = Infant + Length + Diameter + Height + Whole.Weight + Shucked.Weight +Viscera.weight + Shell.weight

We have checked for model violations paying special attention to *Residual vs Fitted value plots and Normal QQ plots*, with the above model and found heteroskedasticity and normality violations. So, we decided to perform a box cox transformation. Box Cox tests gave lambda = -0.30, which is close to -0.5, so we considered a 1/sqrt (Age) transformation for the response variable (fig4).

Next, we fit the model with transformed response variable, *Age* = 1/sqrt (Age)*. There are improvements in the linearity and normality violations. But from the basic EDA analysis we found that there is very high multicollinearity. We checked for VIF (Variance Inflation Factor) of all the predictor variables. Variables *Whole.weight* is highly correlated with other predictor variables (tab2). As there are several *weight* variables, we decided to explore a little bit more about the relationship between weights.

We have used F tests to check if any individual weights could be removed from the model. It appears that none of the weight variables can be individually dropped from the full model. This seems odd since Whole Weight should be a linear combination of the other individual weight components. We further explored the data to see if that appears to be the case. Upon further exploration, we found 79 observations where the whole weight was less than the sum of the individual weight components. This may have been due to data entry errors. We cannot drop such a large number of observations.

In order to find the best weight predictors, we started with a model with only the variables Infant, Length, Diameter, Height. To that, we added each of the individual weight variables and found Shucked Weight to be the most significant contributor to the model. After adding Shucked weight to the model, we added each of the remaining weight variables individually and found Shell Weight to be the most significant addition to the model. As such we landed on the model *1/sqrt (Age) = Infant + Length + Diameter + Shucked Weight + Shell Weight*. We came to the conclusion that Shucked Weight + Shell Weight are the most significant because they were improving the adjusted R_a^2 better than all the remaining combinations.

After addressing the multicollinearity of the weight variables, we addressed the multicollinearity in length and diameter. Since length was the least significant predictor in the previous model

(primary model/first order model), we decided to drop that. That appeared to reduce the multicollinearity in our model, and the VIF values were much improved (tab4).

The model diagnostic plots of said model appear to show no obvious departures of the model assumptions. However, there does seem to be slight nonlinearity in the residuals vs. fitted values plot. We will explore this further by looking at the residuals vs. predictors plot.

We also looked at the plots for residual vs. all two-way interaction terms. Those plots do not show any obvious need for interaction terms but to address the nonlinearity, we now fit a second-order polynomial model, and for completeness, we also include the two-way interactions of the quantitative variables.

The model is as follows: $1/sqrt\ (Age) \sim Infant + (Diameter + Height + Shucked.weight + Shell.weight)^2 + I(Diameter^2) + I(Height^2) + I(Shucked.weight^2) + I(Shell.weight^2)$

The model assumptions appear to hold much better in this model, and we consider it to be our full model moving forward.

3.3 Model Selection

We conducted an exhaustive subset selection using our full model. Used model selection criterion R_a^2, Mallow's C_p, BIC and AIC. All different selection criteria gave us different models.

R_a^2 - Model: 1/sqrt(Age) ~ Infant + Diameter + Height + Shucked.weight +Shell.weight + I(Diameter^2) + I(Height^2) + I(Shucked.weight^2) +Diameter:Shucked.weight +Height:Shucked.weight + Height:Shell.weight +Shucked.weight:Shell.weight

C_p - Model 2: 1/sqrt(Age) ~ Infant + Diameter + Shucked.weight +Shell.weight + I(Diameter^2) + I(Height^2) + I(Shucked.weight^2) +Diameter:Shucked.weight + Height:Shell.weight + Shucked.weight:Shell.weight

BIC - Model 3: 1/sqrt(Age) ~ Infant + Diameter + Shucked.weight +Shell.weight + I(Diameter^2) + I(Height^2) + Diameter:Shucked.weight +Height:Shell.weight

AIC - Model 4 : 1/sqrt(Age) ~ Infant + Diameter + Height + Shucked.weight +Shell.weight + I(Diameter^2) + I(Height^2) + I(Shucked.weight^2) +Diameter:Shucked.weight + Height:Shell.weight + Shucked.weight:Shell.weight

After subset selection, we performed Internal and external validation on the above models with training and test data. (Internal validation metrics). For all four models, SSE is close to $Press_p$, suggesting that there is no serious overfitting. However, The Cp value of Model 3 is much higher than p, suggesting that the model may have large bias. Otherwise, the validity of the models seems adequate. Keeping this in mind, we move forward to external validation with the four models (Diagnostics model1, model2, model3, model4).

From external validation on the test data we observed the following:

- One of the coefficients in Model 4 changed signs from training to validation, so we disregard that model.
- For Models 1, 2, and 3, the regression coefficients did not change signs between training and validation sets.
- Model 3 was the most consistent among the three in terms of the magnitude of coefficients, and model 2 was the least consistent.
- Models 1, 2, and 3 were all comparable in terms of MSPE. The MSPE of all three models were close to training MSE and PRESS/n. We note that these values of MSPE were computed with respect to 1/sqrt(Age). To choose our final model among these three, we will compare the MSPE computed on the actual predicted age.
- Since Model 1 gives the minimum validation MSPE (mspe) based on actual age, we go with Model 1 as our final model and fit it to the entire dataset.

Final Model:

$$\frac{1}{\sqrt{Age}} = 0.52 + 0.01 * Infant_1 - 0.91 * Diameter - 0.24 * Height + 0.39 * Shucked.weight - 0.48 * Shell.weight + 1.31 * Diameter^2 - 1.13 * Height^2 - 0.14 * Shucked.weight^2 - 0.66 * Diameter{:}Shucked.weight + 0.54 * Height{:}Shucked.weight + 0.66 * Height{:}Shell.weight + 0.28 * Shucked.weight{:}Shell.weight$$

After removing high leverage points and significant outliers, our model appears to satisfy the models assumptions quite well (Diagnostics of final model plot, coefficients).

4.Discussion and Conclusion:

From the results of our analysis, we found that due to the slight non-linear relationship between the age of abalone and its physical features, a model with second order polynomial and interaction terms were required to build a good predictor. From our final model, we find Sex = Infant, Diameter, Shucked Weight, and Shell Weight to be the most significant predictors of age. The higher order and interaction terms give our model more explanatory power but are not as significant as the main effect terms. In the end, we obtained a model with a standard error of prediction = 2.27 years, and a 95% prediction interval with an average width of 9.97 years (in terms of actual age of abalone computed from the validation set). This is the best model we could develop using the tools of linear regression.

The most significant limitation in our analysis was the high degree of multicollinearity in our data. We attempted to tackle this issue by removing highly intercorrelated variables to reduce the multicollinearity in our model and had some success in doing so. However, other more advanced methods like Principal Component Regression, and machine learning models may have resulted in a better model. We note those as potential approaches for future studies.

