Final Project Proposal: Abalone Age Prediction

Ran Ma ranma@ucdavis.edu
Sixue Cheng sxcheng@ucdavis.edu
Kevin Su kvnsu@ucdavis.edu

Abstract

Background:

Abalone tends to have increasing value with age and the number of rings it has. Our goal is to construct a model to predict the age of abalone through its rings through indirect measurements such as dimensions and weight rather than having to directly drill through its shell. Additionally, it may be important to note which characteristics are most important in constructing a predictive model of the number of an abalone's rings.

Methods:

Our data comes from the UCI Machine Learning repository, and contains information on various characteristics of abalone: measurements on the weight of the abalone, measurements on the dimensions of the abalone, the sex of the abalone, and the number of rings in the abalone shell (our response variable). We achieved our goal of predicting the number of rings of an abalone shell through the construction of a linear regression model on our data, after accounting for necessary transformations of pertinent variables and selecting our model based on what variables were pertinent using the Akaike Information Criterion (AIC). Additionally, for the purpose of interpretability and stability in our results, we also constructed models in a similar fashion but without some variables that were highly correlated with other explanatory variable inputs.

Results:

We were able to model, with reasonable accuracy, the number of rings in abalone. Our root mean squared error around 0.19 for the log of the number of rings in an abalone. Our Adjusted R-squared for our final models were around 0.66.

Conclusions:

We were able to come up with a reasonably interpretable and predictive model for the number of rings in an abalone shell. However, we were also able to construct slightly more predictive models for the number of rings in an abalone shell at the expense of interpretability and introducing potential instability to changes in the compositions of the test and training datasets. We also found that length and sex (most notably its infancy status) were the most important variables in constructing an effective predictive regression model for the number of rings in an abalone.

Introduction:

Abalone has long been farmed and harvested for numerous uses, notably as food for consumption and for pearls, which have a chance of being produced by their beautiful, iridescent inner shells made of a material known as “mother of pearl,” or nacre, which is also itself prized as a decorative material. The number of layers or rings within the shell of an abalone serves as an effective proxy to the amount of usable nacre in the shell. These layers grow at regular intervals throughout the life of abalone so older abalone tend to have more usable and brilliant nacre. The age of abalone is determined by cutting the shell through the cone, staining it, and counting the number of rings through a microscope – a time-consuming task (Newman).

We seek to predict the number of rings of abalone—and hence its age—using its other more easily measurable features. Additionally, our goal was also to find what characteristics, if any, were more indicative of the number of rings in the abalone shell.

Key Questions:

1. Can we predict the age of the abalone using the other more easily measurable characteristics of the abalone?

2. Which characteristics, if any, are more indicative of the number of rings the abalone has?

Dataset:

The dataset we used is provided by the UCI machine learning Repository and contains 4177 observations of 9 characteristics of individual abalone (Newman). Of these 9 characteristics, three numeric variables correspond to the dimensions of the abalone and four numeric variables correspond to various weight measurements. For the two remaining variables, one is categorical, indicating the sex of the abalone, and one is an integer indicating the number of rings the abalone has. The seven aforementioned numeric variables were divided by 200 as indicated by the UCI machine learning repository (Newman). The response variable we are focusing on in this project is the number of rings that the abalone has, which serves as a proxy for the age of the abalone in years.

Methods:

We begin by conducting exploratory data analysis through finding summary statistics for each of the variables in the dataset and examining correlation plots. We also note that there are no missing values or duplicated records within the dataset.

We began by obtaining summary statistics of the dataset (Figures 3.1 and 3.2). Combined with the box plot of quantitative variables (Figure 3.3), we can see that the largest height of a single

abalone is 1.13×200mm, which is nearly 10 times the upper quartile (0.1650×200mm). A 9 inch height (not length) abalone is unreasonable when measured laying on its side, and all of its other measurements otherwise seem very ordinary (Table 3.4). As such, we will exclude observation 2052 from further analysis as it could be a mistake in data collection or data entry.

We then explored the distribution of each variable. From the pie chart of sex (figure 3.5), we can see that the proportions of infant, female, and male abalone are roughly the same. From the histogram of other variables (figure 3.5), length and diameter are left-skewed. Height, whole weight, shucked weight, viscera weight, shell weight and rings are right skewed.

Because sex is the only categorical variable, then we explored the difference of other measurements due to different sex. We found that the mean of all measurements between female and male abalones are roughly the same, but things are different when it comes to infant abalones (Table 3.6). Furthermore, there is no obvious difference between female and male abalone in all variables, while they have distinct differences with infant abalone (Figure 3.7). All measured values of infant abalone are much lower than that of female and male abalone.

From the correlation and scatter-plot matrix (figure 3.8), We can see that there is no negative relationship between any variable. There's an obvious positive linear relationship between the four types of weight. Length also has a stark positive linear relationship with diameter. There seems to be a logarithmic relationship between the dimension variables (length, diameter, height) and the weight variables.

From the correlation matrix (table 3.9), we can see that aside from rings, all other variables have high correlations with each other.

Following this, we then split our dataset into training and test datasets to be able to effectively evaluate our models. Then, we used Variance Inflation Factor (VIF), a measure of how correlated an explanatory variable is to all other explanatory variables, to examine collinearity as from the correlation and scatter-plot matrix (figure 3.8), there is reason to suspect that the weight measurements of different states of the abalone meat are highly correlated with each other and the physical dimensions of the abalone are also highly correlated with each other. When considering the VIF, we used the equation

$$VIF = \frac{1}{1 - R_k^2}$$

where $R_k^2$is the coefficient of multiple determination when a variable k is regressed onto the remaining explanatory variables. If the highest VIF among the remaining variables was above 10, we dropped the variable before conducting further analysis. For the purpose of comparison of this strategy, we retained and constructed similar models with all variables retained.

We then fit the data using an ordinary least squares model and examine the residuals to check for any necessary transformations of variables, potential interactions, or higher-order terms.

In doing so, we used the Box-Cox procedure to determine that a logarithmic transformation of the number of rings was necessary to maintain a normality of errors assumption in linear regression. For the purpose of regression, we also scaled the remaining explanatory variables to be centered at zero and have unit variance.

We then re-fit the models accordingly and conducted ANOVA (analysis of variance) to find the most important characteristic in predicting the number of rings of the abalone. We used the Akaike Information Condition (AIC) which has the equation

$$AIC = n\, log\frac{SSE}{n} + 2p$$

where n is the number of observations in the sample, p is the number of regression coefficients in the model being considered, and SSE is the sum of squared errors in the model being considered to use as our model selection criterion, where a lower AIC value indicates a superior model.

To select the model, we performed forward stepwise selection to select our model for predicting the number of rings of abalone. To do so, we started with a null model with no explanatory coefficients. We then considered all of the remaining explanatory variables not already within the model, and examined the effects of adding the corresponding coefficient that yielded the model with the lowest AIC out of the candidates if in fact an improvement would occur in the form of a lower AIC.

We then considered all of the existing explanatory variables and whether dropping the corresponding coefficient that would result in the lowest AIC without the coefficient if in fact dropping the coefficient would lower the AIC. Of the coefficients considered to be added or dropped, the procedure that would improve the AIC the most would be conducted, and the process would be repeated until a stable model was reached.

Results

First of all, we apply simple linear regression model using all variables (table 4.1). Based on the given training dataset, the R^2 is 0.5333, the R_a^2 is 0.5318. In presence of other variables, this model ignores the difference caused by sex=M. The p value of length is also large (0.97461), indicating that in presence of other variables, length is not statistically significant.

We applied the model on both training set and test set for prediction, The RMSE and R^2 value on both sets are similar (table 4.2), indicating that the model is somehow stable. In other words, the results would not be significantly different on the training and test dataset.

Now we take a look at residual plots (figure 4.3). From the residual vs fitted plot, there is a a pattern of moderate heteroscedasticity. In Normal Q-Q plot, the distribution of residuals is heavy tailed. Also, the Scale-Location does not show a straight line, which basically means the full model is not adequate. In the Residual vs Leverage plot, no data point has extreme cook's distance, which shows that there are no extreme influential points in our regression model.

In conclusion, the simple linear regression model with all variables is not a great model.

Now we consider scaling the explanatory variables and doing Box-Cox transformation for further analysis. From the Box-Cox plot (figure 4.4), transformation of rings is needed to address heteroscedasticity. When λ is approximately 0, SSE is minimized (or log-likelihood is maximized). As such we apply a log transformation to rings.

On the other hand, the variance inflation factor (VIF) is used to measure the amount of multicollinearity. After calculating VIF, we dropped the variables with the highest VIF iteratively if the variable with the highest VIF has a VIF of over 10, but it ultimately made the models less predictive, even if the coefficients ultimately had more stable standard errors. The remaining variables are sex, length, height, shucked_weight, and shell_weight (table 4.5, table 4.6).

Based on above analysis, there is no need to include all variables in the simple linear regression model. We perform variable selection in two ways. Although both ways perform forward stepwise procedure using AIC criterion, one model uses all variables and the other one drops variables with high VIF.

The selected model with no variables dropped according to VIF is: (table 4.7)

rings ~ height + sex + shucked_weight + diameter + shell_weight + whole_weight + viscera_weight + length

The selected model with variables dropped according to VIF is: (table 4.8)

rings ~ height + sex + shucked_weight + shell_weight + length

Both two models perform similarly on training set and test set (table 4.9, table 4.10).

The Normal Q-Q plots of the two models now look much closer to normal after transforming the data, but there is still some non-linearity as seen in the residuals vs fitted plot (figure 4.11, figure 4.12). All variables seem to be significant both before and after dropping highly correlated variables, but standard errors may be unstable based on training samples if not dropping variables (table 4.7, table 4.8).

To solve the non-linearity issue of residuals, we consider including interaction terms. In this part, we also fit two models: no variables dropped, and variables dropped (according to VIF).

After including the model interactions, the non-linearity issue of residuals seems to have been resolved (figure 4.13, figure 4.14). Our R_a^2 has increased (from 0.5991151 to 0.6624583 for the variable not-dropped model, from 0.5891185 to 0.6439142 for the variable dropped model), AIC has dropped (from -9354.131 to -9830.843, and from -9288.136 to -9690.513 respectively). Our AIC has improved meaningfully after including interaction terms (table 4.15, table 4.16). We can also see that through model selection, height is the variable that is chosen first as it explains most of the variation in the response variable, and sex is chosen second as

it explains the next most of the variation in the number of rings.

Note that the prediction error is not the only criterion for model selection. Although the model built and selected without variable selection according to VIF has slightly better training and test error, it is more unstable and less interpretable due to high standard errors on coefficients (table 4.15, table 4.16).

The reason we did not include second-order terms is that it does not meaningfully improve our model. AIC does very slightly improve, but it adds further instability to the model.

Our more stable model with interaction terms and no higher-order polynomial is

$$\begin{aligned} log(Rings) = {} & 0.0739\, height - 0.0562 sex_I - 0.012 sex_M + 0.365 shellweight \\ & -.3398 shuckedweight + 0.0309 length \\ & +0.0141 height * shellweight + 0.1411\, sex_I * shuckedweight \\ & +0.01 sex_M * shuckedweight - 0.1574 length * shellweight \\ & +0.1713 shuckedweight * length - 0.0668 height * length \\ & -0.0578 sex_I * shellweight + 0.084 sex_M * shellweight \end{aligned}$$

We can also see that through model selection, height and sex are the two variables that explain the most variation in the number of rings and are thus the two most important variables in building a model to predict the number of rings that an abalone has.

Conclusion and Discussion:

Given that our models were able to explain roughly two-thirds of the variation in the number of rings within an abalone shell without encountering overfitting, a model built on dimensions, infancy status, and weight could be useful if the reliability is sufficient when weighed against the labor necessary to measure the number of rings by hand. We can see also through model selection that the characteristics most indicative of the number of rings within the abalone is the height of the abalone followed by the sex (most importantly, whether the abalone is an infant or not).

We also found that our models without the highly correlated variables were more stable than those that included these correlated variables, even if slightly less predictive. Therefore the models without the highly correlated variables probably yield a more generalized result for a standard reference. Additionally, the inclusion of higher-order terms also causes similar issues where the model selection procedures may yield a more predictive model (slightly lower test error) but introduce instability where the model changes depending on the selection of our training dataset.

Works Cited

Newman D, Hettich S, Blake C, Merz C (1998). “UCI Repository of machine learning databases.”

Appendix

Tables and Figures

Table 3.1

	sex
F	1307
I	1342
M	1528

Table 3.2

length	diameter	height	whole_weight	shucked_weight	viscera_weight	shell_weight	rings
Min. :0.075	Min. :0.0550	Min. :0.0000	Min. :0.0020	Min. :0.0010	Min. :0.0005	Min. :0.0015	Min. : 1.000
1st Qu.:0.450	1st Qu.:0.3500	1st Qu.:0.1150	1st Qu.:0.4415	1st Qu.:0.1860	1st Qu.:0.0935	1st Qu.:0.1300	1st Qu.: 8.000
Median :0.545	Median :0.4250	Median :0.1400	Median :0.7995	Median :0.3360	Median :0.1710	Median :0.2340	Median : 9.000
Mean :0.524	Mean :0.4079	Mean :0.1395	Mean :0.8287	Mean :0.3594	Mean :0.1806	Mean :0.2388	Mean : 9.934
3rd Qu.:0.615	3rd Qu.:0.4800	3rd Qu.:0.1650	3rd Qu.:1.1530	3rd Qu.:0.5020	3rd Qu.:0.2530	3rd Qu.:0.3290	3rd Qu.:11.000
Max. :0.815	Max. :0.6500	Max. :1.1300	Max. :2.8255	Max. :1.4880	Max. :0.7600	Max. :1.0050	Max. :29.000

Figure 3.3

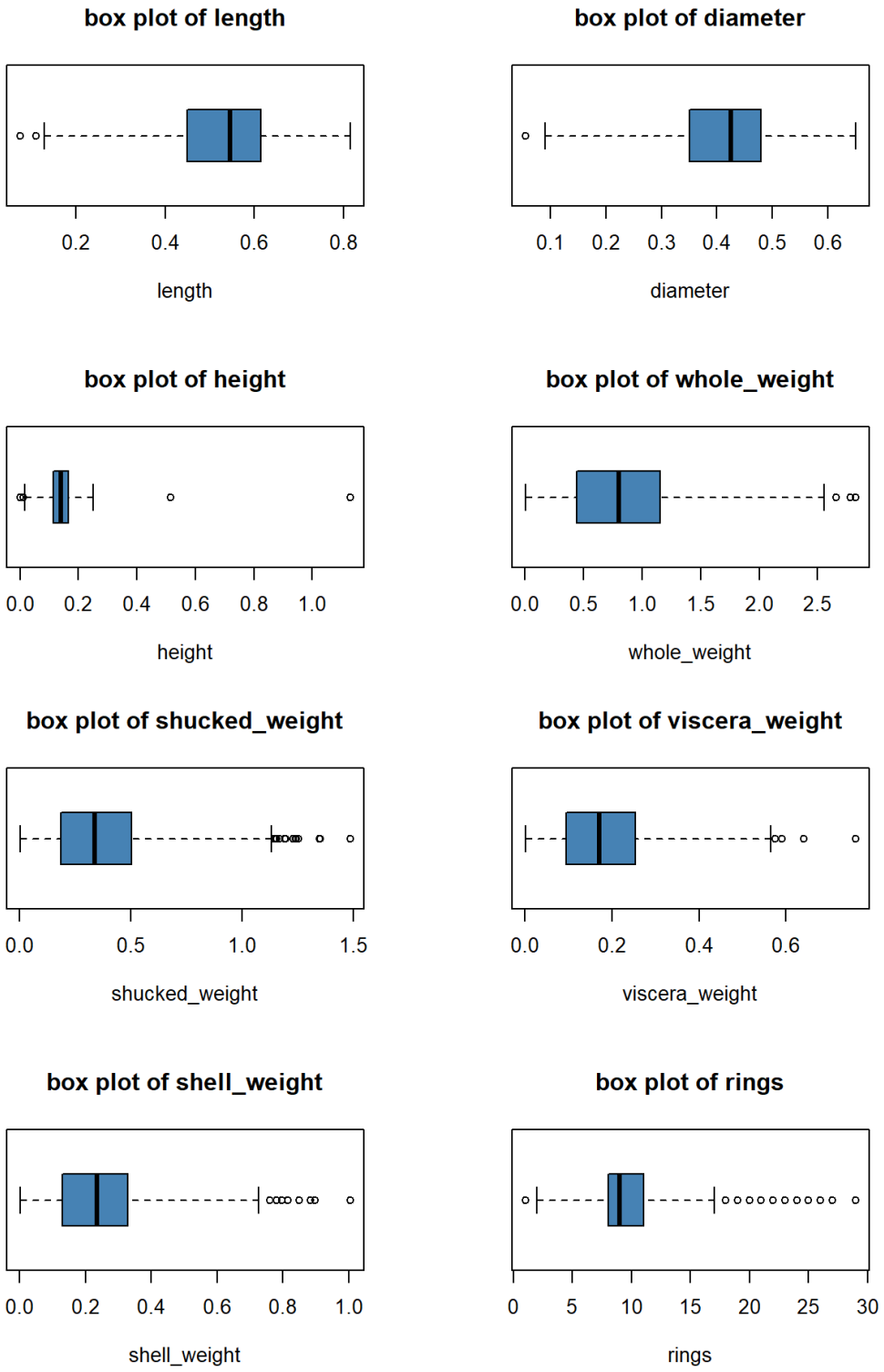


Table 3.4

	sex	length	diameter	height	whole_weight	shucked_weight	viscera_weight	shell_weight	rings
2052	F	0.455	0.355	1.13	0.594	0.332	0.116	0.1335	8

