title: "STAT 602 Project One"
author: "Angela, Anna Leisa, Hacene , Divanshu & Jacob"
date: "2/23/2023"
output:
html_document: default
pdf_document: default

Libraries

```
# Installing required packages and loading libraries

library(ggplot2)
library(GGally)
library(MASS)
library(class)
library(mclust)
library(knitr)
library(dplyr)
library(gridExtra)
library(scales)
library(corrplot)
library(naivebayes)
library(e1071)
library(psych)
```

```
getwd()
```

```
## [1] "/Users/annaleisasauser/Desktop"
```

Step1 : Loading the data, Missing values Check, Summary of the data & Vizually checked the data

```
setwd("/Users/annaleisasauser/Desktop")
```

```
#Loading labeled data set using read.csv

data_bean <- read.csv('labeled.csv', header=TRUE, stringsAsFactor = TRUE)

# Checking dataset format
str(data_bean)
```

```
## 'data.frame':    3000 obs. of  9 variables:
##  $ X              : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Area           : int  30451 69976 34995 47130 68414 57014 41093 26814 35270 34022 ...
##  $ Perimeter      : num  683 1101 852 1040 1241 ...
##  $ MajorAxisLength: num  228 364 258 323 390 ...
##  $ MinorAxisLength: num  140 238 170 209 220 ...
##  $ Eccentricity   : num  0.758 0.807 0.559 0.72 0.858 ...
##  $ ConvexArea     : int  17482 69380 35481 64128 84801 56190 39265 28494 32947 37958 ...
```

```
##  $ Extent         : num  0.763 0.672 0.775 0.726 0.775 ...
##  $ Class          : Factor w/ 6 levels "BOMBAY","CALI",..: 3 2 5 6 2 4 6 3 3 3 ...
```

```
# Missing Value check in the data
sum(is.na(data_bean))
```

```
## [1] 0
```

```
# Summary of the data
summary(data_bean[c(-1,-9)])
```

```
##       Area          Perimeter      MajorAxisLength MinorAxisLength
##  Min.   : 20645   Min.   : 384.2   Min.   :161.5   Min.   :106.0  
##  1st Qu.: 38819   1st Qu.: 760.7   1st Qu.:262.6   1st Qu.:177.6  
##  Median : 48714   Median : 941.9   Median :332.9   Median :202.7  
##  Mean   : 69875   Mean   :1012.2   Mean   :362.0   Mean   :225.2  
##  3rd Qu.: 74690   3rd Qu.:1170.3   3rd Qu.:416.9   3rd Qu.:237.0  
##  Max.   :251320   Max.   :2164.1   Max.   :741.0   Max.   :473.4  
##   Eccentricity      ConvexArea         Extent      
##  Min.   :0.3006   Min.   :  8912   Min.   :0.5710  
##  1st Qu.:0.7135   1st Qu.: 39098   1st Qu.:0.7244  
##  Median :0.7730   Median : 50808   Median :0.7660  
##  Mean   :0.7560   Mean   : 70944   Mean   :0.7528  
##  3rd Qu.:0.8255   3rd Qu.: 76582   3rd Qu.:0.7903  
##  Max.   :0.9449   Max.   :259965   Max.   :0.8502  
```

```
# Removing the X variable from the data
bean_data_updated <- data_bean[-c(1)]
```

The above summary shows that beans data consists of 3000 observations. There are no missing values in the data. On a visual check there are no issues found with the data.

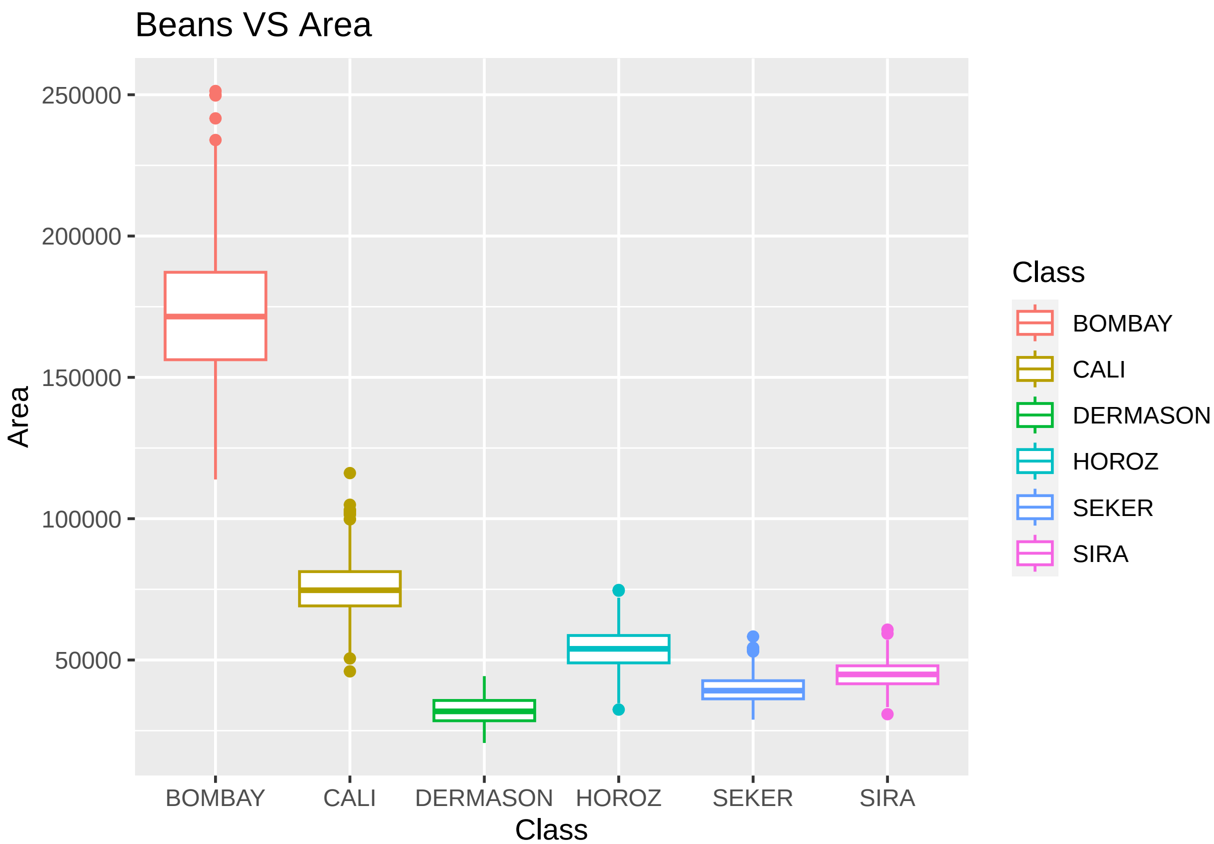
Step2: Exploratory Data Analysis

```
bictest <- Mclust(data_bean)
summary(bictest)
```

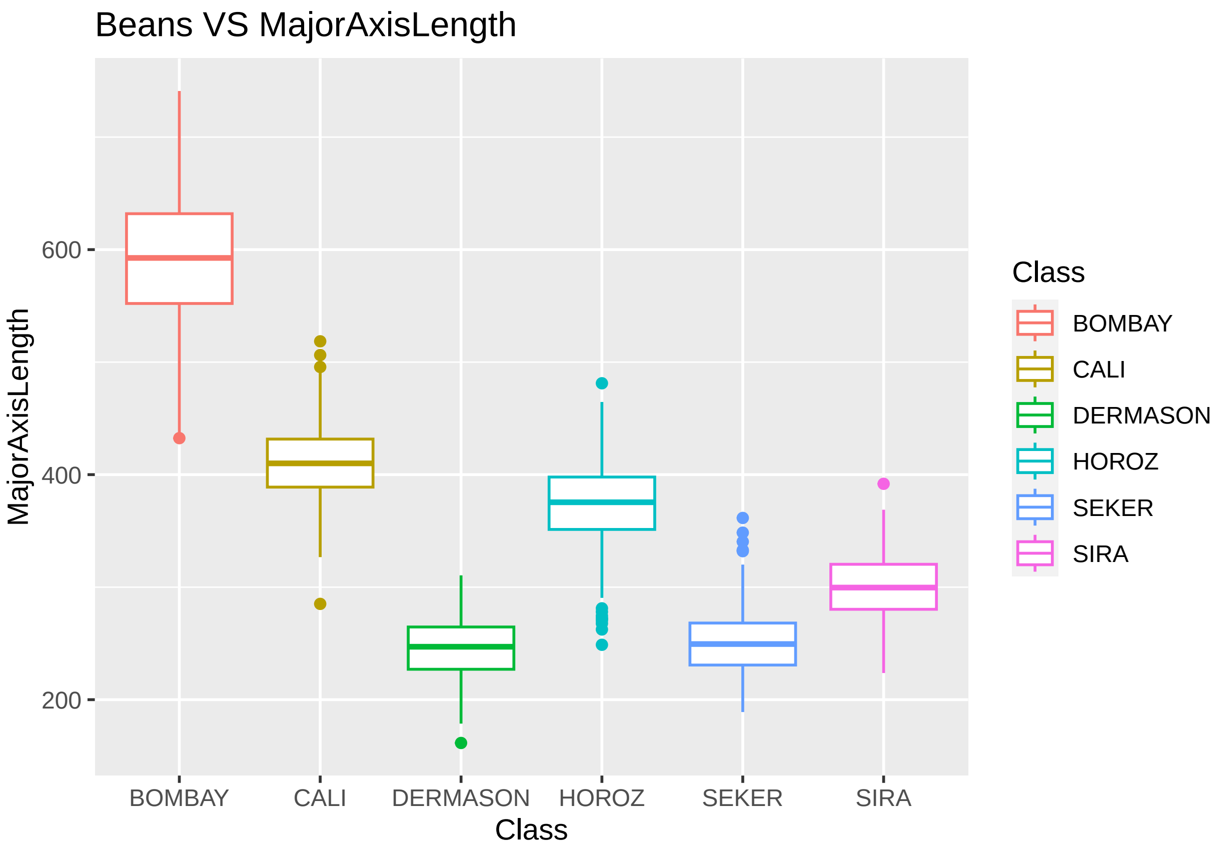
```
## ---------------------------------------------------- 
## Gaussian finite mixture model fitted by EM algorithm 
## ---------------------------------------------------- 
## 
## Mclust VEV (ellipsoidal, equal shape) model with 5 components: 
## 
##  log-likelihood    n  df       BIC       ICL
##       -122017.2 3000 242 -245971.9 -245971.9
## 
## Clustering table:
##    1    2    3    4    5 
##  500  500 1000  500  500 
```

Exploring the data using Box plots & Correlation plots.

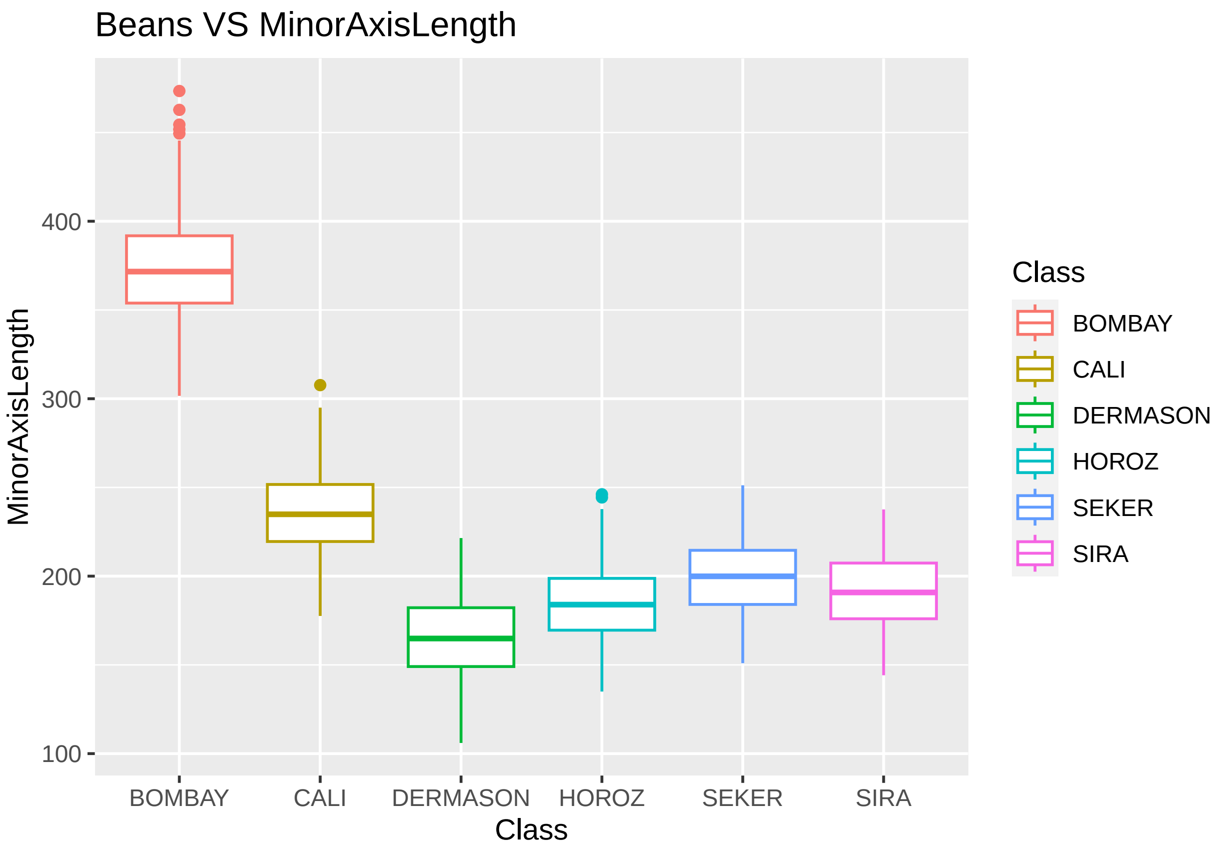
```
# Box plots to show distributions of numeric Predictors vs Response variable.

ggplot(bean_data_updated, aes(x = Class, y = Area, color = Class)) + geom_boxplot() + ggtitle("Beans VS
```



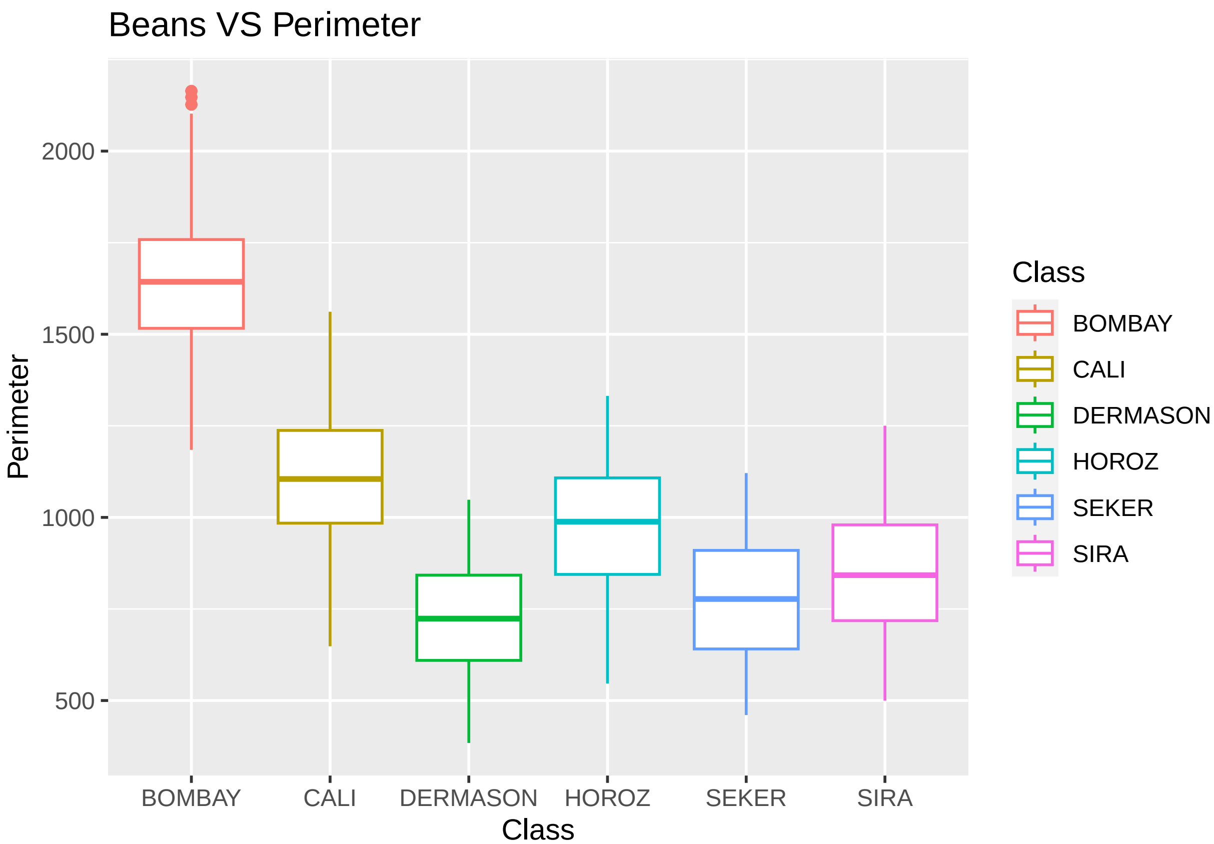

```
ggplot(bean_data_updated, aes(x = Class, y = MajorAxisLength, color = Class)) + geom_boxplot() + ggtitl
```



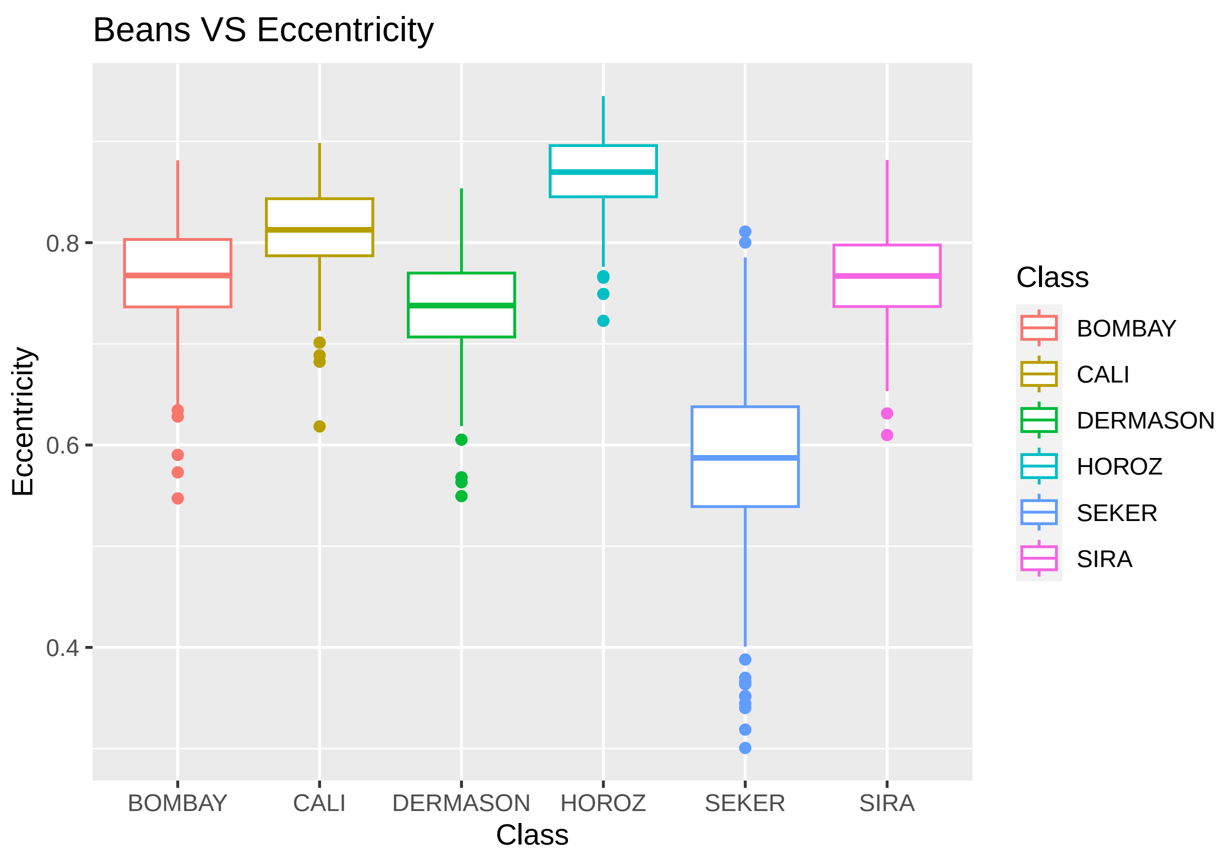

```
ggplot(bean_data_updated, aes(x = Class, y = MinorAxisLength, color = Class)) + geom_boxplot() + ggtitl
```