5 Appendices:

5.1 Images:

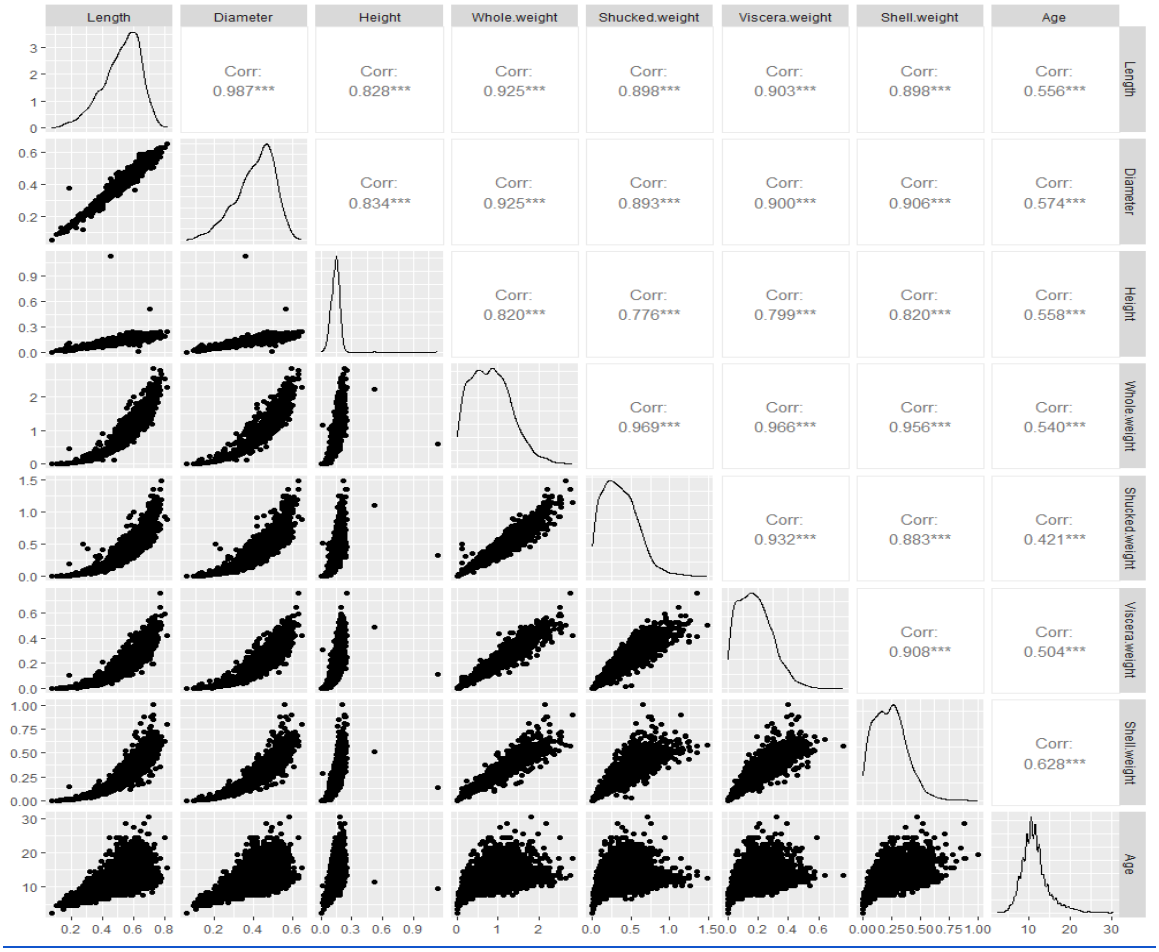


Figure 1: Pairs plot

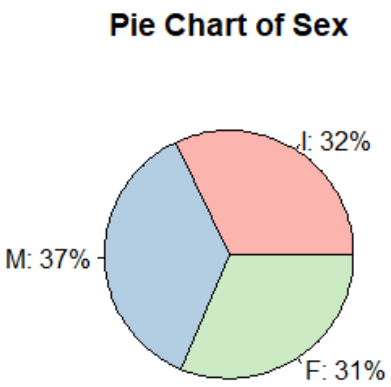


Figure 2: Chart for gender distribution

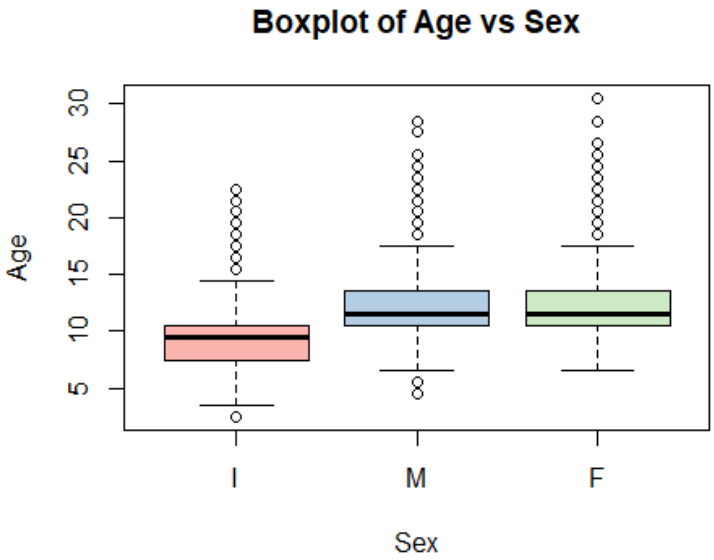
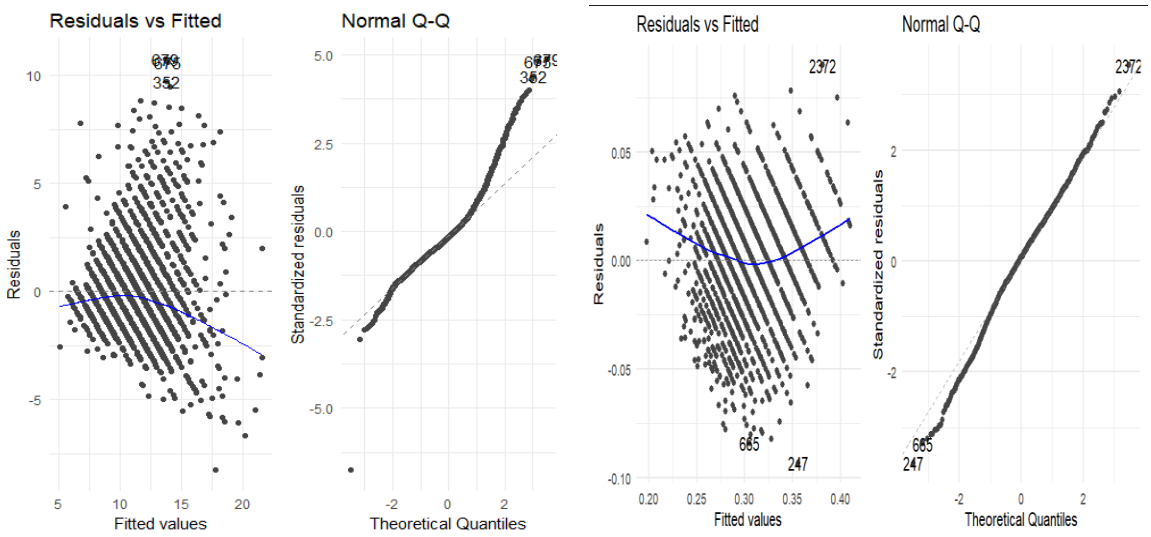


Figure 3: Data distribution of different genders



(a) (b)

Figure 4: Model plots before and after transformation of response variable.