Figure 3.5

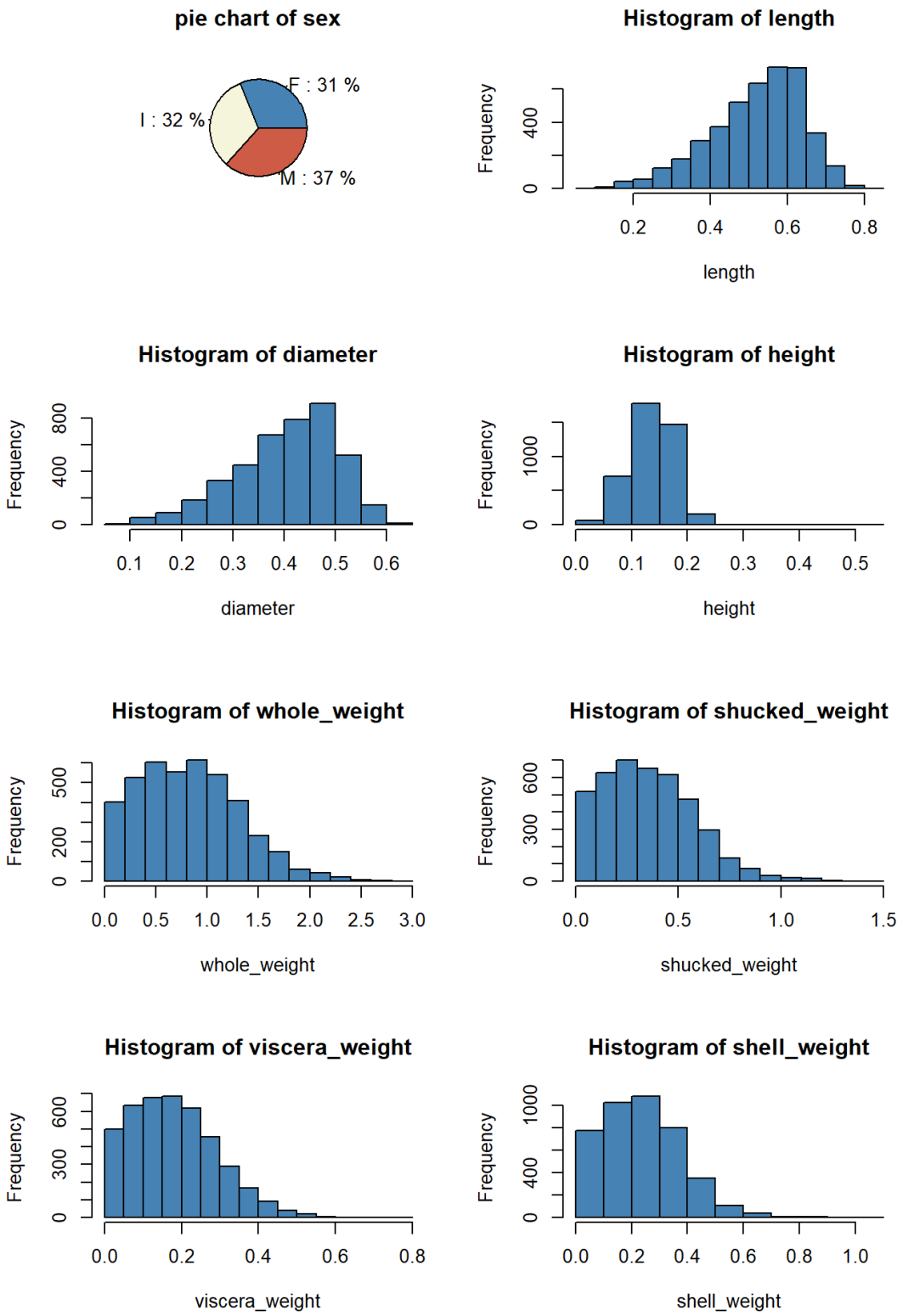

pie chart of sex
F : 31 %
I : 32 %
M : 37 %
Histogram of length
Frequency
0
400
0.2
0.4
0.6
0.8
length
Histogram of diameter
Frequency
0
400
800
0.1
0.2
0.3
0.4
0.5
0.6
diameter
Histogram of height
Frequency
0
1000
0.0
0.1
0.2
0.3
0.4
0.5
height
Histogram of whole_weight
Frequency
0
200
500
0.0
0.5
1.0
1.5
2.0
2.5
3.0
whole_weight
Histogram of shucked_weight
Frequency
0
300
600
0.0
0.5
1.0
1.5
shucked_weight
Histogram of viscera_weight
Frequency
0
300
600
0.0
0.2
0.4
0.6
0.8
viscera_weight
Histogram of shell_weight
Frequency
0
400
1000
0.0
0.2
0.4
0.6
0.8
1.0
shell_weight

Table 3.6

Group.1	length	diameter	height	whole_weight	shucked_weight	viscera_weight	shell_weight	rings
F	0.5791884	0.4548086	0.1572665	1.0468786	0.4462753	0.2307764	0.3021390	11.131700
I	0.4277459	0.3264940	0.1079955	0.4313625	0.1910350	0.0920101	0.1281822	7.890462
M	0.5613907	0.4392866	0.1513809	0.9914594	0.4329460	0.2155445	0.2819692	10.705497

Figure 3.7

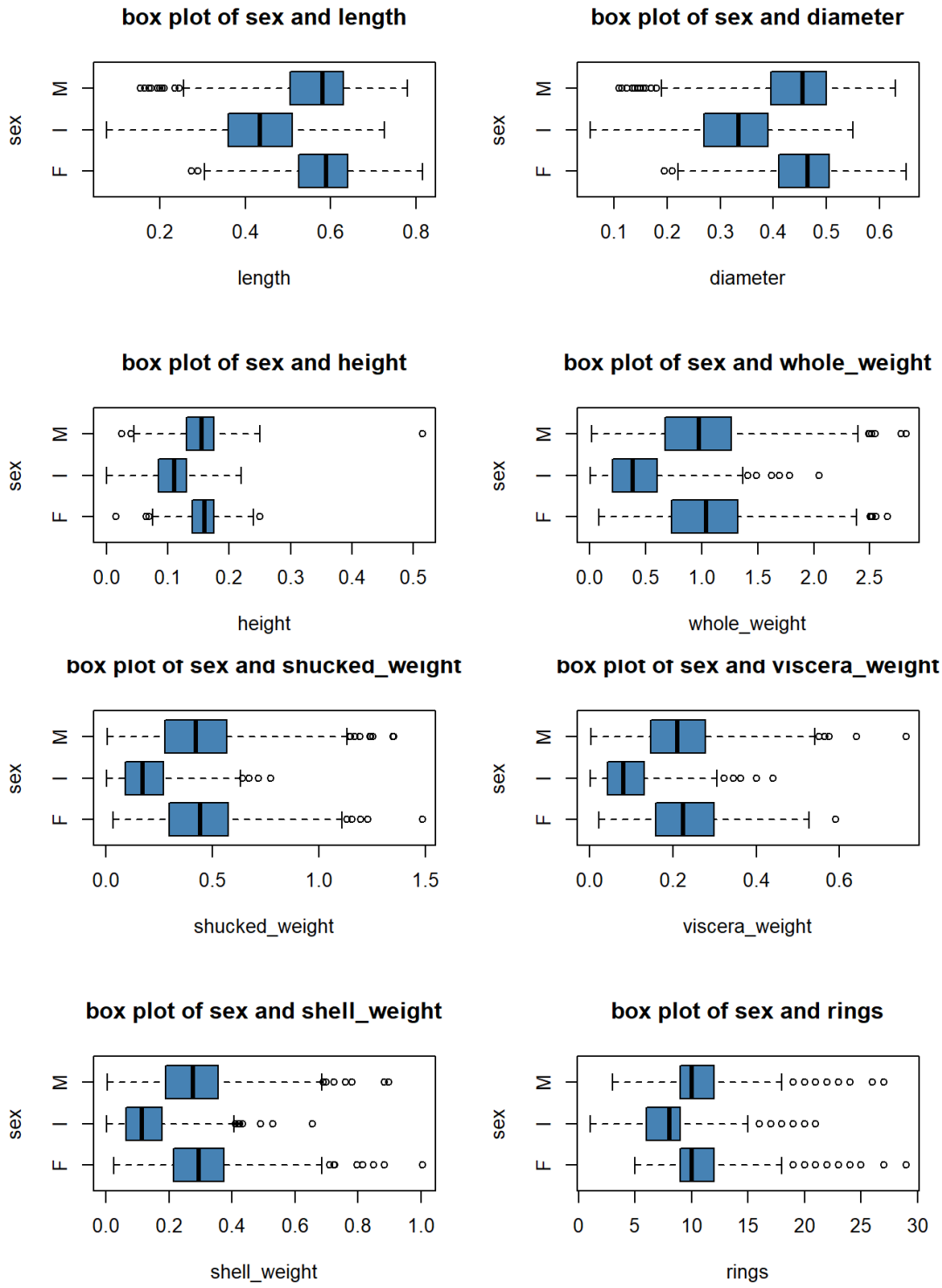


Figure 3.8

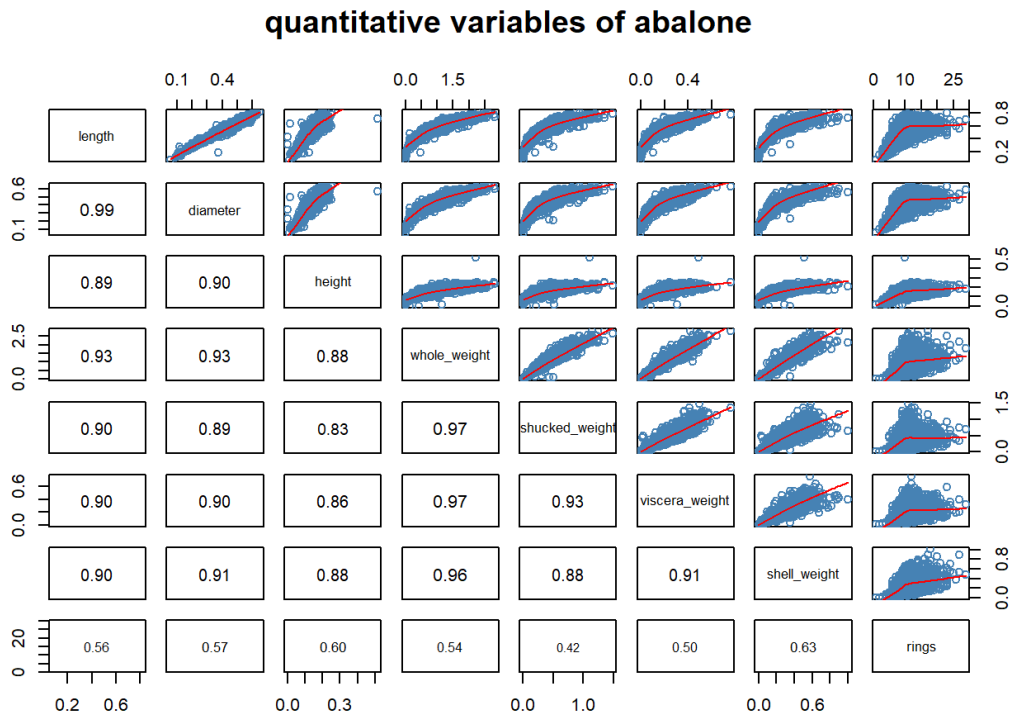


Table 3.9

	length	diameter	height	whole_weight	shucked_weight	viscera_weight	shell_weight	rings
length	1.0000000	0.9868108	0.8929763	0.9252573	0.8979338	0.9030099	0.8976985	0.5566830
diameter	0.9868108	1.0000000	0.8993063	0.9254479	0.8931787	0.8997172	0.9053261	0.5746276
height	0.8929763	0.8993063	1.0000000	0.8834253	0.8336786	0.8616466	0.8831327	0.6028374
whole_weight	0.9252573	0.9254479	0.8834253	1.0000000	0.9694197	0.9663742	0.9553604	0.5403590
shucked_weight	0.8979338	0.8931787	0.8336786	0.9694197	1.0000000	0.9319844	0.8826568	0.4208848
viscera_weight	0.9030099	0.8997172	0.8616466	0.9663742	0.9319844	1.0000000	0.9076495	0.5037772
shell_weight	0.8976985	0.9053261	0.8831327	0.9553604	0.8826568	0.9076495	1.0000000	0.6275354
rings	0.5566830	0.5746276	0.6028374	0.5403590	0.4208848	0.5037772	0.6275354	1.0000000

Table 4.1

```
Call:
lm(formula = rings ~ ., data = train)

Residuals:
    Min      1Q  Median      3Q     Max
-6.6646 -1.3345 -0.3111  0.8938 11.3927

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)     3.482321   0.351186   9.916  < 2e-16 ***
sexI           -0.925011   0.121352  -7.623 3.34e-14 ***
sexM           -0.005074   0.099467  -0.051  0.95932
length         -0.068028   2.137382  -0.032  0.97461
diameter        9.570951   2.624513   3.647  0.00027 ***
height         21.616828   2.531386   8.540  < 2e-16 ***
whole_weight    7.814820   0.867098   9.013  < 2e-16 ***
shucked_weight -19.112471   0.989585 -19.314  < 2e-16 ***
viscera_weight  -9.160202   1.542828  -5.937 3.24e-09 ***
shell_weight    8.089071   1.369178   5.908 3.86e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.182 on 2913 degrees of freedom
Multiple R-squared:  0.5333, Adjusted R-squared:  0.5318
F-statistic: 369.8 on 9 and 2913 DF,  p-value: < 2.2e-16
```

