Code

Data Analysis Report

2022-12-08

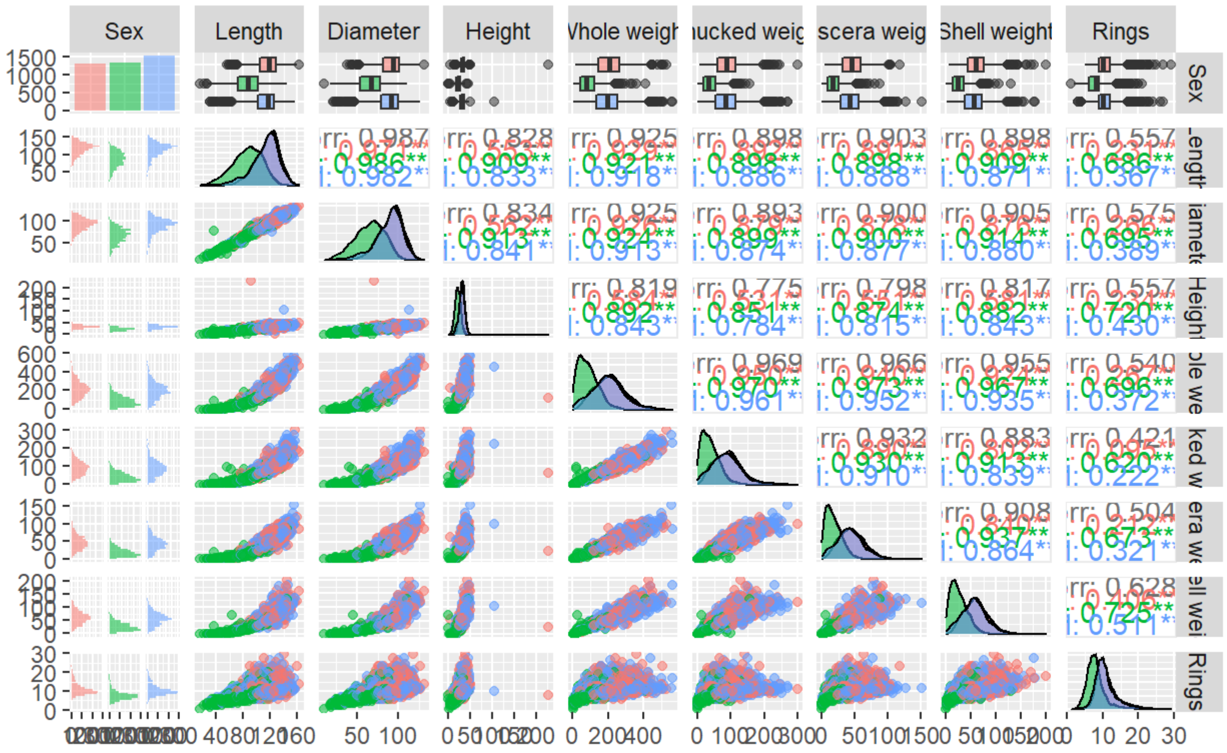
#Exploratory Analysis Firstly, we will look at an overall summary of the variables in the abalone data. This will give us an idea of how the variables are distributed, and if there is a possibility that we have to some preprocessing before fitting our models.

