Abalone Age Prediction

STA206 Final Project

Xiquan Jiang (xiqjiang@ucdavis.edu)
Cindy Li (whali@ucdavis.edu)
Bo Zhang (bzbzhang@ucdavis.edu)

Abstract

Abalones are well known for their rarity and delicacy. Abalones are expensive and their prices are related to their ages. Do you know that estimating the age of abalone is similar to measuring the age of trees? The shells are cut and rings in them are counted. However, the current method for estimating age is costly and inefficient. Thus, researchers are dedicated to finding more scientific ways for better forecasting the age. In this project, the group is interested in developing predictive models for estimating abalone age with higher accuracy, finding the model with best performance, and obtaining the most significant indicators for prediction. A comprehensive data preprocessing is done to tidy the data set, including handling missing values and outliers and train-test split. Exploratory data analysis is conducted to better understand the data, such as the distributions and correlations. After that, four regression models, including multiple linear regression, forward stepwise regression, LASSO regression, and random forest are built, validated, and evaluated. As a result, the stepwise regression, LASSO regression, and random forest models have their own advantages and limitations. Also, further discussions are made for the models, data set, and the entire analysis process at the end of this project.

Introduction

Background

Abalone is a famous kind of sea snail, which is known as one of the rarest and finest delicacies all around the world. They have been considered as a precious source of food for human beings for a long time. Additionally, jewelries made by abalone pearls are popular in some regions, such as Australia and New Zealand. Abalones have to be gathered by hands from the ocean. As a result, their rarity, popularity, and difficulty in obtaining them contribute to the extremely high cost of abalones.

Project Motivation

It is shown in a wide range of research that the price level of abalone is closely correlated with its age. Thus, it is meaningful to determine the age accurately to get better ideas of the price for both customers and producers. Similarly, to the growth of trees, rings are formed in abalone shells as it grows at a common rate of one ring per year and the number of rings is used to determine the age. However, the commonly used aging techniques are costly and time-consuming. The rings are counted manually under microscopes after abalone shells are cut, polished, and stained. This traditional way of estimation is not quite accurate as well, since some rings are hard to be revealed so researchers use a reasonable approximation by adding 1.5 ring count to the results. Therefore, finding better age prediction methods can improve the operation efficiency of the abalone industry and its popularity worldwide.

Questions of Interest

The main objective for this project is to obtain models for predicting the rings (age) of abalone with higher accuracies. Also, the best indicators for forecasting rings can be determined based on the model.

Data Description

The Abalone Age data set is retrieved from UC Davis Canvas. There are nine variables included in the file, with “Sex”, “Length”, “Diameter”, “Height”, “Whole weight”, “Shucked weight”, “Viscera weight”, “Shell weight” as predictors, and “Rings” as response variable. Among the predictors, “Sex” is containing qualitative values with three levels, M, F, and I, while the rest of predictors contain quantitative values. There are 4177 observations in the data set.

Methods

Data Preprocessing & EDA

Missing Values & Outliers

There is no missing value in the abalone dataset, thus there is no adjustment needed. According to the boxplot of this dataset (Figure 1), we could see that there are several outliers which are outside of the range (-3σ, 3σ) for each variable. According to the boxplot for the rings, the range of rings is mainly between 5 to 15, with several older abalone having 15-30 rings, and some younger abalone having less than 5 rings. As we checked these outliers, we decided to keep them since there isn't enough evidence to show that these are incorrect values. Other than that, as we looked into the boxplot of the height, we found that there are several data points with height = 0. However, all other variables for these points are not equal to 0. Therefore, based on our judgement, we decided to remove these data points from our dataset since these should be considered as incorrect values.

Data Transformation

According to the histogram (Figure 2), the distribution of all variables shows that 'Length', 'Whole weight' and 'rings' looks approximately normal; thus, we do not need to make any transformation for these three variables. The Histogram for 'Diameter' is clearly left-tailed (Figure 3), therefore we made a square transformation to make it approximately normal distributed. As for the distribution of 'Shucked weight', 'Viscera weight', and 'Shell weight', (Figure 4, these three weights have similar distributions) which are clearly right tailed, we made a square root transformation to make them approximately normal.

Train-test Split

After the data preprocessing, we spitted the data set into training set(80%) and testing set(20%) in order to construct our model in order to predict the rings, and check the precision of our prediction results.

Data Collinearity

Before constructing our model, we first checked the distribution of all paired predictors and the correlation within each pair (Figure 5a). According to the histogram for each pair of variables based on our categorical variable 'sex' separately in Figure 5, it shows that the distribution for male and female abalone are almost same, but the value of infant abalone on every feature are all smaller than those of male and female. Thus, we decided to combine the male and female categories, reform the sex variable to two categories: infant and non-infant. After that we checked the correlation plot (Figure 5b), as we can see, there exists strong multicollinearity within 'length', 'diameter', 'height', 'whole weight', 'shucked weight', 'viscera weight' and 'shell weight'. There is a clear linear relationship between 'length', 'diameter', and 'height'; 'length' and 'diameter' have a strong nonlinear relationship with four weight variables separately. Thus, we will handle this in our model in order to eliminate the influence of multicollinearity.

Models & Results

Multiple Linear Regression

We have our first model using multiple linear regression using all 8 variables. After fitting the model to the data, we obtained the following result:

$$y_{Rings} = 2.5463 + \quad 0.775x_{Sex} - 8.4163x_{Diameter} + 5.2799x_{Length} + 18.3789x_{Height} + 0.9108x_{Whole\ weight} - 15.9035x_{Shucked\ weight} - 1.8638x_{Viscera\ weight} + 25.8688x_{Shell\ weight}$$

This model generated an MAE = 1.673, an MSE = 5.675, and a NMSE = 0.521, which may have space for improvements. We also used the Variance Inflation Factor to measure how much the variance of the estimated regression coefficient will increase because of collinearity. From the model outputs, we realized that, except for the VIF of 'Sex' and 'Height', all other VIFs are extremely high (> 10), which means attention is needed for the multicollinearity issues within predictors. Based on these facts, it seems reasonable to examine the effects of adding interaction terms of variables to the model and identify the significant ones among all of them.