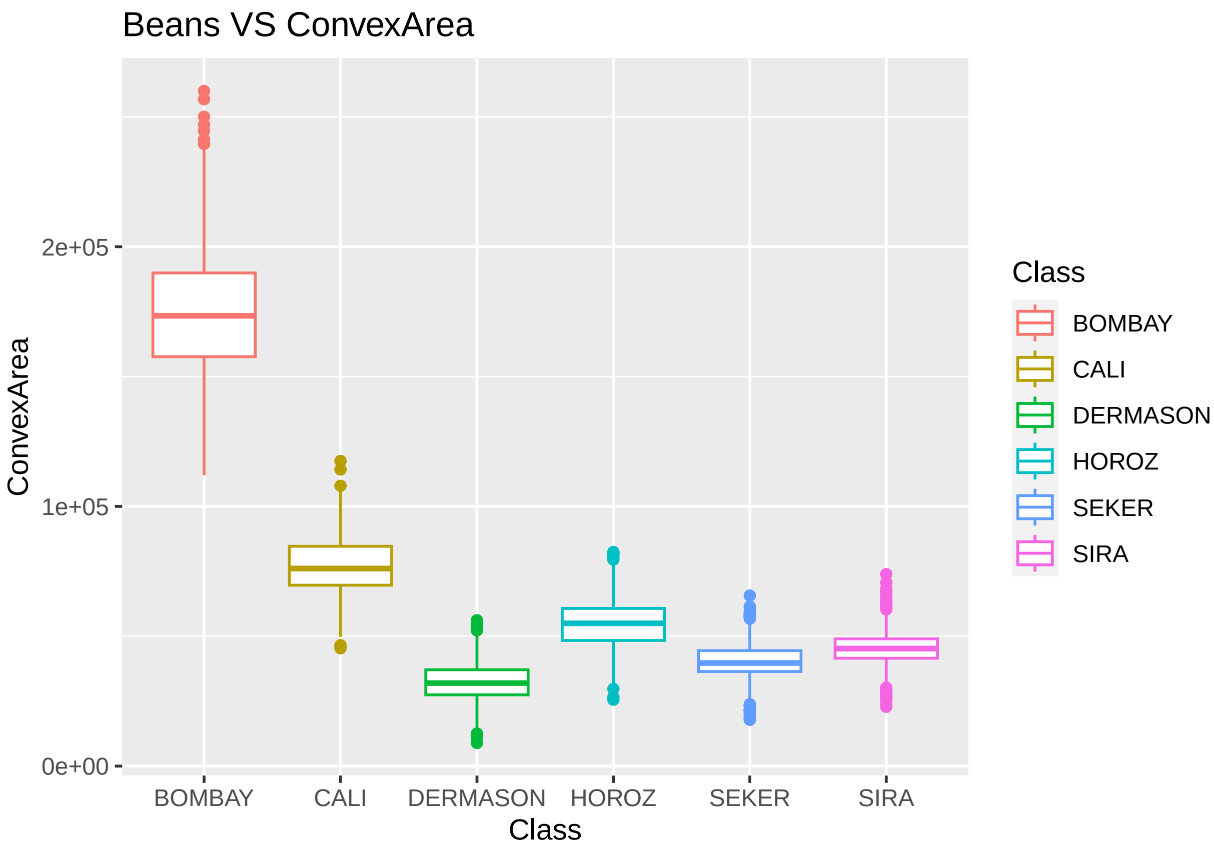

```
ggplot(bean_data_updated, aes(x = Class, y = Perimeter, color = Class)) + geom_boxplot() + ggtitle("Bea
```



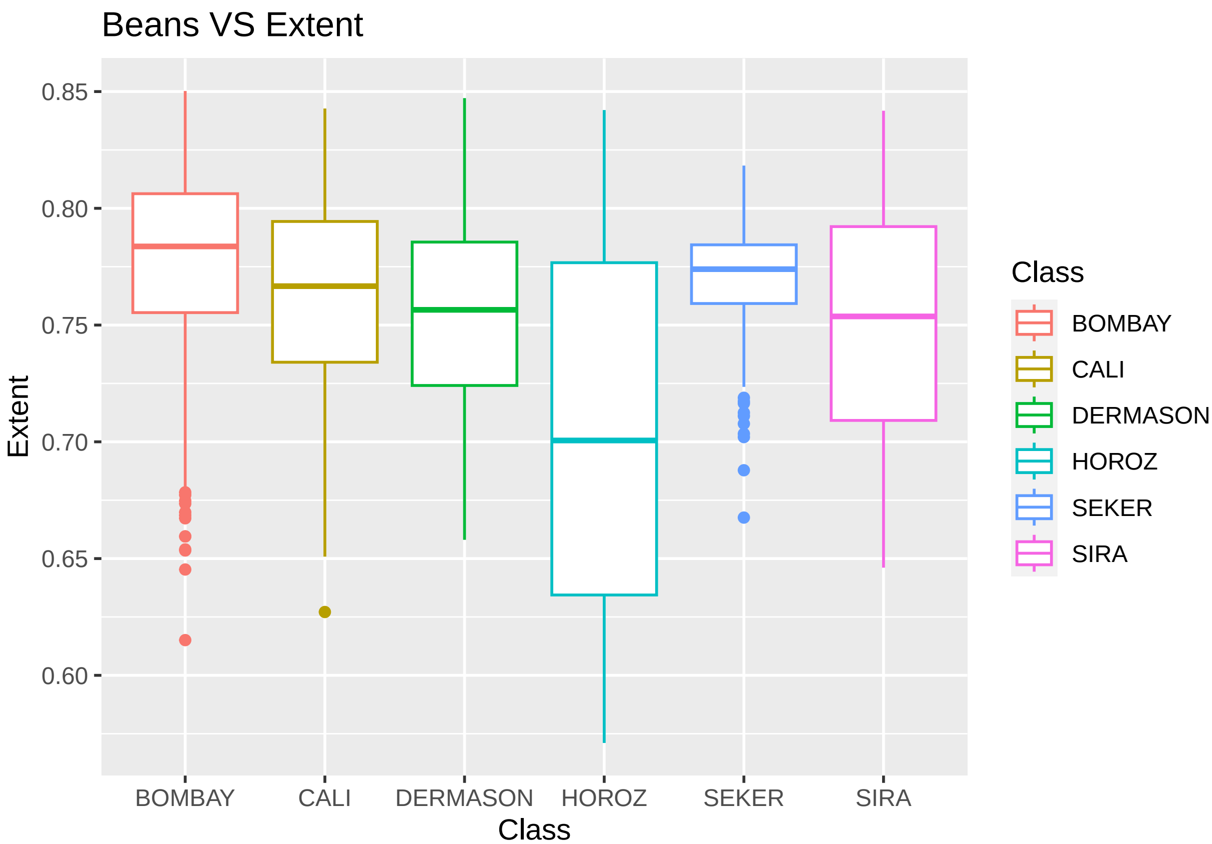

```
ggplot(bean_data_updated, aes(x = Class, y = Eccentricity, color = Class)) + geom_boxplot() + ggtitle("
```




```
ggplot(bean_data_updated, aes(x = Class, y = ConvexArea, color = Class)) + geom_boxplot() + ggtitle("Bea
```




```
ggplot(bean_data_updated, aes(x = Class, y = Extent, color = Class)) + geom_boxplot() + ggtitle("Beans \
```

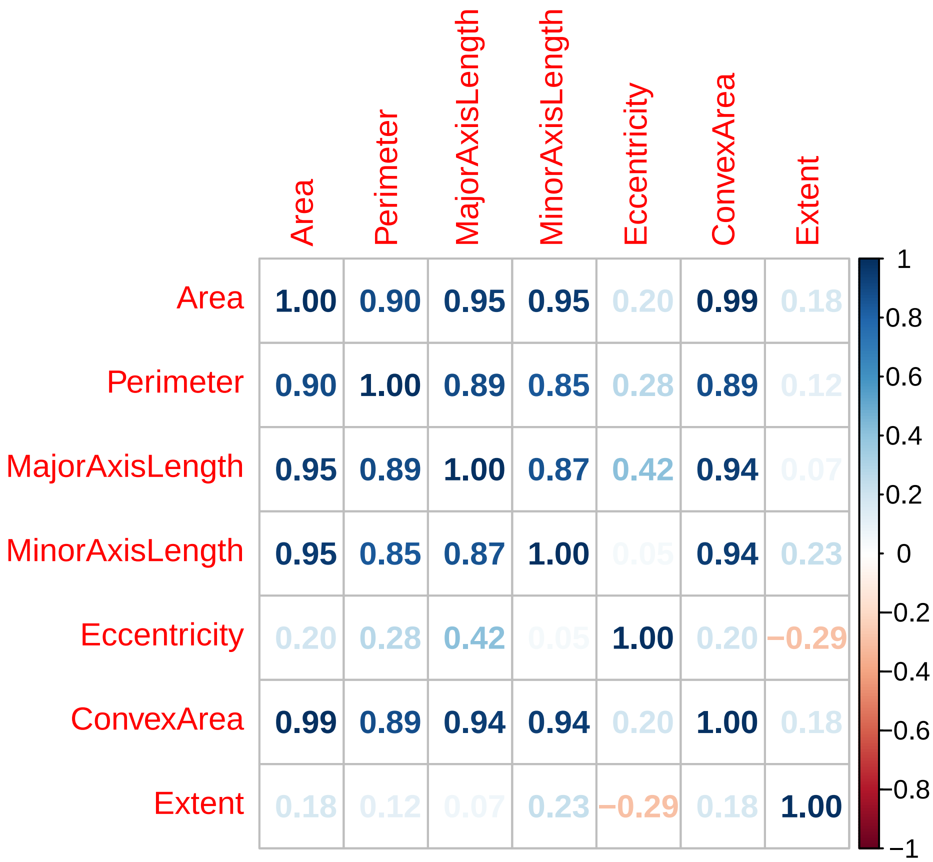


From the above box plots, we can see there are some outliers in the data. For the most part there is not much overlapping in the box plots which shows that there is stronger association with the class variable, but for beans vs extent box plots shows that there is strong overlapping indicates there is less association with the class variable.

Reference: https://ggplot2.tidyverse.org/reference/geom_boxplot.html

Correlation Matrix

```
#  Calculating and Plotting Correlation Matrix
corrplot(cor(bean_data_updated[,-8]), method="number")
```



The Correlation plot shows that variables (Area,Perimeter, MajorAxisLength, MinorAxisLength and ConvexArea) are highly correlated meanwhile Eccentricity and Extent variable are not correlated

Step3 a: Splittin Beans data into a training set and a test set.

```
# splitting the data into train and test

# Set the seed
set.seed(1)

#create ID column
bean_data_updated$seq <- 1:nrow(bean_data_updated)

#Use 70% of data set as training set and 30% as test set
bean_data_updated_train <- bean_data_updated %>% dplyr::sample_frac(0.70)
bean_data_updated_test  <- dplyr::anti_join(bean_data_updated, bean_data_updated_train , by = 'seq')

# Checking training data dimension
cat('Train set:',dim(bean_data_updated_train ), '\n')
```

```
## Train set: 2100 9
```

```
# Checking training data dimension
cat('Test set:',dim(bean_data_updated_test))
```

```
## Test set: 900 9
```

```
# Removing the seq variable from training data set
bean_data_updated_train <- bean_data_updated_train[-c(9)]
```

```r
# Removing the seq variable from testing data set
bean_data_updated_test <- bean_data_updated_test[-c(9)]

# Summary of Training data
summary(bean_data_updated_train[c(-8)])
```

```
##       Area          Perimeter      MajorAxisLength MinorAxisLength
##  Min.   : 20645   Min.   : 384.2   Min.   :161.5   Min.   :106.0  
##  1st Qu.: 38828   1st Qu.: 754.7   1st Qu.:261.5   1st Qu.:177.6  
##  Median : 48535   Median : 941.3   Median :331.2   Median :204.1  
##  Mean   : 70108   Mean   :1011.5   Mean   :362.0   Mean   :226.0  
##  3rd Qu.: 74707   3rd Qu.:1170.0   3rd Qu.:416.9   3rd Qu.:238.2  
##  Max.   :251320   Max.   :2164.1   Max.   :741.0   Max.   :473.4  
##   Eccentricity      ConvexArea         Extent      
##  Min.   :0.3006   Min.   :  8912   Min.   :0.5781  
##  1st Qu.:0.7128   1st Qu.: 39055   1st Qu.:0.7263  
##  Median :0.7722   Median : 50741   Median :0.7668  
##  Mean   :0.7556   Mean   : 71032   Mean   :0.7536  
##  3rd Qu.:0.8243   3rd Qu.: 76991   3rd Qu.:0.7901  
##  Max.   :0.9449   Max.   :259965   Max.   :0.8502
```

After splitting the data, the bean train data have 2100 observations and test have 900 observations

Step 3 b: Actual Price per seed & Weight Calculation

```r
# Price per seed of six classes of beans
price.per.bean<- c( "BOMBAY"=((5.56*1.92)/453.592),
                    "CALI"=((6.02*0.61)/453.592),
                    "DERMASON"=((1.98*0.28)/453.592),
                    "HOROZ"=((2.43*0.52)/453.592),
                    "SEKER"=((2.72*0.49)/453.592),
                    "SIRA"=((5.40*0.38)/453.592))

# Weight per seed of six classes of beans
weight.per.bean <- c("BOMBAY"=(1.92),
                     "CALI"=(0.61),
                     "DERMASON"=(0.28),
                     "HOROZ"=(0.52),
                     "SEKER"=(0.49),
                     "SIRA"=(0.38))

# Actual price of each bean type in test data set
true_bean_price <- data.frame(price.per.bean[bean_data_updated_test$Class])

# Using the class variable from test data and applying the above calculation to get the true Weight
true_bean_weight <-  data.frame(weight.per.bean[bean_data_updated_test$Class])
```

Step4 Approach taken : Since the response variable has more than two classes, we chose LDA for variable selection based on the best accuracy observed by running the following 6 LDA models

a: Variable selection using the box plots, correlation matrix and testing accuracy of LDA models with different predictors

#Lda Model1 with all the predictors

```r
# Fitting the LDA Model using all the Predictors
LDA_beans_mdl1 = lda(Class~., data=bean_data_updated_train )

# Testing the model
LDA_pred1 <-predict(LDA_beans_mdl1,bean_data_updated_test)

# Get the response variable values,model predictions
LDA_Predictions1 <- LDA_pred1$class

# Get the actual response variable values from testing data
true_values  <- bean_data_updated_test$Class

# Get Accuracy Matrix
table(LDA_Predictions1, true_values)
```

```
##                 true_values
## LDA_Predictions1 BOMBAY CALI DERMASON HOROZ SEKER SIRA
##         BOMBAY      149    0        0     0     0    0
##         CALI          1  137        0     7     0    2
##         DERMASON      0    0      113     1    10   18
##         HOROZ         0    4        0   140     0   12
##         SEKER         0    0        8     0   133    1
##         SIRA          0    6       33    17     8  100
```

```r
# Calculating the accuracy
LDA_mdl1_Accr <- mean(LDA_Predictions1 ==true_values)

# Predicting Weights on testing data
LDA_predicted_weight1 <- data.frame(weight.per.bean[LDA_pred1$class])

# Get the weight difference between the actual and predicted values
Weight_diff1 <- abs(sum(true_bean_weight)-sum(LDA_predicted_weight1))

# Predicting the price on test data
LDA_predicted_price1 <- data.frame(price.per.bean[LDA_pred1$class])

#calculating the Sum of square error on price for Model Selection
LDA_Price_compare1 <- cbind(sum(LDA_predicted_price1), sum(true_bean_price))
LDA_Price_compare1
```

```
##          [,1]     [,2]
## [1,] 6.464069 6.413547
```

```r
# Calculate price Price Difference
price_diff1 <- abs( sum(true_bean_price)- sum(LDA_predicted_price1))

# Sum of the price square error or price variance
SSE_LDA1 = sum((true_bean_price- LDA_predicted_price1)^2)

# Printing Results
cat("LDA Accuracy mdl1 is:",LDA_mdl1_Accr)
```

```
## LDA Accuracy mdl1 is: 0.8577778
```

```r
cat(" , ")
```

```
##  ,
```

```r
cat("Sum of squared errors for price:", SSE_LDA1  )
```

```
## Sum of squared errors for price: 0.001372231
```

b: LDA Model2 with Eccentricity , Extent , Area , Perimeter , MajorAxisLength & MinorAxisLength . Removing ConvexArea because of high correlation.

```r
# Fitting the LDA Model using Eccentricity , Extent ,  Area , Perimeter , MajorAxisLength &  MinorAxisL
LDA_beans_mdl2 = lda(Class~ Eccentricity +  Extent + Area +Perimeter + MajorAxisLength + MinorAxisLengt

# Creating a confusion Matrix

LDA_pred2 <-predict(LDA_beans_mdl2,bean_data_updated_test)
LDA_Predictions2 <- LDA_pred2$class
true_values  <- bean_data_updated_test$Class
table(LDA_Predictions2, true_values)
```

```
##                  true_values
## LDA_Predictions2 BOMBAY CALI DERMASON HOROZ SEKER SIRA
##         BOMBAY      149    0        0     0     0    0
##         CALI          1  137        0     7     0    2
##         DERMASON      0    0      113     1    10   18
##         HOROZ         0    4        0   140     0   12
##         SEKER         0    0        8     0   133    1
##         SIRA          0    6       33    17     8  100
```

```r
# Calculating the accuracy

LDA_mdl2_Accr <- mean(LDA_Predictions2 ==true_values)

# Predicting Weights on test data

LDA_predicted_weight2 <- data.frame(weight.per.bean[LDA_pred2$class])

Weight_diff2 <- abs(sum(true_bean_weight)-sum(LDA_predicted_weight2))

# Predicting the price on test data and then calculating the Sum of square error on price for Model Sel

LDA_predicted_price2 <- data.frame(price.per.bean[LDA_pred2$class])
LDA_Price_compare2 <- cbind(sum(LDA_predicted_price2), sum(true_bean_price))
LDA_Price_compare2
```

```
##          [,1]     [,2]
## [1,] 6.464069 6.413547
```

```r
# Price Difference

price_diff2 <- abs(sum(true_bean_price)-sum(LDA_predicted_price2))

# Sum of the price square error or price variance

SSE_LDA2 = sum((true_bean_price-LDA_predicted_price2)^2)
```

```r
# Printing Results
cat("LDA Accuracy mdl2 is:",LDA_mdl2_Accr)
```

```
## LDA Accuracy mdl2 is: 0.8577778
```

```r
cat(" , ")
```

```
##  ,
```

```r
cat("Sum of squared errors for price:", SSE_LDA2 )
```

```
## Sum of squared errors for price: 0.001372231
```

c: LDA Model3 with Eccentricity, Extent, Area, Perimeter & MajorAxisLength . Removing MinorAxisLength & ConvexArea because of high correlation.

```r
# Fitting the LDA Model using Eccentricity, Extent,  Area, Perimeter & MajorAxisLength  Predictors
LDA_beans_mdl3 = lda(Class~ Eccentricity +  Extent + Area +Perimeter + MajorAxisLength , data=bean_data_

# Creating a confusion Matrix

LDA_pred3 <-predict(LDA_beans_mdl3,bean_data_updated_test)
LDA_Predictions3 <- LDA_pred3$class
true_values  <- bean_data_updated_test$Class
table(LDA_Predictions3, true_values)
```

```
##                 true_values
## LDA_Predictions3 BOMBAY CALI DERMASON HOROZ SEKER SIRA
##         BOMBAY      149    0        0     0     0    0
##         CALI          1  134        0     7     0    2
##         DERMASON      0    0      112     2     8   19
##         HOROZ         0    8        0   141     0   13
##         SEKER         0    0        9     0   133    1
##         SIRA          0    5       33    15    10   98
```

```r
# Calculating the accuracy
LDA_mdl3_Accr <- mean(LDA_Predictions3 ==true_values)

# Predicting Weights on test data
LDA_predicted_weight3 <- data.frame(weight.per.bean[LDA_pred3$class])

Weight_diff3 <- abs(sum(true_bean_weight)-sum(LDA_predicted_weight3))


# Predicting the price on test data and then calculating the Sum of square error on price for Model Sel
LDA_predicted_price3 <- data.frame(price.per.bean[LDA_pred3$class])
LDA_Price_compare3 <- cbind(sum(LDA_predicted_price3), sum(true_bean_price))
LDA_Price_compare3
```

```
##          [,1]     [,2]
## [1,] 6.444641 6.413547
```

```r
# Price Difference
price_diff3 <- abs(sum(true_bean_price)-sum(LDA_predicted_price3))

# Sum of the price square error or price variance
```

```r
SSE_LDA3 = sum((true_bean_price-LDA_predicted_price3)^2)

# Printing Results
cat("LDA Accuracy mdl3 is:",LDA_mdl3_Accr)
```

```
## LDA Accuracy mdl3 is: 0.8522222
```

```r
cat(" , ")
```

```
##  ,
```

```r
cat("Sum of squared errors for price:", SSE_LDA3  )
```

```
## Sum of squared errors for price: 0.001484666
```