Looking at the summary statistics we note that standard deviation between variables ranges quite significantly, for example standard deviation for abalone length is 24.02 while standard deviation for abalone height is 8.36. This indicated that we may have to transform some of the variables in order to ensure that there isn't a bias towards those variables during analysis.

```
summary(abalone)

 Sex          Length         Diameter          Height       Whole weight    Shucked
weight     Viscera weight    Shell weight        Rings

 F:1307   Min.   : 15.0   Min.   : 11.00   Min.   :  0.0   Min.   :  0.4   Min.   :
0.20   Min.   :  0.10   Min.   :  0.30   Min.   : 1.000

 I:1342   1st Qu.: 90.0   1st Qu.: 70.00   1st Qu.: 23.0   1st Qu.: 88.3   1st Qu.:
37.20   1st Qu.: 18.70   1st Qu.: 26.00   1st Qu.: 8.000

 M:1528   Median :109.0   Median : 85.00   Median : 28.0   Median :159.9   Median :
67.20   Median : 34.20   Median : 46.80   Median : 9.000

          Mean   :104.8   Mean   : 81.58   Mean   : 27.9   Mean   :165.7   Mean   :
71.87   Mean   : 36.12   Mean   : 47.77   Mean   : 9.934

          3rd Qu.:123.0   3rd Qu.: 96.00   3rd Qu.: 33.0   3rd Qu.:230.6   3rd Qu.:
100.40   3rd Qu.: 50.60   3rd Qu.: 65.80   3rd Qu.:11.000

          Max.   :163.0   Max.   :130.00   Max.   :226.0   Max.   :565.1   Max.   :
297.60   Max.   :152.00   Max.   :201.00   Max.   :29.000

sd(abalone$Length)

[1] 24.01858

sd(abalone$Diameter)

[1] 19.84797

sd(abalone$Height)

[1] 8.365411
```

Running a ggpairs plot we can see correlations, histograms and dot plots of each variable. A summary of findings is below: • The dataset contains roughly similar quantities of each category. This is good as it means that there won't be much of a bias in classifying one category or the other. • Strong positive correlations between variables • Females and males overlap quite a bit in distribution while Infants has less overlap. Indication the differentiation between Male and Females might be tough. • Some outliers found in for Male and Females as seen in the lower diagonal scatter plot.

```
ggpairs(abalone, mapping = aes(col=Sex, alpha=0.3))
```



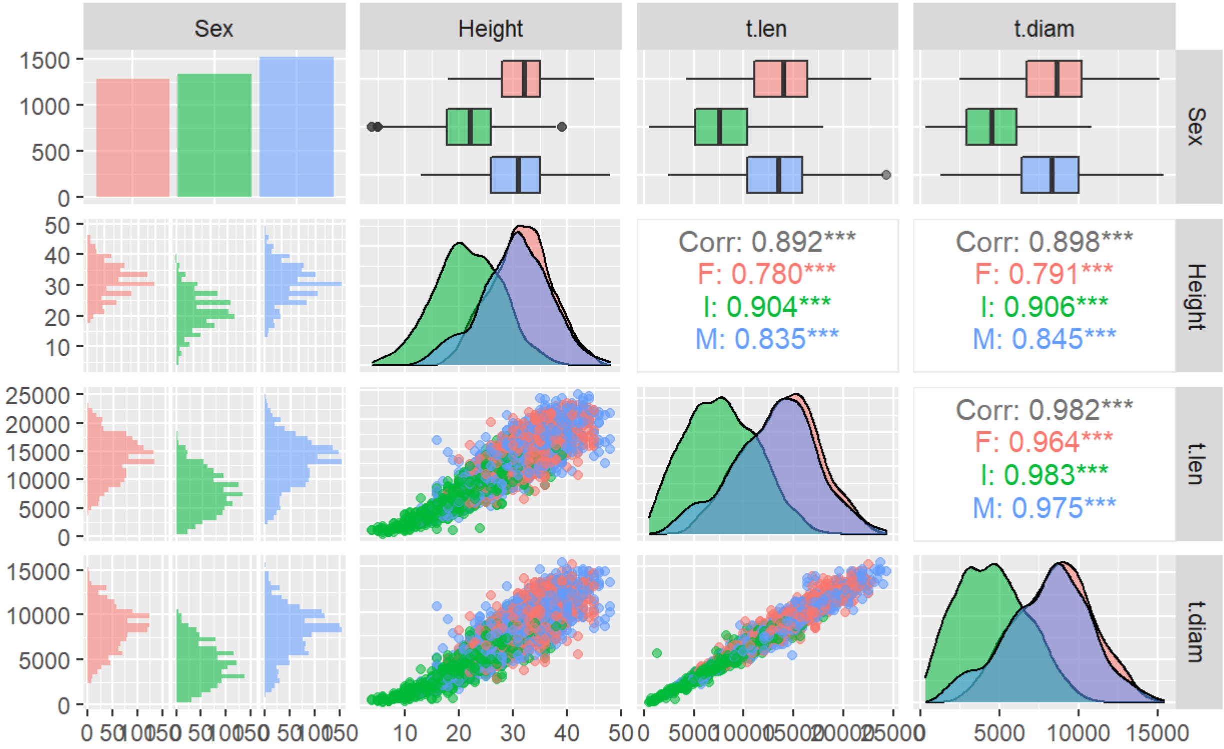
From the exploratory analysis we noted that variances range quite a bit between length and height and diameter and height. We will apply a square transformation on the length and diameter variables to help normalize the longer tail of these variables.