Table 4.2

	RMSE	Rsquare
training set	2.178312	0.5332541
test set	2.192011	0.5602397

Figure 4.3

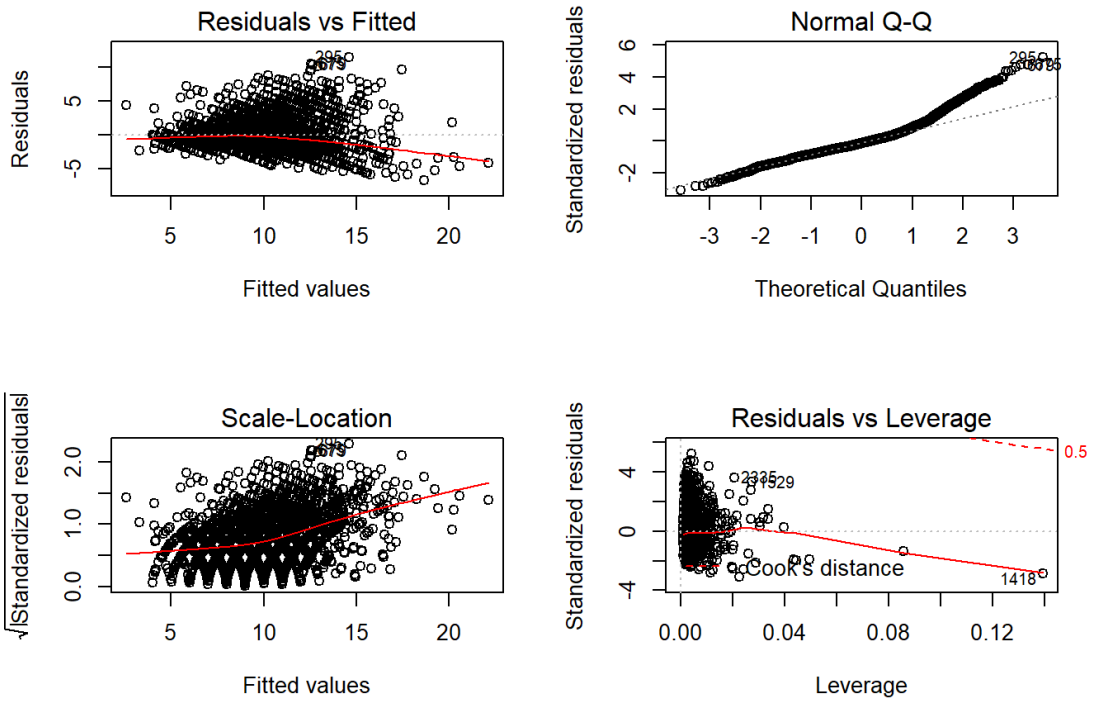


Figure 4,4

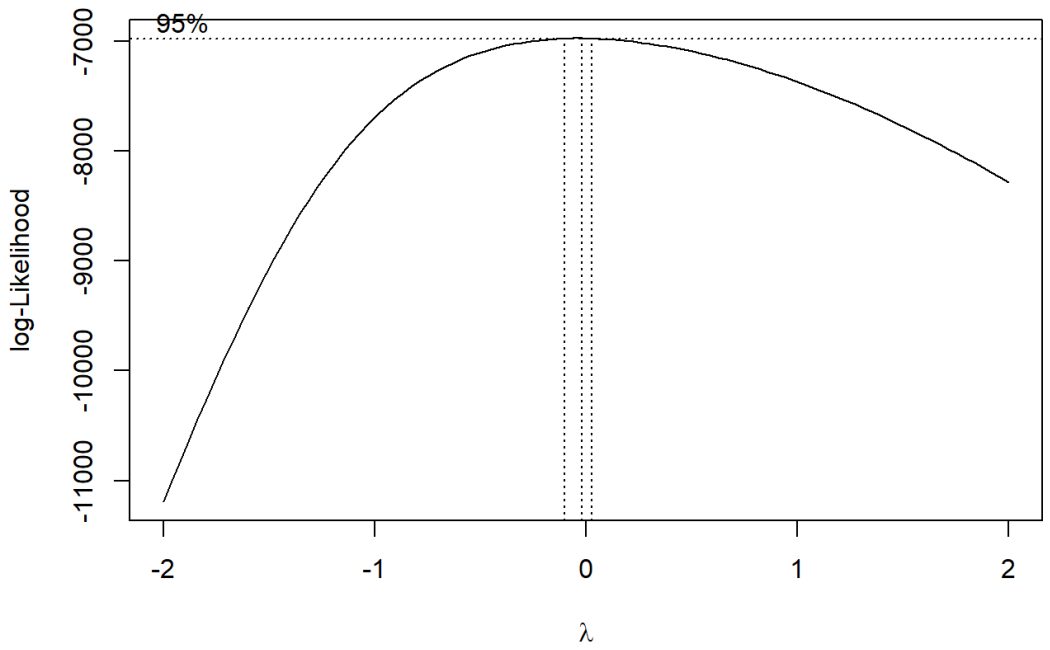


Table 4.5

	GVIF	Df	GVIF^(1/(2*Df))
sex	1.564982	2	1.118478
length	40.684582	1	6.378447
diameter	41.758109	1	6.462051
height	6.008460	1	2.451216
whole_weight	111.225453	1	10.546348
shucked_weight	29.798655	1	5.458814
viscera_weight	17.472538	1	4.180016
shell_weight	22.163857	1	4.707851

Table 4.6

	GVIF	Df	GVIF^(1/(2*Df))
sex	1.511399	2	1.108778
length	8.715446	1	2.952193
height	5.840196	1	2.416650
shucked_weight	6.135415	1	2.476977
shell_weight	7.408978	1	2.721944

Table 4.7

```
Call:
lm(formula = fstniselect$call$formula, data = train.C)

Residuals:
     Min       1Q   Median       3Q      Max
-1.33598 -0.13229 -0.01481  0.11270  0.70562

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)     2.279303   0.007061 322.788  < 2e-16 ***
height          0.092985   0.009139  10.175  < 2e-16 ***
sexI           -0.100397   0.011208  -8.958  < 2e-16 ***
sexM            0.002231   0.009187   0.243  0.80817
shucked_weight -0.354819   0.020352 -17.434  < 2e-16 ***
diameter        0.123898   0.024092   5.143 2.89e-07 ***
shell_weight    0.074408   0.017552   4.239 2.31e-05 ***
whole_weight    0.253446   0.039320   6.446 1.34e-10 ***
viscera_weight -0.084298   0.015584  -5.409 6.85e-08 ***
length          0.067582   0.023781   2.842  0.00452 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2015 on 2913 degrees of freedom
Multiple R-squared:  0.5991, Adjusted R-squared:  0.5979
F-statistic: 483.7 on 9 and 2913 DF,  p-value: < 2.2e-16
```

Table 4.8

```
Call:
lm(formula = fstnivifselect$call$formula, data = train.forVIF)

Residuals:
     Min       1Q   Median       3Q      Max
-1.33903 -0.13294 -0.01671  0.11308  0.73547

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)     2.2821177  0.0070792 322.370   <2e-16 ***
height          0.0991292  0.0091169  10.873   <2e-16 ***
sexI           -0.1071993  0.0111466  -9.617   <2e-16 ***
sexM            0.0006087  0.0092620   0.066    0.948
shucked_weight -0.2577304  0.0093445 -27.581   <2e-16 ***
shell_weight    0.1653873  0.0102686  16.106   <2e-16 ***
length          0.1719438  0.0111372  15.439   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2039 on 2916 degrees of freedom
Multiple R-squared:  0.5891, Adjusted R-squared:  0.5883
F-statistic: 696.8 on 6 and 2916 DF,  p-value: < 2.2e-16
```

Table 4.9

	RMSE	Rsquare
training set	0.2011887	0.5991151
test set	0.2002388	0.6169438

Table 4.10

	RMSE	Rsquare
training set(dropped)	0.2036817	0.5891185
test set(dropped)	0.2045199	0.6003893