d: LDA Model4 with Eccentricity, Extent, Area & Perimeter . Removing MajorAxisLength, MinorAxisLength & ConvexArea because of high correlation.

```r
# Fitting the LDA Model using Eccentricity, Extent,  Area & Perimeter  Predictors
LDA_beans_mdl4 = lda(Class~ Eccentricity +  Extent + Area +Perimeter , data=bean_data_updated_train )

# Creating a confusion Matrix
LDA_pred4 <-predict(LDA_beans_mdl4,bean_data_updated_test)
LDA_Predictions4 <- LDA_pred4$class
true_values  <- bean_data_updated_test$Class
table(LDA_Predictions4, true_values)
```

```
##                  true_values
## LDA_Predictions4 BOMBAY CALI DERMASON HOROZ SEKER SIRA
##         BOMBAY      149    0        0     0     0    0
##         CALI          1  134        0     7     0    0
##         DERMASON      0    0      118     1     7   11
##         HOROZ         0    7        4   137     1   16
##         SEKER         0    0        8     0   135    1
##         SIRA          0    6       24    20     8  105
```

```r
# Calculating the accuracy
LDA_mdl4_Accr <- mean(LDA_Predictions4 ==true_values)

# Predicting Weights on test data
LDA_predicted_weight4 <- data.frame(weight.per.bean[LDA_pred4$class])

# Calculatng weight difference
Weight_diff4 <- abs(sum(true_bean_weight)-sum(LDA_predicted_weight4))

# Predicting the price on test data and then calculating the Sum of square error on price for Model Sel
LDA_predicted_price4 <- data.frame(price.per.bean[LDA_pred4$class])
LDA_Price_compare4 <- cbind(sum(LDA_predicted_price4), sum(true_bean_price))
LDA_Price_compare4
```

```
##          [,1]     [,2]
## [1,] 6.443904 6.413547
```

```r
# Price Difference
price_diff4 <- abs(sum(true_bean_price)-sum(LDA_predicted_price4))

# Sum of the price square error or price variance
```

```r
SSE_LDA4 = sum((true_bean_price-LDA_predicted_price4)^2)

# Printing Results
cat("LDA Accuracy mdl4 is:",LDA_mdl4_Accr)
```

```
## LDA Accuracy mdl4 is: 0.8644444
```

```r
cat(" , ")
```

```
##  ,
```

```r
cat("Sum of squared errors for price:", SSE_LDA4  )
```

```
## Sum of squared errors for price: 0.001279004
```

e: LDA Model5 with Eccentricity, Extent & Area . Removing MajorAxisLength, MinorAxisLength , ConvexArea & Perimeter because of high correlation.

```r
# Fitting the LDA Model using Eccentricity, Extent & Area  Predictors
LDA_beans_mdl5 = lda(Class~ Eccentricity +  Extent + Area   , data=bean_data_updated_train )

# Creating a confusion Matrix
LDA_pred5 <-predict(LDA_beans_mdl5,bean_data_updated_test)
LDA_Predictions5 <- LDA_pred5$class
true_values  <- bean_data_updated_test$Class
table(LDA_Predictions5, true_values)
```

```
##                  true_values
## LDA_Predictions5 BOMBAY CALI DERMASON HOROZ SEKER SIRA
##         BOMBAY      149    0        0     0     0    0
##         CALI          1  134        0     7     0    0
##         DERMASON      0    0      123     0     7   12
##         HOROZ         0    7        5   138     1   11
##         SEKER         0    0        7     0   136    1
##         SIRA          0    6       19    20     7  109
```

```r
# Calculating the accuracy
LDA_mdl5_Accr <- mean(LDA_Predictions5 ==true_values)


# Predicting Weights on test data
LDA_predicted_weight5 <- data.frame(weight.per.bean[LDA_pred5$class])

# Calculating weight difference
Weight_diff5 <- abs(sum(true_bean_weight)-sum(LDA_predicted_weight5))

# Predicting the price on test data and then calculating the Sum of square error on price for Model Sel
LDA_predicted_price5 <- data.frame(price.per.bean[LDA_pred5$class])
LDA_Price_compare5 <- cbind(sum(LDA_predicted_price5), sum(true_bean_price))
LDA_Price_compare5
```

```
##         [,1]     [,2]
## [1,] 6.43261 6.413547
```

```r
# Price Difference
price_diff5 <- abs( sum(true_bean_price)-sum(LDA_predicted_price5))
```

```r
# Sum of the price square error or price variance
SSE_LDA5 = sum((true_bean_price-LDA_predicted_price5)^2)

# Printing Results
cat("LDA Accuracy mdl5 is:",LDA_mdl5_Accr)
```

```
## LDA Accuracy mdl5 is: 0.8766667
```

```r
cat(" , ")
```

```
##  ,
```

```r
cat("Sum of squared errors for price:", SSE_LDA5  )
```

```
## Sum of squared errors for price: 0.001214836
```

Based on the least Sum of square errors on price LDA model (LDA_beans_mdl5) with Eccentricity + Extent + Area predictors is the best model among other LDA models for this analysis. In the next step we performed a 10 fold cross validation to validate our results.

Chacking he model properties

```r
#Display model
LDA_beans_mdl5
```

```
## Call:
## lda(Class ~ Eccentricity + Extent + Area, data = bean_data_updated_train)
##
## Prior probabilities of groups:
##    BOMBAY      CALI  DERMASON     HOROZ     SEKER      SIRA
## 0.1666667 0.1680952 0.1647619 0.1595238 0.1661905 0.1747619
##
## Group means:
##          Eccentricity    Extent      Area
## BOMBAY      0.7676137 0.7782907 174395.20
## CALI        0.8124715 0.7595017  76036.54
## DERMASON    0.7369795 0.7549897  31961.56
## HOROZ       0.8677900 0.7041005  53736.06
## SEKER       0.5864337 0.7704016  39665.08
## SIRA        0.7654264 0.7524302  44804.72
##
## Coefficients of linear discriminants:
##                        LD1           LD2           LD3
## Eccentricity  3.471361e+00  1.898622e+01 -4.452146e+00
## Extent       -1.211768e+00 -2.763255e+00 -2.150174e+01
## Area         -9.144342e-05 -4.276683e-06  6.028941e-06
##
## Proportion of trace:
##    LD1    LD2    LD3
## 0.8746 0.1222 0.0033
```

Based on the prior probabilities of groups results, The training data set is distributed approximately equal for all beans type except for a slight increase for SIRA beans type. The proportion of trace indicates that first linear discriminant(LD1) used by the model has the highest separation percentage 87.46%.

#Step 5- Applying Cross Validation with K=10 for selected LDA mModel

```
'Set seed'
```

```
## [1] "Set seed"
```

```
set.seed(123)

'sample range lies between 1 to 3000 based on dataste rows number'
```

```
## [1] "sample range lies between 1 to 3000 based on dataste rows number"
```

```
var_sample = sample(1:3000, 3000)

'Set the folds size and index'
```

```
## [1] "Set the folds size and index"
```

```
folds=cbind(sort(rep(seq(1, 10, 1), 300))[1:3000], var_sample)

'Set a null variable to hold model accuracy for each iteration'
```

```
## [1] "Set a null variable to hold model accuracy for each iteration"
```

```
var_values = NULL

'Running the loop for CV nethos with K=10'
```

```
## [1] "Running the loop for CV nethos with K=10"
```

```
for (i in 1:10)
{
    'Set the index for selected fold'
    index.i = folds[folds[,1]==i, 2]

    'Get the testing data based on fold index'
    dat_tst = bean_data_updated[index.i,]

    'Get the training data based on fold index'
    dat_trn = bean_data_updated[ - index.i,]

    'Training the model'
    LDA_beans_mdl5 = lda(Class~ Eccentricity +  Extent + Area , data=dat_trn)

    'Testing the model'
    LDA_pred <-predict(LDA_beans_mdl5,dat_tst)

    'Get the predicted class values'
    LDA_Predictions <- LDA_pred$class

    'Get testing data response values data'
    true_values  <- dat_tst$Class

    'Calculate the model accuracy'
    acc_mdl5 <- mean(LDA_Predictions ==true_values)


    'Storing the model accuracy value bsod on the iteration k value'
```

```
    var_values[i]= acc_mdl5
}

'Display the average model accuracy for the Cross Validation with K=10'
```

```
## [1] "Display the average model accuracy for the Cross Validation with K=10"
```

```
mean(var_values)
```

```
## [1] 0.8696667
```

The 10 folds cross validation accuracy for LDA model with Eccentricity + Extent + Area predictors is 87 percent which is very close with the accuracy we calculated above.