We also remove outliers from the data which we highlighted in the review of the ggpairs plot

```
trans.abalone <- abalone[1:4] %>% mutate('t.len' = Length^2,'t.diam' = Diameter^2,.
keep="unused")

scaled.abalone <- cbind(trans.abalone[,1],as.data.frame(scale(trans.abalone[,-1])))

#Remove Outliers

elim <- function(df){

  idx <- c()

  for (j in colnames(df)[-1]){

    for (i in unique(df$Sex)){

      outlier <- boxplot(df[df$Sex==i, j], plot=FALSE)$out

      idx = c(idx, which(as.numeric(df[,j]) %in% outlier & df$Sex == i))

    }

  }

  return(unique(idx))

}


outlier.idx <- elim(scaled.abalone)

final.abalone <- trans.abalone[-outlier.idx,]
```

The output of the preprocess can be seen in the following ggpairs plot. We note that the output of the histogram plots is more ‘normal’ with lower skewness

```
ggpairs(final.abalone,mapping = aes(col=Sex, alpha=0.3))
```



#Part 1 The first part of our analysis focuses on sustainability. We want to find a classification model to predict sex of an abalone using its high, length and diameter dimensions.

We started with the multi-class case, predicting between males, females and Infants. To select the correct model we looked at 3 models, Linear discriminant analysis models, quadratic discriminant models and support vector models. We then selected the model with the highest accuracy score.

```
abalone.lda <- lda(Sex~t.len+t.diam+Height, data=final.abalone, CV=TRUE)

cm.lda <- table(truth=final.abalone$Sex, prediction=abalone.lda$class)

abalone.qda <- qda(Sex~t.len+t.diam+Height, data=final.abalone, CV=TRUE)

cm.qda <- table(truth=final.abalone$Sex, prediction=abalone.qda$class)

svm.radial.tune <- tune.svm(Sex~t.len+t.diam+Height, data=final.abalone,kernal="rad
ial",gamma=10^(-1:1),cost=10^(-1:1))

svm.linear.tune <- tune.svm(Sex~t.len+t.diam+Height, data=final.abalone,kernal="lin
ear",gamma=10^(-1:1),cost=10^(-1:1))


acc_ldaCV <- sum(diag(cm.lda))/sum(cm.lda)

acc_qdaCV <- sum(diag(cm.qda))/sum(cm.qda)

acc_svm.radial <- 1 - svm.radial.tune$best.performance

acc_svm.linear <- 1 - svm.linear.tune$best.performance


#Summary

acc_ldaCV
```

```
[1] 0.525206
acc_qdaCV
[1] 0.5189045
acc_svm.radial
[1] 0.5305443
acc_svm.linear
[1] 0.5254478
```

#A summary of the results for the multi-class case Model Accuracy LDA - 0.525 QDA - 0.519 SVM radial - 0.533 SVM linear - 0.529