	SSE	PRESS	MSE	PRESS/n	Adj.R.sq	MSPE
Training	1.208982	1.226635	0.000583	0.000588	0.65	NA
Validation	1.232858	NA	0.000594	NA	0.66	0.000614

	coef.train	coef.test	pcent.coef	se.train	se.test	pcent.se
(Intercept)	0.5236	0.5196	0.76	0.0128	0.0106	17.19
Infant1	0.0104	0.0110	5.77	0.0014	0.0014	0.00
Diameter	-0.9402	-0.8928	5.04	0.1122	0.0947	15.60
Height	-0.2723	-0.2254	17.22	0.1715	0.0932	45.66
Shucked.weight	0.5010	0.3104	38.04	0.0613	0.0476	22.35
Shell.weight	-0.5715	-0.4159	27.23	0.0586	0.0495	15.53
I(Diameter^2)	1.4506	1.2047	16.95	0.2160	0.1842	14.72
I(Height^2)	-1.6906	-0.5704	66.26	0.8222	0.5211	36.62
I(Shucked.weight^2)	-0.0877	-0.1754	100.00	0.0312	0.0318	1.92
Diameter:Shucked.weight	-0.9084	-0.4634	48.99	0.1898	0.1657	12.70
Height:Shucked.weight	0.3571	0.5629	57.63	0.2590	0.2784	7.49
Height:Shell.weight	1.3801	0.1446	89.52	0.4204	0.3160	24.83
Shucked.weight:Shell.weight	0.2478	0.3095	24.90	0.0741	0.0628	15.25

Figure 5: Comparing coefficients between test set and validation set for Model 1

	SSE	PRESS	MSE	PRESS/n	Adj.R.sq	MSPE
Training	1.211752	1.226393	0.000584	0.000588	0.65	NA
Validation	1.243601	NA	0.000599	NA	0.66	0.000648

	coef.train	coef.test	pcent.coef	se.train	se.test	pcent.se
(Intercept)	0.5199	0.5201	0.04	0.0127	0.0106	16.54
Infant1	0.0105	0.0114	8.57	0.0014	0.0014	0.00
Diameter	-1.0022	-0.9890	1.32	0.1049	0.0869	17.16
Shucked.weight	0.5351	0.3425	35.99	0.0577	0.0463	19.76
Shell.weight	-0.6478	-0.4647	28.26	0.0462	0.0467	1.08
I(Diameter^2)	1.5099	1.2750	15.56	0.2097	0.1741	16.98
I(Height^2)	-2.5906	-0.3119	87.96	0.3438	0.2588	24.72
I(Shucked.weight^2)	-0.0660	-0.1599	142.27	0.0291	0.0311	6.87
Diameter:Shucked.weight	-0.8804	-0.3976	54.84	0.1876	0.1557	17.00
Shell.weight:Height	1.9303	0.2426	87.43	0.3350	0.3020	9.85
Shucked.weight:Shell.weight	0.2156	0.3482	61.50	0.0724	0.0623	13.95

Figure 6: Comparing coefficients between test set and validation set for Model 2

	SSE	PRESS	MSE	PRESS/n	Adj.R.sq	MSPE
Training	1.218587	1.230137	0.000587	0.00059	0.65	NA
Validation	1.269015	NA	0.000610	NA	0.66	0.000668

	coef.train	coef.test	pcent.coef	se.train	se.test	pcent.se
(Intercept)	0.5204	0.5241	0.71	0.0099	0.0091	8.08
Infant1	0.0107	0.0119	11.21	0.0014	0.0014	0.00
Diameter	-0.9842	-1.0120	2.82	0.0685	0.0643	6.13
Shucked.weight	0.5209	0.3375	35.21	0.0345	0.0315	8.70
Shell.weight	-0.6607	-0.4709	28.73	0.0454	0.0470	3.52
I(Diameter^2)	1.4826	1.3682	7.72	0.1046	0.0978	6.50
I(Height^2)	-3.3237	-1.2408	62.67	0.2619	0.2098	19.89
Diameter:Shucked.weight	-0.8508	-0.5057	40.56	0.0684	0.0640	6.43
Shell.weight:Height	2.6692	1.3031	51.18	0.2481	0.2449	1.29

Figure 7: Comparing coefficients between test set and validation set for Model 3

	SSE	PRESS	MSE	PRESS/n	Adj.R.sq	MSPE
Training	1.210091	1.225354	0.000583	0.000587	0.65	NA
Validation	1.235288	NA	0.000595	NA	0.66	0.000619

	coef.train	coef.test	pcent.coef	se.train	se.test	pcent.se
(Intercept)	0.5233	0.5185	0.92	0.0128	0.0106	17.19
Infant1	0.0104	0.0111	6.73	0.0014	0.0014	0.00
Diameter	-0.9346	-0.8594	8.05	0.1122	0.0933	16.84
Height	-0.2887	-0.3107	7.62	0.1711	0.0831	51.43
Shucked.weight	0.5291	0.3335	36.97	0.0578	0.0462	20.07
Shell.weight	-0.6115	-0.4490	26.57	0.0509	0.0468	8.06
I(Diameter^2)	1.4268	1.1141	21.92	0.2154	0.1788	16.99
I(Height^2)	-1.3861	0.2865	120.67	0.7921	0.3037	61.66
I(Shucked.weight^2)	-0.0738	-0.1606	117.62	0.0295	0.0310	5.08
Diameter:Shucked.weight	-0.8682	-0.3486	59.85	0.1876	0.1557	17.00
Height:Shell.weight	1.6325	0.3340	79.54	0.3785	0.3020	20.21
Shucked.weight:Shell.weight	0.2419	0.3124	29.14	0.0740	0.0628	15.14