Step 6 : Validating the test data to make sure all the samples are predicted for price and weight. There is no missing value

```
bean_data_updated_test$LDA_predicted_price <- price.per.bean[LDA_pred5$class]
bean_data_updated_test$LDA_predicted_weight <- weight.per.bean[LDA_pred5$class]

### Missing Value check for the predictions

sum(is.na(bean_data_updated_test))
```

```
## [1] 0
```

```
summary(is.na(bean_data_updated_test))
```

```
##     Area          Perimeter       MajorAxisLength MinorAxisLength
##  Mode :logical   Mode :logical   Mode :logical   Mode :logical  
##  FALSE:900       FALSE:900       FALSE:900       FALSE:900      
##  Eccentricity    ConvexArea        Extent          Class        
##  Mode :logical   Mode :logical   Mode :logical   Mode :logical  
##  FALSE:900       FALSE:900       FALSE:900       FALSE:900      
##  LDA_predicted_price LDA_predicted_weight
##  Mode :logical       Mode :logical       
##  FALSE:900           FALSE:900           
```

```
### Removing the predicted variables from the test data so that it can be used for other models.

bean_data_updated_test<- bean_data_updated_test[-c(9,10)]
bean_data_updated_test
```

```
##       Area Perimeter MajorAxisLength MinorAxisLength Eccentricity ConvexArea
## 1    30451   683.088        228.4132        139.7512    0.7582261      17482
## 2    69976  1101.054        363.6692        238.0052    0.8072377      69380
## 3    57014   923.038        383.1509        185.8885    0.9240707      56190
## 4    36601   721.533        275.0232        165.1726    0.6836473      36793
## 5    35940   549.026        237.5388        170.1975    0.5534404      35949
## 6    45640   863.484        259.7149        214.1477    0.7569310      43498
## 7    41208   785.943        268.8656        212.2343    0.6903172      42626
## 8   202079  1884.155        623.3302        396.2259    0.7989574     205017
## 9    40522   600.013        233.7455        182.5693    0.5792633      39720
## 10   23588   669.166        241.6650        145.6654    0.8120281      26216
## 11   49398   894.443        308.2741        212.3803    0.7597370      48073
## 12   44729  1068.702        334.5964        202.3567    0.7971620      45356
## 13  193747  1531.500        615.2146        366.2294    0.7602122     195768
```

```
## 14  162924  1541.325         573.8804         339.0081    0.7846139     165295
## 15   38344   782.719         258.9193         161.2336    0.7787191      43422
## 16   45712  1101.432         320.5066         186.2023    0.7649676      61398
## 17  146469  1542.745         546.1330         335.3624    0.7803435     148196
## 18   39579   639.780         283.8359         172.3194    0.8026564      46316
## 19   27070   695.449         219.2879         182.2055    0.7892802      30808
## 20   52334   751.699         397.7653         153.2460    0.9210682      50289
## 21   53733   803.728         403.4452         197.5298    0.8291267      73421
## 22   27098   789.195         200.6262         158.7850    0.7872565      23846
## 23   37147   818.538         306.1070         190.0368    0.8449484      38111
## 24   36358   936.816         302.7707         150.2336    0.8612241      37328
## 25   69328  1126.396         411.4595         213.5958    0.8089471      74114
## 26   24826   531.869         216.8466         149.6659    0.7867022      25224
## 27   41107   877.427         251.5994         202.7349    0.7141765      40026
## 28  152745  1706.095         560.8627         363.1339    0.7349160     157801
## 29  165882  1774.347         570.0117         374.3185    0.7437980     165938
## 30   36679   672.336         252.4959         152.0496    0.7072442      34781
## 31   48608   666.347         308.4625         170.5202    0.7941735      50270
## 32   34317   940.022         243.7850         214.4874    0.4710810      34744
## 33   68618   980.504         389.1636         232.7298    0.8047489      69544
## 34   34688   551.620         231.9667         161.1976    0.7630377      34783
## 35  155987  1534.502         544.9326         353.9015    0.7564226     158551
## 36   42600  1072.575         271.6471         187.9238    0.6868758      43480
## 37   28879   848.697         210.9652         143.9499    0.7613125      29829
## 38   35225   894.677         277.3158         205.0230    0.7060285      35997
## 39   39056   559.855         267.0949         200.6917    0.6160991      35886
## 40  156393  1364.939         582.3305         355.8617    0.7400352     162684
## 41  180146  1788.670         650.9387         367.0101    0.8023132     186712
## 42   55925   866.100         419.2038         215.1113    0.9261148      61416
## 43   40672   739.440         274.0464         192.1229    0.6807269      26778
## 44   42666   875.768         271.6689         213.0788    0.6229739      41009
## 45  152884  1400.830         547.8949         360.0819    0.6666823     154335
## 46   84958  1051.355         437.1416         276.5424    0.7873820      86397
## 47   30457   909.164         235.4546         197.1067    0.6723817      31473
## 48   38159   600.005         240.9810         188.5546    0.6003634      38488
## 49   95418  1100.035         452.4401         254.1223    0.7632878      98076
## 50   40572   732.019         224.2359         215.0035    0.5311619      41215
## 51   24205   874.218         222.6128         159.5057    0.7281891      29116
## 52  133881  1270.508         496.7221         304.8746    0.7581820     132204
## 53   79892  1053.163         420.7266         209.8429    0.8468958      96189
## 54   38067   651.946         296.5763         176.9996    0.8033997      38929
## 55   41770   584.896         252.8724         195.0040    0.5585280      43947
## 56   29453   895.150         240.2659         138.7628    0.7344190      33416
## 57   31898   808.399         223.0851         164.4509    0.4863611      36494
## 58  177235  1885.467         619.0084         359.7804    0.7741230     183360
## 59   39121   970.306         228.8805         193.7280    0.5531685      37231
## 60   29163   845.041         211.3226         175.7900    0.7325017      25910
## 61   35135   575.251         214.6340         170.5369    0.4862738      49529
## 62   38191   576.097         255.7538         219.4359    0.6645372      37804
## 63  177101  1521.333         573.0770         369.7453    0.7896907     195860
## 64  205338  1755.579         690.0309         380.4753    0.8501496     206034
## 65   46997   593.706         285.2684         221.3937    0.6401227      44259
## 66   46600   675.427         312.8966         175.8064    0.7687960      36352
## 67  161183  1612.677         571.4507         348.6804    0.7184431     161872
```

```
## 68   28580   913.707          227.2222         142.8686    0.7038115       30422
## 69   94375  1154.160          462.2052         230.0515    0.8347582       93186
## 70   75280   872.746          413.6437         263.0382    0.7650866       73521
## 71   34136   938.579          211.3907         165.9212    0.5228115       38432
## 72  160687  1678.901          572.2235         363.0635    0.7910550      162673
## 73   47067   710.123          281.9025         178.7075    0.7423071       47108
## 74  177133  1693.053          603.7509         358.2124    0.7869377      179074
## 75  156287  1513.760          591.1627         379.0895    0.7877754      159465
## 76   35298   603.878          259.4148         219.7808    0.5429956       38002
## 77   42538   810.112          265.4816         185.1408    0.5775802       42930
## 78   67669   885.331          375.8805         211.5199    0.7839050       69642
## 79   47636   786.333          272.3546         171.4987    0.7954836       45768
## 80  187805  1626.010          605.4125         384.3647    0.7339192      190944
## 81  151975  1430.036          606.9473         312.0388    0.8644619      149285
## 82   41141   907.983          337.4075         189.0945    0.8342813       40072
## 83   59042   900.374          406.7678         216.0812    0.8193840       46600
## 84   43059   713.810          253.6218         185.7253    0.7245886       43341
## 85   71455  1045.997          409.8118         266.5396    0.7948977       72522
## 86   59211   964.677          409.6332         178.1902    0.8168537       60445
## 87  152622  1673.173          530.1406         369.1619    0.7912400      157005
## 88  187188  1837.983          605.8728         396.9938    0.7564110      188239
## 89   44608   777.031          276.1902         187.8392    0.7670811       44970
## 90   30148   746.249          233.9491         162.9781    0.7622659       30251
## 91   33273   688.594          278.3009         144.7815    0.7340386       34270
## 92   42895   600.531          236.3860         198.2638    0.4485543       59832
## 93   55016   715.507          340.1603         179.4957    0.7962381       54782
## 94   32308   813.822          206.8134         172.0127    0.6092652       32549
## 95   44059   723.152          285.1193         166.5263    0.7235678       44307
## 96   41157   730.551          265.6199         196.1206    0.5258288       40316
## 97   50039   744.351          368.5013         184.2928    0.8161560       39923
## 98  159866  1737.573          542.1551         353.0862    0.7894552      160233
## 99   37206   848.899          324.2588         162.0030    0.8833043       38074
## 100  40040   676.182          223.5392         193.7051    0.5773704       40122
## 101  30514   834.047          205.6449         164.0665    0.6609782       33463
## 102  47066  1005.810          369.4285         201.5906    0.8536794       51576
## 103  39815   920.042          338.9573         156.1308    0.8808977       42814
## 104  58121   906.175          342.3618         180.5978    0.7800244       63092
## 105 185953  1549.895          602.2617         428.2813    0.7024615      183816
## 106  76482   942.848          424.4893         221.0477    0.8315587       77846
## 107  78804  1207.097          434.6891         216.5718    0.8478053       94776
## 108 173698  1788.402          598.2622         382.5425    0.7561924      165379
## 109  62236  1235.635          435.3785         189.8659    0.9159224       62206
## 110 164825  1700.762          626.2760         360.6877    0.7659838      154990
## 111  75654  1266.058          435.4200         247.6737    0.8494486       77354
## 112  41526  1014.276          274.3456         221.4851    0.5681120       42850
## 113  58486  1094.413          393.8187         183.3035    0.9056682       60157
## 114 158498  1664.736          537.5938         394.3731    0.7751300      156057
## 115  40323  1001.701          268.4120         188.2977    0.5543672       55716
## 116 171632  1671.371          586.0026         391.7817    0.6983600      185835
## 117 209238  1559.716          645.3555         437.2646    0.7929812      212768
## 118 159363  1476.555          574.4310         336.1414    0.7694626      160408
## 119  71693  1321.699          403.2491         228.1623    0.8161773       87803
## 120  24234   840.243          246.2474         149.4969    0.7337419       26690
## 121  69129  1020.851          379.7252         203.6691    0.7878032       69641
```

```
## 122  32810   575.198          281.2278         197.8185    0.6983672      34956
## 123  33865   612.138          201.9199         167.2217    0.4682839      35034
## 124  26494   568.853          200.9293         132.4825    0.6954465      26889
## 125  80188  1287.022          409.0825         244.7615    0.8409078      79519
## 126  64647  1185.299          375.2117         195.8055    0.8492564      63916
## 127  43970  1074.099          318.8439         186.5547    0.8681841      45076
## 128  38680   977.090          282.0002         180.6631    0.7213902      39970
## 129  33485   830.233          233.2348         146.2086    0.6779357      30599
## 130  85680  1241.205          434.7998         246.0116    0.8848770      87583
## 131  69808  1178.933          374.6385         204.3894    0.8021299      70430
## 132 188631  1728.351          602.9559         385.1260    0.7653108     188358
## 133  77305  1051.818          372.1306         227.8676    0.7993258      75499
## 134  74053   981.343          363.4293         263.5527    0.8211435      70688
## 135  30715   745.402          206.0298         181.1959    0.7029620      28425
## 136  65913  1013.783          441.6322         206.8999    0.8429882      66528
## 137  59910  1104.814          345.5689         196.9399    0.7553250      56727
## 138  76590   951.460          431.9354         225.0159    0.8584107      76448
## 139  70494   997.506          407.3926         245.9295    0.8584506      59103
## 140  24455   696.457          200.7089         134.7670    0.6927759      24707
## 141 145939  1319.110          544.5879         382.7107    0.7584749     148807
## 142  40210   564.773          257.9172         169.0052    0.7816610      40170
## 143 196675  1781.731          667.2024         388.7521    0.8171220     202609
## 144  46775   681.289          281.6582         197.7868    0.7255730      43136
## 145  49045   849.817          291.0760         188.9886    0.7365668      63455
## 146  34398   864.289          272.1286         178.9400    0.7346820      32358
## 147  49661   839.834          317.1629         184.0277    0.8165334      49255
## 148  43615   897.415          336.4523         171.7627    0.8916039      59892
## 149  38373   878.788          288.1612         169.5345    0.7565134      33940
## 150  56872   992.085          382.6190         198.1717    0.9244986      57534
## 151  41574   855.030          355.3643         134.9767    0.9315174      44489
## 152  47584   931.791          292.2263         206.4555    0.7664485      46276
## 153  56270   989.924          390.3848         181.3954    0.9034470      60492
## 154  35249   637.802          259.2329         205.0168    0.7692881      35697
## 155  59260  1123.154          407.8320         182.8444    0.9034434      61914
## 156  42194   745.826          243.7816         225.3507    0.5748792      41474
## 157  42663   786.789          286.3109         201.8518    0.7450827      40399
## 158  83951   917.705          422.0726         243.3288    0.7822302      83025
## 159  45401   908.010          290.4930         199.9740    0.7848009      46173
## 160  86384  1187.887          411.9829         265.4397    0.7692784      87914
## 161  66519   948.457          382.1933         243.0931    0.8362956      70758
## 162  62694   974.585          424.2017         169.9663    0.8538777      63546
## 163 166725  1826.703          571.7961         374.5787    0.7689510     170546
## 164 162019  1401.368          599.6332         365.6871    0.8532453     149855
## 165  74691  1085.564          421.5045         250.3375    0.8559229      75282
## 166  81845  1107.410          449.0190         226.6259    0.8634594      82677
## 167  51873  1166.409          340.4362         182.4062    0.8885970      68221
## 168  25617   740.171          219.1426         151.0635    0.6978712      25934
## 169  26587   882.202          214.5230         151.8861    0.7064312      29255
## 170  58961   807.266          424.6873         214.4168    0.8832908      58392
## 171  41544  1005.776          254.8036         218.5019    0.5825881      40110
## 172  26356   529.883          190.2749         183.7187    0.6432713      27587
## 173  31780   491.955          251.7228         180.6961    0.7790315      30254
## 174  69715  1286.620          368.7075         236.0190    0.7850618      68128
## 175  55827  1144.555          372.5740         192.5900    0.8927472      56340
```

```
## 176  74189   861.181          404.7016          226.5317    0.8373050       76689
## 177  91303  1285.034          428.8628          281.0840    0.8436796       90947
## 178  34341   943.398          241.5415          174.3936    0.7479437       34692
## 179  46323   923.660          289.5278          218.1828    0.4721866       45819
## 180  44926   650.380          271.5141          230.1441    0.5744954       45264
## 181  35121   615.006          271.8738          139.4466    0.8623607       37341
## 182  35518   785.175          286.2411          209.4936    0.7414279       39282
## 183  75185  1317.593          399.3138          251.2542    0.8393017       61283
## 184 157416  1572.519          585.1370          356.4847    0.7638415      160506
## 185 137759  1567.168          512.4144          345.7808    0.7448615      139956
## 186  47113   694.948          301.2660          200.2963    0.7299114       30221
## 187  39734   605.649          277.6717          186.0459    0.4851781       41952
## 188  47423   820.775          321.7098          179.7022    0.7898947       50686
## 189  73522   950.549          400.7832          213.8977    0.7574523       75261
## 190  64379   999.416          408.9811          194.2157    0.8993339       64532
## 191  35497   633.461          273.1563          193.5259    0.7717496       49711
## 192  30458   917.238          228.4848          135.4399    0.7233140       30013
## 193 203832  1924.205          598.0454          449.4471    0.6987852      205078
## 194  38922   886.590          307.4257          193.4122    0.8290637       25661
## 195  76147  1368.498          462.2656          231.5782    0.8825873       80091
## 196  79152   925.913          395.9472          220.1506    0.8132359       79952
## 197  46515  1014.761          273.1339          185.8148    0.7921617       47644
## 198  36988   704.903          230.2998          190.9904    0.5544669       34016
## 199  47489   969.781          302.1459          221.9792    0.5792361       45203
## 200  33891   983.795          282.7738          145.4312    0.6947415       35579
## 201  78331   892.425          415.8117          226.4648    0.8148507       93327
## 202  53115   707.992          334.1212          194.4905    0.7441219       54252
## 203  47858   946.926          330.4251          201.4730    0.7970576       47652
## 204  35807   968.360          231.8866          203.9224    0.5680843       33742
## 205  65546  1172.470          392.2155          212.2641    0.8335337       64773
## 206 224282  1821.258          725.1030          425.6417    0.8152727      211206
## 207  39155   748.296          263.6439          220.8165    0.6155068       34878
## 208  87580  1128.060          463.9213          251.1029    0.8325127       88550
## 209  67203  1280.681          405.2567          208.9797    0.7935087       66763
## 210 149693  1713.231          515.0971          355.0889    0.7783252      152499
## 211 166253  1545.369          585.8423          342.4428    0.8030268      172974
## 212 190108  1837.340          608.2514          417.2912    0.7738634      193506
## 213  31278   779.203          243.0763          134.2025    0.7717708       31230
## 214 211100  1663.261          709.4996          378.6382    0.7618486      213282
## 215 169228  1666.363          569.4960          385.3870    0.7222836      171275
## 216  35893   612.658          268.7215          180.0132    0.7134517       36300
## 217  58365  1162.411          389.0120          238.7191    0.7929780       59746
## 218  80384  1240.793          414.0987          270.0177    0.7477601       80316
## 219  44535   859.779          283.5330          185.3191    0.8182984       43631
## 220  33875   554.399          248.9471          194.2307    0.7923274       33398
## 221  34725   969.686          283.8429          184.4844    0.7829642       35975
## 222  51836   739.292          343.2011          179.4888    0.8386080       37694
## 223  86144  1282.810          495.7000          213.8189    0.8677365       91072
## 224  83420  1075.209          457.0807          223.2137    0.7698430       85854
## 225  34279   739.284          207.1883          203.0481    0.5430527       50252
## 226  55257   861.063          415.8173          167.0669    0.9162217       54606
## 227 152625  1720.391          518.7287          352.2643    0.7855782      154184
## 228  32541   850.971          269.1846          192.6273    0.7378627       33058
## 229  45429   735.825          283.1623          207.2676    0.7615679       44810
```

```
## 230  40603   964.481          353.4201          139.3923    0.8763239       38618
## 231  35325   520.233          258.8059          191.8948    0.7155361       34653
## 232  49145  1009.768          315.0336          158.7504    0.8110259       49439
## 233  67628  1107.266          379.7460          236.7809    0.8110426       84977
## 234  58647  1077.140          412.9913          182.8703    0.9220416       63194
## 235 154875  1558.899          590.3428          314.2425    0.7677404      156460
## 236  70107  1048.403          422.9271          216.8932    0.8858901       73616
## 237  39126  1005.764          264.5199          212.1667    0.7593835       26714
## 238  37096   824.488          292.1070          154.0440    0.7701632       37013
## 239  73192   939.421          385.5731          223.2394    0.8309923       70656
## 240  64449  1079.090          413.0804          194.4641    0.8552266       65091
## 241  32436   657.616          251.4040          153.0325    0.6635209       49217
## 242  79705  1351.396          433.2471          260.5370    0.8646708       99605
## 243  29948   733.340          232.2395          167.7971    0.7135500       34900
## 244  29075   942.853          237.9903          188.8401    0.7030548       32215
## 245  45816   654.541          324.4383          198.5742    0.8206140       45175
## 246  41963   585.399          277.2828          179.5222    0.6262951       27437
## 247  37558   836.922          258.1199          179.7810    0.8354381       22119
## 248  52072   965.205          373.7601          158.5636    0.9021436       54969
## 249  28779   743.125          238.9532          154.7330    0.6564287       44064
## 250  53147   816.306          365.4222          198.9693    0.8202060       68478
## 251  57410  1054.096          418.4756          159.5777    0.8467847       59196
## 252  48708   946.379          274.6663          193.8994    0.6467699       47154
## 253  69860  1185.646          394.0871          194.1924    0.8223477       69868
## 254  34986   581.383          271.4139          187.3284    0.6930588       37359
## 255  48491   921.074          356.8978          158.8818    0.8108159       50647
## 256  41901   764.129          289.1544          174.1294    0.6470044       42317
## 257  59285   966.697          376.1018          190.7388    0.8399761       60898
## 258  61969  1188.659          394.6919          208.8093    0.9170516       76945
## 259  43079   959.362          221.2485          206.0883    0.5399836       27781
## 260  58424  1028.340          392.4723          182.5569    0.8036837       63723
## 261  62574  1022.303          365.7092          216.6559    0.8524348       64772
## 262  90931  1269.657          442.6581          260.3532    0.8569889      107947
## 263  50342  1056.577          326.9796          227.5392    0.7905345       51144
## 264  35227   670.372          265.9658          142.0054    0.7377990       35470
## 265  39584   783.512          294.5293          209.6870    0.7582838       42399
## 266  32709   581.533          235.0613          168.5435    0.6403671       33150
## 267  82835  1167.671          432.9173          238.3289    0.7990995       88356
## 268  32939   719.518          236.0332          136.1433    0.7810645       31982
## 269  32908   943.573          263.9884          123.9457    0.8286407       31447
## 270  77561  1054.423          441.1558          246.0216    0.8883155       65605
## 271  52518   868.852          379.5724          188.7969    0.8951040       53160
## 272  25238   607.430          252.3748          136.3160    0.7376978       23192
## 273  37172   812.024          238.7062          185.6393    0.5822235       39162
## 274  55213   856.641          347.6485          195.0525    0.7861168       55423
## 275  41850  1104.327          325.7934          189.1241    0.8729054       42963
## 276  33185   745.036          269.7365          166.5273    0.7142782       28903
## 277  62678  1152.590          386.3905          203.5816    0.8264439       65537
## 278  44828  1034.567          316.9630          166.8571    0.8120212       46039
## 279 150690  1527.355          559.5186          350.3770    0.8111545      168086
## 280  36107   723.978          254.0179          173.3196    0.6035088       36418
## 281  60268  1126.081          377.9124          173.9182    0.8770688       58569
## 282  63569   962.861          377.8076          228.8948    0.8472092       51918
## 283  38140   978.196          305.9082          159.6031    0.7421443       41000
```

```
## 284 154620  1711.562          566.3462          361.4157    0.7642851      160949
## 285  52598  1098.499          353.9764          218.9900    0.7932607       53665
## 286  66833   992.148          403.6052          221.1101    0.8942752       68800
## 287  64879   863.111          357.4883          244.2561    0.8388577       80909
## 288  39817   799.429          221.0280          216.5497    0.5079522       39463
## 289  55603  1032.568          373.3863          173.4840    0.9257388       56258
## 290  24737   648.815          236.2088          174.8602    0.8259248       31547
## 291  52161  1108.507          406.6027          158.8845    0.8828785       53642
## 292  52264   643.462          286.9962          206.4109    0.5955089       53606
## 293  53957   957.624          403.4881          190.5200    0.9023030       57328
## 294  27449   437.003          256.9796          133.7749    0.7009572       28883
## 295  55225   983.470          413.0304          177.5500    0.9166267       56086
## 296  46118   863.535          302.1701          202.8361    0.7775752       44287
## 297  64053  1218.875          385.3458          227.9841    0.8340467       68434
## 298 183635  1760.958          602.2503          398.2431    0.7443729      185193
## 299  51131   771.322          366.2129          200.6439    0.8533770       49311
## 300  66132  1012.010          368.6143          230.1841    0.7929979       69782
## 301  40571   885.349          225.4265          222.3331    0.5648959       37291
## 302 202555  1699.213          659.5399          428.2591    0.7930955      201479
## 303  46113   709.006          297.1951          176.0895    0.7162128       47908
## 304  45133   989.730          316.0757          228.3089    0.7166700       47053
## 305  39754   957.332          239.8006          185.3478    0.6893131       39796
## 306 178532  1568.952          587.0671          368.5279    0.7253857      180484
## 307  37692   983.039          217.3688          172.4442    0.6594542       39793
## 308  66794  1263.645          389.4224          201.6914    0.8339558       56510
## 309  52585   807.157          403.8221          202.0007    0.8287861       55451
## 310  47425   697.081          336.8835          176.7181    0.7490362       50290
## 311  82508  1268.826          425.7677          236.2929    0.7802344       86339
## 312  58618   754.219          389.1210          199.7339    0.8908728       73883
## 313  26097   598.372          220.3772          120.0990    0.8218244       43269
## 314  43751   741.798          295.5346          165.5826    0.7562453       41841
## 315  33557   452.910          283.6599          147.9399    0.8267984       46136
## 316  50762  1023.609          369.0563          194.7682    0.8869604       53389
## 317 232828  1667.897          726.2194          443.5126    0.8149041      236322
## 318 147800  1625.571          537.6663          339.4271    0.7272175      149662
## 319  47040   974.435          256.6008          211.8488    0.5798175       46674
## 320 185557  1621.210          620.1007          348.8976    0.8356536      182882
## 321  37052   529.779          257.5092          190.3351    0.6494193       21525
## 322  37015   631.783          257.7935          180.0259    0.5338917       35932
## 323 143156  1673.479          530.7330          326.3476    0.7286451      132105
## 324 180826  1629.431          665.0436          347.9129    0.8772818      184707
## 325  49983  1172.355          372.1198          176.7417    0.9253029       53852
## 326  47923  1014.585          305.0624          242.6002    0.6564049       35440
## 327  47770   992.973          291.3696          179.5163    0.7696808       45012
## 328  56849  1098.146          375.1246          160.5099    0.8868301       55160
## 329  46686   652.245          314.1630          194.5731    0.7465655       46393
## 330  41337   895.307          224.1853          193.1762    0.5491389       56794
## 331  22303   395.147          188.7529          154.8194    0.7229560       22575
## 332  32380   840.208          265.5234          177.7253    0.7895932       27976
## 333  46634   743.014          323.3404          145.8884    0.9201833       47660
## 334  37379   923.008          240.5926          206.9687    0.6780941       38159
## 335  35084   857.753          269.3250          171.0803    0.7731667       35469
## 336  68540  1303.560          401.1100          239.2417    0.8489811       55904
## 337  66410  1276.231          358.1010          202.2775    0.8444025       67592
```

```
## 338  56268   751.819          358.0345          231.0493    0.7689117       58879
## 339  50280  1029.145          426.8667          177.3175    0.9234379       54474
## 340 154883  1535.875          504.5626          398.1684    0.7034508      141892
## 341  28429   619.935          231.5129          159.4762    0.7783625       29144
## 342  42244  1011.759          264.5219          210.7939    0.5902541       43006
## 343 134434  1631.772          517.7357          313.6365    0.7420050      133303
## 344  40499   932.462          262.0834          212.6327    0.7094451       44575
## 345 214085  1715.151          698.6109          387.1115    0.7785568      218114
## 346  35700   504.942          258.4597          192.1220    0.7388931       35565
## 347  39340   901.012          288.5465          201.8040    0.7220766       39588
## 348  39837  1037.146          289.0021          178.8722    0.7467122       44456
## 349  36923   973.582          268.7348          157.5038    0.7753144       52861
## 350  22540   607.844          224.0326          156.5330    0.7795643       24255
## 351  51175   991.737          365.5798          158.5586    0.8557754       66511
## 352  38494   948.575          269.4312          203.9067    0.6451443       35322
## 353  44602   909.690          349.4549          141.6688    0.8929595       45420
## 354 186634  1536.534          618.1315          351.4337    0.8152913      188680
## 355  47055   867.220          274.3557          178.8507    0.7389393       61327
## 356  56845   768.281          393.7553          196.4303    0.8780607       52704
## 357  36435   950.185          249.7785          203.0124    0.7362044       37672
## 358 146022  1463.486          530.2979          327.6362    0.7481226      163767
## 359  46892   884.623          269.6431          211.1867    0.7688520       45543
## 360  39477   716.994          300.6648          212.7722    0.7485013       42143
## 361  53445   718.226          367.0625          202.9381    0.8524399       40381
## 362  57366  1223.650          404.5818          201.9107    0.8915172       57301
## 363 160947  1411.838          501.8501          412.1905    0.6876746      160785
## 364 164228  1443.682          597.1694          344.6801    0.8192237      164080
## 365  76898   848.031          426.4862          253.3448    0.8535540       74841
## 366 169645  1860.223          643.3375          318.1908    0.8464599      175137
## 367  55989   744.523          372.9678          183.6899    0.8236364       57045
## 368  60149  1046.277          376.1847          173.3476    0.9098094       59606
## 369  33917   601.704          271.9928          165.1197    0.8090110       37460
## 370  51189  1097.173          346.8365          155.2865    0.8769189       51603
## 371  51936  1145.292          371.1432          189.2745    0.8891133       69460
## 372 168242  1468.813          572.0585          348.1811    0.7913212      171434
## 373  47916  1141.677          373.4307          179.7012    0.8267843       51313
## 374  44260   576.980          255.2882          195.2451    0.5340203       43059
## 375  55627   770.958          365.1830          166.7386    0.8494033       42317
## 376 183718  1591.305          659.1192          391.5652    0.8317911      185905
## 377 249735  2062.606          713.9824          435.0429    0.7680315      250060
## 378  63720   903.912          440.4249          226.5844    0.8842439       66073
## 379  53879   930.577          386.6344          163.6776    0.9090649       53503
## 380  42811   836.287          250.3319          191.2631    0.6818130       45557
## 381  51528  1107.619          366.7241          193.3863    0.8243707       38123
## 382 220294  1789.406          714.2526          364.6791    0.8067740      237536
## 383  41515   760.806          230.7854          224.9316    0.5596423       44090
## 384  52096  1181.491          394.3513          170.5587    0.8728386       53267
## 385  36032   878.252          243.6804          171.7056    0.4666291       36086
## 386  47758   757.637          320.9111          217.8410    0.7186630       47490
## 387  24398   799.790          210.7640          131.4184    0.7430841       26069
## 388  28265   799.385          239.9710          162.7897    0.7207861       47306
## 389  47060   842.567          304.1620          193.5985    0.7097500       47601
## 390  42610   862.418          333.3142          173.9697    0.8539022       40883
## 391  32265   466.849          227.8512          164.5870    0.5480098       17863
```