Forward Stepwise Regression

Since the performance of multiple regression model is not ideal, we decided to train forward stepwise regression on the data with a null model as the lower limit and full model as upper limit, shown as the following:

Null Model: A basic model with no X variable.

$$y_{Rings} = \beta_0 + \varepsilon$$

Full Model: A model with all first order X variables and interaction terms.

$$y_{Rings} = \beta_0 + \quad \beta_1 x_{Sex} + \beta_2 x_{Diameter} + \beta_3 x_{Length} + \beta_4 x_{Height} + \beta_5 x_{Whole\ weight} + \beta_6 x_{Shucked\ weight} + \beta_7 x_{Viscera\ weight} + \beta_8 x_{Shell\ weight} + \beta_9 x_{Sex} * x_{Diameter} + \ldots\ldots + \beta_{36} x_{Viscera\ weight} * x_{Shell\ weight} + \varepsilon$$

Applying the "stepAIC( )" function embedded in R, the model with smallest AIC (=4871.282) value was selected, which is displayed as below:

$$y_{Rings} = 2.133 - 0.013 \quad x_{Sex} - 3.955x_{Length} + 23.559x_{Height} + 12.003x_{Whole\ weight} - 0.618x_{Shucked\ weight} - 11.66x_{Viscera\ weight} + 29.911x_{Shell\ weight} - 4.508x_{Shucked\ weight}x_{Sex} - 69.652x_{Shell\ weight}x_{Shucked\ weight} + 17.294x_{Shell\ weight}x_{Sex} + 29.66x_{Shell\ weight}x_{Viscera\ weight} - 4.33x_{Sex}x_{Whole\ weight} + 12.867x_{Shucked\ weight}x_{Whole\ weight} - 12.986x_{Viscera\ weight}x_{Whole\ weight} - 13.004x_{Sex} * x_{Height}$$

From the stepwise regression, we obtained an MAE = 1.620, MSE = 5.074, and NMSE = 0.466, which are slightly better than what we got from the MLR model. However, the extremely high VIF values showed that there was still significant multicollinearity within predictors, which indicated that there were reductions for model complexity and multicollinearity required for the model.

Ridge Regression

Since there exists strong multicollinearity within predictors based on our previous results, we could apply the ridge regression and LASSO Regression on our dataset, in order to reduce the model complexity and eliminate the influence of multicollinearity. The key to ridge regression is to find a reasonable λ value to balance the variance and deviation of the model. The larger λ makes the estimated value of β more deviated from the true value, and the deviation of the model larger. According to the cost function of Ridge regression,

$$\sum(y_i - \widehat{y_i})^2 = \sum(y_i - \sum w_j \times x_{ij})^2 + \lambda \sum w_j^2$$

Using the cross-validation method to choose λ, we select the λ with the minimum value and the λ with 0.5 times the standard error. However, it can be found that the ridge regression cannot eliminate the variables and is not significantly useful for building the model, so we decided to apply the LASSO regression based on the results of ridge regression, for the purpose of getting a more efficient model.

LASSO Regression

Based on the previous results, the LASSO regression model changes the penalty term from the L2 norm to the L1 norm, which can reduce some unimportant regression coefficients to 0 to achieve the purpose of eliminating multicollinearity and doing the feature selection. Using the cost function of LASSO regression,

$$\sum(y_i - \hat{y_i})^2 = \sum(y_i - \sum w_j \times x_{ij})^2 + \lambda \sum |w_j|$$

We directly use the cross-validation method to select λ. Firstly, we selected the minimum value of λ and found that there is no variable to be filtered out. Then, we tried λ with the standard error to filter out 'SexI', 'SexNI', 'Height', 'Shucked weight' and 'Shell weight' as predictors. Compared with Ridge Regression, LASSO regression can select those features which could be completely neglected in the model. With LASSO regression, we obtained MAE = 1.683, MSE = 5.692, and NMSE = 0.523.

Random Forest

Random forest uses bootstrap and bagging to train the data, which are algorithms for improving accuracy of machine learning models. Since the performance of the previous models is not quite ideal, we decided to fit another random forest algorithm on the data set. After plotting the error vs trees graph, we realized that the model error decreases with increasing tree number. By trial and error, we set the number of trees to 800. The random forest model gave us an MAE = 1.570, MSE = 4.950, and a NMSE = 0.454.

Results

We decided to use LASSO regression as the final model, which has the best performance in reducing multicollinearity. The model diagnostic plots (Figure 12) proved that the assumptions for the regression model are satisfied: the residuals are approximately normally distributed, independent of one another, and have a constant variance. According to the ANOVA table and the summary of our model, all the p-values for each parameter are extremely small, thus we could conclude that sex, height, shucked weight, and shell weight are significant for our model.

However, since the AIC value of the LASSO regression model is too high and it tends to choose some variables over others, resulting in model instabilities. We also used an extra random forest model to help validate the results we got from LASSO regression. It turns out that the random forest model has the best MAE, MSE, and NMSE among all four models. The variable significance level (Table 3) is ranked as: Shucked weight > Shell weight > SexNI > SexI > Viscera weight > Height > Whole weight > Diameter > Length. We also design a new measure parameter, if the absolute difference between actual and predicted ages is smaller than or equal to 1, we would consider it as a good age prediction, and get the accuracy percentage by using the amount of good age prediction over the total number of data points. We could clearly see that for abalone that have less than 12 rings, the prediction accuracy is 61.46%; the prediction accuracy for those that have more than 12 rings is 27.59%.

Conclusions & Discussion

Based on our model, our results show that the rings(age) are highly correlated with 'Sex(infant/non-infant)', 'height', 'shucked weight', 'shell weight'. Which means that as the height increases, the shucked weight and shell weight also increases, and the rings number will increase. In general, as the abalone grows up, the size of it is growing up, and the weight is not surprisingly increasing, which would be a reasonable relationship for us to predict the age of an abalone during growth period using these parameters.

We first conclude that after the abalone grow up as an adult abalone, it doesn't matter what sex it is, it all shows a similar pattern for all the dimensions of the size. These results show that all the dimensions for abalone would increase until approximately 12 rings, after that the increased slope gets smaller or even has a declined trend.

As we looked into our final predictions, the results proved our previous inference. We could clearly see that for abalone that have less than 12 rings, the prediction accuracy is more than 61%, but as compared to those that have more than 12 rings, the prediction accuracy is around 28%. Thus, our model is in general not very accurate for predicting the age of abalone, however, it's doing better for predicting those younger abalone. This may be because abalone keeps growth in every dimension until it achieves 12 rings; after that, as the abalone gets older, the size of it would shrink and decline. Nevertheless, since there isn't much data for abalone with more than 15 rings in our dataset (261 data), we could conclude that the decrease in every dimension of abalone may also have other 2 reasons:

1. Larger abalones are already being caught when they first obtain the threshold size for capture, since considering the commercial purpose for abalone, people would harvest it once they reach the capture size in order to decrease the cost and increase the profit.
2. Dead elder abalone has a larger proportion than younger ones, thus the sample for large amounts of rings are less.

Our project still has several limitations. First of all, if we could get more data for the elder abalone and build the model for abalones which already passed the growing period separately, it may be more accurate to predict the age for them. Besides, we could include other parameters when we prepare the dataset which are more significant for elder abalone in order to determine the age more accurately, for example, find what characteristics does elder abalone have, and make a new parameter to measure it. As for our model building process, we could also include other advanced technologies to build our model and do the optimization of the model, since all our three model having limitations: the running time of random forest is unexpected long, the multicollinearity of stepwise regression is too significant, and LASSO has largest AIC among these three models, and will neglect several features which may lead to the unstable of the results.