Figure 8: Comparing coefficients between test set and validation set for Model 4

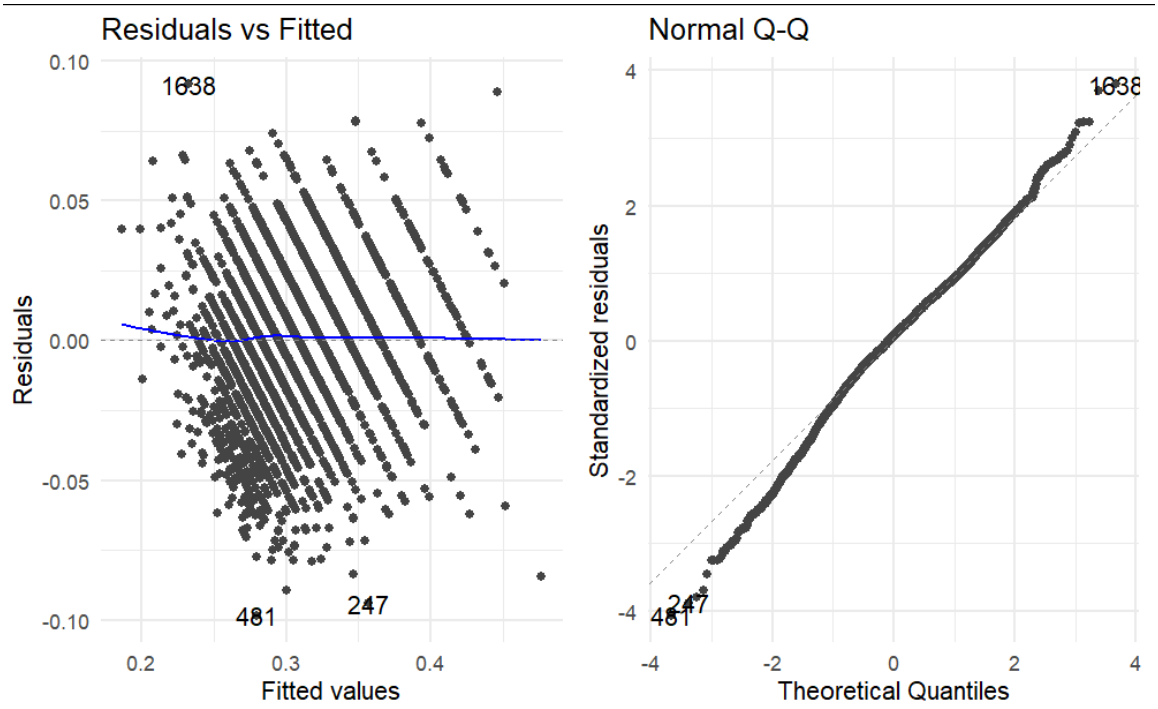


Figure 9: Final model plots

	MSPE1	MSPE2	MSPE3
Age	5.168737	75321.57	358.6284

Figure 10: MSPE Data

5. 2 Tables:

Dataset Description

Variable Name	Variable Type	Description (Units)
Sex	Qualitative	M, F, and I (infant)
Length	Quantitative	longest shell measurement(mm)
Diameter	Quantitative	perpendicular to length(mm)
Height	Quantitative	with meat in shell (mm)
Whole weight	Quantitative	Weight of whole abalone (grams)
Shucked weigh	Quantitative	weight of meat(grams)
Viscera weight	Quantitative	gut weight-after bleeding(grams)
Shell weight	Quantitative	after being dried(grams)
Rings: integer	Quantitative	+1.5 gives the age in years

Table 1: The Variable Description of the ‘abalone’ data set

Infant	1.534042
Length	42.723218
Diameter	43.254056
Height	6.799503
Whole.weight	113.337340
Shucked.weight	29.660170
Viscera.weight	18.020649
Shell.weight	22.897003

Table 2: VIF values of all the predictors

	Coefficient	2.5%	97.5%
(Intercept)	0.52008030	0.503974244	0.536186353
Infant1	0.01079368	0.008812048	0.012775311
Diameter	-0.90569353	-1.051264906	-0.760122148
Height	-0.24683271	-0.502974650	0.009309238
Shucked.weight	0.38542811	0.312172682	0.458683547
Shell.weight	-0.48252900	-0.560864432	-0.404193568
I(Diameter^2)	1.31207568	1.036136319	1.588015050
I(Height^2)	-1.13395741	-2.369216079	0.101301254
I(Shucked.weight^2)	-0.13939943	-0.182430497	-0.096368363
Diameter:Shucked.weight	-0.65858195	-0.899023695	-0.418140209
Height:Shucked.weight	0.53858066	0.173652108	0.903509202
Height:Shell.weight	0.65666259	0.102769137	1.210556053
Shucked.weight:Shell.weight	0.28333168	0.187110439	0.379552923

Table 3: 95% confidence interval of regression coefficients.

Infant	1.505514
Diameter	9.732181
Height	6.675547
Shucked.weight	6.011256
Shell.weight	7.980674

Table 4: VIF after remove values with high multicollinearity

Model	p	C_p	SSE	$Press_p$
Model 1	13	10.424	1.209	1.2266
Model 2	11	11.1678	1.2118	1.2264
Model 3	9	18.8719	1.2186	1.2301
Model 4	12	10.3221	1.2101	1.2254

Table 5: Internal Validation Metrics

5.3 R codes

Data set Information: https://archive.ics.uci.edu/ml/datasets/abalone

```
# Reading the data
data = read.csv('abalone.csv')
abalone = data
data$Sex = factor(data$Sex, levels = c("I", "M", "F"))
```

EDA

```
sapply(data, class)
summary(data)
```

Observations from data summary:

- Minimum height of abalone is 0 in the dataset. That is highly implausible. Let's explore that further.

```
data[c(which(data$Height==0)), ]
```

- The heights of the above two abalones were most likely incorrectly recorded, since their other measurements suggest it's not possible for the height to be 0. We will set these values of height as 'NA' and drop the corresponding observations.

```
data$Height[which(data$Height==0)] = NA
data = na.omit(data)
# creating response variable Age from rings column
data$Rings = data$Rings+1.5
names(data)[9] = "Age"
hist(data$Age, main="Histogram of Age", xlab = "Age")
```

- The distribution of the target variable Age appears slightly right tailed. We will keep this in consideration when deciding if any transformations are required after preliminary model fitting.

```
# plotting histograms to understand the distribution of all the predictor variables
par(mfrow = c(2, 2))
for(i in 2:8) {
hist(data[, i], main=paste("Histogram of", names(data)[i]), xlab = paste(names(data)[i]))}
```

- Length and Diameter appear to have left tailed distributions.
- All variables related to weight appear to have right tailed distributions.
- The histogram of Height suggests a highly irregular distribution. We need to explore this further.

```
boxplot(data$Height)
```

- Two observations of Height appear to be much larger than the upper fence value. We will now observe the distribution of Height without these two observations.

```
Height_outlier=which(data$Height>0.4)
par(mfrow=c(1,2))
hist(data$Height, main="Histogram of Height")
hist(data$Height[-Height_outlier], main="Histogram of Height Without Outliers")
```