Figure 4.11

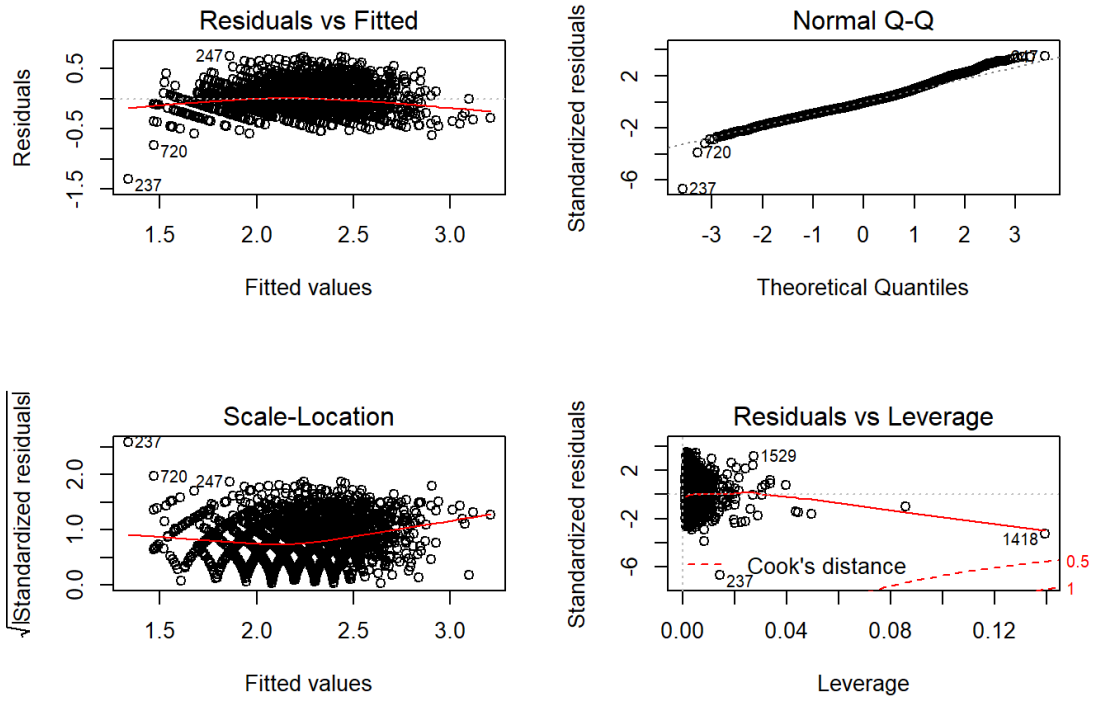


Figure 4.12

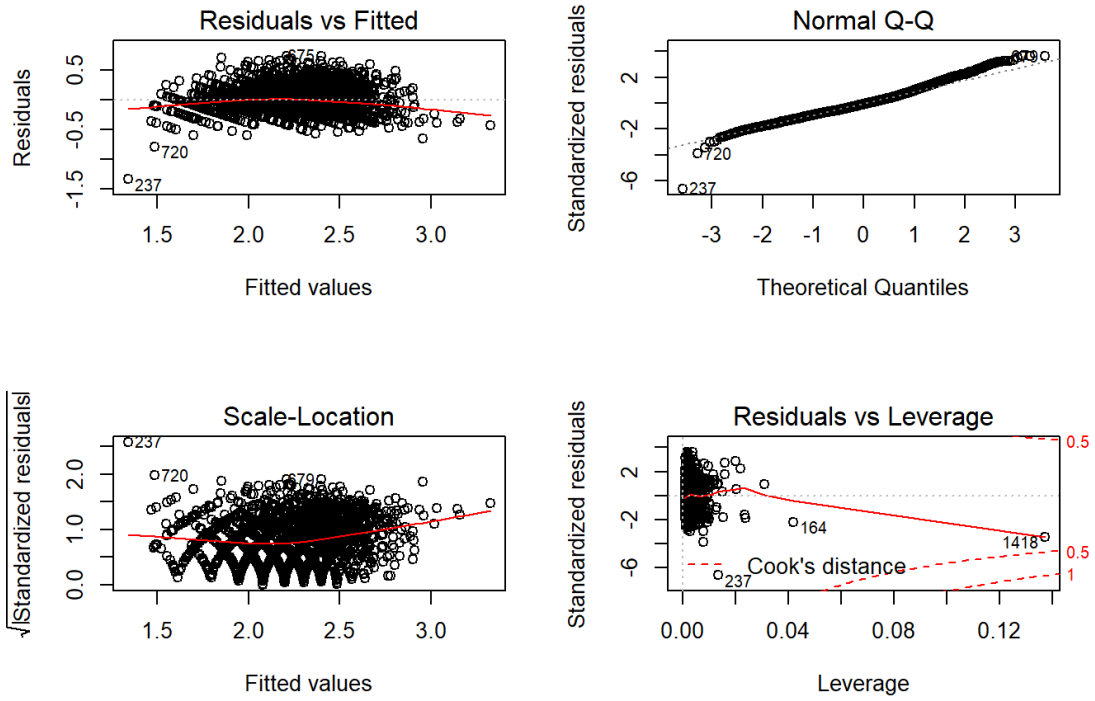


Figure 4.13

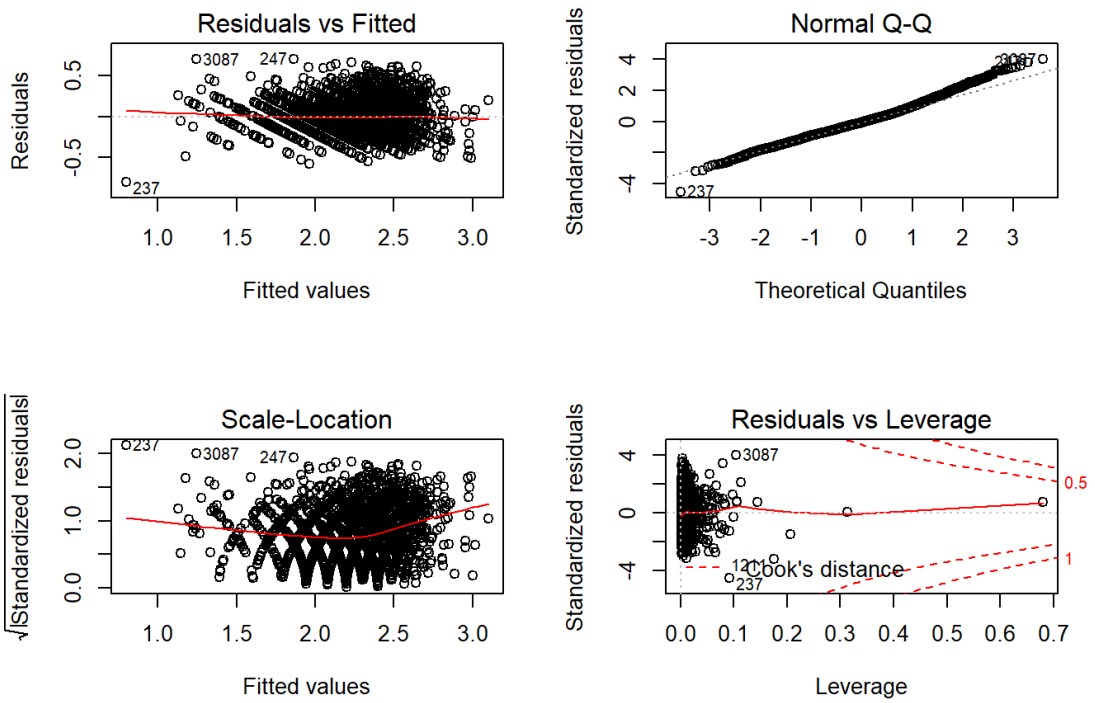


Figure 4.14

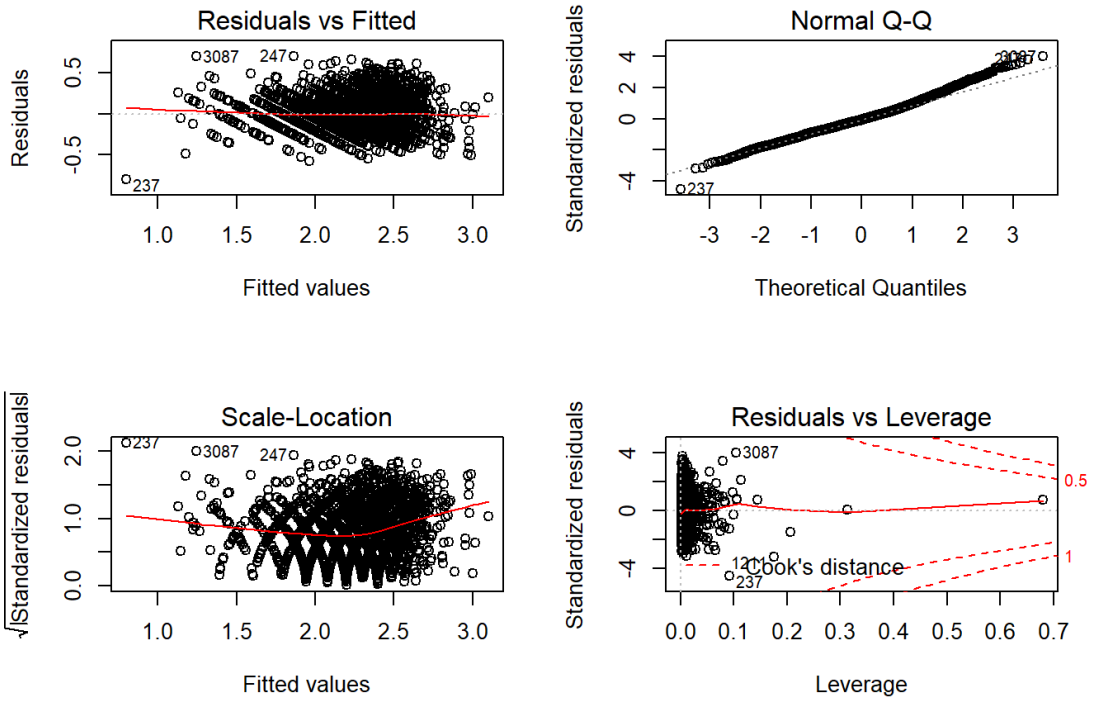


Table 4.15

```
Stepwise Model Path
Analysis of Deviance Table

Initial Model:
rings ~ 1

Final Model:
rings ~ height + sex + shell_weight + shucked_weight + whole_weight +
    viscera_weight + diameter + length + shucked_weight:whole_weight +
    shell_weight:whole_weight + sex:shucked_weight + shucked_weight:diameter +
    whole_weight:diameter + shell_weight:shucked_weight + sex:diameter +
    diameter:length + shell_weight:length + height:shucked_weight +
    shucked_weight:length


                           Step Df   Deviance Resid. Df Resid. Dev       AIC
28 - whole_weight:viscera_weight  1 0.03773443      2900   99.61938 -9830.843
```

Table 4.16

```
Stepwise Model Path
Analysis of Deviance Table

Initial Model:
rings ~ 1

Final Model:
rings ~ height + sex + shell_weight + shucked_weight + length +
    height:shell_weight + sex:shucked_weight + shell_weight:length +
    shucked_weight:length + height:length + sex:shell_weight


                            Step Df   Deviance Resid. Df Resid. Dev       AIC
18 - shell_weight:shucked_weight  1 0.06868626      2908   105.0923 -9690.513
```

206 final project

Kevin Su, Ran Ma, Sixue Cheng

11/23/2021

Background

Abalone has long been farmed and harvested for numerous uses, notably as food for consumption and for pearls, which have a chance of being produced by their beautiful, iridescent inner shells made of a material known as "mother of pearl," or nacre, which is also itself prized as a decorative material.

The number of layers or rings within the shell of an abalone serves as an effective proxy to the amount of usable nacre in the shell. These layers grow at regular intervals throughout the life of abalone so older abalone tend to have more usable and brilliant nacre. The age of abalone is determined by cutting the shell through the cone, staining it, and counting the number of rings through a microscope – a time-consuming task. This project instead seeks to predict the number of rings of abalone (and hence its age) using its other more easily measurable features.

Key Questions

There are two key questions we are trying to answer in this project:

1. Can we predict the age of the abalone using the other more easily measurable characteristics of the abalone?
2. Which characteristics, if any, are more indicative of the number of rings the abalone has?

Dataset

The dataset we used is provided by the UCI machine learning Repository called "Abalone".

```
abalone_ori = read.table("abalone.txt",sep=",", header=FALSE)
knitr::kable(cbind(length(abalone_ori), nrow(abalone_ori)), col.names = c('observations','variables'))
```

observations	variables
9	4177

```
colnames(abalone_ori) = c("sex","length","diameter","height","whole_weight","shucked_weight","viscera_w
knitr::kable(t(sapply(abalone_ori, class)))
```

sex	length	diameter	height	whole_weight	shucked_weight	viscera_weight	shell_weight	rings
character	numeric	numeric	numeric	numeric	numeric	numeric	numeric	integer

```
abalone_ori['sex'] = factor(abalone_ori$sex)
knitr::kable(head(abalone_ori))
```

sex	length	diameter	height	whole_weight	shucked_weight	viscera_weight	shell_weight	rings
M	0.455	0.365	0.095	0.5140	0.2245	0.1010	0.150	15
M	0.350	0.265	0.090	0.2255	0.0995	0.0485	0.070	7
F	0.530	0.420	0.135	0.6770	0.2565	0.1415	0.210	9
M	0.440	0.365	0.125	0.5160	0.2155	0.1140	0.155	10
I	0.330	0.255	0.080	0.2050	0.0895	0.0395	0.055	7
I	0.425	0.300	0.095	0.3515	0.1410	0.0775	0.120	8

The dataset contains 4177 observations of 9 characteristics of individual abalone. Of these 9 characteristics, only 1 variable is categorical(`sex`), indicating the sex of the abalone, all other variables are quantitive variables. 3 numeric variables correspond to the dimensions of the abalone (`length`,`diameter` and `height`), 4 numeric variables correspond to various weight measurements (`whole_weight`, `shucked_weight`, `viscera_weight` and `shell_weight`). The remaining variable is an integer(`rings`), our response variable, indicating the number of rings the abalone has.

Note that the original documentation from UCI data repository indicates that continuous explanatory variables were divided by 200.

Methodologies:

We achieved our goal by applying linear regression model. We conducted exploratory data analysis to find summary statistics for each variable and to check correlations among variables. The reason to suspect multicollinearity is that the weight measurements of the abalone can be highly correlated with each other. Then we applied linear regression model on training and test set. After scaling, we checked Variance inflation factor (VIF). The further model building is split to two cases, one case is using all variables, and the other case is dropping variables with extreme VIF. We select our model based on AIC criteria in each case. We also considered interaction terms and second-order terms.

Exploratory Data Analysis

```
knitr::kable(t(sapply(abalone_ori,function(x) {length(which(is.na(x)))})))
```

sex	length	diameter	height	whole_weight	shucked_weight	viscera_weight	shell_weight	rings
0	0	0	0	0	0	0	0	0

```
duplicate_value = table(duplicated(abalone_ori))
knitr::kable(as.data.frame(rbind(duplicate_value)))
```

	FALSE
duplicate_value	4177

There are no missing values or duplicated records in the dataset.

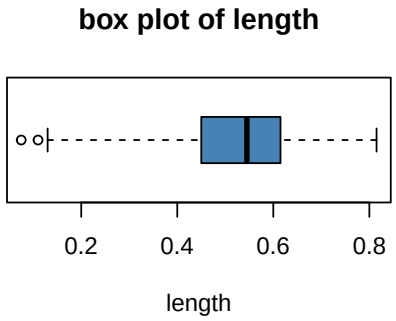
```
knitr::kable(summary(abalone_ori$sex), col.names = 'sex')
```

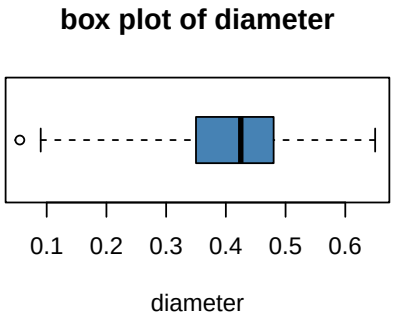
	sex
F	1307
I	1342
M	1528

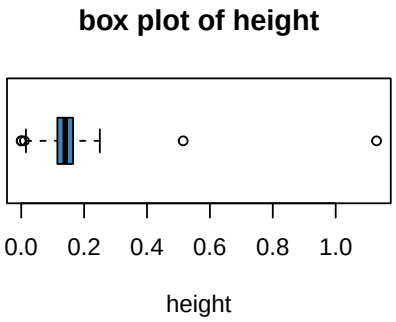
```
knitr::kable(summary(abalone_ori[2:9]))
```

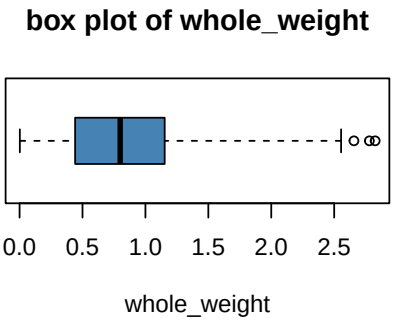
length	diameter	height	whole_weight	shucked_weight	viscera_weight	shell_weight	rings
Min. :0.075	Min. :0.0550	Min. :0.0000	Min. :0.0020	Min. :0.0010	Min. :0.0005	Min. :0.0015	Min. : 1.000
1st Qu.:0.450	1st Qu.:0.3500	1st Qu.:0.1150	1st Qu.:0.4415	1st Qu.:0.1860	1st Qu.:0.0935	1st Qu.:0.1300	1st Qu.: 8.000
Median :0.545	Median :0.4250	Median :0.1400	Median :0.7995	Median :0.3360	Median :0.1710	Median :0.2340	Median : 9.000
Mean :0.524	Mean :0.4079	Mean :0.1395	Mean :0.8287	Mean :0.3594	Mean :0.1806	Mean :0.2388	Mean : 9.934
3rd Qu.:0.615	3rd Qu.:0.4800	3rd Qu.:0.1650	3rd Qu.:1.1530	3rd Qu.:0.5020	3rd Qu.:0.2530	3rd Qu.:0.3290	3rd Qu.:11.000
Max. :0.815	Max. :0.6500	Max. :1.1300	Max. :2.8255	Max. :1.4880	Max. :0.7600	Max. :1.0050	Max. :29.000

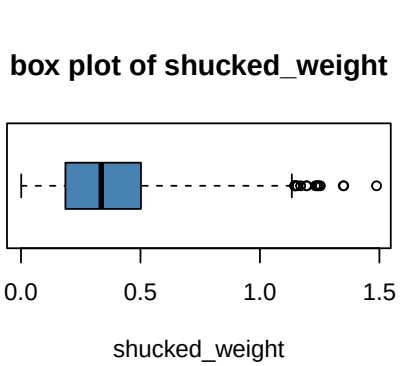
```
par(mfrow = c(2,2))
for (i in 2:9) {
  box = boxplot(abalone_ori[,i],range = 2,horizontal = TRUE, col='steelblue', main = paste('box plot of
```

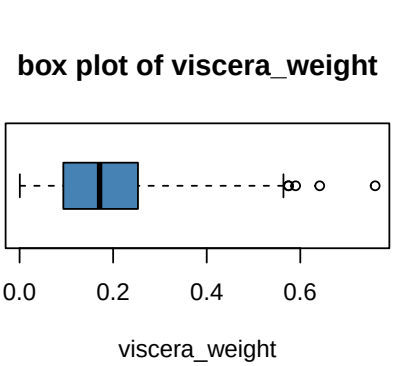


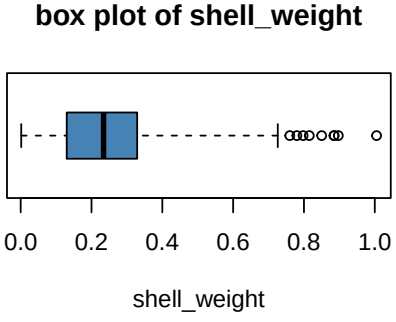


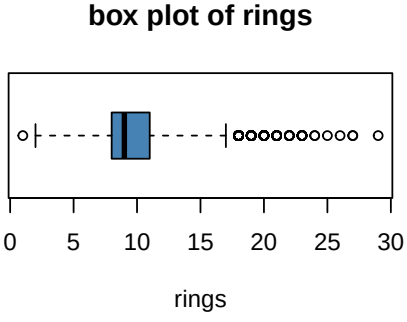













```
max_height = abalone_ori[which.max(abalone_ori$height),]
knitr::kable(max_height)
```

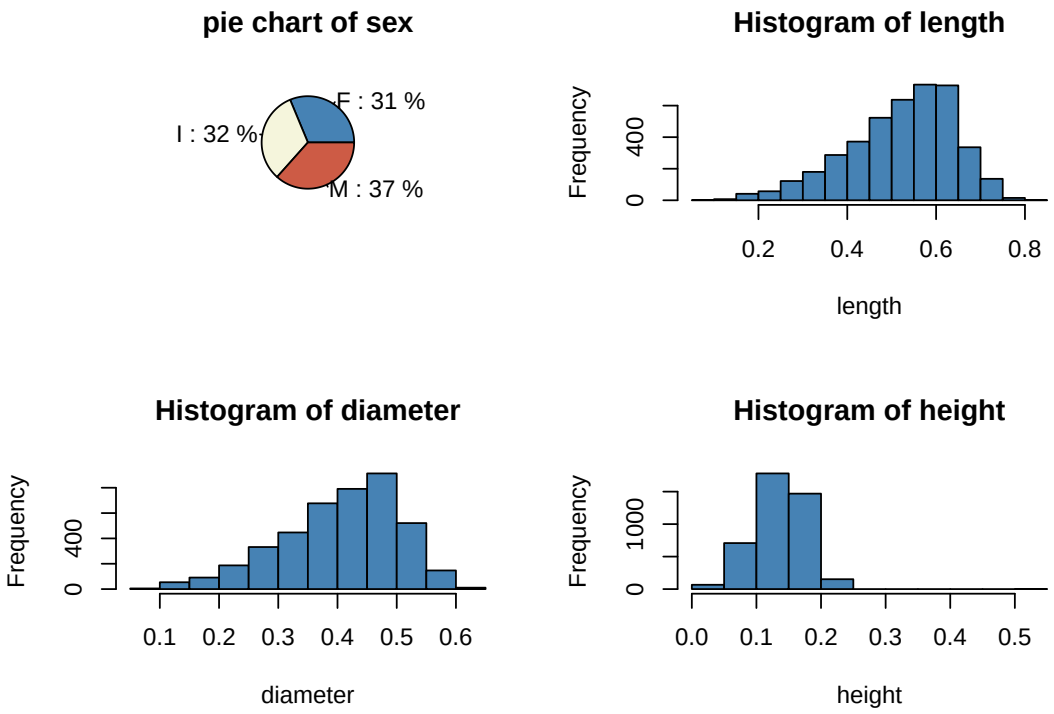
	sex	length	diameter	height	whole_weight	shucked_weight	viscera_weight	shell_weight	rings
2052	F	0.455	0.355	1.13	0.594	0.332	0.116	0.1335	8