#Infant Case Next we tested the same set of models for the Infants

```
infants.abalone <- final.abalone
levels(infants.abalone$Sex) = c('Non I','I','Non I')

#LDA
infant.lda <- lda(Sex~., data=infants.abalone, CV=TRUE)
cm.lda.infants <- table(infants.abalone$Sex, infant.lda$class)
acc_lda.infants <- sum(diag(cm.lda.infants))/sum(cm.lda.infants)

#QDA
infant.qda <- qda(Sex~., data=infants.abalone, CV=TRUE)
cm.qda.infants <- table(infants.abalone$Sex, infant.qda$class)
acc_qda.infants <- sum(diag(cm.qda.infants))/sum(cm.qda.infants)

#SVM
svm.radial.tune.infant <- tune.svm(Sex~t.len+t.diam+Height, data=infants.abalone,kernal="radial",gamma=10^(-1:1),cost=10^(-1:1))
acc_svm.radial.infant <- 1 - svm.radial.tune.infant$best.performance

svm.linear.tune.infant <- tune.svm(Sex~t.len+t.diam+Height, data=infants.abalone,kernal="linear",gamma=10^(-1:1),cost=10^(-1:1))
acc_svm.linear.infant <- 1 - svm.linear.tune.infant$best.performance
```

#A summary of the results for the Infant-class case Model Accuracy LDA 0.8 QDA 0.793 SVM radial 0.803 SVM linear 0.806

```
#Females
females.abalone <- final.abalone
levels(females.abalone$Sex) = c('F','Non F','Non F')

#LDA
females.lda <- lda(Sex~., data=females.abalone, CV=TRUE)
```

```
cm.lda.females <- table(females.abalone$Sex, females.lda$class)
acc_lda.females <- sum(diag(cm.lda.females))/sum(cm.lda.females)

#QDA
females.qda <- qda(Sex~., data=females.abalone, CV=TRUE)
cm.qda.females <- table(females.abalone$Sex, females.qda$class)
acc_qda.females <- sum(diag(cm.qda.females))/sum(cm.qda.females)

#SVM
svm.radial.tune.females <- tune.svm(Sex~t.len+t.diam+Height, data=females.abalone,kernal="radial",gamma=10^(-1:1),cost=10^(-1:1))
acc_svm.radial.females <- 1 - svm.radial.tune.females$best.performance

svm.linear.tune.females <- tune.svm(Sex~t.len+t.diam+Height, data=females.abalone,kernal="linear",gamma=10^(-1:1),cost=10^(-1:1))
acc_svm.linear.females <- 1 - svm.linear.tune.females$best.performance

#Males
male.abalone <- final.abalone
levels(male.abalone$Sex) = c('Non M','Non M','M')

#LDA
male.lda <- lda(Sex~., data=male.abalone, CV=TRUE)
cm.lda.male <- table(male.abalone$Sex, male.lda$class)
acc_lda.male <- sum(diag(cm.lda.male))/sum(cm.lda.male)

#QDA
male.qda <- qda(Sex~., data=male.abalone, CV=TRUE)
cm.qda.male <- table(male.abalone$Sex, male.qda$class)
acc_qda.male <- sum(diag(cm.qda.male))/sum(cm.qda.male)

#SVM
svm.radial.tune.male <- tune.svm(Sex~t.len+t.diam+Height, data=male.abalone,kernal="radial",gamma=10^(-1:1),cost=10^(-1:1))
acc_svm.radial.male <- 1 - svm.radial.tune.male$best.performance

svm.linear.tune.male <- tune.svm(Sex~t.len+t.diam+Height, data=male.abalone,kernal="linear",gamma=10^(-1:1),cost=10^(-1:1))
acc_svm.linear.male <- 1 - svm.linear.tune.male$best.performance
```