```
## 392 146008 1547.287          526.6134          339.0442     0.7392978      133888
## 393  35797  671.153          266.7045          163.6196     0.6161315       32344
## 394  38088  962.832          324.6865          151.3640     0.8463682       41105
## 395  42021  921.955          320.9203          201.5516     0.8393790       42449
## 396  68469 1177.963          412.1621          255.0445     0.8149272       71341
## 397 179006 1867.030          650.6493          382.0850     0.8464007      182340
## 398  41640  767.596          291.0091          188.9095     0.7480440       58725
## 399  41958  925.818          274.0112          229.0812     0.5051812       37809
## 400  49730  952.531          390.0399          185.0005     0.8375965       49241
## 401  45897  956.536          289.7472          168.0019     0.8150151       47850
## 402  40426  547.721          272.0250          197.8980     0.5498702       40644
## 403  40202  706.123          299.6430          145.0322     0.7826140       39737
## 404  41744  622.339          318.6852          158.7988     0.7645828       44567
## 405  44696  857.341          385.8355          146.1182     0.8809505       47883
## 406  37875  559.614          255.5349          158.9399     0.7603656       38407
## 407  51573  749.351          372.6308          180.4936     0.8575654       52116
## 408  83592 1039.133          459.0070          251.4176     0.7922438       85869
## 409  45355  811.575          241.0159          226.2283     0.5734948       46772
## 410  35895  887.494          284.7949          160.3113     0.7588214       36717
## 411 136948 1299.737          496.3459          317.9229     0.7421471      138115
## 412 159821 1475.636          541.2617          347.7258     0.7917036      168310
## 413  38524  546.644          267.7360          206.1234     0.7826100       55040
## 414  54800 1094.706          371.0314          167.3673     0.8587330       59211
## 415 131928 1258.848          517.0036          357.5870     0.7826045      134335
## 416  35584  751.029          239.5259          173.4069     0.7056926       22825
## 417  91484 1058.063          486.5263          279.5823     0.8270301       93308
## 418  76371 1089.041          404.8564          232.8270     0.8544160       76051
## 419  40647  598.768          225.5200          232.4094     0.4008117       40123
## 420  98782 1363.887          467.2333          260.8310     0.8262465      101548
## 421  57433  814.921          395.3629          191.4208     0.8732886       58524
## 422  36805  538.234          249.4310          178.1164     0.5559462       37650
## 423  37520  863.998          275.9357          178.3487     0.6344298       38810
## 424  29340  620.438          206.8622          180.8385     0.7034402       30481
## 425  38652  533.042          289.1696          169.1542     0.8079375       38136
## 426  40848  732.683          332.4887          175.0873     0.9026905       41575
## 427  39444 1016.332          268.7458          220.6651     0.6230346       41087
## 428  81247 1377.145          408.7835          211.9023     0.8760159       82704
## 429  67190 1273.223          389.8705          214.3860     0.8417936       82798
## 430  37395  662.182          283.8289          208.0121     0.6779018       37017
## 431 200390 1966.068          643.0124          395.3047     0.7736344      206402
## 432  82093  995.106          434.4270          227.4749     0.8220766       97622
## 433  58413  802.873          381.4775          213.3675     0.8914542       60605
## 434  37628  887.425          274.8429          171.7378     0.6285196       42050
## 435  45737  860.838          309.7908          169.6991     0.7345367       45658
## 436  41459  776.411          256.2861          224.4085     0.5758188       41384
## 437  78303 1041.988          430.1746          221.4661     0.7812581       81002
## 438 209433 1683.390          684.3173          379.3523     0.8164365      211979
## 439  77548 1274.352          393.2707          266.7925     0.7913815       77389
## 440  50292  844.359          348.8614          170.9970     0.9045446       55600
## 441  38803  914.200          258.9901          194.7805     0.6507419       39879
## 442  52233  977.459          383.6671          159.7187     0.8419007       53030
## 443  36578  821.229          304.8855          144.2484     0.8890186       37258
## 444  44051  899.410          322.4946          172.5635     0.7980616       44477
## 445  54089 1052.862          376.7647          186.6801     0.8809992       57738
```

```
## 446  49124   894.916          336.1319          152.5379   0.8607709      54115
## 447  34695   875.415          261.8349          164.4137   0.6035628      33357
## 448  37235   906.002          301.4923          185.0096   0.7532553      26656
## 449  77415  1080.262          402.5913          236.3321   0.8082343      94085
## 450  63739  1116.894          383.8037          222.3532   0.8242151      63456
## 451  30240   489.289          226.2420          149.0556   0.7574704      46082
## 452  44277   768.119          289.7034          206.6690   0.7529194      41255
## 453  24473   603.183          232.4501          133.5446   0.7415483      25768
## 454  40960   899.424          337.3981          190.5949   0.8662591      41219
## 455 190285  1550.268          595.3704          383.1187   0.7289976     189267
## 456  49386   775.494          314.7132          156.6216   0.8698515      35591
## 457  49109   662.257          332.1357          228.4712   0.7796800      47022
## 458  29076   776.353          226.2847          169.1205   0.6497801      25631
## 459  60952   815.442          400.1212          173.2700   0.9067783      64793
## 460  45072   835.693          284.1367          246.9852   0.5747006      47645
## 461  51855  1102.676          337.4547          200.4970   0.7751683      51670
## 462  36247   683.685          251.3632          185.4752   0.4938789      37982
## 463  57874  1304.838          380.1049          177.8209   0.7760973      57644
## 464  98270  1483.677          471.9857          253.5066   0.8112805     100194
## 465  54678   970.211          391.8545          158.7127   0.9286646      54375
## 466  36409   762.928          243.0928          174.3779   0.6784956      33130
## 467 175635  1754.204          592.6398          351.1553   0.7819272     178063
## 468  38667   913.569          297.0581          158.3755   0.8299504      40725
## 469  67939  1172.165          393.9947          249.0846   0.7997240      66561
## 470  77758  1268.664          417.7930          205.8006   0.8040547      74992
## 471  40908   626.124          289.9027          190.8445   0.7831300      60080
## 472  24856   578.518          178.7249          179.3847   0.6814092      24779
## 473  32884   794.005          240.7917          167.9515   0.4943963      47647
## 474  32721   827.359          246.6107          192.2743   0.7597115      18011
## 475  37699   708.428          218.6279          209.3968   0.5392219      51947
## 476  41049   657.956          266.1656          204.1273   0.7624179      42412
## 477  56622  1032.303          368.8240          209.0765   0.8347949      73945
## 478  47482   926.995          327.5211          174.8515   0.8403116      44767
## 479 177582  1596.787          589.7043          384.9329   0.7262067     183294
## 480  57280  1153.790          383.2832          216.4353   0.8887615      71734
## 481  40622  1018.766          230.4429          206.6139   0.5955848      40308
## 482 155543  1667.661          544.9501          354.6856   0.7472059     174014
## 483  38197   947.391          244.0495          201.9185   0.7208701      38301
## 484  43968   912.930          331.9771          209.2071   0.7091184      49294
## 485 198387  1932.776          676.2527          334.7792   0.8395814     202378
## 486 213640  1941.540          642.0649          420.7663   0.7558714     217702
## 487  36181   482.552          238.0492          205.9417   0.6206716      39071
## 488  35243   607.678          209.7820          200.0775   0.5838210      35537
## 489  77688   918.299          401.6555          251.1759   0.8029981      83363
## 490  68896  1239.043          391.7123          255.0746   0.8036297      71154
## 491  45105  1098.958          307.9845          180.8431   0.7982239      47053
## 492  61745  1150.976          389.4481          195.4168   0.8989720      61546
## 493  81699  1230.290          433.9337          230.2199   0.8062643      80463
## 494  44894   878.455          293.0212          183.2681   0.7665772      46731
## 495  51882  1178.407          398.8073          179.7371   0.8702198      51603
## 496  28888   811.530          237.2527          159.3179   0.8374826      29285
## 497  43179  1019.327          278.5582          191.1445   0.7220008      40436
## 498  71572  1151.165          379.3453          219.1732   0.8762368      72575
## 499  38726   826.680          289.1674          167.7547   0.7806703      38307
```

```
## 500  44297  1023.622         281.3037         148.4305    0.8092784       38279
## 501  31497   534.385         256.3430         137.2253    0.7483524       32941
## 502  48377  1080.662         299.4062         177.8507    0.7489456       49465
## 503  77548  1148.576         376.9206         253.5511    0.7573314       64894
## 504  40935   992.741         285.9825         210.5502    0.6457737       43158
## 505  50789  1065.324         369.5757         151.2356    0.8763811       56150
## 506  34972   832.179         247.6647         202.8834    0.5807127       32866
## 507  76768   952.231         390.7370         212.8038    0.7975542       74661
## 508  89396  1240.168         445.2546         277.1509    0.7778914       88613
## 509  87264  1285.597         471.7791         255.5356    0.8128121       85146
## 510  53229  1079.116         307.2338         229.7524    0.7540529       53762
## 511  60233  1259.146         338.9371         217.2196    0.7529517       68154
## 512 171104  1409.496         590.6257         349.4927    0.7501133      172188
## 513  54226   798.708         384.5934         160.9300    0.8402205       52172
## 514  30452   887.672         235.6093         149.7297    0.6437848       31570
## 515 185497  1756.149         595.0286         399.5375    0.7335807      187914
## 516  78661   962.095         434.5890         228.5338    0.7968205       80278
## 517  32107   733.953         265.8134         150.1619    0.7751555       35628
## 518  47993   861.757         280.8658         207.5887    0.5901660       48280
## 519  34164   673.037         264.5790         147.2754    0.7278678       36948
## 520  39928   959.267         263.0531         195.1238    0.8246212       27030
## 521  50205  1097.428         319.2999         215.0235    0.6779306       51041
## 522  69338  1042.956         430.1284         238.4171    0.7957049       55610
## 523  58084   841.930         376.3860         181.7511    0.8324255       54303
## 524  39885   617.387         239.6879         223.2868    0.6142175       44058
## 525  47796  1104.419         305.8504         203.8704    0.7074596       49047
## 526  37225   807.989         273.4560         168.7984    0.6186456       38940
## 527  37600   589.195         239.3575         221.6656    0.6141184       38509
## 528  31459   777.179         256.0299         146.7855    0.7503278       28546
## 529 143498  1542.572         519.1268         327.2987    0.7605048      146822
## 530  55140   895.072         369.5710         211.0135    0.8510535       41102
## 531  36368   579.034         265.5064         202.1416    0.7467598       33903
## 532  27100   706.749         234.8153         140.9335    0.7217935       27811
## 533  34888   593.000         279.4393         145.6495    0.8242549       32388
## 534  75936  1070.503         417.8350         240.8155    0.8206605       76106
## 535  54063   825.036         271.4838         211.1359    0.5955477       54040
## 536  45826   735.403         338.1822         185.5944    0.8015287       47584
## 537  43725  1034.886         270.4840         190.0199    0.7344243       60716
## 538  83085   919.784         416.1931         270.0432    0.7444125       81559
## 539  45560   925.849         296.2763         198.2019    0.7594753       45214
## 540 139710  1358.215         500.8133         356.0357    0.7432230      142169
## 541  40612   537.060         256.9880         181.0254    0.6270407       39979
## 542 155430  1637.212         593.3180         342.6016    0.7584112      155124
## 543  80131  1199.867         419.2811         241.7833    0.7821933       81786
## 544  41048   688.818         262.1423         209.2891    0.7523825       42272
## 545  67250  1038.047         434.5144         186.4879    0.8852534       72306
## 546 143781  1545.041         517.6599         353.1618    0.7623974      147240
## 547  51558   725.673         339.9277         176.0906    0.8416394       54025
## 548  28599   886.634         255.5155         162.6265    0.7867919       27125
## 549  58995  1115.556         420.6906         190.8871    0.8703503       74252
## 550  35518   705.645         261.6322         190.0529    0.6315345       38273
## 551  42071   563.005         289.6905         217.2284    0.6284440       57074
## 552  35247   903.676         251.9407         175.5833    0.6521170       33456
## 553  72500   842.840         419.7337         199.3077    0.8216412       72175
```

```
## 554 179929  1615.406          593.0100          349.7780    0.7746718      180754
## 555 174293  1888.752          642.3852          353.3157    0.8549183      176391
## 556  30968   501.594          258.1017          151.8034    0.7458098       30034
## 557  69523  1174.885          360.5400          257.5006    0.8277302       71773
## 558  44070   611.939          303.8471          177.7594    0.7397032       44406
## 559  30562   733.971          262.9495          142.4445    0.7001361       31814
## 560  28847   939.742          251.1291          188.7013    0.7143352       47905
## 561  48916  1037.073          334.7266          183.0927    0.8712546       51616
## 562  34045   853.822          260.6494          181.0813    0.8377674       49359
## 563 165420  1822.062          600.9803          335.7467    0.7510061      169130
## 564 154010  1417.810          550.4389          385.1870    0.7765504      140209
## 565  59437   867.725          432.5650          188.2210    0.8912549       61916
## 566  30516   550.887          271.9511          185.8329    0.7621474       34150
## 567  58480   859.763          362.9967          182.6512    0.8405059       77852
## 568  22630   552.409          221.7278          153.9376    0.7368451       22828
## 569  63278  1089.993          375.3604          221.0352    0.7998947       63722
## 570 200060  1956.082          587.6283          406.3252    0.7336534      189083
## 571  67811  1121.943          372.4281          207.5084    0.7449543       66104
## 572  28585   710.423          259.8651          141.3601    0.7650500       28012
## 573  29202   633.390          241.4616          196.4065    0.6904549       31809
## 574  47344   990.775          331.5870          185.7763    0.7741169       33480
## 575  59527  1193.450          412.2103          202.3971    0.8425747       58941
## 576 183919  1843.140          578.8831          384.5299    0.7853768      187679
## 577 193309  1667.095          651.5148          403.9754    0.8354532      195489
## 578  41723   832.772          284.5642          206.1477    0.6113280       57511
## 579 165746  1474.298          577.3555          374.8546    0.7815468      166544
## 580  34829   823.317          215.7210          185.0817    0.4851978       38275
## 581  35161   983.413          212.9920          196.8999    0.4654405       35102
## 582 139370  1359.333          539.4482          356.6197    0.7878780      143772
## 583 154912  1703.724          559.1126          348.8163    0.7304680      158478
## 584  42674   721.252          286.7873          213.9092    0.7456524       42275
## 585  37322   971.200          206.2769          214.4111    0.4062331       36489
## 586  75687  1328.421          413.2995          237.2165    0.8832753       76183
## 587  38970   695.028          260.0054          195.2712    0.5689384       55051
## 588 149656  1733.206          533.6631          398.9406    0.7060158      156809
## 589 182494  1557.777          602.6230          415.9801    0.7318862      188539
## 590  40815   687.558          307.0811          157.5660    0.7638890       37253
## 591  39712   994.407          292.3470          212.0913    0.7617222       43126
## 592  41407   990.092          267.7325          198.2335    0.7871522       41902
## 593  48266  1116.257          345.2928          147.1937    0.8244857       49718
## 594  54116  1043.237          376.5947          215.4424    0.8592042       55266
## 595  42603  1031.973          289.4615          209.2309    0.7111240       42187
## 596  39256   653.973          252.3059          167.8313    0.8105678       39847
## 597 208346  1651.978          638.6771          429.1540    0.7720193      211549
## 598 196761  1502.779          642.9903          384.6192    0.7302345      183696
## 599 148457  1575.512          522.1945          335.4377    0.8047005      146234
## 600  43918   819.149          322.6501          172.4629    0.9111586       47588
## 601  57217  1174.101          426.3584          177.4971    0.9045762       57163
## 602  25606   652.907          196.7897          180.3980    0.7448019       29690
## 603  47078   762.287          339.0011          165.1169    0.8324870       45267
## 604  49224   807.032          326.8414          188.1028    0.7794707       49355
## 605  38513   562.964          260.5711          211.4960    0.5450152       53762
## 606  28063   471.615          213.5311          125.5994    0.7150178       27562
## 607  43553   945.961          274.5416          163.5986    0.8086281       39438
```

```
## 608  44488   650.116          297.4555          205.2276    0.7381315      46816
## 609  58635  1194.184          404.2915          211.5638    0.8867488      60168
## 610  39065   995.225          266.5030          154.4683    0.7899656      35338
## 611 181323  1554.945          645.8096          387.6012    0.8367964     181684
## 612  35685   587.776          226.6353          182.2684    0.6477945      37823
## 613 201313  1773.226          635.1119          395.8985    0.7748016     200337
## 614  65846   968.135          386.6718          222.4063    0.8327617      52414
## 615  66670  1173.004          405.2426          227.9262    0.7816959      70301
## 616  55580  1072.629          336.9405          229.5821    0.8047740      56742
## 617  31394   878.096          241.6937          198.9210    0.6298322      31198
## 618  48949   925.712          356.5720          182.1724    0.9119475      49587
## 619  33578   460.451          221.3175          210.8887    0.5538851      33651
## 620  32231   665.168          256.6440          141.0072    0.7955090      30156
## 621  46108   810.985          317.4067          153.3801    0.8697500      32646
## 622  45791   884.896          296.0341          185.7366    0.6753881      48564
## 623  79402  1177.949          406.5889          234.7017    0.7800435      79988
## 624  38474   899.125          246.5954          183.7807    0.6370205      39413
## 625  93074  1156.392          436.3256          274.9230    0.7696551      95501
## 626  67051   935.915          416.7621          237.0403    0.8005413      66574
## 627  38702   946.383          223.6091          224.5550    0.6355226      24694
## 628  43276   710.285          315.1467          183.7673    0.7915676      44879
## 629  61528  1267.263          410.9823          195.7153    0.8607838      59008
## 630  41720   693.965          275.1738          174.1640    0.7398267      41681
## 631  67442  1156.294          356.0450          257.5225    0.8008035      66438
## 632  74359  1058.691          392.9518          220.5480    0.8392564      62355
## 633  55382   819.783          369.3126          194.1936    0.7930354      43096
## 634  36980   785.456          227.1325          208.0016    0.3661537      36451
## 635  44928   656.497          270.1861          243.1288    0.5523120      51760
## 636  49322   825.304          299.4790          178.6941    0.7247398      49794
## 637  80919  1077.878          438.9849          239.4192    0.8152527      67015
## 638  45890  1057.735          283.2403          188.5632    0.6358271      49550
## 639  81610  1208.327          479.2523          251.7283    0.8890402      83610
## 640  29557   776.545          252.4868          172.6177    0.6526281      48652
## 641  57101  1150.255          385.8010          174.1864    0.8847168      55046
## 642  37866   821.283          298.0023          149.8192    0.7157217      35230
## 643  40975   829.327          224.3024          239.1011    0.3187789      45341
## 644  66118   839.399          370.8523          237.7552    0.8434752      83578
## 645  51688  1113.242          360.0790          168.1433    0.9353118      50523
## 646  56761  1100.587          409.1563          228.7561    0.8876009      76485
## 647  34868   946.654          260.1857          168.3293    0.7210013      36343
## 648  48063   828.524          305.9761          212.4966    0.6315372      48441
## 649  31776   899.928          219.8857          186.9115    0.5494947      29873
## 650  29790   905.199          263.9172          182.7993    0.7365111      28742
## 651  75296  1201.259          414.1586          217.2364    0.7838796      78452
## 652 134075  1613.882          486.2329          361.8664    0.6382360     131855
## 653  45660   787.663          373.8682          185.4263    0.9019375      63299
## 654  39525   767.613          259.4727          202.9434    0.6237744      56269
## 655  50669   988.675          300.3497          192.1252    0.8001247      50547
## 656  37106   705.325          247.7183          197.0515    0.3699794      52393
## 657  42138   599.841          267.2470          161.6984    0.7460560      41473
## 658  35684   942.692          271.5882          164.9882    0.7884020      52481
## 659  38279   861.716          252.5199          182.1103    0.6136508      37528
## 660 136227  1679.215          498.8051          341.7147    0.6802030     140258
## 661  29805   616.393          258.4597          135.6010    0.8207437      28616
```

```
## 662  51109  1122.267          351.8440          184.7568    0.7698453       50520
## 663  75139  1344.203          423.0667          207.0941    0.8435958       77642
## 664  50824  1032.407          350.9646          198.7481    0.8096260       50023
## 665  74083  1208.155          388.6351          243.9356    0.8410404       76571
## 666  46201   840.965          264.9320          232.1648    0.6879742       45992
## 667  61522   998.923          402.5973          195.7702    0.8749931       57870
## 668  40358   912.333          236.2828          169.3433    0.6073384       38276
## 669  36128   501.450          231.6781          179.3484    0.5807068       32352
## 670  41909   830.363          293.3744          208.5172    0.7972855       26570
## 671  41229  1008.376          248.6964          183.2054    0.6372128       41302
## 672  59922   907.423          439.1916          163.1640    0.9184221       49510
## 673  68448   988.166          368.6728          248.9993    0.7402152       70536
## 674  76524  1146.189          394.8534          255.7916    0.7852816       75918
## 675  79435  1091.826          455.6239          249.5647    0.8324468       82794
## 676  52205   945.662          341.2053          189.2632    0.8061326       56709
## 677  35729   838.985          304.1943          202.9423    0.7714552       39672
## 678 225832  1698.037          650.1348          399.5840    0.7388249      227469
## 679  29831   789.435          232.9713          146.1483    0.6850538       30125
## 680  73065   847.056          396.6510          204.7719    0.7992582       74500
## 681  49968   682.866          325.1364          182.9342    0.7860986       51233
## 682  27566   879.832          216.2399          185.2801    0.6418003       30160
## 683  52509   898.446          382.5706          192.0146    0.8518140       65451
## 684 141385  1586.014          510.0347          352.3335    0.7659418      144027
## 685  50240  1147.707          348.0048          189.1180    0.8982332       50949
## 686  55401   960.169          403.8108          185.8112    0.8494317       56083
## 687  81343   983.549          422.5054          205.9441    0.8539154       82483
## 688  41361   915.837          317.7655          147.8604    0.8394786       41892
## 689  43400   702.944          280.9054          192.1291    0.7633956       43389
## 690  31686   723.474          269.1687          166.6040    0.7433894       18937
## 691  43359   895.333          317.1677          195.4671    0.7817590       45113
## 692  39313   769.620          307.3533          189.0245    0.7581490       55944
## 693  30188   534.524          224.1347          164.8586    0.7838018       28249
## 694  25285   591.694          219.4680          160.3215    0.6917966       27433
## 695  50728  1045.386          329.6624          194.6435    0.7736125       47841
## 696  49070   954.255          364.0100          195.2224    0.8474792       51589
## 697  73143  1009.977          389.7383          216.9465    0.7717608       73814
## 698  50278   822.772          333.6897          213.7677    0.7630222       46680
## 699  48619  1142.958          368.9922          174.3158    0.8450827       47715
## 700 147197  1294.066          567.9384          359.3278    0.7316935      147342
## 701  68770  1237.289          428.1024          197.1599    0.7829601       70217
## 702  40062   973.539          272.5653          171.2238    0.7469789       54625
## 703 161859  1521.564          562.2198          372.1281    0.7025637      179137
## 704  55056  1021.561          324.4019          182.7663    0.8470174       55963
## 705  41769   816.881          311.4392          198.2284    0.8112585       43981
## 706  46195   751.614          311.0540          166.8762    0.7511936       47472
## 707  32815   652.769          249.7707          179.5117    0.6562224       35599
## 708  55122   943.685          393.7342          167.5183    0.8822938       54408
## 709  72386  1321.815          410.7370          225.9718    0.8859166       59390
## 710  32512   748.830          288.9574          129.3525    0.8224594       19169
## 711  75919  1224.433          440.0390          248.0276    0.8684991       74763
## 712  31829   769.158          235.2155          157.3428    0.7757252       18802
## 713  40747   838.235          256.1911          171.4523    0.6765130       55244
## 714  47382  1029.185          368.3777          200.4038    0.8415019       47656
## 715  48154   873.985          317.7625          166.2189    0.8019129       48538
```

```
## 716  36130  762.638          292.5778         155.7446    0.7949508     38200
## 717  46639  815.760          263.2633         236.6419    0.5335189     44336
## 718  38360  798.123          263.6960         186.7998    0.7978174     35609
## 719  39150  582.485          265.8513         220.6714    0.5339474     43619
## 720 134120 1587.187          514.6420         356.8769    0.8233281    138527
## 721  34347  602.720          230.5971         180.1750    0.6640752     37017
## 722  49016 1142.417          322.1578         191.8560    0.8006958     52446
## 723  53051 1150.129          336.4271         191.8165    0.8816856     54269
## 724  49150  990.620          337.4990         174.3098    0.8981001     65013
## 725  52857 1007.599          400.6579         201.1516    0.9120882     53159
## 726  54780  971.586          378.2111         170.5949    0.8617046     57896
## 727  44677 1027.497          282.4239         215.0497    0.5544899     44817
## 728  51218  822.127          388.1063         182.1196    0.8868048     53475
## 729  82659 1323.765          439.2215         226.8347    0.8523649     98973
## 730  46517 1017.459          314.9060         215.5668    0.7958225     45249
## 731  28875  541.721          258.6217         169.2389    0.7425510     30581
## 732  80147 1334.248          402.6693         246.2522    0.7777974     80387
## 733  41709  756.821          224.1626         198.0214    0.5406292     42025
## 734  44012  708.254          258.0482         217.5587    0.6221692     45238
## 735  26209  764.975          206.5995         186.7256    0.6981133     28945
## 736  49405  768.660          374.2870         187.5011    0.9034829     51986
## 737  26901  645.149          244.3801         160.7833    0.7513360     27306
## 738  74205  862.636          386.0913         234.9346    0.8367490     75942
## 739  41973  864.050          305.3359         175.2638    0.8247006     45035
## 740 148427 1655.117          543.2698         386.0832    0.6875439    148407
## 741 182429 1690.174          633.0209         399.8378    0.8283085    186471
## 742  73311 1293.887          376.8362         212.3622    0.8645530     74744
## 743  67918  873.174          391.4218         229.7311    0.7632798     69234
## 744  39467  923.921          335.2380         157.5132    0.8843038     40488
## 745  45406  641.843          294.3235         213.0241    0.7678364     46182
## 746 202592 1509.525          605.1885         445.5121    0.7256405    208344
## 747  42234  984.740          266.4281         222.6268    0.5539234     44227
## 748  36622  798.891          226.4202         198.3951    0.6708215     37108
## 749  70927  979.002          455.9187         208.4052    0.8397711     90627
## 750  66641 1015.898          349.6331         217.6780    0.8316604     67566
## 751  60467 1075.218          404.0081         204.6683    0.8394315     58453
## 752 185030 1682.618          657.7765         366.8214    0.7844327    190332
## 753 163716 1432.943          560.8693         355.1632    0.7651865    152573
## 754  64524  863.749          336.4593         204.3391    0.7359622     79767
## 755 113873 1558.469          432.4385         341.9643    0.6996721    112002
## 756  51306  964.887          347.1675         182.4487    0.7895252     51612
## 757  37162  825.318          289.0164         174.8150    0.7646242     39986
## 758  44200  766.897          307.3040         173.4775    0.8484815     46820
## 759  38187  982.005          223.6720         194.0993    0.6807576     37505
## 760  25787  701.217          245.3512         177.7262    0.7320449     30436
## 761  36967  926.969          265.9617         174.3649    0.6693897     37950
## 762 173165 1485.844          549.4320         405.5811    0.7672372    175193
## 763  69659 1050.114          419.9850         213.0247    0.8940288     69867
## 764  40519  793.521          272.5633         217.6421    0.5936766     42587
## 765  43681  717.613          277.7295         166.1042    0.7942027     46158
## 766 155585 1614.485          514.1321         345.9143    0.7369475    156545
## 767  35285  578.069          199.7193         204.7835    0.4628187     33018
## 768  67999 1009.366          377.0580         229.0344    0.7887875     70685
## 769  35685  855.700          268.4886         184.8397    0.7357390     35557
```

```