Appendices

Appendix 1

Figure 1 Box plot for quantitative predictors

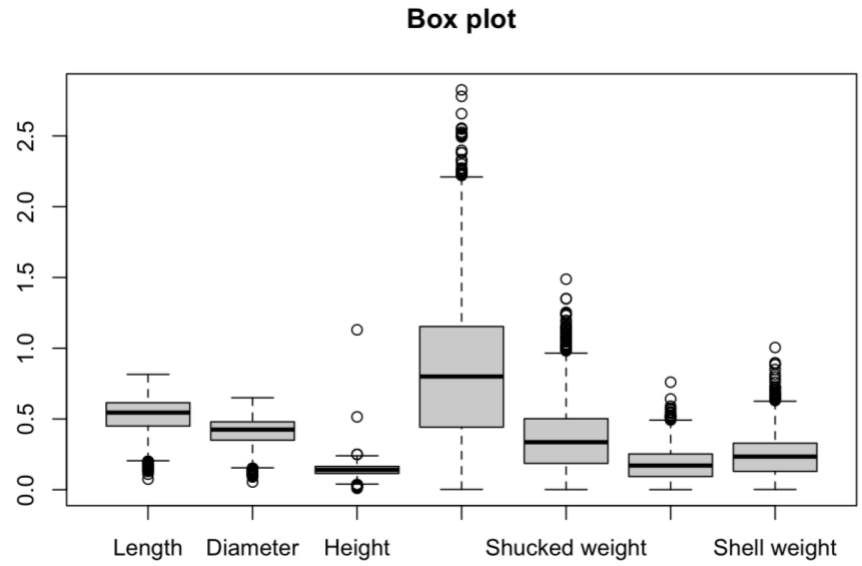

Figure 2 Histogram of quantitative variables

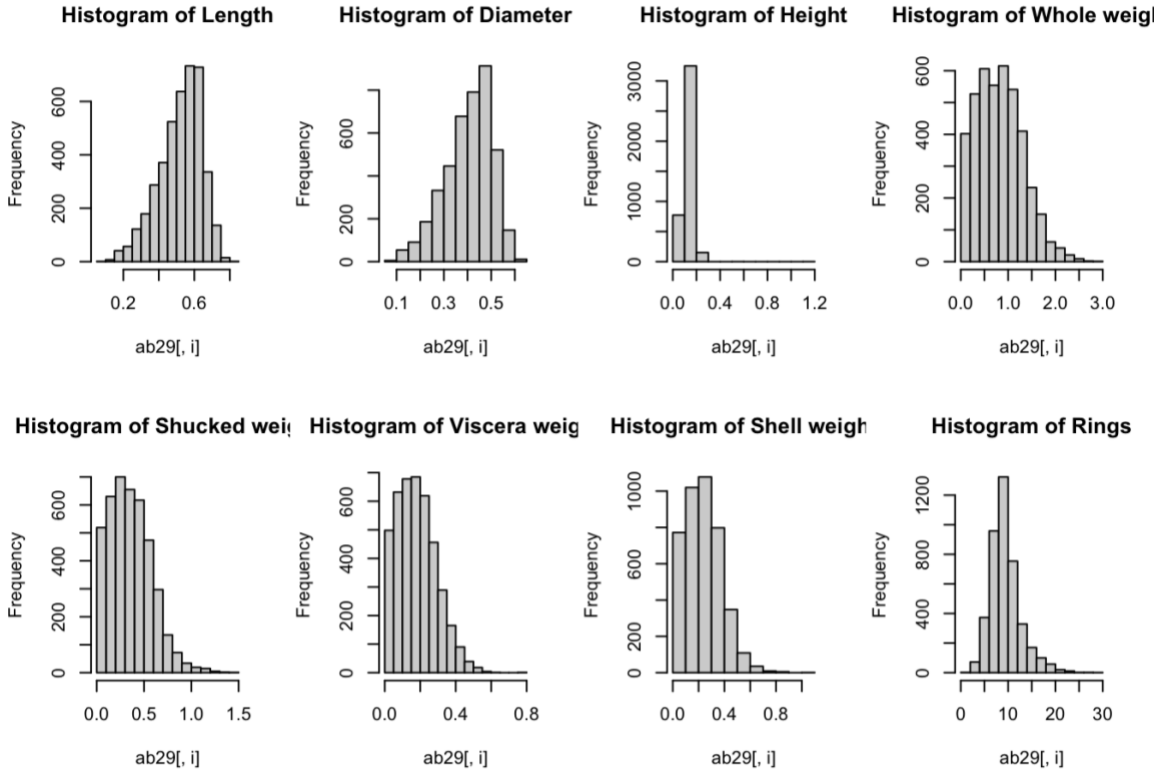

Figure 3 Transformation for variable ‘Diameter’

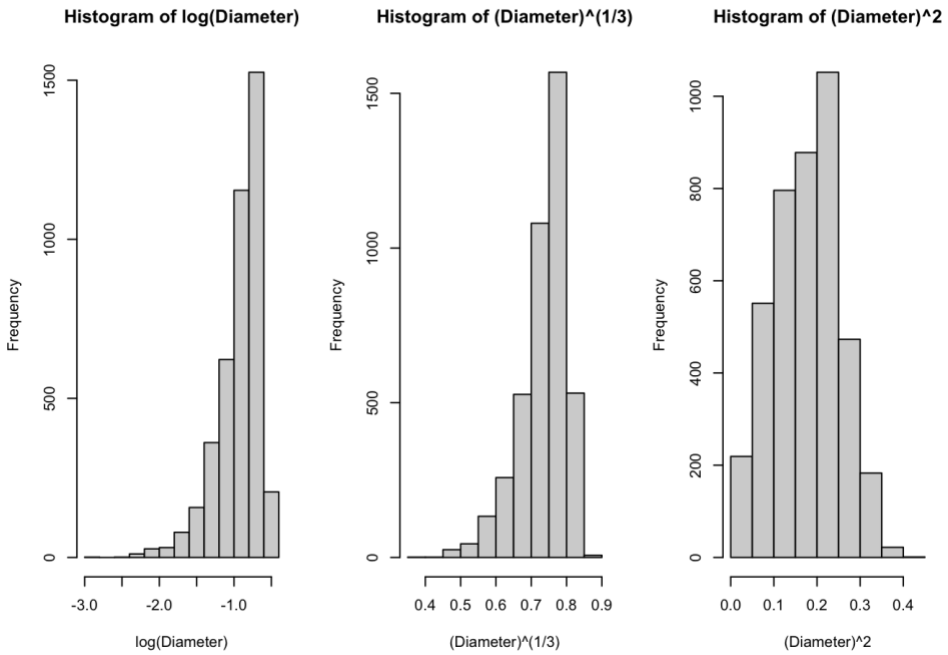

Figure 4 Transformation for variable ‘Shucked weight’