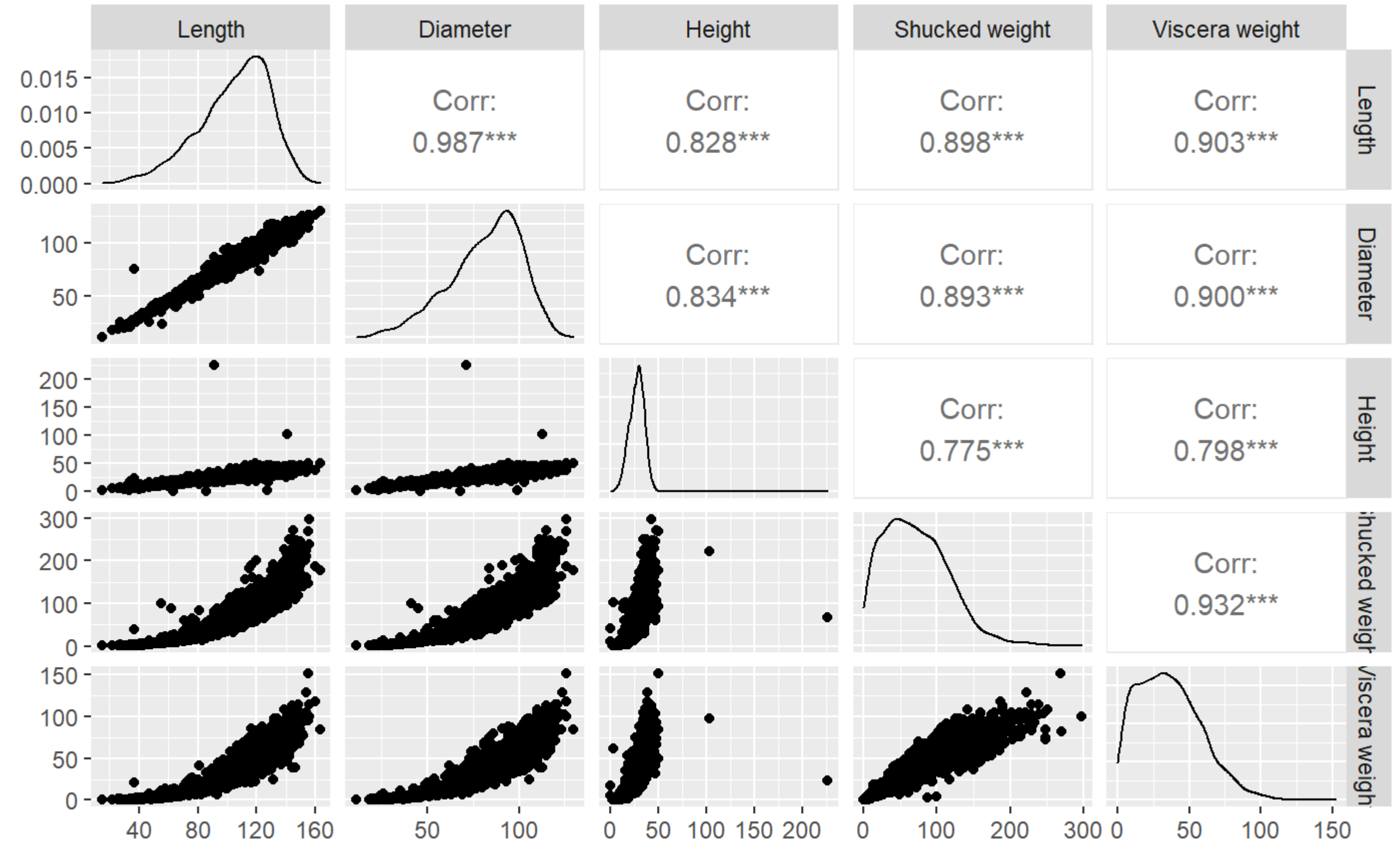
Following the same set of model testing for the Females and males produced the following summaries of accuracy scores #A summary of the results for the Female-class case Model

Accuracy LDA 0.692 QDA 0.681 SVM radial 0.693 SVM linear 0.692 #A summary of the results for the Female-class case Model Accuracy LDA 0.63 QDA 0.62 SVM radial 0.64 SVM linear 0.64