## 770  79534  1261.032          414.9752          251.6505    0.7944106       92612
## 771  62680  1129.208          409.6410          182.1332    0.8465169       57061
## 772  27073   663.904          234.9398          118.4034    0.7486176       29717
## 773 159394  1315.729          567.1796          371.6367    0.7830635      160691
## 774 146007  1335.163          521.9862          364.8319    0.7450517      147014
## 775  38148   838.758          232.5748          195.6291    0.5927680       37434
## 776  39393   784.640          237.7682          225.1424    0.4723673       39818
## 777 203657  1638.185          633.8912          411.8815    0.6544741      206515
## 778 204001  1667.463          642.5193          409.8346    0.7512824      206831
## 779  77999   963.638          392.5638          245.0749    0.7501294       77484
## 780 170897  1479.290          539.2903          373.9306    0.7395043      173105
## 781  36619   669.170          294.9505          185.1227    0.7543081       38609
## 782  38274   862.891          243.7939          173.6711    0.6743253       23864
## 783 200381  1731.120          651.0215          403.0658    0.8483121      206008
## 784  28127   924.983          260.7829          139.0422    0.6962185       31115
## 785  49759   967.034          338.0984          171.5255    0.8159006       35325
## 786  49359   782.326          278.9213          201.2459    0.7441768       49068
## 787  41983   551.059          277.1988          201.5367    0.5656869       35883
## 788  39938   912.880          217.2359          219.9554    0.4553456       40046
## 789  53955   784.965          395.1458          166.5598    0.8972691       52331
## 790 157712  1364.650          493.7012          437.5306    0.5472634      178469
## 791 152598  1421.586          595.7981          369.9428    0.8144651      158494
## 792  27982   714.964          213.3242          152.9451    0.7902946       26133
## 793  53938  1098.276          352.0416          169.0417    0.8182790       71115
## 794  33103   691.654          280.1010          140.3458    0.7266217       33789
## 795  65294  1280.051          426.9208          191.0123    0.8627872       66338
## 796 176932  1515.825          629.4302          391.3069    0.7825170      192338
## 797  50654  1117.317          349.7472          159.9209    0.8544052       66701
## 798  73404  1277.096          413.6404          212.8953    0.8747118       74019
## 799  64335  1221.500          384.7834          178.1819    0.9035973       82148
## 800  59338   826.621          380.6689          217.3157    0.8260214       61310
## 801  33107   566.592          224.2791          152.1315    0.7457736       19196
## 802  43222  1084.847          318.8018          197.3050    0.7513866       45190
## 803  60364  1080.521          360.6216          221.0007    0.7937025       47666
## 804 142509  1572.902          536.3023          354.9181    0.7748678      141250
## 805  41275   952.744          243.8133          183.0561    0.5873316       40906
## 806 181335  1798.340          614.8143          373.5686    0.8050903      181843
## 807  39093   725.403          216.0916          169.1275    0.5701826       33011
## 808 127978  1432.967          500.1283          351.8034    0.7404370      131003
## 809  76191  1160.138          405.1286          227.5860    0.8704411       78591
## 810 171325  1715.094          572.4178          401.3933    0.7193781      180152
## 811  52253   730.117          404.0644          164.7617    0.8492298       53329
## 812  30396   674.360          249.4744          186.7467    0.7769480       33503
## 813  36727   754.302          268.1055          183.8919    0.6058814       36297
## 814  46729  1077.358          288.7316          184.0061    0.7230676       46956
## 815 171784  1718.413          604.6297          333.6529    0.8223156      174813
## 816  31895   758.821          209.7143          182.9200    0.6476180       32846
## 817  56981   898.083          367.6524          209.7731    0.8931531       59030
## 818  75574   922.550          395.9175          219.4382    0.8434083       91298
## 819  77121   971.744          365.2095          270.5196    0.8141475       75879
## 820  65042  1034.670          394.5440          207.3949    0.8302606       69621
## 821  28892   849.853          190.1466          168.5511    0.5937191       29212
## 822  31193   791.954          247.5479          161.9274    0.7616853       31456
## 823  43870   937.159          245.4293          223.3583    0.4918362       41609
```

```
## 824  42474   612.220          245.6541          195.4178    0.7255279      39124
## 825  39557   894.795          289.5034          217.3364    0.7371197      43763
## 826  50638  1140.175          311.0451          176.7623    0.7759515      48075
## 827  44620   956.620          249.9468          206.4100    0.6521962      44497
## 828  48251  1081.033          329.6734          155.6299    0.8232212      44813
## 829  80449   963.224          448.4181          273.0559    0.8037180      82633
## 830  29089   536.730          246.4712          128.1000    0.7201348      24858
## 831 158301  1724.550          560.8001          368.4621    0.6995935     146174
## 832  28513   556.046          228.5562          167.7368    0.7489687      28064
## 833  36420   793.073          241.1855          188.8901    0.7236176      38379
## 834  34644   568.361          239.2546          205.0088    0.6169211      51972
## 835 151714  1699.135          579.7745          353.1857    0.7772238     167421
## 836 146933  1713.606          572.5500          362.4591    0.8016040     149091
## 837  41277   620.585          237.0563          194.8058    0.5617019      38980
## 838  61730   917.107          392.2187          172.6737    0.8466094      59144
## 839  46970  1086.119          294.8525          190.6519    0.7434855      45263
## 840 169223  1705.924          613.7935          365.7673    0.7729450     172644
## 841  37714   831.876          248.1514          192.2229    0.5017897      37616
## 842  66290  1159.642          369.3685          214.8204    0.8661942      71919
## 843  41899   669.018          247.7412          208.0366    0.6432130      44588
## 844  45785   790.732          325.4259          190.6458    0.9024320      46247
## 845  43610   685.440          261.8911          183.3038    0.5393132      40221
## 846  62885   870.621          340.0262          221.7198    0.7646718      63439
## 847 139659  1566.072          472.5364          360.2534    0.7014185     138857
## 848  74888  1285.952          389.5112          257.0138    0.8292625      72948
## 849  21351   822.488          207.9946          140.6809    0.7465792      26320
## 850  33924   940.271          213.1808          204.7352    0.6167039      36540
## 851  70975   858.123          425.3772          218.5636    0.7773406      59560
## 852  71538  1257.632          410.2528          230.3950    0.8019596      70544
## 853  48027   677.878          273.6817          179.2179    0.6791535      47254
## 854 172427  1708.583          642.7954          379.4193    0.8132278     174655
## 855  34608   879.429          222.9284          192.1903    0.4717877      35011
## 856  36122   591.949          226.2835          204.6270    0.5347981      36730
## 857  38665   704.996          254.4247          164.4525    0.6789923      39055
## 858 154707  1552.174          547.6281          387.7554    0.7147431     154353
## 859 174904  1590.819          572.1430          384.4449    0.7641014     181383
## 860  57166   851.915          370.1194          167.7914    0.8363796      58986
## 861  68221   899.172          370.3504          224.9004    0.7578018      69668
## 862 167616  1405.871          615.1367          374.8135    0.7983951     172107
## 863  58165   848.586          390.4966          198.6992    0.8923081      46291
## 864  35382   601.575          263.6605          199.7235    0.7816589      34842
## 865 197649  1704.337          625.8028          386.4809    0.8021836     204252
## 866  45380   879.819          330.2440          181.3191    0.7329199      45897
## 867  55598  1157.204          365.5032          211.8789    0.9226426      54675
## 868 152314  1682.086          586.4924          317.9587    0.7564272     154137
## 869 233467  1907.805          666.5132          413.2578    0.7411758     239576
## 870  67378  1005.650          394.8397          201.7454    0.8277634      69451
## 871  57363  1268.727          387.8373          170.7662    0.8972867      58548
## 872  78053  1278.689          431.4128          249.6563    0.8155243      81765
## 873  69380  1103.357          438.1080          174.1614    0.8709735      55660
## 874  59793  1243.099          427.4641          205.7316    0.8535605      62029
## 875  39892   673.063          248.6961          194.2766    0.5149310      57142
## 876  38868   948.122          217.8836          192.8845    0.5665858      40261
## 877 153572  1711.681          513.3897          347.9288    0.6886587     155084
```

```
## 878 186988  1834.795        558.1505        379.0443    0.6929720     190651
## 879  39496   691.475        278.6621        213.0757    0.6936271      43755
## 880  61645   888.013        396.3153        222.3577    0.8051504      67004
## 881  44812   618.868        239.1603        200.9916    0.5830951      45988
## 882 207430  1742.909        689.8913        372.8643    0.8214148     207142
## 883 152636  1603.765        564.6882        359.9326    0.6992901     152456
## 884  53877   776.564        386.9977        154.7154    0.9008314      71779
## 885  27718   434.840        243.8112        138.3250    0.7562185      29367
## 886  33072   540.775        190.1888        202.2539    0.5257528      33112
## 887  70816  1232.805        435.3441        244.3642    0.8851088      71910
## 888  32008   522.199        262.9406        179.2175    0.7288438      32282
## 889 186854  1702.696        636.0983        378.0377    0.8215381     189077
## 890  61716  1245.013        436.0164        205.0324    0.8590949      59939
## 891  51988  1044.216        368.4772        227.7369    0.7975910      55236
## 892  36893   845.638        228.3865        184.7432    0.4566399      33489
## 893  87844  1358.893        438.5492        236.3283    0.7756735      94001
## 894  43880   905.325        286.5407        193.3580    0.7382430      43479
## 895  44374   974.651        292.9726        187.0697    0.7305259      44068
## 896  48584   972.861        314.6011        215.5551    0.7671780      47564
## 897  46355  1074.768        295.7148        185.7986    0.7419488      46269
## 898  39001   820.133        262.3379        147.8304    0.6930030      39414
## 899  76086  1299.813        436.0021        207.9042    0.8124714      73389
## 900  53811   696.989        348.5611        208.4517    0.8591610      53117
##        Extent    Class
## 1   0.7626678 DERMASON
## 2   0.6715971     CALI
## 3   0.7393049    HOROZ
## 4   0.7809190    SEKER
## 5   0.7881844    SEKER
## 6   0.6987021     SIRA
## 7   0.7908370    SEKER
## 8   0.7487831   BOMBAY
## 9   0.7779757    SEKER
## 10  0.7775096 DERMASON
## 11  0.6952869     SIRA
## 12  0.7050646    HOROZ
## 13  0.7912839   BOMBAY
## 14  0.7953518   BOMBAY
## 15  0.6924858     SIRA
## 16  0.6584441     SIRA
## 17  0.7360260   BOMBAY
## 18  0.8016399     SIRA
## 19  0.7621549 DERMASON
## 20  0.6863633    HOROZ
## 21  0.7923825    HOROZ
## 22  0.8238295 DERMASON
## 23  0.7376287    HOROZ
## 24  0.7733479    HOROZ
## 25  0.7207822     CALI
## 26  0.7375375 DERMASON
## 27  0.7174249     SIRA
## 28  0.8116939   BOMBAY
## 29  0.7817080   BOMBAY
## 30  0.7999544 DERMASON
```

```
## 31  0.6983322     SIRA
## 32  0.7627716    SEKER
## 33  0.7626833     CALI
## 34  0.7681155 DERMASON
## 35  0.7676833   BOMBAY
## 36  0.7693940     SIRA
## 37  0.7835592 DERMASON
## 38  0.7970984 DERMASON
## 39  0.7615178    SEKER
## 40  0.7784635   BOMBAY
## 41  0.7548790   BOMBAY
## 42  0.8125408    HOROZ
## 43  0.7845288    SEKER
## 44  0.7906900    SEKER
## 45  0.7675683   BOMBAY
## 46  0.7605066     CALI
## 47  0.7038208 DERMASON
## 48  0.7570183    SEKER
## 49  0.8119955     CALI
## 50  0.7584039    SEKER
## 51  0.7241787 DERMASON
## 52  0.7550807   BOMBAY
## 53  0.8178021     CALI
## 54  0.7902255     SIRA
## 55  0.7519916    SEKER
## 56  0.7248985 DERMASON
## 57  0.7862080    SEKER
## 58  0.7146833   BOMBAY
## 59  0.8071503    SEKER
## 60  0.6971772 DERMASON
## 61  0.7894085    SEKER
## 62  0.7811448    SEKER
## 63  0.7720119   BOMBAY
## 64  0.7971509   BOMBAY
## 65  0.7562718    SEKER
## 66  0.7260410     SIRA
## 67  0.7590435   BOMBAY
## 68  0.7619999 DERMASON
## 69  0.8108603     CALI
## 70  0.7910552     CALI
## 71  0.7738763    SEKER
## 72  0.7810013   BOMBAY
## 73  0.7489206     SIRA
## 74  0.7412329   BOMBAY
## 75  0.8095117   BOMBAY
## 76  0.7569039    SEKER
## 77  0.7370016    SEKER
## 78  0.7676727     CALI
## 79  0.7597530     SIRA
## 80  0.7045298   BOMBAY
## 81  0.7864186   BOMBAY
## 82  0.6504258    HOROZ
## 83  0.6583593    HOROZ
## 84  0.7712027    SEKER
```

```
## 85  0.8160674     CALI
## 86  0.6753528    HOROZ
## 87  0.7683713   BOMBAY
## 88  0.8441560   BOMBAY
## 89  0.7836683     SIRA
## 90  0.7156315 DERMASON
## 91  0.7695102 DERMASON
## 92  0.7722206    SEKER
## 93  0.6599033    HOROZ
## 94  0.7750645    SEKER
## 95  0.7089477     SIRA
## 96  0.7703075    SEKER
## 97  0.6914454     SIRA
## 98  0.8140619   BOMBAY
## 99  0.5969378    HOROZ
## 100 0.7637022    SEKER
## 101 0.7216649 DERMASON
## 102 0.8240002    HOROZ
## 103 0.7042509    HOROZ
## 104 0.7236726     CALI
## 105 0.8121461   BOMBAY
## 106 0.7479443     CALI
## 107 0.7719130     CALI
## 108 0.7488264   BOMBAY
## 109 0.8307897    HOROZ
## 110 0.6969944   BOMBAY
## 111 0.7892169     CALI
## 112 0.7503503 DERMASON
## 113 0.6323397    HOROZ
## 114 0.7880293   BOMBAY
## 115 0.7962432    SEKER
## 116 0.8313028   BOMBAY
## 117 0.8158818   BOMBAY
## 118 0.8016024   BOMBAY
## 119 0.7902436     CALI
## 120 0.7896131 DERMASON
## 121 0.7134387     CALI
## 122 0.7378845 DERMASON
## 123 0.7666192    SEKER
## 124 0.8159580 DERMASON
## 125 0.7479501     CALI
## 126 0.6238088    HOROZ
## 127 0.7804390    HOROZ
## 128 0.7843101 DERMASON
## 129 0.7216874 DERMASON
## 130 0.7042428     CALI
## 131 0.7696799     CALI
## 132 0.7623494   BOMBAY
## 133 0.7947153     CALI
## 134 0.8127932     CALI
## 135 0.7180023 DERMASON
## 136 0.7932044    HOROZ
## 137 0.7096670     CALI
## 138 0.8021698     CALI
```

```
## 139 0.7173294     CALI
## 140 0.6968480 DERMASON
## 141 0.8126799   BOMBAY
## 142 0.7712033     SIRA
## 143 0.8369892   BOMBAY
## 144 0.7484727     SIRA
## 145 0.7119445     SIRA
## 146 0.7357782 DERMASON
## 147 0.7657709     SIRA
## 148 0.7124782    HOROZ
## 149 0.7234933 DERMASON
## 150 0.7762851    HOROZ
## 151 0.7246796    HOROZ
## 152 0.7504802     SIRA
## 153 0.5808701    HOROZ
## 154 0.8010084     SIRA
## 155 0.6427744    HOROZ
## 156 0.7587290    SEKER
## 157 0.7307424 DERMASON
## 158 0.8064889     CALI
## 159 0.6893145    HOROZ
## 160 0.7789358     CALI
## 161 0.6804773     CALI
## 162 0.6917946    HOROZ
## 163 0.8016923   BOMBAY
## 164 0.7103128   BOMBAY
## 165 0.7036643     CALI
## 166 0.6655746     CALI
## 167 0.6418705    HOROZ
## 168 0.7423779 DERMASON
## 169 0.7110162 DERMASON
## 170 0.7801545    HOROZ
## 171 0.7439757    SEKER
## 172 0.7516647 DERMASON
## 173 0.7747102 DERMASON
## 174 0.7767901     CALI
## 175 0.5973746    HOROZ
## 176 0.7536285     CALI
## 177 0.8036333     CALI
## 178 0.7706925 DERMASON
## 179 0.7649835    SEKER
## 180 0.7739981    SEKER
## 181 0.7676373    HOROZ
## 182 0.7361076 DERMASON
## 183 0.7337567     CALI
## 184 0.7286337   BOMBAY
## 185 0.7867353   BOMBAY
## 186 0.8089174     SIRA
## 187 0.7769874    SEKER
## 188 0.7032127     SIRA
## 189 0.8191883     CALI
## 190 0.7468758    HOROZ
## 191 0.7864335 DERMASON
## 192 0.7303177 DERMASON
```

```
## 193 0.7175643   BOMBAY
## 194 0.7564883     SIRA
## 195 0.8139964     CALI
## 196 0.6889017     CALI
## 197 0.6868750     SIRA
## 198 0.7590548    SEKER
## 199 0.7660721    SEKER
## 200 0.7010359 DERMASON
## 201 0.8182326     CALI
## 202 0.8082006     SIRA
## 203 0.6859342     SIRA
## 204 0.7674373    SEKER
## 205 0.8110837     CALI
## 206 0.8341095   BOMBAY
## 207 0.7771815    SEKER
## 208 0.7194571     CALI
## 209 0.7228416     CALI
## 210 0.8156840   BOMBAY
## 211 0.6534593   BOMBAY
## 212 0.7951245   BOMBAY
## 213 0.8102407 DERMASON
## 214 0.8076371   BOMBAY
## 215 0.8067293   BOMBAY
## 216 0.7270416 DERMASON
## 217 0.8114644     CALI
## 218 0.7861964     CALI
## 219 0.7813898     SIRA
## 220 0.7086983 DERMASON
## 221 0.7898104 DERMASON
## 222 0.7082223     SIRA
## 223 0.6846161     CALI
## 224 0.7962781     CALI
## 225 0.8054553    SEKER
## 226 0.7932186    HOROZ
## 227 0.7270304   BOMBAY
## 228 0.8056415 DERMASON
## 229 0.7900489     SIRA
## 230 0.6360109    HOROZ
## 231 0.8156130 DERMASON
## 232 0.7010413    HOROZ
## 233 0.7281140     CALI
## 234 0.8389902    HOROZ
## 235 0.8029681   BOMBAY
## 236 0.7951137     CALI
## 237 0.7196649     SIRA
## 238 0.7247050 DERMASON
## 239 0.8082827     CALI
## 240 0.6735819    HOROZ
## 241 0.8160325 DERMASON
## 242 0.7308237     CALI
## 243 0.7349507 DERMASON
## 244 0.7735284 DERMASON
## 245 0.6460995     SIRA
## 246 0.7628195    SEKER
```

```
## 247 0.6768654 DERMASON
## 248 0.7448810    HOROZ
## 249 0.7333812 DERMASON
## 250 0.6724006    HOROZ
## 251 0.6069686    HOROZ
## 252 0.7863523    SEKER
## 253 0.7919081    HOROZ
## 254 0.7803460     SIRA
## 255 0.7274048    HOROZ
## 256 0.7509498    SEKER
## 257 0.8135391    HOROZ
## 258 0.7351328    HOROZ
## 259 0.7808447    SEKER
## 260 0.7510107     CALI
## 261 0.7731276    HOROZ
## 262 0.7907863     CALI
## 263 0.7890125     SIRA
## 264 0.7770345 DERMASON
## 265 0.7525944 DERMASON
## 266 0.7980575    SEKER
## 267 0.6992260     CALI
## 268 0.7783649 DERMASON
## 269 0.7279062 DERMASON
## 270 0.6572211     CALI
## 271 0.6093769    HOROZ
## 272 0.7914240 DERMASON
## 273 0.7833926    SEKER
## 274 0.7731125    HOROZ
## 275 0.6107728    HOROZ
## 276 0.6980403 DERMASON
## 277 0.6790006     CALI
## 278 0.6857992     SIRA
## 279 0.7141174   BOMBAY
## 280 0.7605231    SEKER
## 281 0.6813921    HOROZ
## 282 0.6513806    HOROZ
## 283 0.7076990 DERMASON
## 284 0.7690297   BOMBAY
## 285 0.8250161     SIRA
## 286 0.7349437     CALI
## 287 0.6839755     CALI
## 288 0.7677461    SEKER
## 289 0.6917009    HOROZ
## 290 0.7561608 DERMASON
## 291 0.5925747    HOROZ
## 292 0.7840755    SEKER
## 293 0.7378108    HOROZ
## 294 0.7330851 DERMASON
## 295 0.6048316    HOROZ
## 296 0.7596292     SIRA
## 297 0.8087077     CALI
## 298 0.7659472   BOMBAY
## 299 0.6218422    HOROZ
## 300 0.7889875     CALI
```

```
## 301 0.7622685    SEKER
## 302 0.7970768   BOMBAY
## 303 0.7052905     SIRA
## 304 0.7162567     SIRA
## 305 0.7904779    SEKER
## 306 0.7744755   BOMBAY
## 307 0.7937529    SEKER
## 308 0.7470968     CALI
## 309 0.6180229    HOROZ
## 310 0.6793780     SIRA
## 311 0.8190385     CALI
## 312 0.7961249    HOROZ
## 313 0.7820333 DERMASON
## 314 0.7767313     SIRA
## 315 0.7313089 DERMASON
## 316 0.6885833    HOROZ
## 317 0.8215599   BOMBAY
## 318 0.7463643   BOMBAY
## 319 0.7684894    SEKER
## 320 0.7648305   BOMBAY
## 321 0.7889284    SEKER
## 322 0.7925176    SEKER
## 323 0.6995823   BOMBAY
## 324 0.8232694   BOMBAY
## 325 0.7540324    HOROZ
## 326 0.7905286    SEKER
## 327 0.7920167    SEKER
## 328 0.8021569    HOROZ
## 329 0.7453401     SIRA
## 330 0.7778938    SEKER
## 331 0.7701740 DERMASON
## 332 0.7466156 DERMASON
## 333 0.6505196    HOROZ
## 334 0.7463492     SIRA
## 335 0.7669345     SIRA
## 336 0.7698863     CALI
## 337 0.8066012     CALI
## 338 0.7477320     SIRA
## 339 0.6117016    HOROZ
## 340 0.7618897   BOMBAY
## 341 0.7741042 DERMASON
## 342 0.8010633    SEKER
## 343 0.7864208   BOMBAY
## 344 0.7868532     SIRA
## 345 0.8101493   BOMBAY
## 346 0.8065163 DERMASON
## 347 0.7228709 DERMASON
## 348 0.8030519     SIRA
## 349 0.7736354 DERMASON
## 350 0.7872120 DERMASON
## 351 0.7254473    HOROZ
## 352 0.7353704    SEKER
## 353 0.7516491    HOROZ
## 354 0.8132239   BOMBAY
```

```
## 355 0.7907665    SEKER
## 356 0.6304017    HOROZ
## 357 0.7891989 DERMASON
## 358 0.8185895   BOMBAY
## 359 0.7152808     SIRA
## 360 0.6986253     SIRA
## 361 0.7610784     SIRA
## 362 0.6182928    HOROZ
## 363 0.7806135   BOMBAY
## 364 0.8043847   BOMBAY
## 365 0.8096028     CALI
## 366 0.7138184   BOMBAY
## 367 0.7772987    HOROZ
## 368 0.5983668    HOROZ
## 369 0.7793574 DERMASON
## 370 0.7433915    HOROZ
## 371 0.6412172    HOROZ
## 372 0.6783844   BOMBAY
## 373 0.7068981    HOROZ
## 374 0.7849725    SEKER
## 375 0.7829064    HOROZ
## 376 0.8398065   BOMBAY
## 377 0.8103320   BOMBAY
## 378 0.6484550    HOROZ
## 379 0.5961712    HOROZ
## 380 0.7368971    SEKER
## 381 0.6228698    HOROZ
## 382 0.7407868   BOMBAY
## 383 0.7630741    SEKER
## 384 0.6535307    HOROZ
## 385 0.7824555    SEKER
## 386 0.8019256     SIRA
## 387 0.7706427 DERMASON
## 388 0.8030944 DERMASON
## 389 0.7035563     SIRA
## 390 0.8158502    HOROZ
## 391 0.7805009    SEKER
## 392 0.8240037   BOMBAY
## 393 0.7643257    SEKER
## 394 0.6375332    HOROZ
## 395 0.7831243     SIRA
## 396 0.6609612     CALI
## 397 0.6747284   BOMBAY
## 398 0.7027607     SIRA
## 399 0.7737097    SEKER
## 400 0.7899987    HOROZ
## 401 0.7846588     SIRA
## 402 0.7529440    SEKER
## 403 0.6778832 DERMASON
## 404 0.6900196     SIRA
## 405 0.7374619    HOROZ
## 406 0.7851376 DERMASON
## 407 0.6009787    HOROZ
## 408 0.7818829     CALI
```

```
## 409 0.7973714    SEKER
## 410 0.7674651 DERMASON
## 411 0.7942664   BOMBAY
## 412 0.7350256   BOMBAY
## 413 0.6913661    HOROZ
## 414 0.8054384    HOROZ
## 415 0.6772264   BOMBAY
## 416 0.8418021     SIRA
## 417 0.7156377     CALI
## 418 0.7106411     CALI
## 419 0.7726311    SEKER
## 420 0.8093847     CALI
## 421 0.7492800    HOROZ
## 422 0.7778354    SEKER
## 423 0.7961561    SEKER
## 424 0.7035073 DERMASON
## 425 0.7750195     SIRA
## 426 0.6723698    HOROZ
## 427 0.7514445    SEKER
## 428 0.7856676     CALI
## 429 0.8009230     CALI
## 430 0.7555267     SIRA
## 431 0.7833974   BOMBAY
## 432 0.8071517     CALI
## 433 0.6171918    HOROZ
## 434 0.7815639    SEKER
## 435 0.7161249     SIRA
## 436 0.7946520    SEKER
## 437 0.8022894     CALI
## 438 0.7873483   BOMBAY
## 439 0.7427319     CALI
## 440 0.7237688    HOROZ
## 441 0.8243687 DERMASON
## 442 0.6969009    HOROZ
## 443 0.6827907    HOROZ
## 444 0.7809463     SIRA
## 445 0.6965066    HOROZ
## 446 0.7705595    HOROZ
## 447 0.7520558    SEKER
## 448 0.7075683     SIRA
## 449 0.7783436     CALI
## 450 0.6313805    HOROZ
## 451 0.7620809 DERMASON
## 452 0.8044099     SIRA
## 453 0.7493610 DERMASON
## 454 0.6907937    HOROZ
## 455 0.7856968   BOMBAY
## 456 0.7954196    HOROZ
## 457 0.7849796     SIRA
## 458 0.7343114 DERMASON
## 459 0.8000016    HOROZ
## 460 0.7560756    SEKER
## 461 0.8136920     SIRA
## 462 0.7987028    SEKER
```

```
## 463 0.6473352    HOROZ
## 464 0.7194494     CALI
## 465 0.8233714    HOROZ
## 466 0.7239728 DERMASON
## 467 0.8224070   BOMBAY
## 468 0.6945170 DERMASON
## 469 0.7354056     CALI
## 470 0.8184477     CALI
## 471 0.8172166     SIRA
## 472 0.7212412 DERMASON
## 473 0.7618069    SEKER
## 474 0.7947340 DERMASON
## 475 0.7757534    SEKER
## 476 0.7357930     SIRA
## 477 0.6720547     SIRA
## 478 0.6987418     SIRA
## 479 0.7656243   BOMBAY
## 480 0.7647697    HOROZ
## 481 0.7709921    SEKER
## 482 0.6851101   BOMBAY
## 483 0.7585037 DERMASON
## 484 0.7513033     SIRA
## 485 0.7759324   BOMBAY
## 486 0.7363333   BOMBAY
## 487 0.7748103    SEKER
## 488 0.7925152    SEKER
## 489 0.7659680     CALI
## 490 0.7567860     CALI
## 491 0.6982883     SIRA
## 492 0.5786897    HOROZ
## 493 0.7670723     CALI
## 494 0.7126907     SIRA
## 495 0.5710291    HOROZ
## 496 0.7239318 DERMASON
## 497 0.7144299 DERMASON
## 498 0.7928647     CALI
## 499 0.7521486     SIRA
## 500 0.7939166 DERMASON
## 501 0.7577378 DERMASON
## 502 0.6991446     SIRA
## 503 0.7384447     CALI
## 504 0.7554743    SEKER
## 505 0.6094275    HOROZ
## 506 0.7609622    SEKER
## 507 0.8117695     CALI
## 508 0.7887261     CALI
## 509 0.7815693     CALI
## 510 0.7948908    SEKER
## 511 0.7767959     CALI
## 512 0.7455373   BOMBAY
## 513 0.6823787    HOROZ
## 514 0.7233064 DERMASON
## 515 0.8378846   BOMBAY
## 516 0.8015815     CALI
```

```
## 517 0.7933400 DERMASON
## 518 0.7457623    SEKER
## 519 0.7519215 DERMASON
## 520 0.8045593     SIRA
## 521 0.7768516    SEKER
## 522 0.7280624     CALI
## 523 0.6957168    HOROZ
## 524 0.7836846    SEKER
## 525 0.6973924     SIRA
## 526 0.7990220    SEKER
## 527 0.7668461    SEKER
## 528 0.7326130 DERMASON
## 529 0.7073423   BOMBAY
## 530 0.6029540    HOROZ
## 531 0.8226515 DERMASON
## 532 0.7974156 DERMASON
## 533 0.6857548 DERMASON
## 534 0.7702421     CALI
## 535 0.7660720    SEKER
## 536 0.6932100     SIRA
## 537 0.6876567     SIRA
## 538 0.8176599     CALI
## 539 0.6862708     SIRA
## 540 0.7723317   BOMBAY
## 541 0.7710510    SEKER
## 542 0.7843468   BOMBAY
## 543 0.7749564     CALI
## 544 0.7290487 DERMASON
## 545 0.7931886    HOROZ
## 546 0.7647646   BOMBAY
## 547 0.8326913    HOROZ
## 548 0.7077624 DERMASON
## 549 0.6326190    HOROZ
## 550 0.7369715    SEKER
## 551 0.7343718    SEKER
## 552 0.7642550 DERMASON
## 553 0.8042387     CALI
## 554 0.7959198   BOMBAY
## 555 0.7256407   BOMBAY
## 556 0.7519933 DERMASON
## 557 0.7352110     CALI
## 558 0.7320713    SEKER
## 559 0.7818931 DERMASON
## 560 0.7855303 DERMASON
## 561 0.6515940    HOROZ
## 562 0.6797713 DERMASON
## 563 0.7823203   BOMBAY
## 564 0.8410152   BOMBAY
## 565 0.7064115    HOROZ
## 566 0.7445795 DERMASON
## 567 0.6387065    HOROZ
## 568 0.6898954 DERMASON
## 569 0.7630198     CALI
## 570 0.7878331   BOMBAY
```

```
## 571 0.7758667     CALI
## 572 0.6956607 DERMASON
## 573 0.7677723 DERMASON
## 574 0.7747559     SIRA
## 575 0.7271096    HOROZ
## 576 0.8183852   BOMBAY
## 577 0.8213525   BOMBAY
## 578 0.7717373    SEKER
## 579 0.8067362   BOMBAY
## 580 0.7949254    SEKER
## 581 0.7776719    SEKER
## 582 0.7631775   BOMBAY
## 583 0.7528551   BOMBAY
## 584 0.7635746    SEKER
## 585 0.7887606    SEKER
## 586 0.8002937     CALI
## 587 0.7565430    SEKER
## 588 0.7931218   BOMBAY
## 589 0.7404332   BOMBAY
## 590 0.6653568     SIRA
## 591 0.7825257     SIRA
## 592 0.7478045     SIRA
## 593 0.7812731    HOROZ
## 594 0.6167799    HOROZ
## 595 0.7123548 DERMASON
## 596 0.7988735     SIRA
## 597 0.7859764   BOMBAY
## 598 0.8285194   BOMBAY
## 599 0.6978238   BOMBAY
## 600 0.6944437    HOROZ
## 601 0.7035679    HOROZ
## 602 0.7991997 DERMASON
## 603 0.8306812    HOROZ
## 604 0.7175346     SIRA
## 605 0.7428195    SEKER
## 606 0.7920552 DERMASON
## 607 0.7444095     SIRA
## 608 0.7034520     SIRA
## 609 0.8076279    HOROZ
## 610 0.8193161 DERMASON
## 611 0.7974459   BOMBAY
## 612 0.7826155    SEKER
## 613 0.7920565   BOMBAY
## 614 0.7617679     CALI
## 615 0.7190484     CALI
## 616 0.7242936     SIRA
## 617 0.7311638 DERMASON
## 618 0.6177584    HOROZ
## 619 0.7870651    SEKER
## 620 0.7732697 DERMASON
## 621 0.6645936     SIRA
## 622 0.7246917     SIRA
## 623 0.6825464     CALI
## 624 0.7908711    SEKER
```

```
## 625 0.7157517     CALI
## 626 0.7227973     CALI
## 627 0.7452788    SEKER
## 628 0.6998281     SIRA
## 629 0.5758827    HOROZ
## 630 0.7559181     SIRA
## 631 0.8077730     CALI
## 632 0.7607671     CALI
## 633 0.7363657    HOROZ
## 634 0.7988602    SEKER
## 635 0.7740128    SEKER
## 636 0.7656046     SIRA
## 637 0.7505572     CALI
## 638 0.7328199    SEKER
## 639 0.8018125     CALI
## 640 0.7117215 DERMASON
## 641 0.7961695    HOROZ
## 642 0.7724295 DERMASON
## 643 0.7958668    SEKER
## 644 0.7691054     CALI
## 645 0.6527142    HOROZ
## 646 0.6342486    HOROZ
## 647 0.7625946 DERMASON
## 648 0.7538896    SEKER
## 649 0.7716027 DERMASON
## 650 0.7557395 DERMASON
## 651 0.8117218     CALI
## 652 0.7420929   BOMBAY
## 653 0.6976102    HOROZ
## 654 0.7565362    SEKER
## 655 0.6878258    SEKER
## 656 0.7587864    SEKER
## 657 0.7759201     SIRA
## 658 0.7062184 DERMASON
## 659 0.7721583    SEKER
## 660 0.7392702   BOMBAY
## 661 0.6926786 DERMASON
## 662 0.8205864     SIRA
## 663 0.7970703     CALI
## 664 0.8197454    HOROZ
## 665 0.7576134     CALI
## 666 0.7564581     SIRA
## 667 0.6017266    HOROZ
## 668 0.7926707    SEKER
## 669 0.7908468    SEKER
## 670 0.7245458     SIRA
## 671 0.7785599    SEKER
## 672 0.7946386    HOROZ
## 673 0.8113907     CALI
## 674 0.7091637     CALI
## 675 0.7337763     CALI
## 676 0.7903686     CALI
## 677 0.6607087     SIRA
## 678 0.7909196   BOMBAY
```

```
## 679 0.7854081 DERMASON
## 680 0.6691944     CALI
## 681 0.7524201     SIRA
## 682 0.7771689 DERMASON
## 683 0.7625548    HOROZ
## 684 0.7170931   BOMBAY
## 685 0.7558507    HOROZ
## 686 0.5978495    HOROZ
## 687 0.7038064     CALI
## 688 0.8096423     SIRA
## 689 0.7831414     SIRA
## 690 0.6916404 DERMASON
## 691 0.7918591     SIRA
## 692 0.7358505 DERMASON
## 693 0.7962923 DERMASON
## 694 0.6993522 DERMASON
## 695 0.7082029     SIRA
## 696 0.6261609    HOROZ
## 697 0.7428171     CALI
## 698 0.8134680     SIRA
## 699 0.6132323    HOROZ
## 700 0.7659691   BOMBAY
## 701 0.8186609     CALI
## 702 0.7940332     SIRA
## 703 0.7252246   BOMBAY
## 704 0.6622282     SIRA
## 705 0.8090501     SIRA
## 706 0.6945514     SIRA
## 707 0.7802536    SEKER
## 708 0.8184362    HOROZ
## 709 0.6543501     CALI
## 710 0.6970390 DERMASON
## 711 0.8206193     CALI
## 712 0.7012543 DERMASON
## 713 0.7715256    SEKER
## 714 0.7416261    HOROZ
## 715 0.7986902     SIRA
## 716 0.7996546 DERMASON
## 717 0.7798065    SEKER
## 718 0.7174799 DERMASON
## 719 0.8135847    SEKER
## 720 0.7864822   BOMBAY
## 721 0.7302867 DERMASON
## 722 0.7200162     SIRA
## 723 0.7740405     SIRA
## 724 0.6998903    HOROZ
## 725 0.6829633    HOROZ
## 726 0.6809763    HOROZ
## 727 0.8049308    SEKER
## 728 0.7323392    HOROZ
## 729 0.7398497     CALI
## 730 0.8130273     SIRA
## 731 0.7743606 DERMASON
## 732 0.7522528     CALI
```

```
## 733 0.7881431    SEKER
## 734 0.7775381    SEKER
## 735 0.7709856 DERMASON
## 736 0.8067621    HOROZ
## 737 0.7543859 DERMASON
## 738 0.7399135     CALI
## 739 0.8393583     SIRA
## 740 0.7660180   BOMBAY
## 741 0.7689113   BOMBAY
## 742 0.7382263     CALI
## 743 0.7945911     CALI
## 744 0.7006663    HOROZ
## 745 0.7520064     SIRA
## 746 0.7985590   BOMBAY
## 747 0.7797938    SEKER
## 748 0.7748991 DERMASON
## 749 0.7819773     CALI
## 750 0.8042442     CALI
## 751 0.6459626    HOROZ
## 752 0.7985951   BOMBAY
## 753 0.6849615   BOMBAY
## 754 0.8074902     CALI
## 755 0.7399024   BOMBAY
## 756 0.7703809     SIRA
## 757 0.7396643 DERMASON
## 758 0.6508655    HOROZ
## 759 0.7647805     SIRA
## 760 0.6921149 DERMASON
## 761 0.7745850 DERMASON
## 762 0.7992796   BOMBAY
## 763 0.6604659    HOROZ
## 764 0.7730008    SEKER
## 765 0.7234267     SIRA
## 766 0.7282119   BOMBAY
## 767 0.7543288    SEKER
## 768 0.6968291     CALI
## 769 0.8017797 DERMASON
## 770 0.7660310     CALI
## 771 0.6448851    HOROZ
## 772 0.8157466 DERMASON
## 773 0.7721090   BOMBAY
## 774 0.7801207   BOMBAY
## 775 0.7749604    SEKER
## 776 0.7840219    SEKER
## 777 0.7986217   BOMBAY
## 778 0.8004242   BOMBAY
## 779 0.6735039     CALI
## 780 0.7891145   BOMBAY
## 781 0.7535620     SIRA
## 782 0.7747245    SEKER
## 783 0.8076305   BOMBAY
## 784 0.7829625 DERMASON
## 785 0.7803817    HOROZ
## 786 0.7514597     SIRA
```

```
## 787 0.7652406    SEKER
## 788 0.7647849    SEKER
## 789 0.6350520    HOROZ
## 790 0.8156077   BOMBAY
## 791 0.7926280   BOMBAY
## 792 0.7941526 DERMASON
## 793 0.8336889    HOROZ
## 794 0.7459471 DERMASON
## 795 0.7147562    HOROZ
## 796 0.8027947   BOMBAY
## 797 0.6405466    HOROZ
## 798 0.7786109     CALI
## 799 0.7845072    HOROZ
## 800 0.7856539     CALI
## 801 0.8060158 DERMASON
## 802 0.7954033     SIRA
## 803 0.7018636    HOROZ
## 804 0.8131967   BOMBAY
## 805 0.8083792    SEKER
## 806 0.7871811   BOMBAY
## 807 0.7744360    SEKER
## 808 0.7947284   BOMBAY
## 809 0.7424461     CALI
## 810 0.7618930   BOMBAY
## 811 0.7719935    HOROZ
## 812 0.7882993 DERMASON
## 813 0.7394490    SEKER
## 814 0.7733685     SIRA
## 815 0.7794865   BOMBAY
## 816 0.7840863    SEKER
## 817 0.6433128    HOROZ
## 818 0.7855470     CALI
## 819 0.8092349     CALI
## 820 0.8268432     CALI
## 821 0.7702923    SEKER
## 822 0.6786725 DERMASON
## 823 0.7892384    SEKER
## 824 0.7935833 DERMASON
## 825 0.7240036     SIRA
## 826 0.7088811     SIRA
## 827 0.7799621    SEKER
## 828 0.8290353    HOROZ
## 829 0.7972845     CALI
## 830 0.7017097 DERMASON
## 831 0.7119470   BOMBAY
## 832 0.7503024 DERMASON
## 833 0.7663944     SIRA
## 834 0.7593362    SEKER
## 835 0.7871961   BOMBAY
## 836 0.8149563   BOMBAY
## 837 0.7519216    SEKER
## 838 0.6832278    HOROZ
## 839 0.7969707     SIRA
## 840 0.8171443   BOMBAY
```

```
## 841 0.7715747    SEKER
## 842 0.7722147     CALI
## 843 0.7787516    SEKER
## 844 0.6191178    HOROZ
## 845 0.8065545    SEKER
## 846 0.7943558     CALI
## 847 0.7867571   BOMBAY
## 848 0.8124195     CALI
## 849 0.7095908 DERMASON
## 850 0.7545598    SEKER
## 851 0.6812210     CALI
## 852 0.7593747     CALI
## 853 0.7504467     SIRA
## 854 0.6805151   BOMBAY
## 855 0.8092111    SEKER
## 856 0.7896887    SEKER
## 857 0.7704508 DERMASON
## 858 0.7734924   BOMBAY
## 859 0.7536495   BOMBAY
## 860 0.6053074    HOROZ
## 861 0.7433952     CALI
## 862 0.7979296   BOMBAY
## 863 0.6528386    HOROZ
## 864 0.6872498 DERMASON
## 865 0.8007517   BOMBAY
## 866 0.7991903     SIRA
## 867 0.7791845    HOROZ
## 868 0.6594873   BOMBAY
## 869 0.7898997   BOMBAY
## 870 0.7766162     CALI
## 871 0.6117144    HOROZ
## 872 0.7518556     CALI
## 873 0.6548229    HOROZ
## 874 0.8168870    HOROZ
## 875 0.7839855    SEKER
## 876 0.7525826    SEKER
## 877 0.7925105   BOMBAY
## 878 0.7636737   BOMBAY
## 879 0.7983970     SIRA
## 880 0.7193589     CALI
## 881 0.7487453    SEKER
## 882 0.7232606   BOMBAY
## 883 0.7530405   BOMBAY
## 884 0.6511909    HOROZ
## 885 0.7817553 DERMASON
## 886 0.7832928    SEKER
## 887 0.7831285    HOROZ
## 888 0.7632940 DERMASON
## 889 0.8179475   BOMBAY
## 890 0.5794764    HOROZ
## 891 0.6802410     SIRA
## 892 0.7903458    SEKER
## 893 0.7939372     CALI
## 894 0.7130574     SIRA
```

```
## 895 0.7056213     SIRA
## 896 0.7330393     SIRA
## 897 0.7072689     SIRA
## 898 0.8027022 DERMASON
## 899 0.8006726     CALI
## 900 0.8104071    HOROZ
```