- Ignoring the said outliers, Height appears to follow a mostly normal distribution.
- We will decide if anything needs to be done to deal with these observations after some preliminary model fitting. If either, or both, of these observations turn out out be high leverage points, we will have to handle them accordingly.

```
# creating pairs plot to understand the distributions, bivariate relationships and correlation between predictor variables.
pairplot = ggpairs(data[-1], progress = FALSE)
pairplot
# Creating pie chart to understand the distribution of sex in the dataset
n = nrow(data)
lab_Sex = levels(data$Sex)
pcent_Sex = round(100*table(data$Sex)/n)
lab_Sex = paste0(lab_Sex, ": ", pcent_Sex, "%")

par(mfrow = c(1, 1))
pie(table(data$Sex), labels = lab_Sex, main = "Pie Chart of Sex", col = palette.colors(palette = "Pastel 1"))
```

```
# creating box plot to understand distribution of different genders
boxplot(data$Age~data$Sex, main = "Boxplot of Age vs Sex", xlab = "Sex", ylab = "Age", col = palette.colors(palette = "Pastel 1"))
```

```
# creating box plots to understand how different sex/gender is distributed with all other predictor variables
par(mfrow = c(2, 2))
for(i in 2:8) {
boxplot(data[, i]~data$Sex, main=paste("Boxplot of", names(data)[i], "vs. Sex"), xlab = "Sex", ylab = paste(names(data)[i]), col = palette.colors(palette = "Pastel 1"))}
# Since M and F has similar distribution with other variables. We are creating a new predictor named Infant that has value 1 and 0 for classes M and F.
data$Sex = ifelse(data$Sex=="I", 1, 0)
names(data)[1] = "Infant"
data$Infant = factor(data$Infant)
# splitting the dataset into test and train.
set.seed(1)
n = nrow(data)/2
split = sample(1:(2*n), n, replace = F)
```

```
train = data[split, ]
test = data[-split, ]
# Checking whether the distribution of all the variables in the test and train sets is similar or not.
par(mfrow = c(2, 2))

for (i in 2:ncol(data)) {
  boxplot(train[, i], test[, i],
    col = palette.colors(palette = "Pastel 1"),
    main = paste("Boxplot of", names(data)[i]),
    ylab = paste(names(data)[i])
  )
  axis(1, at = c(1, 2), labels = c('Training', 'Validation'))
}
```

```
par(mfrow = c(1,2))

n = nrow(train)
lab_Infant = levels(train$Infant)
pcent_Infant = round(100*table(train$Infant)/n)
lab_Infant = paste0(lab_Infant, ": ", pcent_Infant, "%")

pie(table(train$Infant), labels = lab_Infant, main = "Pie Chart of Infant in Train", col =
palette.colors(palette = "Pastel 1"))

n = nrow(test)
lab_Infant = levels(test$Infant)
pcent_Infant = round(100*table(test$Infant)/n)
lab_Infant = paste0(lab_Infant, ": ", pcent_Infant, "%")

pie(table(train$Infant), labels = lab_Infant, main = "Pie Chart of Infant in Validation", col =
palette.colors(palette = "Pastel 1"))
```

Model Fitting

```
# fitting the first order model with all the predictor variables
fit1=lm(Age~., train)
summary(fit1)
# model plots and box cox transformation check
par(mfrow = c(2,2))
autoplot(fit1, which = 1:3) + theme_minimal()
boxcox(fit1)
# High leverage point
index = which(rownames(train) == "2052")
train[index, ]
round(sapply(train, rank)[index,-1]/nrow(train), 2)
train$Height[index] = 0.113
```

```
# Plots to check, if there is any non linearity with predictors using residuals vs each predictor plot
par(mfrow=c(2,2))
for (i in c(3:9)){
  plot(fit1$model[,i], fit1$residuals,
       main = paste("Residual vs",names(fit1$model)[i]),
       xlab = paste(names(fit1$model)[i]),
       ylab = paste("Residuals"))
}
```

```
# Plots to check, if there is any non linearity with interaction terms using residuals vs interaction terms plot
par(mfrow = c(2,3))
for (i in 3:8){
  for (j in (i+1):9){
    plot(fit1$model[, i]*fit1$model[, j], fit1$residuals,
    main = paste0("Residuals vs ", names(fit1$model)[i],"*",names(fit1$model)[j]),
    xlab = paste0(names(fit1$model)[i],"*",names(fit1$model)[j]),
    ylab = "Residuals"
    )
  }
}
```

There doesn't appear to be any need for higher order or interaction terms.

Let's define our full model now

```
# As there is some non linearity in the residuals vs fitted values plot, we are performing a box cox to find the best transformation on the response variable.
bc=boxcox(fit1)
bc$x[which.max(bc$y)]
## [1] -0.3030303
# we are considering a 1/sq-rt(Age) as the transformed variable as per the lambda value from box cox
fit1.3 = lm(1/sqrt(Age)~., train)
summary(fit1.3)
par(mfrow = c(2,2))
autoplot(fit1.3, which = c(1:2,5)) + theme_minimal()
boxcox(fit1.3)
vif(fit1.3)
##        Infant        Length      Diameter        Height  Whole.weight
##      1.535527      42.903502      43.514373       6.824037     113.482626
## Shucked.weight Viscera.weight   Shell.weight
##      29.689586      18.035240      22.908606
```

There appears to be 1 significant outlier (observation 237). We will refit model 1.3 after removing said observation.

```
index = which(rownames(train) == "237")
train = train[-index, ]
fit1.3 = lm(1/sqrt(Age)~., train)
summary(fit1.3)
anova(fit1.3)
par (mfrow = c(2,2))
autoplot(fit1.3, which = c(1:2,5)) + theme_minimal()
boxcox(fit1.3)
vif(fit1.3)
par(mfrow=c(2,2))
for (i in c(3:9)){
  plot(fit1.3$model[,i], fit1.3$residuals,
       main = paste("Residual vs",names(fit1.3$model)[i]),
       xlab = paste(names(fit1.3$model)[i]),
       ylab = paste("Residuals"))
}
# check variance inflation factor between predictors
vif(fit1.3)
#As weight parameters are highly correlated, we are trying to find the best weight parameters that could be used in the model
fit2.1 = lm(1/sqrt(Age)~. - Whole.weight, train)
anova(fit1.3, fit2.1)
fit2.2 = lm(1/sqrt(Age)~. - Shucked.weight, train)
anova(fit1.3, fit2.2)
fit2.3 = lm(1/sqrt(Age)~. - Viscera.weight, train)
anova(fit1.3, fit2.3)
fit2.4 = lm(1/sqrt(Age)~. - Shell.weight, train)
anova(fit1.3, fit2.4)
```

It appears that none of the weight variables can be individually dropped from the full model. This seems odd since Whole Weight should be a linear combination of the other individual weight components. We can try to explore the data to see if that appears to be the case.

```
par(mfrow = c(2,2))
plot(train$Whole.weight, (train$Shucked.weight+train$Shell.weight+train$Viscera.weight),
abline(0,1))
plot(train$Whole.weight, train$Shucked.weight, abline(0,1))
plot(train$Whole.weight, train$Shell.weight, abline(0,1))
plot(train$Whole.weight, train$Viscera.weight, abline(0,1))
```