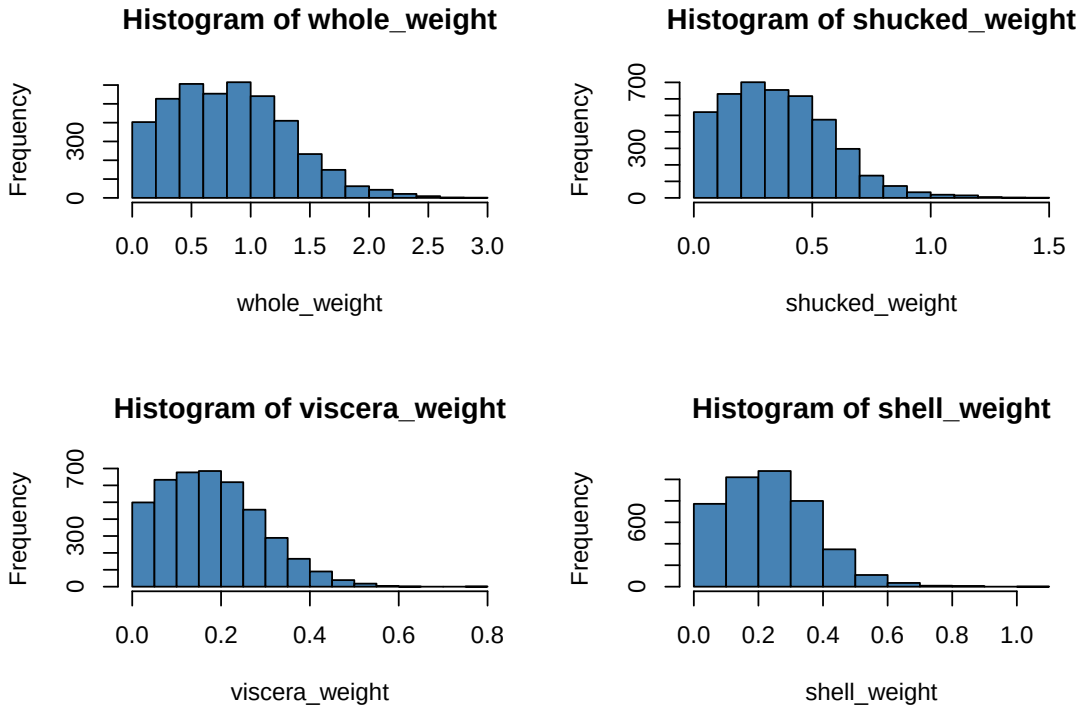
From the summary and the box plot of quantitative variables, we can see that the largest `height` is 1.1300×200mm, which is nearly 10 times of the upper quartile (0.1650×200mm).A 9 inch height (not length) abalone is unreasonable when measured laying on its side, and all of its other measurements otherwise seem very ordinary. It could be a mistake in data collection or data entry. As such, we will exclude observation 2052 from further analysis.

```
abalone = abalone_ori[-which.max(abalone_ori$height),]
```

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

n = nrow(abalone)
lbls = names(table(abalone$sex))
pct = round(100*table(abalone$sex)/n)
lab = paste(lbls,":", pct, "%", sep=' ')

pie(table(abalone$sex),labels=lab,col=c('steelblue','beige','coral3'), main='pie chart of sex')

for (i in 2:8) {
  hist(abalone[,i], main = paste('Histogram of', colnames(abalone[i]), sep = ' '), xlab = colnames(abal
```

The Distribution of Variables





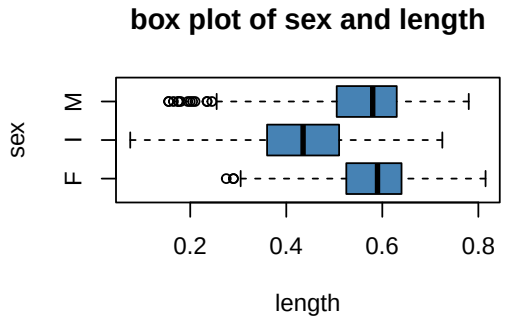
The proportions of infant, female, and male abalone are roughly the same.`length` and `diameter` are left-skewed. `height`, `whole weight`, `shucked weight`, `viscera weight`, `shell weight` and `rings` are right skewed.

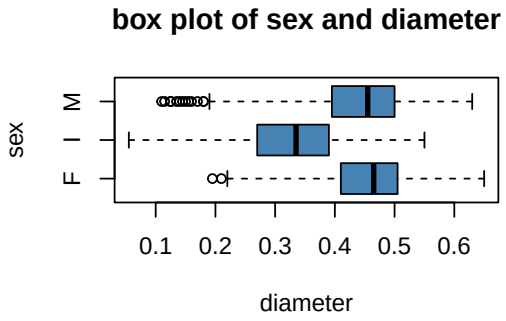
```
mean_sex = aggregate(abalone[,-1], list(abalone[,1]), FUN=mean)
knitr::kable(mean_sex)
```

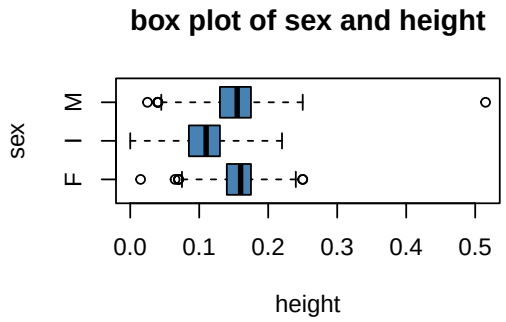
Relationship Between Variables

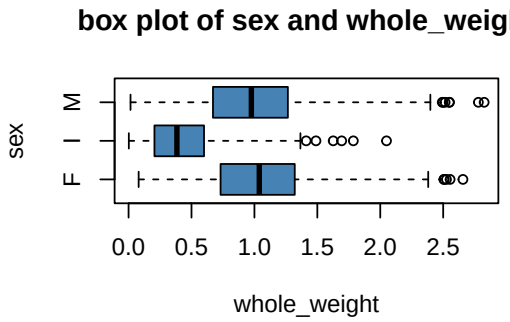
Group.1	length	diameter	height	whole_weight	shucked_weight	viscera_weight	shell_weight	rings
F	0.5791884	0.4548086	0.1572665	1.0468786	0.4462753	0.2307764	0.3021390	11.131700
I	0.4277459	0.3264940	0.1079955	0.4313625	0.1910350	0.0920101	0.1281822	7.890462
M	0.5613907	0.4392866	0.1513809	0.9914594	0.4329460	0.2155445	0.2819692	10.705497

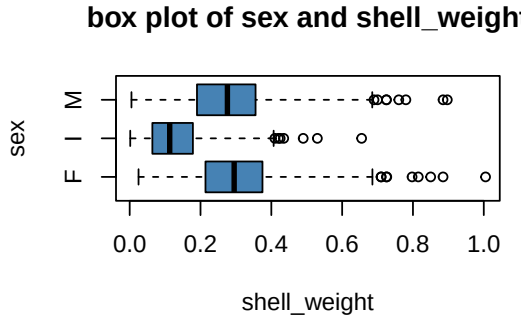
```
par(mfrow = c(2,2))

for (i in 2:9) {
  boxplot(abalone[,i] ~ abalone$sex ,range = 2,horizontal = TRUE, col='steelblue', main = paste('box pl
```

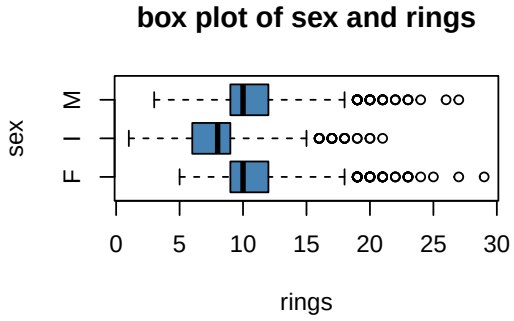






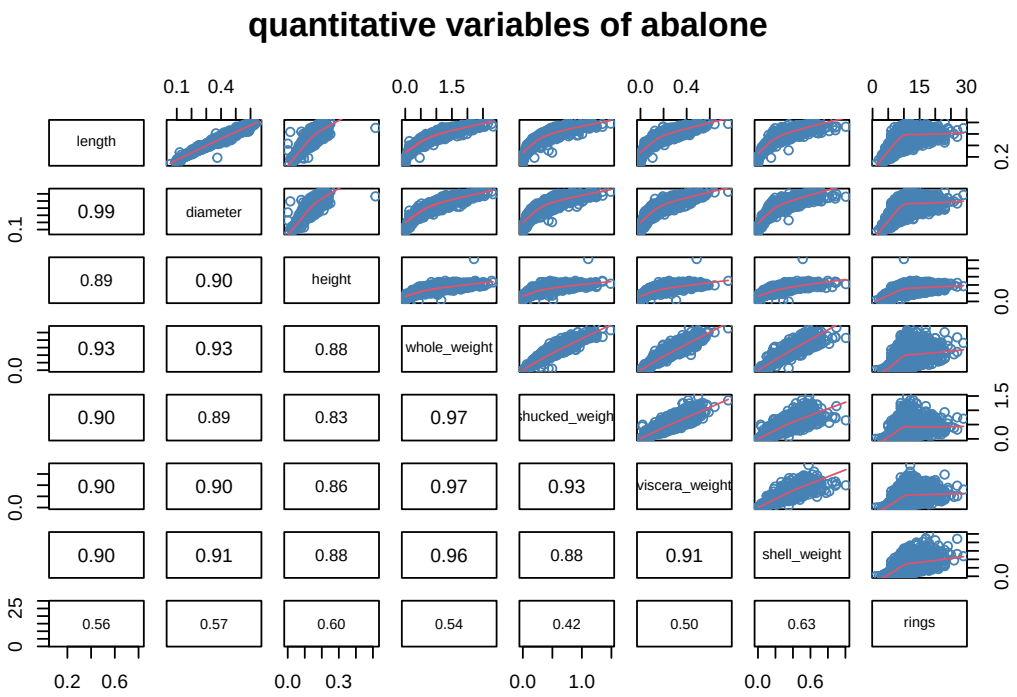






There is no obvious difference between female and male abalone in all variables, while they have distinct difference with infant abalone. All measured values of infant abalone are much lower than that of female and male abalone respectively.

```
panel.cor <- function(x, y, digits = 2, prefix = "", cex.cor, ...) {
  usr <- par("usr")
  on.exit(par(usr))
  par(usr = c(0, 1, 0, 1))
  r <- abs(cor(x, y, use = "complete.obs"))
  txt <- format(c(r, 0.123456789), digits = digits)[1]
  txt <- paste(prefix, txt, sep = "")
  if (missing(cex.cor)) cex.cor <- 0.8/strwidth(txt)
  text(0.5, 0.5, txt, cex =  cex.cor * (1 + r) / 2)
}


pairs(abalone[,c(-1)],  col = 'steelblue', cex = 1, panel = panel.smooth, lower.panel = panel.cor, main
```



We can see that there's no negative relationship between variables. There's obvious positive linear relationship between the four types of weight.`length` also has obvious positive linear relationship with `diameter`. There's a logarithmic relationship between the dimension variables (length, diameter, height) and the weight variables.

```
correlation = cor(abalone[,-1], method = "pearson",use = "complete.obs")
knitr::kable(correlation)
```

	length	diameter	height	whole_weight	shucked_weight	viscera_weight	shell_weight	rings
length	1.0000000	0.9868108	0.8929763	0.9252573	0.8979338	0.9030099	0.8976985	0.5566830
diameter	0.9868108	1.0000000	0.8993063	0.9254479	0.8931787	0.8997172	0.9053261	0.5746276
height	0.8929763	0.8993063	1.0000000	0.8834253	0.8336786	0.8616466	0.8831327	0.6028374

	length	diameter	height	whole_weight	shucked_weight	viscera_weight	shell_weight	rings
whole_weight	0.9252573	0.9254479	0.8834253	1.0000000	0.9694197	0.9663742	0.9553604	0.5403590
shucked_weight	0.8979338	0.8931787	0.8336786	0.9694197	1.0000000	0.9319844	0.8826568	0.4208848
viscera_weight	0.9030099	0.8997172	0.8616466	0.9663742	0.9319844	1.0000000	0.9076495	0.5037772
shell_weight	0.8976985	0.9053261	0.8831327	0.9553604	0.8826568	0.9076495	1.0000000	0.6275354
rings	0.5566830	0.5746276	0.6028374	0.5403590	0.4208848	0.5037772	0.6275354	1.0000000

From the correlation matrix, we can see that except `rings`, all other variables has pretty high correlation with each other. Such high correlation coefficients among features can result into multicollinearity. We further investigate multicollinearity later and select variables through examining VIF.

Linear Regression

From the Exploratory Data Analysis, we can apply linear regression on the dataset.

Training Set and Test Set We select 70% whole dataset as the training set for model building, the other 30% as the test set for model testing.

```
set.seed(1)
len = dim(abalone)[1]
train_ind = sample(1:len,0.7*len,replace = F)
#cv_ind <- sample(train_ind, 0.4*length(train_ind), replace = F)
train = abalone[train_ind,]
#cv <- abalone[cv_ind,]
test = abalone[-train_ind,]
tt = as.data.frame(rbind(dim(train),dim(test)),row.names = c('training set','test set'))
colnames(tt) = c('observations','variables')
knitr::kable(tt)
```

	observations	variables
training set	2923	9
test set	1253	9

Simple Linear Regression Model with All Variables First of all, we apply simple linear regression model using all variables.

```
eval_results = function(true, predicted, df) {
  SSE = sum((predicted - true)^2)
  SSTO = sum((true - mean(true))^2)
  R_square = 1 - SSE / SSTO
  RMSE = sqrt(SSE/nrow(df))
  # Model performance metrics
  data.frame(RMSE = RMSE,Rsquare = R_square)
}

fit = lm(rings~., data = train)
summary(fit)
```

```
##
## Call:
## lm(formula = rings ~ ., data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.6646 -1.3345 -0.3111  0.8938 11.3927
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       3.482321   0.351186   9.916  < 2e-16 ***
## sexI             -0.925011   0.121352  -7.623 3.34e-14 ***
## sexM             -0.005074   0.099467  -0.051  0.95932
## length           -0.068028   2.137382  -0.032  0.97461
## diameter          9.570951   2.624513   3.647  0.00027 ***
## height           21.616828   2.531386   8.540  < 2e-16 ***
## whole_weight      7.814820   0.867098   9.013  < 2e-16 ***
## shucked_weight -19.112471   0.989585 -19.314  < 2e-16 ***
## viscera_weight  -9.160202   1.542828  -5.937 3.24e-09 ***
## shell_weight     8.089071   1.369178   5.908 3.86e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.182 on 2913 degrees of freedom
## Multiple R-squared:  0.5333, Adjusted R-squared:  0.5318
## F-statistic: 369.8 on 9 and 2913 DF,  p-value: < 2.2e-16
```

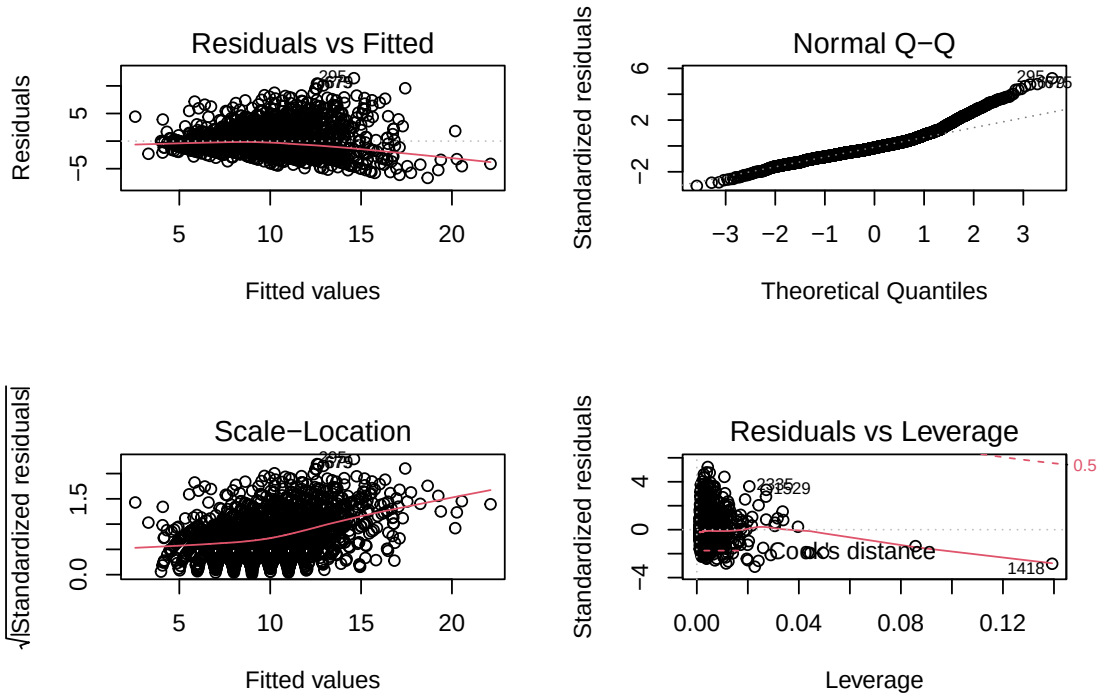
```
lm.train.pred = predict(fit, newdata = train[,-9]) #Edited to work properly
lm.test.pred = predict(fit, newdata = test[,-9]) #Edited to work properly
knitr::kable(as.data.frame(rbind(eval_results(train[,9], lm.train.pred, train),eval_results(test[,9], lm
```

	RMSE	Rsquare
training set	2.178312	0.5332541
test set	2.192011	0.5602397

Based on the given training dataset (`set.seed(1)`) , the R^2 is 0.5333, the R^2_α is 0.5318. In presence of other variables, this model ignores the difference caused by `sex=F`. The p value of `length` is also pretty large, indicating that in presence of other variables, `length` is not statistically significant.

We applied the model on both training set and test set for prediction, The $RMSE$ and R^2 value on both sets are similar, indicating that the model is somehow stable. In other words, the results would not be significantly different on the training and test dataset.