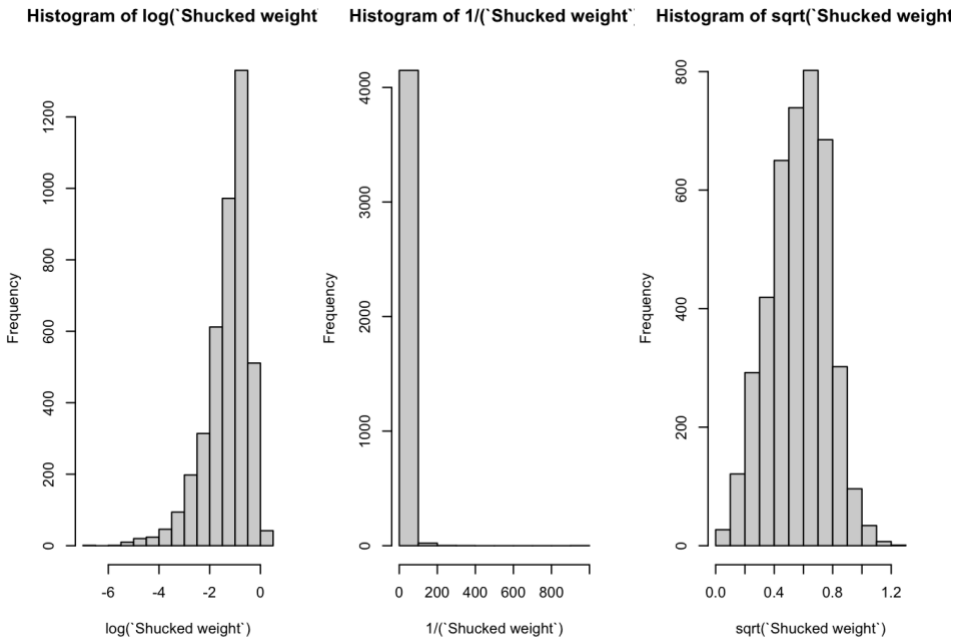

Figure 5a Pair Plot

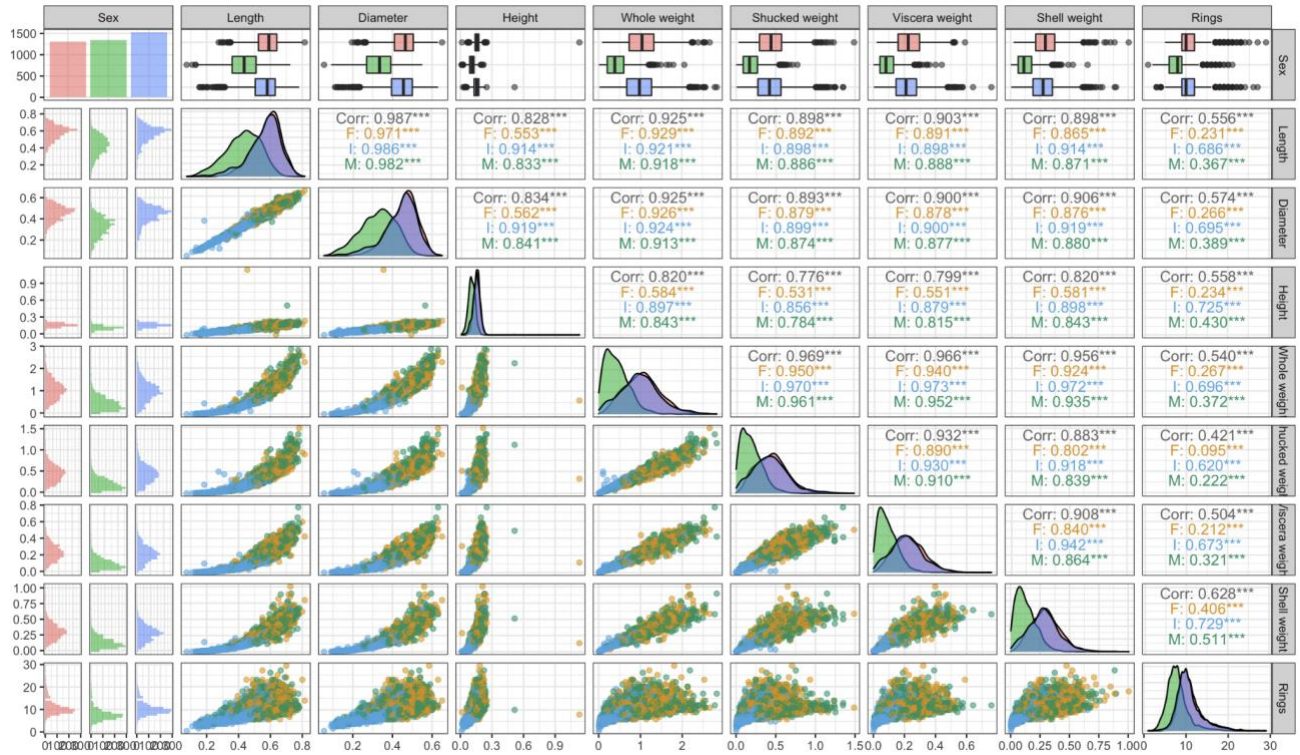

Figure 5b Pair Plot after combined Male and Female

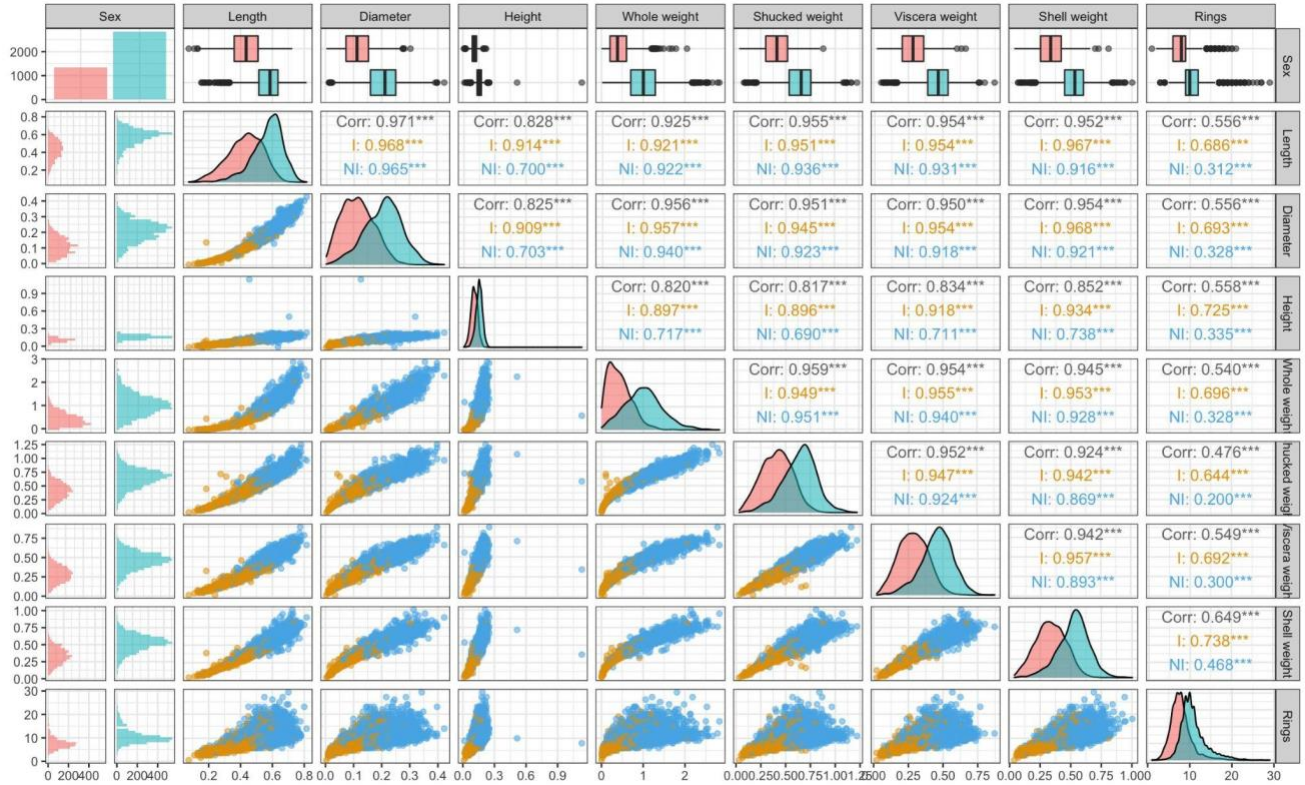

Figure 6 Scatterplot

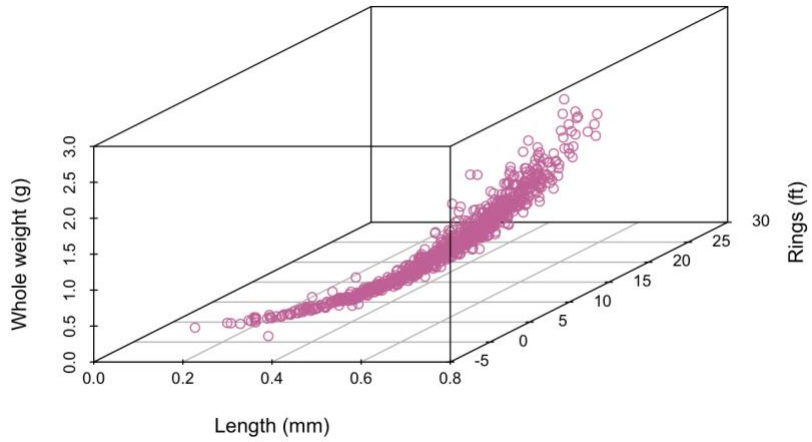

Figure 7 Ridge Regression

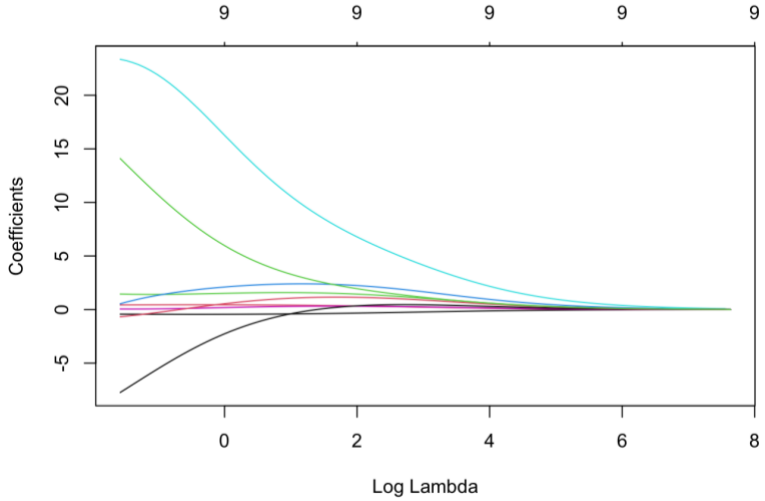

Figure 8 MSE Ridge