#Conclusion After testing all the models of the four different classification cases we produce the following sets of final models Classification Model Chosen Multi Class = SVM radial Infants = SVM linear Females = SVM radial Males = SVM linear

#Part 2 In part 2 we are focusing on profitability. We are wanting to predict shucked weight and viscera based on hieght, length and diameter and use thoes predictions to determine the value of the abalone. First we create a ggpairs plot to analyse the distribution of predictors in relation to the dependant variables.

```
data <- cbind(abalone[,c("Length","Diameter","Height")],abalone[,c("Shucked weight"
, "Viscera weight")])

ggpairs(data)
```



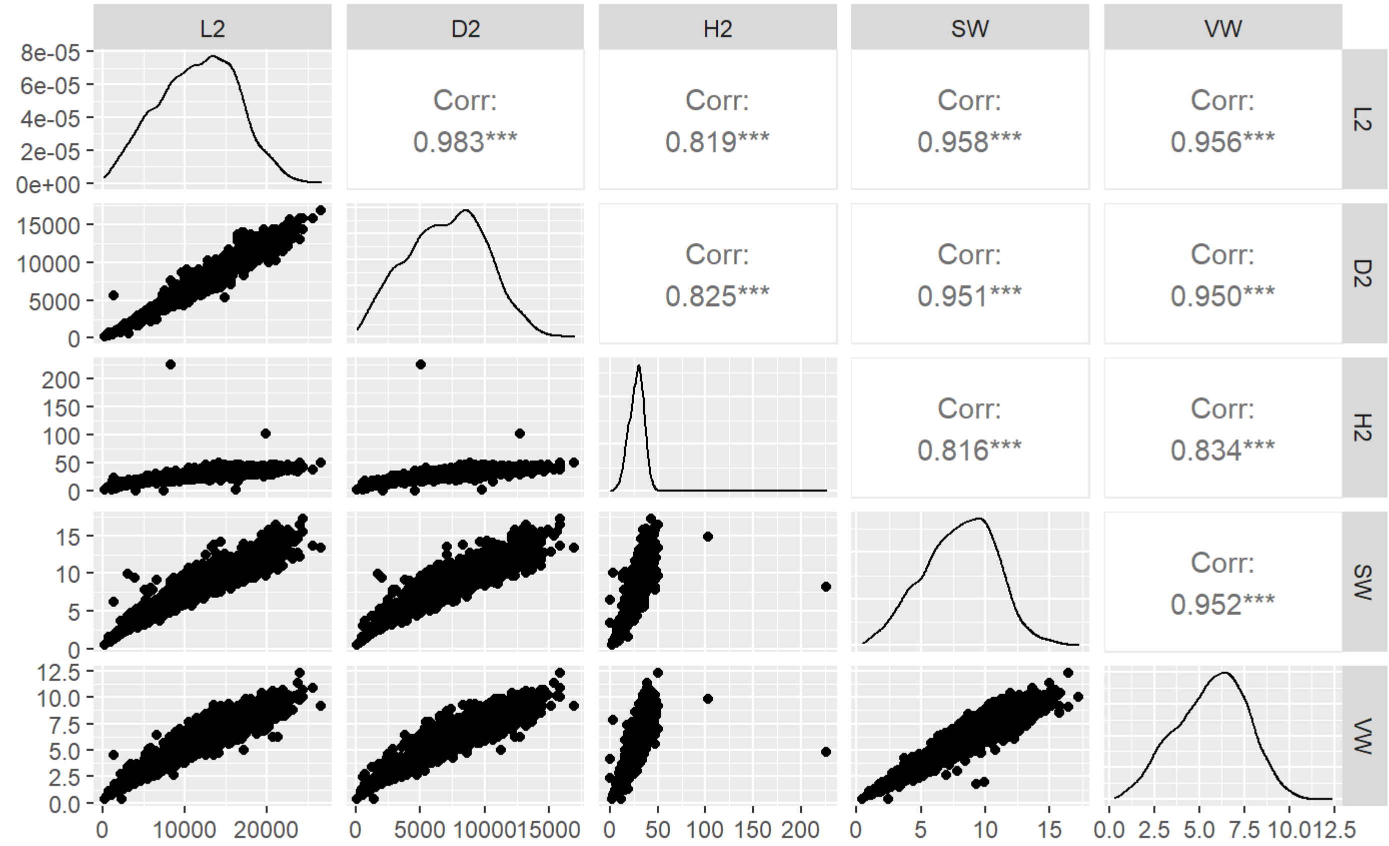
We note that there is a non-linear relationship the independant and dependant variables. This indicates that we will need to make some transformations on these variables so that they are linear. We also note the outliers and remove them from the dataset.

```
library(stringr)

names(data)<-str_replace_all(names(data),c(" "="."))

#We see some non-linear relationships in the Data, need to transform the data

new.data <- data %>% mutate("L2"=Length^2,"D2"=Diameter^2,"H2"=Height,"SW"=sqrt(Shu
cked.weight),"VW"=sqrt(Viscera.weight), .keep="unused")



#Remove Outliers
```

```
elim.mlm <- function(df){
  idx <- c()
  for (j in colnames(df)){
    outlier <- boxplot(df[,j], plot=FALSE)$out
    idx = c(idx, which(as.numeric(df[,j]) %in% outlier))
  }
  return(unique(idx))
}

result <- elim.mlm(new.data)
new.data <- new.data[-result]
```