```
par(mfrow = c(2,2))
plot(fit)
```



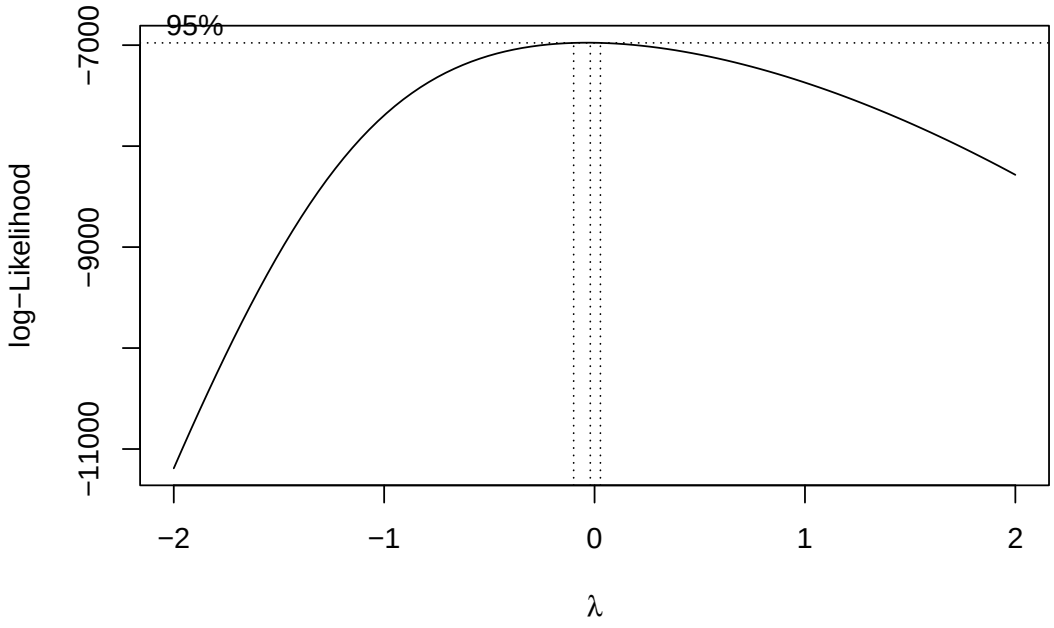
From the residual vs fitted plot, there is a a pattern of moderate heteroscedasticity. In Normal Q-Q plot, the distribution of residuals is heavy tailed. Also, the Scale-Location does not show a straight line, which basically means the full model is not adequate. In the Residual vs Leverage plot, no data point has extreme cook's distance, which shows that there are no extreme influential points in our regression model.

In conclusion, the simple linear regression model with all variables is not a great model. Now we consider to scale the explanatory variables and do Box-Cox transformation for further analysis.

```
train.C = train
train.C[,2:8] = scale(train.C[,2:8]) #scale explanatory variables
test.C = test
test.C[,2:8] = scale(test.C[,2:8])
fit.scaled = lm(rings~., data = train.C)
```

```
boxcoxY = boxcox(fit.scaled)
```

Scaling, Box-Cox Transformation and VIF




```
lambda = boxcoxY$x[which.max(boxcoxY$y)]
```

From the Box-Cox plot, transformation of `rings` is needed to address heteroscedasticity. When λ is approximately 0, SSE is minimized (or log-likelihood is maximized). As such we apply a log transformation to `rings`.

$Y_i^* = log(Y_i)$

```
train.C$rings = log(train.C$rings)
test.C$rings = log(test.C$rings)
```

The variance inflation factor (VIF) is used to measure the amount of multicollinearity. Now we calculate VIF.

```
train.forVIF = train.C
test.forVIF = test.C

fit.vif = lm(rings~., data = train.forVIF)
testVIF = as.data.frame(vif(fit.vif))
knitr::kable(testVIF)
```

	GVIF	Df	GVIF^(1/(2*Df))
sex	1.564982	2	1.118478
length	40.684582	1	6.378447
diameter	41.758109	1	6.462051
height	6.008460	1	2.451216

	GVIF	Df	GVIF^(1/(2*Df))
whole_weight	111.225453	1	10.546348
shucked_weight	29.798655	1	5.458814
viscera_weight	17.472538	1	4.180016
shell_weight	22.163857	1	4.707851

```
while(max(testVIF$GVIF) >= 10)
{
varTBE = rownames(testVIF[testVIF$GVIF == max(testVIF$GVIF),])
train.forVIF = train.forVIF[,!names(train.forVIF)%in%varTBE]
test.forVIF = test.forVIF[,!names(test.forVIF)%in%varTBE]
fit.vif = lm(rings~., data = train.forVIF)
testVIF = as.data.frame(vif(fit.vif))
}

knitr::kable(testVIF)
```

	GVIF	Df	GVIF^(1/(2*Df))
sex	1.511399	2	1.108778
length	8.715446	1	2.952193
height	5.840196	1	2.416650
shucked_weight	6.135415	1	2.476977
shell_weight	7.408978	1	2.721944

We dropped the variables with the highest VIF iteratively if the variable with the highest VIF has a VIF of over 10, but it ultimately made the models less predictive, even if the coefficients ultimately had more stable standard errors. The remaining variables are `sex`, `length`, `height`, `shucked_weight`, and `shell_weight`.

Refit The Model From the analysis above, we know that there is no need to include all variables in the simple linear regression model. Now we perform variable selection in two ways. Although both ways perform forward stepwise procedure using AIC criterion, one model uses all variables and the other one drops variables with high VIF.

```
# variables not dropped
fstnoint <- lm(rings~.,data = train.C)
fstniselect <- stepAIC(lm(rings~1, data = train.C), scope = list(upper = fstnoint, lower = ~1), directi
```

```
## Start:  AIC=-6700.27
## rings ~ 1
##
##                   Df Sum of Sq    RSS     AIC
## + height           1   130.667 164.47 -8407.4
## + diameter         1   129.413 165.72 -8385.2
## + shell_weight     1   126.276 168.86 -8330.4
## + length           1   124.380 170.75 -8297.8
## + whole_weight     1    99.982 195.15 -7907.4
## + viscera_weight   1    92.171 202.96 -7792.7
## + sex              2    72.773 222.36 -7523.9
## + shucked_weight   1    67.743 227.39 -7460.5
```

```
## <none>                       295.13 -6700.3
##
## Step:  AIC=-8407.42
## rings ~ height
##
##                   Df Sum of Sq    RSS     AIC
## + sex              2     6.743 157.72 -8525.8
## + diameter         1     6.504 157.96 -8523.4
## + shell_weight     1     6.020 158.44 -8514.4
## + shucked_weight   1     5.521 158.94 -8505.2
## + length           1     4.529 159.94 -8487.0
## + viscera_weight   1     0.226 164.24 -8409.4
## <none>                         164.47 -8407.4
## + whole_weight     1     0.027 164.44 -8405.9
## - height           1   130.667 295.13 -6700.3
##
## Step:  AIC=-8525.79
## rings ~ height + sex
##
##                   Df Sum of Sq    RSS     AIC
## + shucked_weight   1     7.361 150.36 -8663.5
## + diameter         1     4.333 153.39 -8605.2
## + shell_weight     1     4.292 153.43 -8604.4
## + length           1     2.971 154.75 -8579.4
## + viscera_weight   1     1.060 156.66 -8543.5
## + whole_weight     1     0.424 157.30 -8531.7
## <none>                         157.72 -8525.8
## - sex              2     6.743 164.47 -8407.4
## - height           1    64.637 222.36 -7523.9
##
## Step:  AIC=-8663.5
## rings ~ height + sex + shucked_weight
##
##                   Df Sum of Sq    RSS     AIC
## + diameter         1    20.567 129.79 -9091.4
## + shell_weight     1    19.185 131.18 -9060.5
## + length           1    18.309 132.05 -9041.0
## + whole_weight     1    17.245 133.12 -9017.6
## + viscera_weight   1     2.559 147.80 -8711.7
## <none>                         150.36 -8663.5
## - shucked_weight   1     7.361 157.72 -8525.8
## - sex              2     8.584 158.94 -8505.2
## - height           1    52.561 202.92 -7789.2
##
## Step:  AIC=-9091.45
## rings ~ height + sex + shucked_weight + diameter
##
##                   Df Sum of Sq    RSS     AIC
## + shell_weight     1    9.3473 120.45 -9307.9
## + whole_weight     1    8.5685 121.22 -9289.1
## + viscera_weight   1    0.4197 129.37 -9098.9
## + length           1    0.2427 129.55 -9094.9
## <none>                         129.79 -9091.4
## - sex              2    5.0299 134.82 -8984.3
```

```
## - height          1   10.6193 140.41 -8863.6
## - diameter        1   20.5672 150.36 -8663.5
## - shucked_weight  1   23.5950 153.39 -8605.2
##
## Step:  AIC=-9307.92
## rings ~ height + sex + shucked_weight + diameter + shell_weight
##
##                  Df Sum of Sq    RSS     AIC
## + whole_weight    1     0.715 119.73 -9323.3
## + length          1     0.287 120.16 -9312.9
## + viscera_weight  1     0.112 120.33 -9308.6
## <none>                        120.45 -9307.9
## - sex             2     4.364 124.81 -9207.9
## - height          1     4.400 124.85 -9205.0
## - shell_weight    1     9.347 129.79 -9091.4
## - diameter        1    10.730 131.18 -9060.5
## - shucked_weight  1    31.718 152.16 -8626.6
##
## Step:  AIC=-9323.31
## rings ~ height + sex + shucked_weight + diameter + shell_weight +
##     whole_weight
##
##                  Df Sum of Sq    RSS     AIC
## + viscera_weight  1    1.0896 118.64 -9348.0
## + length          1    0.2293 119.50 -9326.9
## <none>                        119.73 -9323.3
## - whole_weight    1    0.7147 120.45 -9307.9
## - shell_weight    1    1.4935 121.22 -9289.1
## - sex             2    3.9492 123.68 -9232.5
## - height          1    4.0739 123.81 -9227.5
## - diameter        1   10.3539 130.09 -9082.9
## - shucked_weight  1   11.0992 130.83 -9066.2
##
## Step:  AIC=-9348.04
## rings ~ height + sex + shucked_weight + diameter + shell_weight +
##     whole_weight + viscera_weight
##
##                  Df Sum of Sq    RSS     AIC
## + length          1    0.3280 118.31 -9354.1
## <none>                        118.64 -9348.0
## - shell_weight    1    0.6990 119.34 -9332.9
## - viscera_weight  1    1.0896 119.73 -9323.3
## - whole_weight    1    1.6923 120.33 -9308.6
## - sex             2    4.1919 122.83 -9250.5
## - height          1    4.3473 122.99 -9244.8
## - diameter        1   10.8706 129.51 -9093.8
## - shucked_weight  1   12.1508 130.79 -9065.0
##
## Step:  AIC=-9354.13
## rings ~ height + sex + shucked_weight + diameter + shell_weight +
##     whole_weight + viscera_weight + length
##
##                  Df Sum of Sq    RSS     AIC
## <none>                        118.31 -9354.1
```

```
## - length          1    0.3280 118.64 -9348.0
## - shell_weight    1    0.7299 119.04 -9338.2
## - diameter        1    1.0742 119.39 -9329.7
## - viscera_weight  1    1.1884 119.50 -9326.9
## - whole_weight    1    1.6875 120.00 -9314.7
## - height          1    4.2047 122.52 -9254.1
## - sex             2    4.3258 122.64 -9253.2
## - shucked_weight  1   12.3452 130.66 -9066.0
```

```
fstniaic <- lm(fstniselect$call$formula, data = train.C)

summary(fstniaic)
```

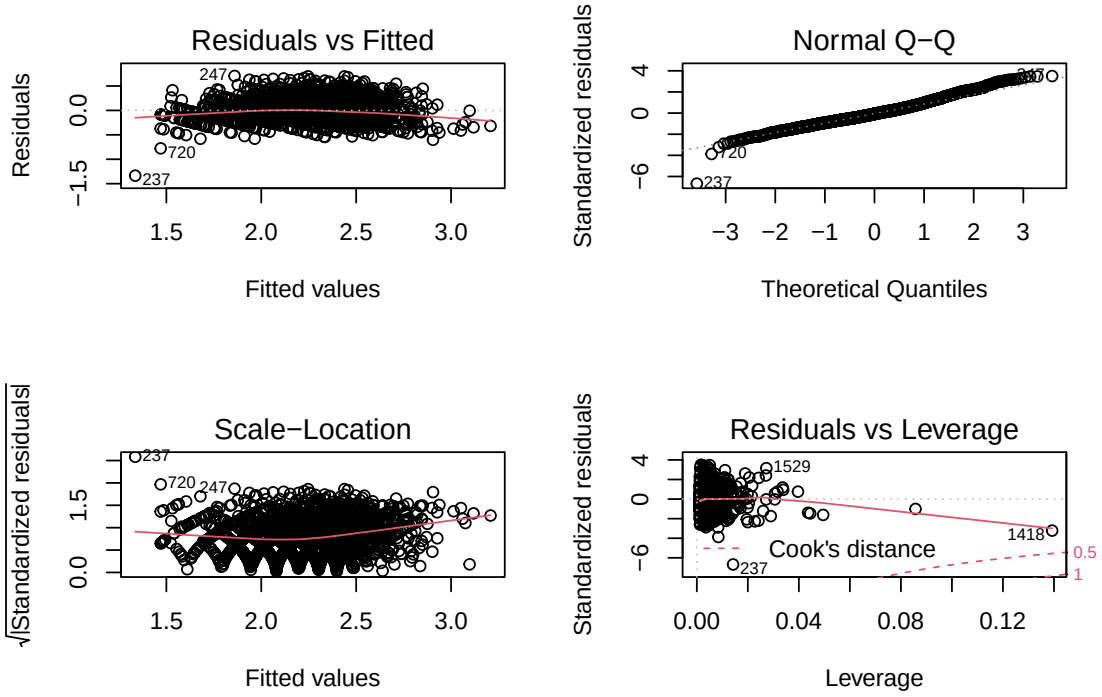
```
##
## Call:
## lm(formula = fstniselect$call$formula, data = train.C)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.33598 -0.13229 -0.01481  0.11270  0.70562
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     2.279303   0.007061 322.788  < 2e-16 ***
## height          0.092985   0.009139  10.175  < 2e-16 ***
## sexI           -0.100397   0.011208  -8.958  < 2e-16 ***
## sexM            0.002231   0.009187   0.243  0.80817
## shucked_weight -0.354819   0.020352 -17.434  < 2e-16 ***
## diameter        0.123898   0.024092   5.143 2.89e-07 ***
## shell_weight    0.074408   0.017552   4.239 2.31e-05 ***
## whole_weight    0.253446   0.039320   6.446 1.34e-10 ***
## viscera_weight -0.084298   0.015584  -5.409 6.85e-08 ***
## length          0.067582   0.023781   2.842  0.00452 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2015 on 2913 degrees of freedom
## Multiple R-squared:  0.5991, Adjusted R-squared:  0.5979
## F-statistic: 483.7 on 9 and 2913 DF,  p-value: < 2.2e-16
```