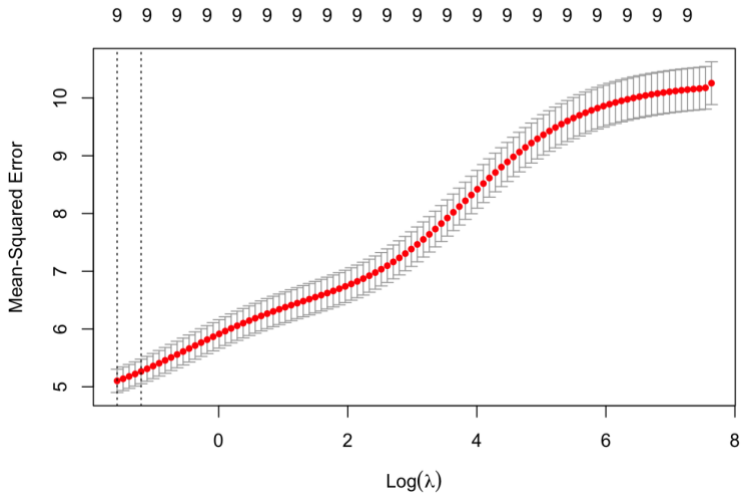

Figure 9 Lasso Regression

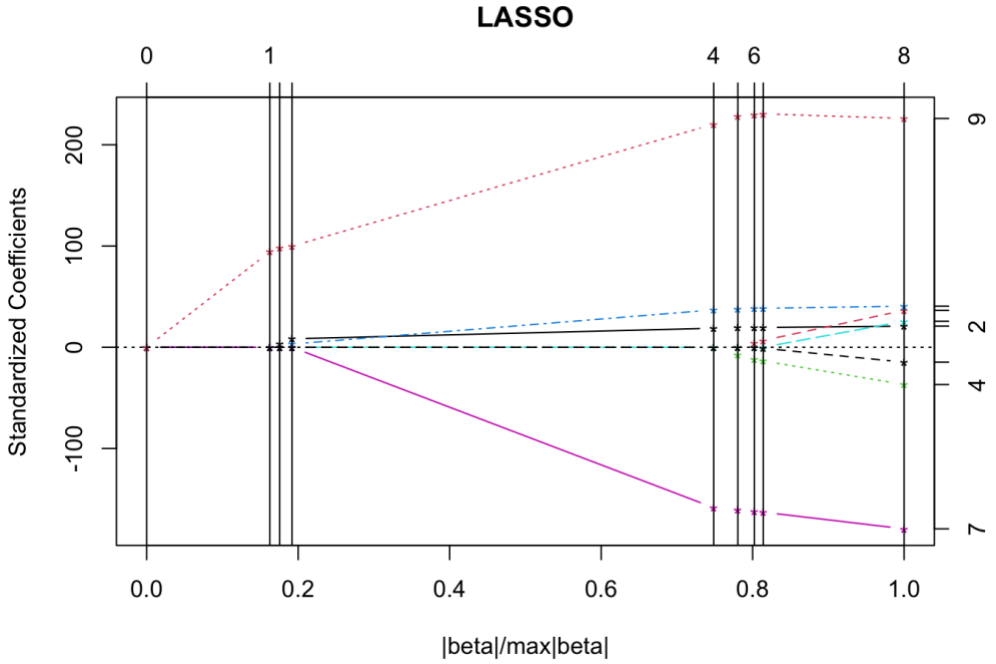

Figure 10 MSE of LASSO Regression

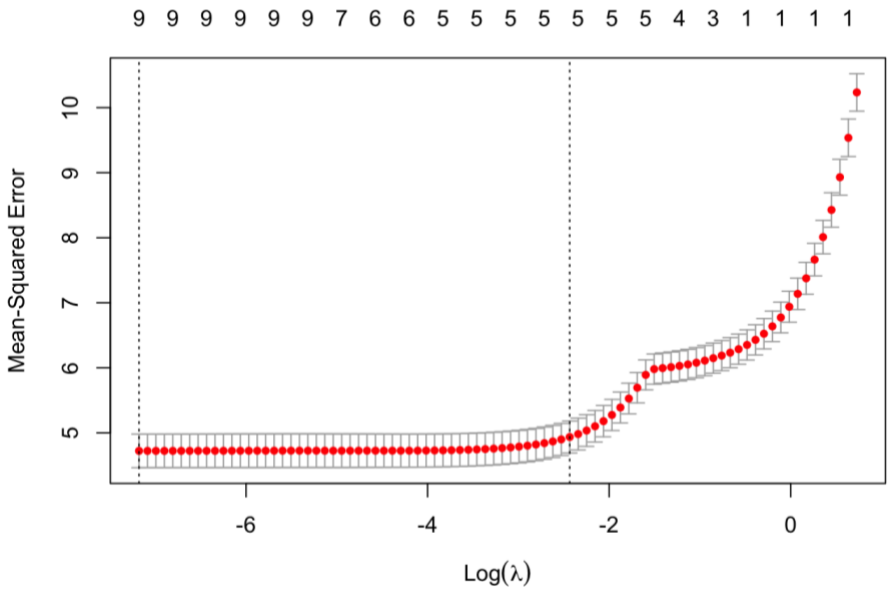

Figure 11 Error Curve of Random Forest

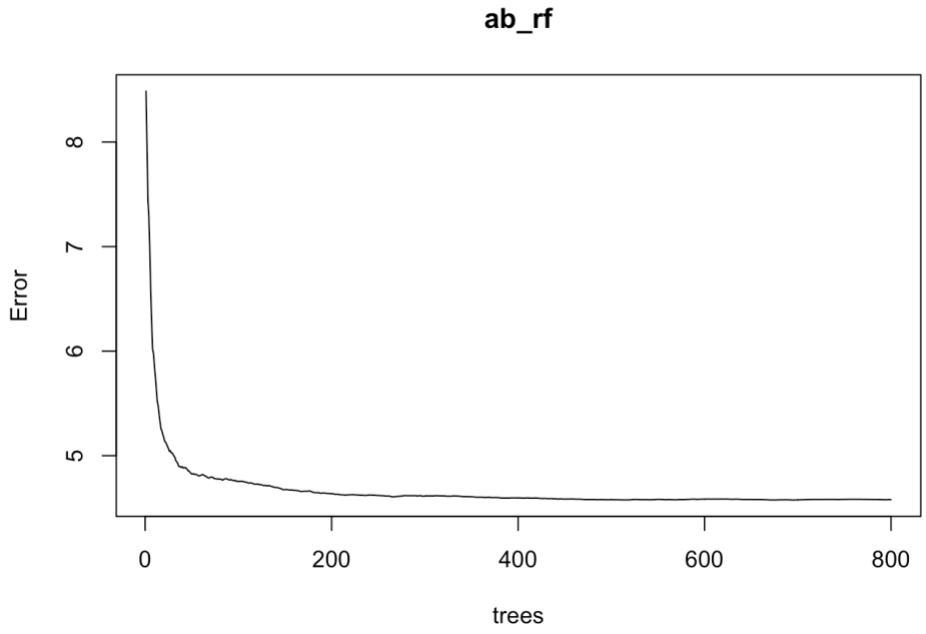

Figure 12 Residual Plot and Q-Q Plot

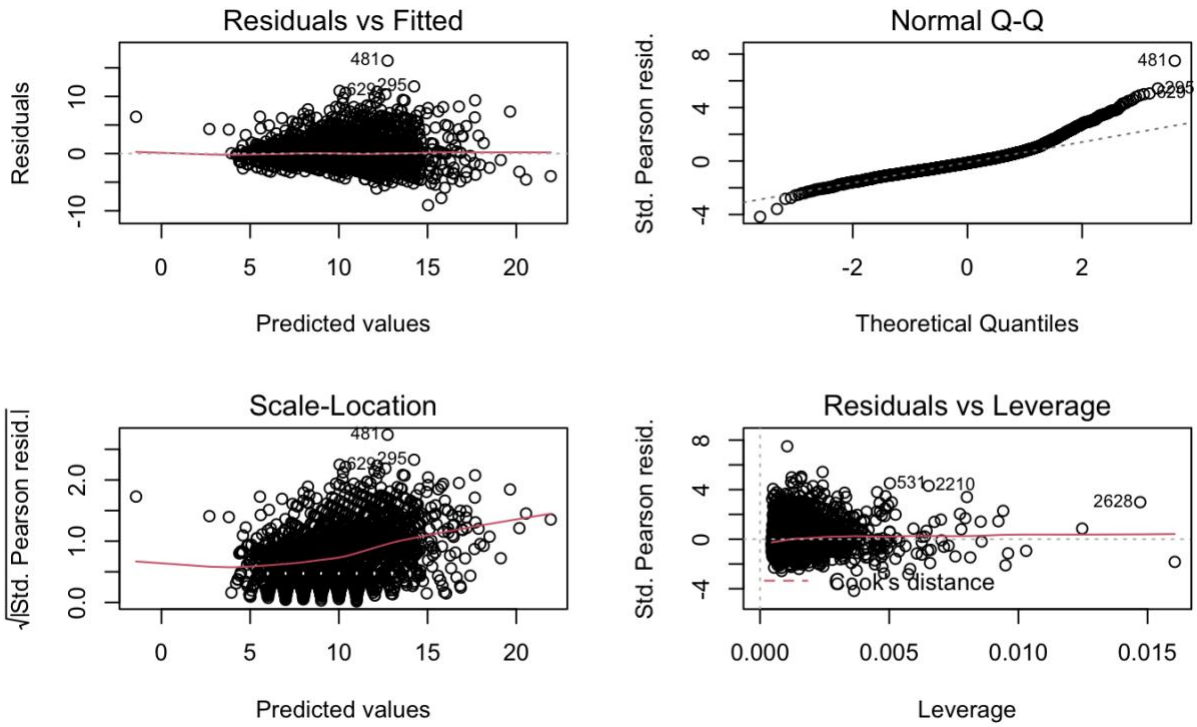

Table 1 VIF for linear regression

```
##              Sex         Diameter           Length           Height
##         1.587514        30.998651        38.191040         7.756325
##   `Whole weight` `Shucked weight` `Viscera weight`   `Shell weight`
##        30.456240        23.922272        20.080857        21.745193
```