From the above missing value check we can see all the 900 prediction were made properly for price and weight.

Step 7: Beginning QDA Analysis. Looking at QDA with the selected variables eccentricity, extent & area

```r
set.seed(1234)

# Fitting the QDA Model using Eccentricity, Extent & Area Predictors

QDAmodel = qda(Class~Eccentricity+Extent+Area, data=bean_data_updated_train)

# Creating a confusion Matrix
QDA_pred <-predict(QDAmodel,bean_data_updated_test)

#Labels for predictions'
QDA_predictions <- QDA_pred$class

true_values <- bean_data_updated_test$Class
#Confusion matrix'
table(QDA_predictions, true_values)
```

```
##                  true_values
## QDA_predictions BOMBAY CALI DERMASON HOROZ SEKER SIRA
##        BOMBAY      150    0        0     0     0    0
##        CALI          0  140        0     4     0    1
##        DERMASON      0    0      132     2     4    8
##        HOROZ         0    6        2   146     0    9
##        SEKER         0    0        6     0   140    6
##        SIRA          0    1       14    13     7  109
```

```r
# Calculating the accuracy
qdamodel_Accr <- mean(QDA_predictions ==true_values)

# Predicting Weights on test data

QDA_predicted_weight <- data.frame(weight.per.bean[QDA_pred$class])

QDA_Weight_diff <- abs(sum(true_bean_weight)-sum(QDA_predicted_weight))

# Predicting the price on test data and then calculating the Sum of square error on price

QDA_predicted_price <- data.frame(price.per.bean[QDA_pred$class])
QDA_Price_compare <- cbind(sum(QDA_predicted_price), sum(true_bean_price))


# Price Difference
QDA_price_diff <- abs( sum(true_bean_price)-sum(QDA_predicted_price))
```

```r
# Sum of the price square error or price variance
SSE_QDA = sum((true_bean_price-QDA_predicted_price)^2)

# Printing Results
cat("QDA Accuracy is:",qdamodel_Accr)
```

```
## QDA Accuracy is: 0.9077778
```

```r
cat(" , ")
```

```
##  ,
```

```r
cat("Sum of squared errors for price:", SSE_QDA )
```

```
## Sum of squared errors for price: 0.0006856773
```