```
fstniselect$anova[nrow(fstniselect$anova), ]
```

```
## Stepwise Model Path
## Analysis of Deviance Table
##
## Initial Model:
## rings ~ 1
##
## Final Model:
## rings ~ height + sex + shucked_weight + diameter + shell_weight +
##     whole_weight + viscera_weight + length
##
##
```

```
##       Step Df  Deviance Resid. Df Resid. Dev       AIC
## 9 + length  1 0.3280274      2913    118.314 -9354.131
```

```
par(mfrow = c(2,2))
plot(fstniaic)
```




```
fstniaic.train.pred <- predict(fstniaic, newdata = train.C[,!names(train.C)%in%"rings"])
fstniaic.test.pred <- predict(fstniaic, newdata = test.C)
fstniaic_result = rbind(eval_results(train.C$rings, fstniaic.train.pred, train.C),
eval_results(test.C$rings, fstniaic.test.pred, test.C))

knitr::kable(as.data.frame(fstniaic_result,row.names = c('training set','test set')))
```

	RMSE	Rsquare
training set	0.2011887	0.5991151
test set	0.2002388	0.6169438

```
# variables dropped
fstnointVIF <- lm(rings~., data = train.forVIF)
fstnivifselect <- stepAIC(lm(rings~1, data = train.forVIF), scope = list(upper = fstnointVIF, lower = ~
```

```
## Start:  AIC=-6700.27
## rings ~ 1
##
##                  Df Sum of Sq     RSS      AIC
```

```
## + height          1   130.667 164.47 -8407.4
## + shell_weight    1   126.276 168.86 -8330.4
## + length          1   124.380 170.75 -8297.8
## + sex             2    72.773 222.36 -7523.9
## + shucked_weight  1    67.743 227.39 -7460.5
## <none>                        295.13 -6700.3
##
## Step:  AIC=-8407.42
## rings ~ height
##
##                  Df Sum of Sq    RSS     AIC
## + sex             2     6.743 157.72 -8525.8
## + shell_weight    1     6.020 158.44 -8514.4
## + shucked_weight  1     5.521 158.94 -8505.2
## + length          1     4.529 159.94 -8487.0
## <none>                        164.47 -8407.4
## - height          1   130.667 295.13 -6700.3
##
## Step:  AIC=-8525.79
## rings ~ height + sex
##
##                  Df Sum of Sq    RSS     AIC
## + shucked_weight  1     7.361 150.36 -8663.5
## + shell_weight    1     4.292 153.43 -8604.4
## + length          1     2.971 154.75 -8579.4
## <none>                        157.72 -8525.8
## - sex             2     6.743 164.47 -8407.4
## - height          1    64.637 222.36 -7523.9
##
## Step:  AIC=-8663.5
## rings ~ height + sex + shucked_weight
##
##                  Df Sum of Sq    RSS     AIC
## + shell_weight    1    19.185 131.18 -9060.5
## + length          1    18.309 132.05 -9041.0
## <none>                        150.36 -8663.5
## - shucked_weight  1     7.361 157.72 -8525.8
## - sex             2     8.584 158.94 -8505.2
## - height          1    52.561 202.92 -7789.2
##
## Step:  AIC=-9060.47
## rings ~ height + sex + shucked_weight + shell_weight
##
##                  Df Sum of Sq    RSS     AIC
## + length          1    9.9121 121.26 -9288.1
## <none>                        131.18 -9060.5
## - sex             2    6.1843 137.36 -8929.8
## - height          1   14.3814 145.56 -8758.4
## - shell_weight    1   19.1845 150.36 -8663.5
## - shucked_weight  1   22.2535 153.43 -8604.4
##
## Step:  AIC=-9288.14
## rings ~ height + sex + shucked_weight + shell_weight + length
##
```

```
##                   Df Sum of Sq    RSS     AIC
## <none>                         121.26 -9288.1
## - sex              2     4.970 126.23 -9174.7
## - height           1     4.916 126.18 -9174.0
## - length           1     9.912 131.18 -9060.5
## - shell_weight     1    10.788 132.05 -9041.0
## - shucked_weight   1    31.635 152.90 -8612.6
```

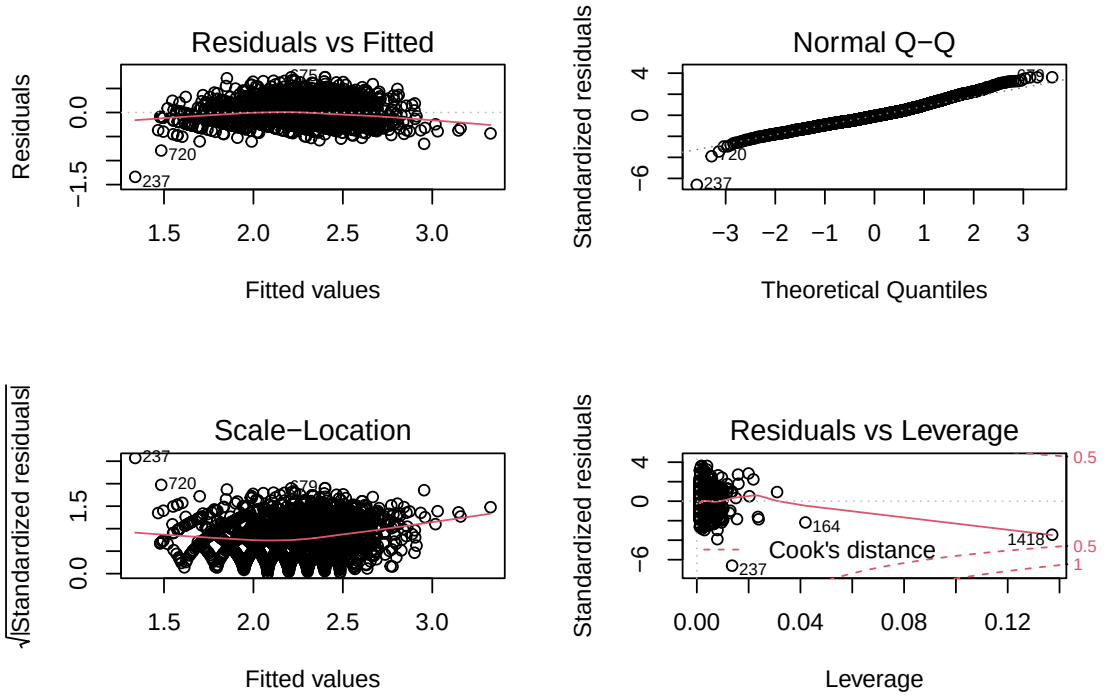
```
fstniaicvif <- lm(fstnivifselect$call$formula, data = train.forVIF)

summary(fstniaicvif)
```

```
##
## Call:
## lm(formula = fstnivifselect$call$formula, data = train.forVIF)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.33903 -0.13294 -0.01671  0.11308  0.73547
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     2.2821177  0.0070792 322.370   <2e-16 ***
## height          0.0991292  0.0091169  10.873   <2e-16 ***
## sexI           -0.1071993  0.0111466  -9.617   <2e-16 ***
## sexM            0.0006087  0.0092620   0.066    0.948
## shucked_weight -0.2577304  0.0093445 -27.581   <2e-16 ***
## shell_weight    0.1653873  0.0102686  16.106   <2e-16 ***
## length          0.1719438  0.0111372  15.439   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2039 on 2916 degrees of freedom
## Multiple R-squared:  0.5891, Adjusted R-squared:  0.5883
## F-statistic: 696.8 on 6 and 2916 DF,  p-value: < 2.2e-16
```

```
fstnivifselect$anova[nrow(fstnivifselect$anova), ]
```

```
## Stepwise Model Path
## Analysis of Deviance Table
##
## Initial Model:
## rings ~ 1
##
## Final Model:
## rings ~ height + sex + shucked_weight + shell_weight + length
##
##
##       Step Df Deviance Resid. Df Resid. Dev       AIC
## 6 + length  1 9.912059      2916   121.2643 -9288.136
```

```
par(mfrow = c(2,2))
plot(fstniaicvif)
```




```
fstniaicvif.train.pred <- predict(fstniaicvif, newdata = train.forVIF[,!names(train.forVIF)%in%"rings"])
fstniaicvif.test.pred <- predict(fstniaicvif, newdata = test.forVIF[,!names(test.forVIF)%in%"rings"])

fstniaicvif_result = rbind(eval_results(train.forVIF$rings, fstniaicvif.train.pred, train.forVIF), eval

knitr::kable(as.data.frame(fstniaicvif_result,row.names = c('training set(dropped)','test set(dropped)'
```

	RMSE	Rsquare
training set(dropped)	0.2036817	0.5891185
test set(dropped)	0.2045199	0.6003893

The Normal Q-Q plot now looks much closer to normal after transforming the data, but there may still be some non-linearity as seen in the residuals vs fitted plot. All variables seem to be significant both before and after dropping highly correlated variables, but standard errors may be unstable based on training sample if not dropping variables.

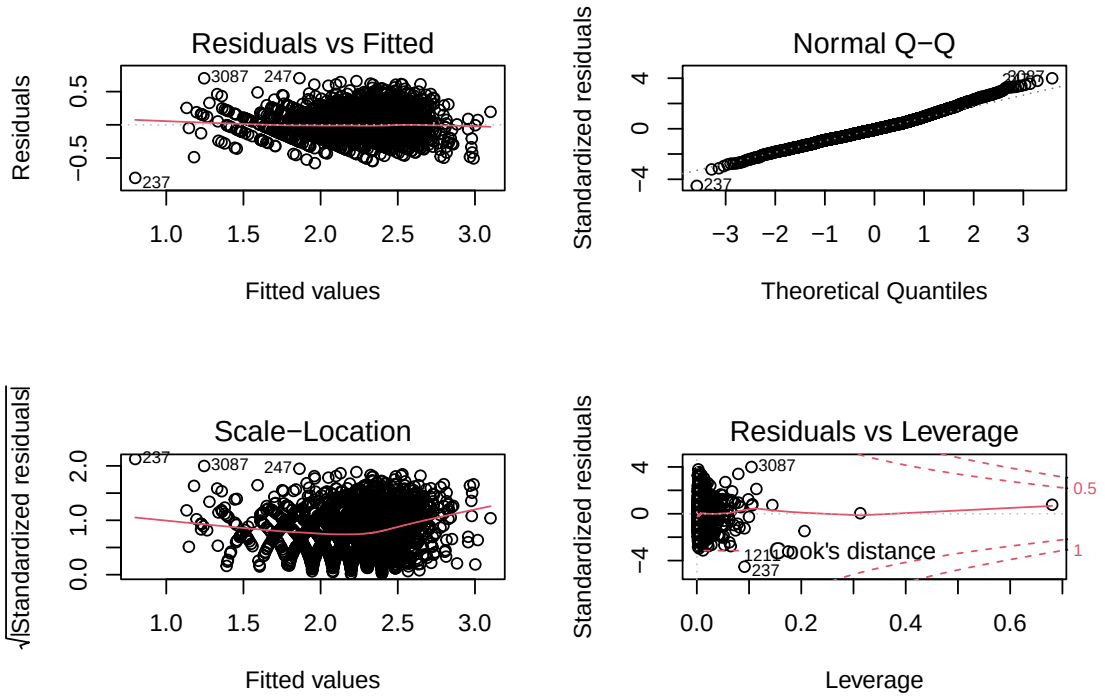
Add Interaction Terms To solve the non-linearity issue of residuals, we consider including interaction terms first. In this part, we also fit two models: all variables included and variables dropped according to VIF.

```r
#variables not dropped
fstintC <- lm(rings~.^2, data = train.C)

fst1C <- stepAIC(lm(rings~1, data = train.C), scope = list(upper = fstintC, lower = ~1), direction = "b
fst1CAIC <- lm(fst1C$call$formula, data = train.C)
```

```r
summary(fst1CAIC)
```

```
##
## Call:
## lm(formula = fst1C$call$formula, data = train.C)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.79885 -0.12242 -0.01285  0.10301  0.70184
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   2.327301   0.008655 268.888  < 2e-16 ***
## height                        0.051764   0.009621   5.380 8.03e-08 ***
## sexI                         -0.021430   0.013526  -1.584 0.113207
## sexM                         -0.004330   0.009772  -0.443 0.657701
## shell_weight                  0.199921   0.024238   8.248 2.41e-16 ***
## shucked_weight               -0.544683   0.031097 -17.516  < 2e-16 ***
## whole_weight                  0.431973   0.050057   8.630  < 2e-16 ***
## viscera_weight               -0.050117   0.014758  -3.396 0.000693 ***
## diameter                      0.078450   0.028290   2.773 0.005589 **
## length                       -0.047180   0.023880  -1.976 0.048281 *
## shucked_weight:whole_weight   0.046546   0.022753   2.046 0.040880 *
## shell_weight:whole_weight    -0.060263   0.026019  -2.316 0.020621 *
## sexI:shucked_weight           0.186863   0.032739   5.708 1.26e-08 ***
## sexM:shucked_weight           0.030191   0.018571   1.626 0.104120
## shucked_weight:diameter       0.196847   0.050273   3.916 9.23e-05 ***
## whole_weight:diameter        -0.093928   0.047621  -1.972 0.048659 *
## shell_weight:shucked_weight -0.066913   0.029616  -2.259 0.023937 *
## sexI:diameter                -0.087408   0.029930  -2.920 0.003523 **
## sexM:diameter                -0.020988   0.023091  -0.909 0.363459
## diameter:length              -0.156941   0.017174  -9.138  < 2e-16 ***
## shell_weight:length           0.053552   0.037785   1.417 0.156504
## height:shucked_weight        -0.023772   0.006190  -3.841 0.000125 ***
## shucked_weight:length         0.065878   0.037453   1.759 0.078691 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1853 on 2900 degrees of freedom
## Multiple R-squared:  0.6625, Adjusted R-squared:  0.6599
## F-statistic: 258.7 on 22 and 2900 DF,  p-value: < 2.2e-16
```