Table 2 VIF for stepwise regression

```
##                   `Shell weight`                 `Shucked weight`
##                        158.86003                        331.03725
##                              Sex                           Height
##                         73.06363                         37.06941
##                   `Whole weight`                 `Viscera weight`
##                        288.63170                        389.43241
##                           Length             `Shucked weight`:Sex
##                         27.47201                        298.53009
## `Shell weight`:`Shucked weight`               `Shell weight`:Sex
##                        3424.87359                        430.99777
## `Shell weight`:`Viscera weight`               Sex:`Whole weight`
##                        3577.30646                        359.47918
## `Shucked weight`:`Whole weight` `Whole weight`:`Viscera weight`
##                        1833.81362                       2059.89893
##                       Sex:Height
##                         184.35343
```

Table 3 LASSO Regression Coefficients

```
## 10 x 1 sparse Matrix of class "dgCMatrix"
##                                  s1
## (Intercept)         4.228361e+00
## SexI               -5.541274e-01
## SexNI               1.415551e-12
## Length              .
## Diameter            .
## Height              1.115687e+01
## X.Whole.weight.     .
## X.Shucked.weight. -8.808777e+00
## X.Viscera.weight.   .
## X.Shell.weight.     2.004406e+01
```

Table 4 random forest importance

```
##                    %IncMSE
## SexI              31.66244
## SexNI             32.58261
## Length            22.43976
## Diameter          23.22770
## Height            28.85438
## X.Whole.weight.   26.07148
## X.Shucked.weight. 50.32488
## X.Viscera.weight. 29.05404
## X.Shell.weight.   41.93567
```

Table 5 Model Evaluation Results

Model\Metrics	MAE	MSE	NMSE
Multiple Linear Regr	1.673	5.675	0.521
Forward Stepwise Regr	1.620	5.074	0.465
LASSO Regression	1.683	5.692	0.522
Random Forest	1.570	4.950	0.454

Table 6 ANOVA for LASSO

```
## Analysis of Deviance Table
##
## Model: gaussian, link: identity
##
## Response: Rings
##
## Terms added sequentially (first to last)
##
##
##                    Df Deviance Resid. Df Resid. Dev
## NULL                                3339      34252
## SexI                1   6450.9       3338      27801
## SexNI               0      0.0       3338      27801
## Height              1   6664.9       3337      21136
## X.Shucked.weight.   1    847.4       3336      20288
## X.Shell.weight.     1   4589.6       3335      15699
```

Table 7 Summary for LASSO

```
##
## Call:
## glm(formula = Rings ~ SexI + SexNI + Height + X.Shucked.weight. +
##     X.Shell.weight., family = "gaussian", data = newdata)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -9.0301  -1.3582  -0.3096   0.8752  16.2527
##
## Coefficients: (1 not defined because of singularities)
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)         4.03376    0.18799  21.458  < 2e-16 ***
## SexI               -0.73071    0.09926  -7.362 2.28e-13 ***
## SexNI                    NA         NA      NA       NA
## Height             17.89870    2.68035   6.678 2.83e-11 ***
## X.Shucked.weight. -15.14161    0.51760 -29.253  < 2e-16 ***
## X.Shell.weight.    26.29170    0.84200  31.225  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 4.707257)
##
##     Null deviance: 34252  on 3339  degrees of freedom
## Residual deviance: 15699  on 3335  degrees of freedom
## AIC: 14660
##
## Number of Fisher Scoring iterations: 2
```