From the data description, intuitively, whole weight can not be less than shucked weight + shell weight. There are clearly some cases where shucked weight + shell weight are greater than whole weight. Those may be errors and we need to explore those further.

```
index = which(train$Whole.weight <
(train$Shucked.weight+train$Shell.weight+train$Viscera.weight))
train[index, ]
```

```
fit3.1 = lm(1/sqrt(Age) ~ Infant + Length + Diameter + Height + Whole.weight, train)
fit3.2 = lm(1/sqrt(Age) ~ Infant + Length + Diameter + Height + Shucked.weight, train)
fit3.3 = lm(1/sqrt(Age) ~ Infant + Length + Diameter + Height + Viscera.weight, train)
fit3.4 = lm(1/sqrt(Age) ~ Infant + Length + Diameter + Height + Shell.weight, train)

summary(fit3.1)
anova(fit3.1)
summary(fit3.2)
summary(fit3.3)
anova(fit3.3)
summary(fit3.4)
anova(fit3.4)
fit4.1 = lm(1/sqrt(Age) ~ Infant + Length + Diameter + Height + Shucked.weight +
Whole.weight, train)
fit4.2 = lm(1/sqrt(Age) ~ Infant + Length + Diameter + Height + Shucked.weight +
Viscera.weight, train)
fit4.3 = lm(1/sqrt(Age) ~ Infant + Length + Diameter + Height + Shucked.weight + Shell.weight,
train)

summary(fit4.1)
anova(fit4.1)
summary(fit4.2)
anova(fit4.2)
summary(fit4.3)
anova(fit4.3)
vif(fit4.1)
vif(fit4.2)
vif(fit4.3) # best so far
##        Infant        Length      Diameter        Height Shucked.weight
##      1.512374      42.156352      43.144506       6.727005       6.369325
##   Shell.weight
##      7.986173
fit4.4 = lm(1/sqrt(Age) ~ Infant + Diameter + Height + Shucked.weight + Shell.weight, train)
summary(fit4.4)
anova(fit4.3, fit4.4)
vif(fit4.4)
##        Infant      Diameter        Height Shucked.weight   Shell.weight
##      1.505514      9.732181       6.675547       6.011256       7.980674
par(mfrow = c(2,2))
autoplot(fit4.4, which = c(1:3, 5)) + theme_minimal()
boxcox(fit4.4)
par(mfrow=c(2,2))
for (i in c(3:6)){
  plot(fit4.4$model[,i], fit4.4$residuals,
       main = paste("Residual vs",names(fit4.4$model)[i]),
       xlab = paste(names(fit4.4$model)[i]),
```

```
      ylab = paste("Residuals"))
}
```

```
par(mfrow = c(2,3))
for (i in 3:5){
  for (j in (i+1):6){
    plot(fit4.4$model[, i]*fit4.4$model[, j], fit4.4$residuals,
    main = paste0("Residuals vs ", names(fit4.4$model)[i],"*",names(fit4.4$model)[j]),
    xlab = paste0(names(fit4.4$model)[i],"*",names(fit4.4$model)[j]),
    ylab = "Residuals"
    )
  }
}
```

```
fit4.5 = lm(1/sqrt(Age) ~ Infant + (Diameter + Height + Shucked.weight + Shell.weight)^2 +
I(Diameter^2) + I(Height^2) + I(Shucked.weight^2) + I(Shell.weight^2), train)
summary(fit4.5)
anova(fit4.5)
par(mfrow = c(2,2))
autoplot(fit4.5, which = c(1:3, 5)) + theme_minimal()
boxcox(fit4.5)
par(mfrow=c(2,3))
for (i in c(3:10)){
  plot(fit4.5$model[,i], fit4.5$residuals,
       main = paste("Residual vs",names(fit4.5$model)[i]),
       xlab = paste(names(fit4.5$model)[i]),
       ylab = paste("Residuals"))
}
```

Model Selection

```
modelF = fit4.5
# Performing an exhaustive search using regsubsets to find the best model
subsets = regsubsets(1/sqrt(Age) ~ Infant + (Diameter + Height + Shucked.weight +
Shell.weight)^2 + I(Diameter^2) + I(Height^2) + I(Shucked.weight^2) + I(Shell.weight^2), train,
nbest = 1, nvmax = 15, method = 'exhaustive')
sum_sub = summary(subsets)
sum_sub
# calculating model selection criterion for all the models suggested by regsubsets.
n = nrow(train)
p.m = rowSums(sum_sub$which)
ssto = sum((1/sqrt(train$Age) - mean(1/sqrt(train$Age))) ^ 2)
sse = (1 - sum_sub$rsq) * ssto
aic = n * log(sse / n) + 2 * p.m
bic = n * log(sse / n) + log(n) * p.m
```

```
res_sub = cbind(sum_sub$which, sse, sum_sub$rsq, sum_sub$adjr2, sum_sub$cp, bic, aic)

fit0 = lm(1/sqrt(Age) ~ 1, data = train)
sse0 = sum(fit0$residuals ^ 2)
p0 = 1
c0 = sse0 / summary(fit1)$sigma ^ 2 - (n - 2 * p0)
aic0 = n * log(sse0 / n) + 2 * p0
bic0 = n * log(sse0 / n) + log(n) * p0
none = c(1, rep(0, 15),sse0, 0, 0, c0, bic0, aic0)

res_sub = rbind(none, res_sub)
colnames(res_sub) = c(colnames(sum_sub$which), "sse", "R^2", "R^2_a", "Cp", "bic", "aic")

res_sub = round(res_sub, 4)
res_sub = cbind(p=c(1, p.m), res_sub)

for (i in 19:20) {
  ind = which(res_sub[, i] == max(as.numeric(res_sub[, i])))
  res_sub[ind, i] = paste0(res_sub[ind, i], '*')
}

for (i in c(18, 22:23)) {
  ind = which(res_sub[, i] == min(as.numeric(res_sub[, i])))
  res_sub[ind, i] = paste0(res_sub[ind, i], '*')
}

ind = which(abs(as.numeric(res_sub[-16,1]) - as.numeric(res_sub[-16, 21])) ==
min(abs(as.numeric(res_sub[-16,1]) - as.numeric(res_sub[-16, 21]))))
res_sub[ind, 21] = paste0(res_sub[ind, 21], '*')

res_sub = noquote(res_sub)
res_sub
##    p  (Intercept) Infant1 Diameter Height Shucked.weight Shell.weight
##    1  1         0      0        0      0             0
## 1  2  1         0      0        1      0             0
## 2  3  1         0      0        0      0             1
## 3  4  1         0      0        1      0             1
## 4  5  1         0      0        1      0             1
## 5  6  1         0      0        1      1             1
## 6  7  1         1      0        1      1             1
## 7  8  1         1      1        1      1             1
## 8  9  1         1      1        0      1             1
## 9  10 1         1      1        0      1             1
## 10 11 1         1      1        0      1             1
## 11 12 1         1      1        1      1             1
## 12 13 1         1      1        1      1             1
```

```
## 13 14 1          1      1      1     1            1
## 14 15 1          1      1      1     1            1
## 15 16 1          1      1      1     1            1
##    I(Diameter^2) I(Height^2) I(Shucked.weight^2) I(Shell.weight^2)
##    0             0           0                   0
## 1  0             0           0                   0
## 2  0             0           0                   0
## 3  0             0           0                   0
## 4  0             0           1                   0
## 5  0             0           1                   0
## 6  0             0           1                   0
## 7  0             0           1                   0
## 8  1             1           0                   0
## 9  1             1           0                   0
## 10 1             1           1                   0
## 11 1             1           1                   0
## 12 1             1           1                   0
## 13 1             1           1                   0
## 14 1             1           1                   1
## 15 1             1           1                   1
##    Diameter:Height Diameter:Shucked.weight Diameter:Shell.weight
##    0               0                       0
## 1  0               0                       0
## 2  0               0                       0
## 3  0               0                       0
## 4  0               0                       0
## 5  0               0                       0
## 6  0               0                       0
## 7  0               0                       1
## 8  0               1                       0
## 9  0               1                       0
## 10 0               1                       0
## 11 0               1                       0
## 12 0               1                       0
## 13 1               1                       0
## 14 1               1                       0
## 15 1               1                       1
##    Height:Shucked.weight Height:Shell.weight Shucked.weight:Shell.weight
##    0                     0                   0
## 1  0                     0                   0
## 2  0                     0                   1
## 3  1                     0                   0
## 4  1                     0                   0
## 5  0                     1                   0
## 6  0                     1                   0
## 7  0                     0                   0
```

```
## 8  0                 1                 0
## 9  0                 1                 1
## 10 0                  1                 1
## 11 0                  1                 1
## 12 1                  1                 1
## 13 1                  1                 1
## 14 1                  1                 1
## 15 1                  1                 1
##    sse     R^2     R^2_a   Cp        bic          aic
##    3.5064  0       0       -2083.2816 -13318.5966  -13324.2396
## 1  1.8344  0.4768  0.4766  1059.4357  -14662.432   -14673.718
## 2  1.5863  0.5476  0.5472  636.5336   -14957.9333  -14974.8623
## 3  1.4565  0.5846  0.584   416.3721   -15128.2671  -15150.8391
## 4  1.3507  0.6148  0.6141  237.0427   -15278.0611  -15306.2761
## 5  1.2949  0.6307  0.6298  143.4807   -15358.4309  -15392.2889
## 6  1.2666  0.6388  0.6377  97.1317    -15396.7733  -15436.2743
## 7  1.2423  0.6457  0.6445  57.4202    -15429.6338  -15474.7779
## 8  1.2186  0.6525  0.6511  18.8719    -15462.1335* -15512.9206
## 9  1.2147  0.6536  0.6521  14.2903    -15461.0797  -15517.5097
## 10 1.2118  0.6544  0.6528  11.1678*   -15458.5796  -15520.6527
## 11 1.2101  0.6549  0.6531  10.3221    -15453.7992  -15521.5152*
## 12 1.209   0.6552  0.6532* 10.424     -15448.0677  -15521.4267
## 13 1.2088  0.6552  0.6531  12.1702    -15440.6803  -15519.6824
## 14 1.2088  0.6553* 0.6529  14.0345    -15433.1741  -15517.8192
## 15 1.2087* 0.6553* 0.6528  16         -15425.5659  -15515.8539
```