```r
par(mfrow = c(2,2))
plot(fst1CAIC)
```




```
fst1CAIC.train.pred <- predict(fst1CAIC, newdata = train.C[,!names(train.C)%in%"rings"])
fst1CAIC.test.pred <- predict(fst1CAIC, newdata = test.C)

fst1CAIC_result = rbind(eval_results(train.C$rings, fst1CAIC.train.pred, train.C), eval_results(test.C$

knitr::kable(as.data.frame(fst1CAIC_result, row.names = c('training set(interaction terms added)','test
```

	RMSE	Rsquare
training set(interaction terms added)	0.184611	0.6624583
test set(interaction terms added)	0.196552	0.6309197

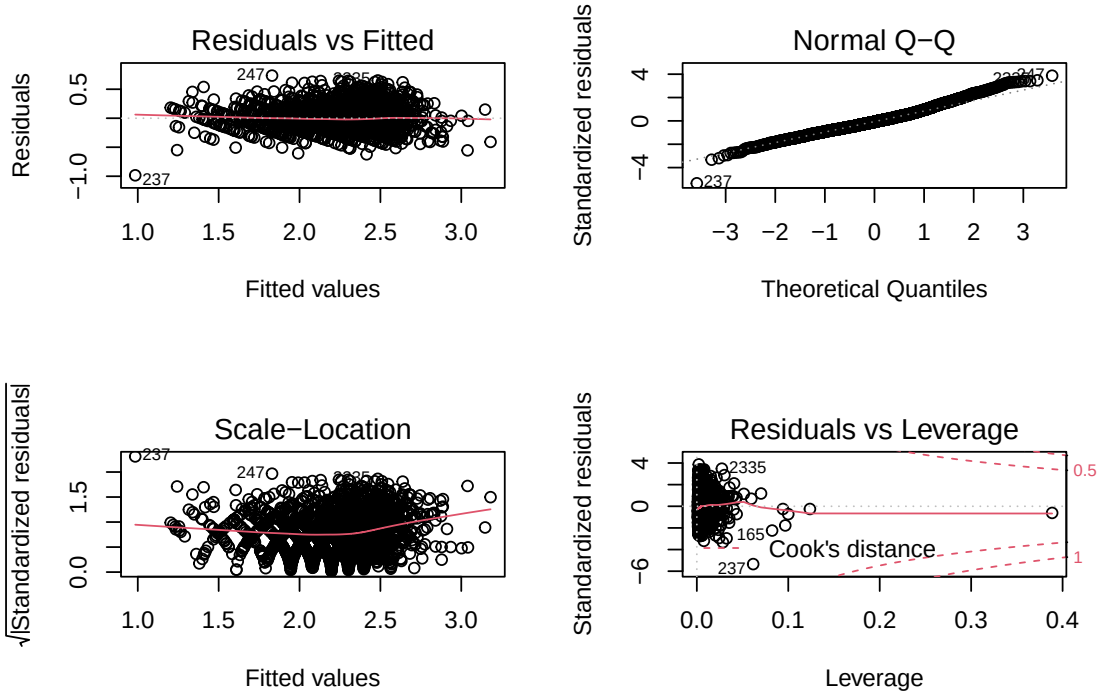
```
#variables dropped
fit.scaled.transformed <- lm(rings~.^2, data = train.forVIF)

fst1 <- stepAIC(lm(rings~1, data = train.forVIF), scope = list(upper = fit.scaled.transformed, lower =
fst1aic <- lm(fst1$call$formula, data = train.forVIF) #after model selection
```

```
summary(fst1aic)
```

```
##
## Call:
## lm(formula = fst1$call$formula, data = train.forVIF)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -0.98423 -0.12627 -0.01538  0.10375  0.73484
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)              2.322885   0.008278 280.598  < 2e-16 ***
## height                   0.073952   0.009562   7.734 1.43e-14 ***
## sexI                    -0.056190   0.012733  -4.413 1.06e-05 ***
## sexM                    -0.012015   0.009433  -1.274   0.2029
## shell_weight             0.364972   0.019321  18.890  < 2e-16 ***
## shucked_weight          -0.339797   0.018541 -18.327  < 2e-16 ***
## length                   0.030900   0.015996   1.932   0.0535 .
## height:shell_weight      0.014119   0.008549   1.652   0.0987 .
## sexI:shucked_weight      0.141113   0.032909   4.288 1.86e-05 ***
## sexM:shucked_weight      0.009964   0.016026   0.622   0.5341
## shell_weight:length     -0.157353   0.014418 -10.914  < 2e-16 ***
## shucked_weight:length    0.171287   0.012524  13.676  < 2e-16 ***
## height:length           -0.066820   0.007877  -8.483  < 2e-16 ***
## sexI:shell_weight       -0.057843   0.031469  -1.838   0.0661 .
## sexM:shell_weight        0.008434   0.016635   0.507   0.6122
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1901 on 2908 degrees of freedom
## Multiple R-squared:  0.6439, Adjusted R-squared:  0.6422
## F-statistic: 375.6 on 14 and 2908 DF,  p-value: < 2.2e-16
```

```
par(mfrow = c(2,2))
plot(fst1aic)
```



```
fst1aic.train.pred <- predict(fst1aic, newdata = train.forVIF[,!names(train.forVIF)%in%"rings"]) #varia
fst1aic.test.pred <- predict(fst1aic, newdata = test.forVIF[,!names(test.forVIF)%in%"rings"])

fst1aic_result = rbind(eval_results(train.forVIF$rings, fst1aic.train.pred, train.forVIF),eval_results(

knitr::kable(as.data.frame(fst1aic_result, row.names = c('training set(variable dropped,interaction ter
```

	RMSE	Rsquare
training set(variable dropped,interaction terms added)	0.1896143	0.6439142
test set(variable dropped,interaction terms added)	0.1988591	0.6222046

After including the model interactions, the non-linearity issue pf residuals seems to have been resolved. Our R^2_α has increased and AIC has dropped. We see that through model selection, `height` is the variable that is chosen first as it explains most of the variation in the response variable, `rings`.

Note that as the prediction error is not the only criterion for model selection, although the model built and selected without variable selection has slightly better training and test error, it is more unstable and less interpretable due to high standard errors on coefficients.

```
fst1$anova[nrow(fst1$anova),] #after variable selection
```

```
## Stepwise Model Path
## Analysis of Deviance Table
##
## Initial Model:
## rings ~ 1
##
## Final Model:
## rings ~ height + sex + shell_weight + shucked_weight + length +
##     height:shell_weight + sex:shucked_weight + shell_weight:length +
##     shucked_weight:length + height:length + sex:shell_weight
##
##
##                              Step Df   Deviance Resid. Df Resid. Dev       AIC
## 18 - shell_weight:shucked_weight  1 0.06868626      2908   105.0923 -9690.513
```

```
fst1C$anova[nrow(fst1C$anova),] #all variables included
```

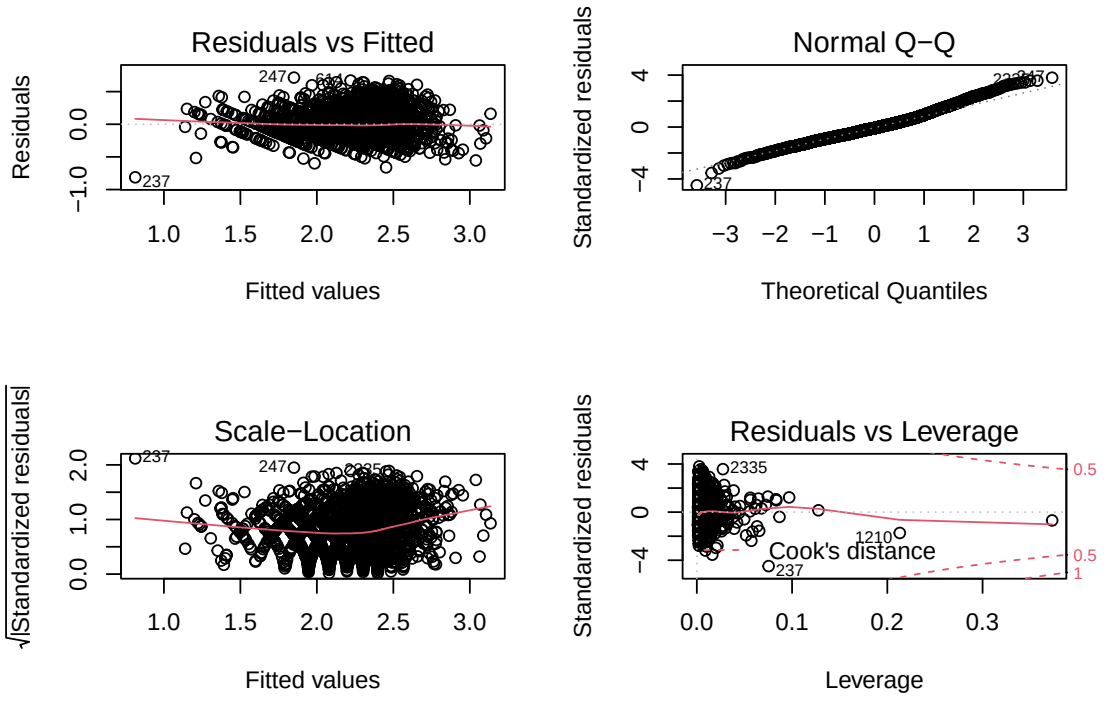
```
## Stepwise Model Path
## Analysis of Deviance Table
##
## Initial Model:
## rings ~ 1
##
## Final Model:
## rings ~ height + sex + shell_weight + shucked_weight + whole_weight +
##     viscera_weight + diameter + length + shucked_weight:whole_weight +
##     shell_weight:whole_weight + sex:shucked_weight + shucked_weight:diameter +
##     whole_weight:diameter + shell_weight:shucked_weight + sex:diameter +
##     diameter:length + shell_weight:length + height:shucked_weight +
```

```
##     shucked_weight:length
##
##
##                               Step Df   Deviance Resid. Df Resid. Dev       AIC
## 28 - whole_weight:viscera_weight  1 0.03773443      2900   99.61938 -9830.843
```

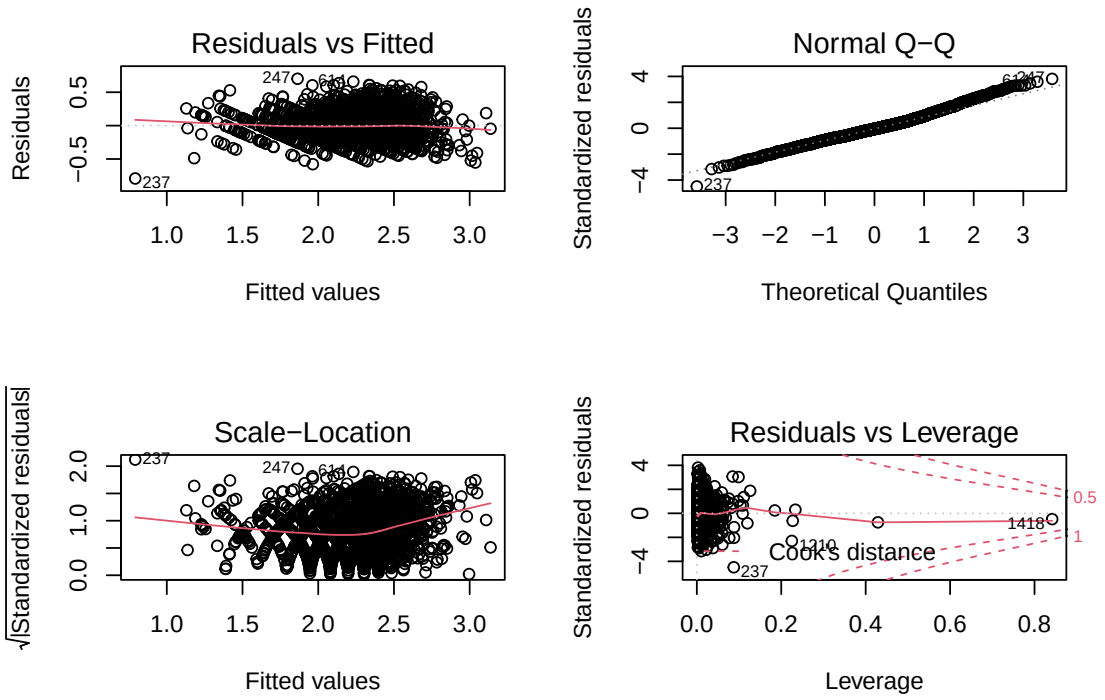
We can see that our AIC has improved meaningfully after including interaction terms.

Note that the reason we did not include second-order terms is that it does not meaningfully improve our model. *AIC* does very slightly improve, but things get slightly worse if using an information condition that had a higher penalty for more terms such as *BIC* (same function but with $k = log(n)$).

```
fitsecond.scaled.transformed = lm(as.formula(paste('rings ~ .^2 +', paste('I(', colnames(train.forVIF[,
fstnodrop <- lm(as.formula(paste('rings~.^2 + ', paste('I(', colnames(train.C[,!names(train.C) %in%c("r

fstaicvif <- stepAIC(lm(rings~1, data = train.forVIF), scope = list(upper = fitsecond.scaled.transforme
fstvif <- lm(fstaicvif$call$formula, data = train.forVIF)

fstaicc <- stepAIC(lm(rings~1, data = train.C), scope = list(upper = fstnodrop, lower = ~1), direction =
fstc <- lm(fstaicc$call$formula, data = train.C)

#summary(fitsecond.scaled.transformed)
summary(fstvif)

#summary(fstnodrop)
summary(fstc)

par(mfrow = c(2,2))
plot(fstvif)
```



```
plot(fstc)
```




```
fitsecond.train.pred <- predict(fstvif, newdata = train.forVIF[,!names(train.forVIF)%in%"rings"])
fitsecond.test.pred <- predict(fstvif, newdata = test.forVIF[,!names(test.forVIF)%in%"rings"])
eval_results(train.forVIF$rings, fitsecond.train.pred, train.forVIF)
eval_results(test.forVIF$rings, fitsecond.test.pred, test.forVIF)

fitsecond.train.pred <- predict(fstc, newdata = train.C[,!names(train.C)%in%"rings"])
fitsecond.test.pred <- predict(fstc, newdata = test.C[,!names(test.C)%in%"rings"])
eval_results(train.C$rings, fitsecond.train.pred, train.C)
eval_results(test.C$rings, fitsecond.test.pred, test.C)
```

```
fstaicvif$anova[nrow(fstaicvif$anova),]
```

```
## Stepwise Model Path
## Analysis of Deviance Table
##
## Initial Model:
## rings ~ 1
##
## Final Model:
## rings ~ height + I(length^2) + shell_weight + shucked_weight +
##     sex + I(shucked_weight^2) + length + height:shell_weight +
##     shell_weight:shucked_weight + shucked_weight:sex + shucked_weight:length +
##     sex:length
##
##
```

```
##            Step Df  Deviance Resid. Df Resid. Dev       AIC
## 13 + sex:length  2 0.4027871      2907   103.0695 -9745.325
```

```
fstaicc$anova[nrow(fstaicc$anova),]
```

```
## Stepwise Model Path
## Analysis of Deviance Table
##
## Initial Model:
## rings ~ 1
##
## Final Model:
## rings ~ height + I(length^2) + shell_weight + shucked_weight +
##     sex + whole_weight + I(shucked_weight^2) + viscera_weight +
##     I(viscera_weight^2) + diameter + I(diameter^2) + length +
##     I(height^2) + shell_weight:whole_weight + shucked_weight:sex +
##     height:shucked_weight + shucked_weight:diameter + shucked_weight:whole_weight +
##     whole_weight:length + sex:length + viscera_weight:diameter
##
##
##             Step Df  Deviance Resid. Df Resid. Dev       AIC
## 22 + I(height^2)  1 0.1151671      2898   98.47267 -9860.685
```

Conclusion