Appendix 2

```
install.packages("GGally")
install.packages("caret")
install.packages("scatterplot3d")
install.packages("Metrics")
install.packages("lars")
install.packages("glmnet")
install.packages("randomForest")
install.packages("stats")

#Data Preprocessing
ab<-read.table("abalone.txt",sep = ",")
names(ab) = c("Sex","Length","Diameter","Height","Whole weight","Shucked weight","Viscera weight","Shell weight","Rings")

#Missing values
sum(is.na(ab))
#Quality test
summary(ab$Height==0)#2 data points,which height=0, whole weight>0 的 data point
ab<-ab[which(ab$'Height' != 0),]
#Outliers
sapply(ab, class)
boxplot(ab[,2:8],main='Box plot')

#Data transformation
par(mfrow=c(2,4))
ab29<-ab[,2:9]
for(i in 1:8) {
hist(ab29[, i], main=paste("Histogram of", names(ab29)[i]))}

#choose how to transform `Diameter`
par(mfrow=c(1,3))
with(ab29,{
  hist(log(`Diameter`))
  hist((`Diameter`)^(1/3))
  hist((`Diameter`)^2)
})
ab$`Diameter`<-(ab$`Diameter`)^2#(`Diameter`)^2
```

```
#choose how to transform `Shucked weight`
par(mfrow=c(1,3))
with(ab29,{
  hist(log(`Shucked weight`))
  hist(1/(`Shucked weight`))
  hist(sqrt(`Shucked weight`))
})
ab$`Shucked weight`<-sqrt(ab$`Shucked weight`)#sqrt(`Shucked weight`)
ab$`Viscera weight`<-sqrt(ab$`Viscera weight`)#sqrt(`Viscera weight`)
ab$`Shell weight`<-sqrt(ab$`Shell weight`)#sqrt(`Shell weight`)

#ggpairs
library(GGally)
ggpairs(ab,aes(color=Sex,alpha=0.8))+theme_grey(base_size=9)+theme_bw()+scale_colour_ma
nual(values=c("#E69F00", "#56B4E9", "#009E73"))

#combine female and male as noninfant
ab$'Sex'[which(ab$'Sex' != 'I')]<- 'NI'

#Train test split
set.seed(10)
sub<-sample(1:nrow(ab),round(nrow(ab)*4/5))
length(sub)
ab_train<-ab[sub,]#take 4/5 of the data as the training set
ab_test<-ab[-sub,]#take 1/5 of the data as the testing set

#Linear Model
ab_lm <- lm(`Rings`~`Sex`+`Diameter`+`Length`+`Height`+`Whole weight`+`Shucked
weight`+`Viscera weight`+`Shell weight`,data=ab_train)
summary(ab_lm)
ab_lm_rings<-predict(ab_lm, ab_test)
ab_lm_prage<-round(ab_lm_rings+1.5)
ab_test_age<-round(ab_test$Rings+1.5)

# calculate MAE, MSE, RMSE and VIF
library(car)
maefun<-function(pred,obs) mean(abs(pred-obs))
msefun<-function(pred,obs) mean((pred-obs)^2)
nmsefun<-function(pred,obs) mean((pred-obs)^2)/mean((mean(obs)-obs)^2)
```

```
lm_mae=maefun(ab_lm_prage,ab_test_age)
paste0('MAE_lm=',lm_mae)
lm_mse=msefun(ab_lm_prage,ab_test_age)
paste0('MSE_lm=',lm_mse)
lm_nmse=nmsefun(ab_lm_prage,ab_test_age)
paste0('NMSE=',lm_nmse)
lm_vif<-vif(ab_lm)
lm_vif

library(scatterplot3d)
fit_2_sp<- scatterplot3d(ab_test[,2],ab_lm_rings, ab_test[,5], angle = 50, color = "#CC79A7",
pch = 1, ylab = "Rings (ft)", xlab = "Length (mm)", zlab = "Whole weight (g)")

#AIC stepforward
none_mod <- lm(Rings ~ 1, data = ab_train)
full_mod <- lm(Rings ~(.)^2, data = ab_train)
library(MASS)
ab_fs= stepAIC(none_mod, scope = list(upper = full_mod, lower = ~1), direction = "forward", k
= 2, trace = FALSE)
ab_fs$anova
summary(ab_fs)

ab_fs_rings<-predict(ab_fs, ab_test)
ab_fs_prage<-round(ab_fs_rings+1.5)

# calculate MAE, MSE, RMSE and VIF
fs_mae=maefun(ab_fs_prage,ab_test_age)
paste0('MAE_las=',fs_mae)
fs_mse=msefun(ab_fs_prage,ab_test_age)
paste0('MSE_las=',fs_mse)
fs_nmse=nmsefun(ab_fs_prage,ab_test_age)
paste0('NMSE=',fs_nmse)
fs_vif<-vif(ab_fs)
fs_vif

#one-hot encoding for sex variable
library(caret)
dummy <- dummyVars('~.', data = ab_train)
newdata <- data.frame(predict(dummy, newdata = ab_train))
dummy2 <- dummyVars('~.', data = ab_test)
```

```
newtest <- data.frame(predict(dummy2, newdata = ab_test))
x<-as.matrix(newdata[,1:9])
y<-as.matrix(newdata[,10])

#Ridge regression
library(glmnet)
r1<-glmnet(x=x,y=y,family = "gaussian",alpha = 0)
plot(r1,xvar="lambda")
r1cv<-cv.glmnet(x=x,y=y,family="gaussian",alpha=0,nfolds = 10)
plot(r1cv)
rimin<-glmnet(x=x,y=y,family = "gaussian",alpha = 0,lambda = r1cv$lambda.min)
coef(rimin)
rimin<-glmnet(x=x,y=y,family = "gaussian",alpha = 0,lambda = r1cv$lambda.1se)
coef(rimin)
#library(ridge)
#mod <- linearRidge(Rings ~ ., data = newdata,lambda = r1cv$lambda.min)
#summary(mod)

#LASSO regression
library(lars)
ab_lar<-lars(x,y,type="lasso")
plot(ab_lar)

f1 = glmnet(x, y, family="mgaussian", nlambda=100, alpha=1)
print(f1)
plot(f1, xvar="lambda", label=TRUE)
cvfit=cv.glmnet(x,y)
plot(cvfit)
l.coef1<-coef(cvfit$glmnet.fit,s=cvfit$lambda.min,exact=F)
l.coef1
l.coef2<-coef(cvfit$glmnet.fit,s=cvfit$lambda.1se,exact=F)
l.coef2
ab_las<-glm(Rings~.,family="gaussian",data=newdata)
ab_las<-
glm(Rings~`SexI`+`SexNI`+`Height`+`X.Shucked.weight.`+`X.Shell.weight.`,family="gaussian
",data=newdata)

ab_las_rings<-predict(ab_las, newtest)
ab_las_prage<-round(ab_las_rings+1.5)
```

```
# calculate MAE, MSE, RMSE and VIF
las_mae=maefun(ab_las_prage,ab_test_age)
paste0('MAE_las=',las_mae)
las_mse=msefun(ab_las_prage,ab_test_age)
paste0('MSE_las=',las_mse)
las_nmse=nmsefun(ab_las_prage,ab_test_age)
paste0('NMSE=',las_nmse)

library(randomForest)
set.seed(100)
ab_rf=randomForest(Rings~.,data=newdata,ntree=800,importance=TRUE,proximity=TRUE)
print(ab_rf)
importance(ab_rf,type=1)
plot(ab_rf)

ab_rf_rings<-predict(ab_rf, newtest)
ab_rf_prage<-round(ab_rf_rings+1.5)

# calculate MAE, MSE, RMSE and VIF
rf_mae=maefun(ab_rf_prage,ab_test_age)
paste0('MAE_las=',rf_mae)
rf_mse=msefun(ab_rf_prage,ab_test_age)
paste0('MSE_las=',rf_mse)
rf_nmse=nmsefun(ab_rf_prage,ab_test_age)
paste0('NMSE=',rf_nmse)

good_number<-as.matrix(which(abs(as.matrix(ab_rf_rings)-ab_test$Rings)<2))
bad_number<-as.matrix(which(abs(as.matrix(ab_rf_rings)-ab_test$Rings)>2))
good_age <- c()
bad_age <- c()
for (i in (1:nrow(good_number))){
  good_age[i] <- ab_test$Rings[good_number[i, 1]]
}
for (j in (1:nrow(bad_number))){
  bad_age[j] <- ab_test$Rings[bad_number[j, 1]]
}

hist(good_age)
hist(bad_age)
```

```
length(good_age[good_age>12])/length(ab_test$Rings[ab_test$Rings>12])
length(good_age[good_age<12])/length(ab_test$Rings[ab_test$Rings<12])
length(bad_age[bad_age>12])/length(ab_test$Rings[ab_test$Rings>12])
length(bad_age[bad_age<12])/length(ab_test$Rings[ab_test$Rings<12])

#qq plot, residual plot, anova, summary
par(mfrow=c(2,2))
plot(ab_las)
anova(ab_las)
summary(ab_las)
```

References

Abalone age - University of California, Davis. (n.d.). Retrieved December 7, 2021, from https://anson.ucdavis.edu/~haochen/abalone_description.pdf.

Kaur, S. (n.d.). *Why is abalone so expensive?* Luxury Viewer. Retrieved December 7, 2021, from https://luxuryviewer.com/why-is-abalone-so-expensive/.

Mobarak, H., & Niaz Murshed, C. (n.d.). *Econometric ways to estimate the age and price of abalone*. Retrieved December 7, 2021, from https://mpra.ub.uni-muenchen.de/91210/1/MPRA_paper_91210.pdf.

Donges, N. (n.d.). *A complete guide to the random forest algorithm. Built In.* Retrieved December 7, 2021, from https://builtin.com/data-science/random-forest-algorithm.

Bhattacharyya, S. (2020, September 28). *Ridge and lasso regression: L1 and L2 regularization*. Medium. Retrieved December 7, 2021, from https://towardsdatascience.com/ridge-and-lasso-regression-a-complete-guide-with-python-scikit-learn-e20e34bcbf0b.

M, S. (2021, September 14). *Introduction to random forest in R. Simplilearn.com*. Retrieved December 7, 2021, from https://www.simplilearn.com/tutorials/data-science-tutorial/random-forest-in-r.

Singh, D. (2019, November 12). *Linear, Lasso, and Ridge Regression with R*. Pluralsight. Retrieved December 7, 2021, from https://www.pluralsight.com/guides/linear-lasso-and-ridge-regression-with-r.