Results of best subset selection

By Ra2:

```
model1 = lm(1/sqrt(Age) ~ Infant + Diameter + Height + Shucked.weight + Shell.weight +
I(Diameter^2) + I(Height^2) + I(Shucked.weight^2) + Diameter:Shucked.weight +
Height:Shucked.weight + Height:Shell.weight + Shucked.weight:Shell.weight, train)
#summary(model1)
model1
```

By Cp:

```
model2 = lm(1/sqrt(Age) ~ Infant + Diameter + Shucked.weight + Shell.weight + I(Diameter^2)
+ I(Height^2) + I(Shucked.weight^2) + Diameter:Shucked.weight + Height:Shell.weight +
Shucked.weight:Shell.weight, train)
#summary(model2)
model2
```

By BIC:

```
model3 = lm(1/sqrt(Age) ~ Infant + Diameter + Shucked.weight + Shell.weight + I(Diameter^2)
+ I(Height^2) + Diameter:Shucked.weight + Height:Shell.weight, train)
```

```
#summary(model3)
model3
```

By AIC:

```
model4 = lm(1/sqrt(Age) ~ Infant + Diameter + Height + Shucked.weight + Shell.weight +
I(Diameter^2) + I(Height^2) + I(Shucked.weight^2) + Diameter:Shucked.weight +
Height:Shell.weight + Shucked.weight:Shell.weight, train)
#summary(model4)
model4
summary(model1)
anova(model1)
par(mfrow = c(2,2))
autoplot(model1, which = c(1:3, 5)) + theme_minimal()
boxplot(model1$residuals, main = "Residuals Boxplot")
summary(model2)
anova(model2)
par(mfrow = c(2,2))
autoplot(model2, which = c(1:3, 5)) + theme_minimal()
boxplot(model2$residuals, main = "Residuals Boxplot")
anova(model3)par(mfrow = c(2,2))



boxplot(model3$residuals, main = "Residuals Boxplot")
```

```
par(mfrow = c(2,2))
autoplot(model4, which = c(1:3, 5)) + theme_minimal()
```

```
boxplot(model4$residuals, main = "Residuals Boxplot")
```

Validation

Internal Validation

```
paste("Model 1:", "p =", length(coef(model1)),
      "Cp =", round(ols_mallows_cp(model1, modelF),4),
      "SSE =", round(anova(model1)["Residuals",2],4),
      "Press =", round(ols_press(model1),4))
## [1] "Model 1: p = 13 Cp = 10.424 SSE = 1.209 Press = 1.2266"
paste("Model 2:", "p =", length(coef(model2)),
      "Cp =", round(ols_mallows_cp(model2, modelF),4),
      "SSE =", round(anova(model2)["Residuals",2],4),
      "Press =", round(ols_press(model2),4))
## [1] "Model 2: p = 11 Cp = 11.1678 SSE = 1.2118 Press = 1.2264"
paste("Model 3:", "p =", length(coef(model3)),
      "Cp =", round(ols_mallows_cp(model3, modelF),4),
```

```
    "SSE =", round(anova(model3)["Residuals",2],4),
    "Press =", round(ols_press(model3),4))
## [1] "Model 3: p = 9 Cp = 18.8719 SSE = 1.2186 Press = 1.2301"
paste("Model 4:", "p =", length(coef(model4)),
    "Cp =", round(ols_mallows_cp(model4, modelF),4),
    "SSE =", round(anova(model4)["Residuals",2],4),
    "Press =", round(ols_press(model4),4))
## [1] "Model 4: p = 12 Cp = 10.3221 SSE = 1.2101 Press = 1.2254"
```

External Validation

Model 1

```
model1.ev = lm(model1, test)
coef.train = round(coef(model1), 4)
se.train = round(summary(model1)$coefficients[,"Std. Error"], 4)
coef.test = round(coef(model1.ev), 4)
se.test = round(summary(model1.ev)$coefficients[,"Std. Error"], 4)

pcent.coef = round(abs((coef.train - coef.test)/coef.train)*100, 2)
pcent.se = round(abs((se.train - se.test)/se.train)*100, 2)

cbind(coef.train, coef.test, pcent.coef, se.train, se.test, pcent.se)
Training = c("SSE" = round(anova(model1)["Residuals",2],6),
          "PRESS" = round(ols_press(model1),6),
          "MSE" = round(anova(model1)["Residuals",3],6),
          "PRESS/n" = round(ols_press(model1)/nrow(train),6),
          "Adj.R.sq" = round(summary(model1)$adj.r.sq, 2),
          "MSPE" = NA)

pred = predict.lm(model1, test[-9])
true = 1/sqrt(test[9])
sq.diff = (pred-true)^2
Validation = c("SSE" = round(anova(model1.ev)["Residuals",2],6),
          "PRESS" = NA,
          "MSE" = round(anova(model1.ev)["Residuals",3],6),
          "PRESS/n" = NA,
          "Adj.R.sq" = round(summary(model1.ev)$adj.r.sq, 2),
          "MSPE" = round(sapply(sq.diff, mean),6))

rbind(Training, Validation)
##              SSE    PRESS      MSE  PRESS/n Adj.R.sq     MSPE
## Training   1.208982 1.226635 0.000583 0.000588     0.65       NA
## Validation 1.232858       NA 0.000594       NA     0.66 0.000614
```