Running a ggpairs plot on the transformed data shows that the variables produce more linear plots and the density plots appear much more symetrical and normal

```
ggpairs(new.data)
```



Now we fit a multi-linear model A summary of the output is below:

```
class(abalone.mlm <- lm(cbind(SW,VW)~L2+D2+H2,data=new.data))
[1] "mlm" "lm"
summary(abalone.mlm)
Response SW :
```

```
Call:
lm(formula = SW ~ L2 + D2 + H2, data = new.data)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.5448 -0.4651 -0.0760  0.4041  6.7684 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 1.266e+00  4.329e-02   29.25   <2e-16 ***
L2          3.988e-04  1.374e-05   29.02   <2e-16 ***
D2          1.956e-04  2.166e-05    9.03   <2e-16 ***
H2          2.716e-02  2.527e-03   10.75   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.77 on 4173 degrees of freedom
Multiple R-squared:  0.9228,	Adjusted R-squared:  0.9227 
F-statistic: 1.662e+04 on 3 and 4173 DF,  p-value: < 2.2e-16


Response VW :

Call:
lm(formula = VW ~ L2 + D2 + H2, data = new.data)

Residuals:
    Min      1Q  Median      3Q     Max 
-6.1515 -0.3184 -0.0212  0.2935  2.6868 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 8.121e-01  3.016e-02  26.929   <2e-16 ***
L2          2.545e-04  9.573e-06  26.584   <2e-16 ***
D2          1.461e-04  1.509e-05   9.682   <2e-16 ***
H2          3.235e-02  1.761e-03  18.377   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
Residual standard error: 0.5364 on 4173 degrees of freedom
Multiple R-squared:  0.924, Adjusted R-squared:  0.924
F-statistic: 1.691e+04 on 3 and 4173 DF,  p-value: < 2.2e-16
```

We can see that all the predicted coefficients are significant between bother shucked weight and viscera weight, so we will use for creating our abalone value function

```
coeffs <- t(coef(abalone.mlm))
```

#Function for predicting Abalone value The following function takes in the dimensions of the abalone as well as shucked weight value per 1 gram and the viscera weight value per 1 gram

```
abalone_value <- function(l,d,h,shuck_val,visc_val){
  measures<-matrix(c(1,l,d,h,1,l,d,h),nrow=4,ncol=2)
  x1=coeffs["SW",]%*%measures[,1]
  x2=coeffs["VW",]%*%measures[,2]
  ab_value = shuck_val*x1+visc_val*x2
  return(ab_value)
}
```