Model 2

```
model2.ev = lm(model2, test)
coef.train = round(coef(model2), 4)
```

```
se.train = round(summary(model2)$coefficients[,"Std. Error"], 4)
coef.test = round(coef(model2.ev), 4)
se.test = round(summary(model2.ev)$coefficients[,"Std. Error"], 4)

pcent.coef = round(abs((coef.train - coef.test)/coef.train)*100, 2)
pcent.se = round(abs((se.train - se.test)/se.train)*100, 2)

cbind(coef.train, coef.test, pcent.coef, se.train, se.test, pcent.se)
Training = c("SSE" = round(anova(model2)["Residuals",2],6),
             "PRESS" = round(ols_press(model2),6),
             "MSE" = round(anova(model2)["Residuals",3],6),
             "PRESS/n" = round(ols_press(model2)/nrow(train),6),
             "Adj.R.sq" = round(summary(model2)$adj.r.sq, 2),
             "MSPE" = NA)

pred = predict.lm(model2, test[-9])
true = 1/sqrt(test[9])
sq.diff = (pred-true)^2
Validation = c("SSE" = round(anova(model2.ev)["Residuals",2],6),
             "PRESS" = NA,
             "MSE" = round(anova(model2.ev)["Residuals",3],6),
             "PRESS/n" = NA,
             "Adj.R.sq" = round(summary(model2.ev)$adj.r.sq, 2),
             "MSPE" = round(sapply(sq.diff, mean),6))

rbind(Training, Validation)
##               SSE    PRESS      MSE  PRESS/n Adj.R.sq     MSPE
## Training   1.211752 1.226393 0.000584 0.000588     0.65       NA
## Validation 1.243601       NA 0.000599       NA     0.66 0.000648
```

Model 3

```
model3.ev = lm(model3, test)
coef.train = round(coef(model3), 4)
se.train = round(summary(model3)$coefficients[,"Std. Error"], 4)
coef.test = round(coef(model3.ev), 4)
se.test = round(summary(model3.ev)$coefficients[,"Std. Error"], 4)

pcent.coef = round(abs((coef.train - coef.test)/coef.train)*100, 2)
pcent.se = round(abs((se.train - se.test)/se.train)*100, 2)

cbind(coef.train, coef.test, pcent.coef, se.train, se.test, pcent.se)
Training = c("SSE" = round(anova(model3)["Residuals",2],6),
             "PRESS" = round(ols_press(model3),6),
             "MSE" = round(anova(model3)["Residuals",3],6),
             "PRESS/n" = round(ols_press(model3)/nrow(train),6),
             "Adj.R.sq" = round(summary(model3)$adj.r.sq, 2),
             "MSPE" = NA)
```

```
pred = predict.lm(model3, test[-9])
true = 1/sqrt(test[9])
sq.diff = (pred-true)^2
Validation = c("SSE" = round(anova(model3.ev)["Residuals",2],6),
          "PRESS" = NA,
          "MSE" = round(anova(model3.ev)["Residuals",3],6),
          "PRESS/n" = NA,
          "Adj.R.sq" = round(summary(model3.ev)$adj.r.sq, 2),
          "MSPE" = round(sapply(sq.diff, mean),6))

rbind(Training, Validation)
##               SSE    PRESS      MSE PRESS/n Adj.R.sq     MSPE
## Training   1.218587 1.230137 0.000587 0.00059     0.65       NA
## Validation 1.269015       NA 0.000610      NA     0.66 0.000668
```

####Model 4

```
model4.ev = lm(model4, test)
coef.train = round(coef(model4), 4)
se.train = round(summary(model4)$coefficients[,"Std. Error"], 4)
coef.test = round(coef(model4.ev), 4)
se.test = round(summary(model4.ev)$coefficients[,"Std. Error"], 4)

pcent.coef = round(abs((coef.train - coef.test)/coef.train)*100, 2)
pcent.se = round(abs((se.train - se.test)/se.train)*100, 2)

cbind(coef.train, coef.test, pcent.coef, se.train, se.test, pcent.se)
Training = c("SSE" = round(anova(model4)["Residuals",2],6),
          "PRESS" = round(ols_press(model4),6),
          "MSE" = round(anova(model4)["Residuals",3],6),
          "PRESS/n" = round(ols_press(model4)/nrow(train),6),
          "Adj.R.sq" = round(summary(model4)$adj.r.sq, 2),
          "MSPE" = NA)

pred = predict.lm(model4, test[-9])
true = 1/sqrt(test[9])
sq.diff = (pred-true)^2
Validation = c("SSE" = round(anova(model4.ev)["Residuals",2],6),
          "PRESS" = NA,
          "MSE" = round(anova(model4.ev)["Residuals",3],6),
          "PRESS/n" = NA,
          "Adj.R.sq" = round(summary(model4.ev)$adj.r.sq, 2),
          "MSPE" = round(sapply(sq.diff, mean),6))

rbind(Training, Validation)
```

```
##              SSE    PRESS      MSE  PRESS/n Adj.R.sq     MSPE
## Training   1.210091 1.225354 0.000583 0.000587     0.65       NA
## Validation 1.235288       NA 0.000595       NA     0.66 0.000619
```

```
AgeP1 = round((1 / predict.lm(model1, test[-9]))^2, 2)
AgeP2 = round((1 / predict.lm(model2, test[-9]))^2, 2)
AgeP3 = round((1 / predict.lm(model3, test[-9]))^2, 2)
Age = test[9]

# AgePredictions = data.frame(AgeP1, AgeP2, AgeP3, Age)

MSPE1 = sapply((Age - AgeP1)^2, mean)
MSPE2 = sapply((Age - AgeP2)^2, mean)
MSPE3 = sapply((Age - AgeP3)^2, mean)

cbind(MSPE1, MSPE2, MSPE3)
##       MSPE1    MSPE2    MSPE3
## Age 5.168737 75321.57 358.6284
```

Final model:

```
par(mfrow = c(2,2))
plot(model.Final, which = 1)
plot(model.Final, which = 2)
plot(model.Final, which = 5)
boxplot(model.Final$residuals, main = "Residuals Boxplot")
```

```
index = which(rownames(data) == c("2052"))
data$Height[index] = 0.113
index = which(rownames(data) == c("2184"))
data = data[-index, ]
index = which(rownames(data) == c("237"))
data = data[-index, ]
model.Final = lm(model.Final, data)
par(mfrow = c(2,2))
plot(model.Final, which = 1)
plot(model.Final, which = 2)
plot(model.Final, which = 5)
summary(model.Final)
anova(model.Final)
boxplot(model.Final$residuals, main = "Residuals Boxplot")
autoplot(model.Final, which = 1:2) + theme_minimal()
```