The accuracy of the QDA model increase over LDA by 4%. The accuracy of QDA model is 91%. The sum of square error for price is less than LDA model5.

Step 8 KNN model

```r
set.seed(123)

#Scale the dataset
bean_data_updated_knn <- as.data.frame(scale(bean_data_updated[,- c(2,3,4,6,8,9)]))

#Set predictores dataset
bean_data_updated_knn$Class <- bean_data_updated[,8]

#Get rows dataset number
numrow = nrow(bean_data_updated_knn)

#Set training index datas et to get 70% of main data set
trn_ind = sample(1:numrow,size = as.integer(0.7*numrow))

#Set training data set
train_df  <- bean_data_updated_knn[trn_ind,]
'Set testing dataset'
```

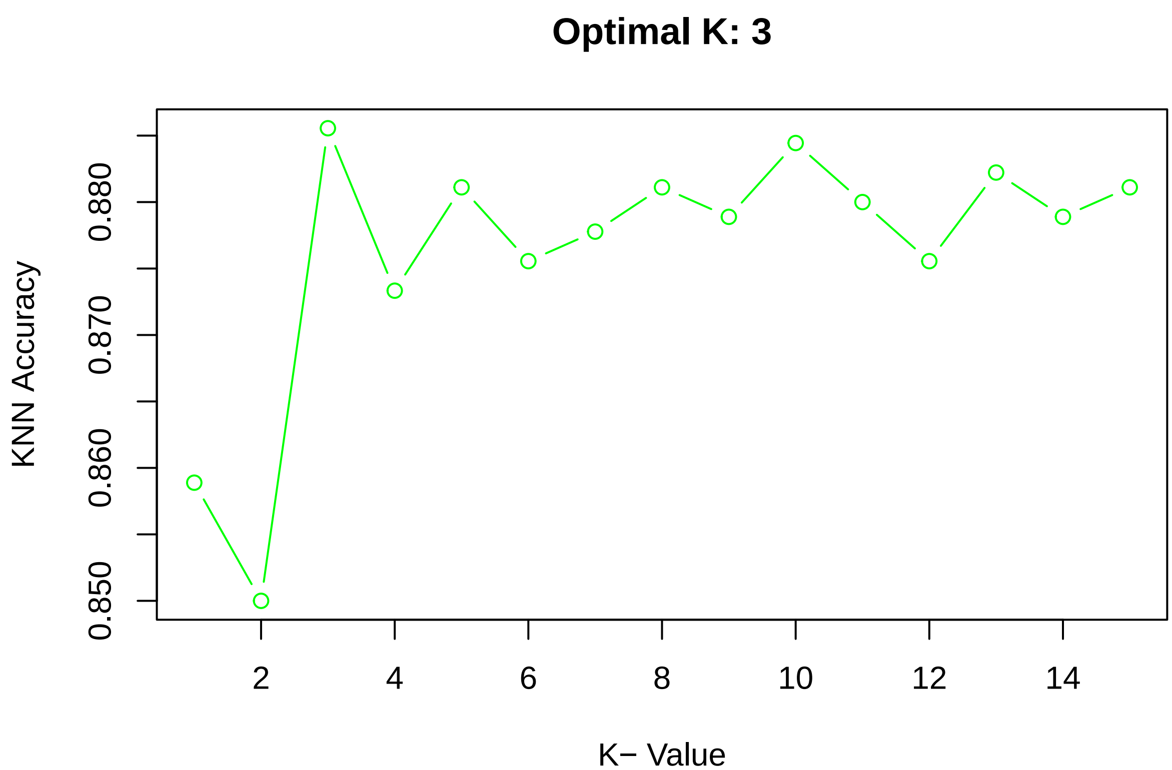
```
## [1] "Set testing dataset"
```

```r
test_df   <- bean_data_updated_knn[-trn_ind,]


#Set response variable values for training data set
train_labels <- train_df[,4]

#Set response variable values for testing data set
test_labels <- test_df[,4]

#Set Predictors dataset for training data
data_train <- train_df[,-4]

#Set Predictors data set for testing data
data_test <- test_df[,-4]
```

```r
#A varible to hold CV model accuraccies
var_acc_knn=NULL

for (i in 1:15){

  #Get model predictions'
  knn_preds <- knn( data_train,data_test, cl = train_labels, k= i)

   #Store accuracy results'
   var_acc_knn[i] <- mean(knn_preds == test_labels)

}

#Calculate the mean accuracy for CV
mean(var_acc_knn)
```

```
## [1] 0.8762963
```

```r
#Plotting the accuracy of the KNN

plot(var_acc_knn, type="b", xlab="K- Value",ylab="KNN Accuracy",col = "Green", main=paste("Optimal K:",
```



From the above plot, we can the maximum accuracy is achieved at K =3.

Step 8b: Creating the KNN model using K = 3.

```r
set.seed(12345)

knn_pred <- knn(data_train,data_test, cl = train_labels, k=3)
```

```r
# Confusion Matrix
 table(knn_pred , test_labels)
```

```
##           test_labels
## knn_pred   BOMBAY CALI DERMASON HOROZ SEKER SIRA
##   BOMBAY      156    0        0     0     0    0
##   CALI          0  135        0     7     0    1
##   DERMASON      0    0      116     0     3   11
##   HOROZ         0    6        1   142     0    9
##   SEKER         0    1       10     1   126    9
##   SIRA          0    3       28    11     7  117
```

```r
# Checking accuracy of kNN model

KNN_Accr <- mean(knn_pred==test_labels)

cat("Accuracy of kNN Model:", KNN_Accr)
```

```
## Accuracy of kNN Model: 0.88
```

```r
# KNN Actual price of each bean type in test data set

KNN_true_bean_price <- data.frame(price.per.bean[test_labels])
KNN_mean_true_bean_price <- mean(price.per.bean[test_labels])

# using the class variable from test data and applying the above calculation to get the true Weight.

KNN_true_bean_weight <-  data.frame(weight.per.bean[test_labels])


# Predicting Weights on test data

KNN_predicted_weight <- data.frame(weight.per.bean[knn_pred])

KNN_Weight_diff <- abs(sum(KNN_true_bean_weight)-sum(KNN_predicted_weight))

# Predicting the price on test data and then calculating the Sum of square error on price

KNN_predicted_price <- data.frame(price.per.bean[knn_pred])
KNN_Price_compare <- cbind(sum(KNN_predicted_price), sum(KNN_true_bean_price))
KNN_Price_compare
```

```
##          [,1]     [,2]
## [1,] 6.611074 6.547903
```

```r
# Price Difference

KNN_price_diff <- abs(sum(true_bean_price)-sum(KNN_predicted_price))
KNN_price_diff
```

```
## [1] 0.1975264
```

```r
# Sum of the price square error or price variance

SSE_KNN = sum((KNN_true_bean_price-KNN_predicted_price)^2)
```

```r
# Printing Results
cat("KNN Accuracy is:",KNN_Accr)
```

```
## KNN Accuracy is: 0.88
```

```r
cat(" , ")
```

```
##  ,
```

```r
cat("Sum of squared errors for price:", SSE_KNN )
```

```
## Sum of squared errors for price: 0.001010724
```

Step 9 : MCLUSTDA

```r
'Set seed'
```

```
## [1] "Set seed"
```

```r
set.seed(123)

#separate the predictors we will be using in the model
mclustTrain <- bean_data_updated_train[ , c("Area","Extent","Eccentricity")]
mclustTest <- bean_data_updated_test[ , c("Area","Extent","Eccentricity")]

#separate the class variable from the training data
mclustClass <- bean_data_updated_train[ , c("Class")]

# Fitting the MclustDA Model using Eccentricity, Extent & Area Predictors with G= 1
mclust.mod <- MclustDA(mclustTrain, mclustClass, G = 1)

#Print summary
summary(mclust.mod)
```

```
## ------------------------------------------------
## Gaussian finite mixture model for classification
## ------------------------------------------------
##
## MclustDA model summary:
##
##  log-likelihood    n df       BIC
##       -17645.72 2100 45 -35635.67
##
## Classes      n     % Model G
##   BOMBAY   350 16.67   XXX 1
##   CALI     353 16.81   XXI 1
##   DERMASON 346 16.48   XXI 1
##   HOROZ    335 15.95   XXI 1
##   SEKER    349 16.62   XXX 1
##   SIRA     367 17.48   XXX 1
##
## Training confusion matrix:
##           Predicted
## Class      BOMBAY CALI DERMASON HOROZ SEKER SIRA
##   BOMBAY      350    0        0     0     0    0
##   CALI          0  336        0    14     2    1
```

```
##   DERMASON      0    0      293     2    11   40
##   HOROZ         0   19        3   284     0   29
##   SEKER         0    1        9     0   315   24
##   SIRA          0    2       28    24    11  302
## Classification error = 0.1048
## Brier score          = 0.0749
```

```r
# Predictions on test data

MclustDA_pred <- predict.MclustDA(mclust.mod, newdata = mclustTest)
true_values <- bean_data_updated_test$Class

# Confusion Matrix of MclustDA model
table(MclustDA_pred$class, bean_data_updated_test$Class)
```

```
##
##            BOMBAY CALI DERMASON HOROZ SEKER SIRA
##   BOMBAY      150    0        0     0     0    0
##   CALI          0  141        0     6     0    0
##   DERMASON      0    0      130     2     4    8
##   HOROZ         0    5        1   143     0    9
##   SEKER         0    0        6     0   140    6
##   SIRA          0    1       17    14     7  110
```

```r
# Finding accuracy of MclustDA model

MclustDA_accr <- mean(MclustDA_pred$class==bean_data_updated_test$Class)
```

Note: we tried to implement Mclust with group =2,3,4 & 5, but due to hours of processing time we were not able to succeed.We only implemented Mclust with Group = 1.

Step 9 a: Cross validation of MClust model

```r
set.seed(1234)

#subsetting the bean data to use the 4 variables
bean.cv = bean_data_updated[ , c("Class", "Area","Extent","Eccentricity")]
bean.cv$Class = paste(bean_data_updated$Class)

#head(bean.cv)

#summary(bean.cv)

#Cross Validation
premut.cv = sample(1:3000, 3000)

folds = cbind(sort(rep(seq(1, 10, 1), 301))[1:3000], premut.cv)

accuracy = NULL

for (i in 1:10)
{
    index.i = folds[folds[,1]==i, 2]
    beanTest.i =  bean.cv[index.i,]
    beanTrain.i = bean.cv[ - index.i,]
```

```
    bean.dat = beanTrain.i[, -1]
    class.dat = beanTrain.i[, 1]
    mod.i = MclustDA(bean.dat, class.dat, G = 1)
    results.i = cbind(paste(predict.MclustDA(mod.i,
                      newdata = beanTest.i[, -1])$classification),
                      paste(beanTest.i[, 1]))
    accuracy=c(accuracy,mean(results.i[, 1] == results.i[, 2]))
}


#Overall accuracy of the model
cat('10 fold cross validation accuracy:',(mean(accuracy)*100))
```

```
## 10 fold cross validation accuracy: 89.77441
```

Step9 b : Predicting the price, Weight & Sum of Square Errors on Price using the MclustDA Model with Eccentricity, Extent & Area

```
# Predicting Weights on test data

MclustDA_predicted_weight <- data.frame(weight.per.bean[MclustDA_pred$class])

MclustDA_Weight_diff <- abs(sum(true_bean_weight)-sum(MclustDA_predicted_weight))

# Predicting the price on test data and then calculating the Sum of square error on price

MclustDA_predicted_price <- data.frame(price.per.bean[MclustDA_pred$class])
MclustDA_Price_compare <- cbind(sum(MclustDA_predicted_price), sum(true_bean_price))
MclustDA_Price_compare
```

```
##          [,1]     [,2]
## [1,] 6.457145 6.413547
```

```
# Price Difference

MclustDA_price_diff <- abs(sum(true_bean_price)-sum(MclustDA_predicted_price))
MclustDA_price_diff
```

```
## [1] 0.04359777
```

```
# Sum of the price square error or price variance

SSE_MclustDA = sum((true_bean_price - MclustDA_predicted_price)^2)

# Printing Results
cat("MclustDA Accuracy is:",MclustDA_accr)
```

```
## MclustDA Accuracy is: 0.9044444
```

```
cat(" , ")
```

```
##  ,
```

```
cat("Sum of squared errors for price:", SSE_MclustDA )
```

```
## Sum of squared errors for price: 0.0007343944
```

Step 10 a: Comparison of true predicted values of beans by each model

```
# True predicted values of beans by LDA
LDA_count <- rowSums(table(LDA_Predictions5, true_values))
# True predicted values of beans by QDA
QDA_count <- rowSums(table(QDA_predictions, true_values) )
# True predicted values of beans by KNN
Knn_count <- rowSums(table(knn_pred , test_labels))
# True predicted values of beans by Mclust
Mclust_count <- rowSums(table(MclustDA_pred$class, bean_data_updated_test$Class))

# Results Printing

LDA_count
```

```
##   BOMBAY     CALI DERMASON    HOROZ    SEKER     SIRA
##      149      142      142      162      144      161
```

```
QDA_count
```

```
##   BOMBAY     CALI DERMASON    HOROZ    SEKER     SIRA
##      150      145      146      163      152      144
```

```
Knn_count
```

```
##   BOMBAY     CALI DERMASON    HOROZ    SEKER     SIRA
##      156      143      130      158      147      166
```

```
Mclust_count
```

```
##   BOMBAY     CALI DERMASON    HOROZ    SEKER     SIRA
##      150      147      144      158      152      149
```

Step 10: Creating table for final prediction:

Bean test data results in the form of a table

```
# Creating a Table for the final results from all the models

Test_Model_Metrics <- data.frame(cbind(Models = c("LDA", "QDA", "kNN", "MClustDA"),
                                       Accuracy                 = round(c(LDA_mdl5_Accr, qdamodel_Accr
                                       True_Price               = round(c(sum(true_bean_price), sum(tru
                                       Predicted_Price          = round(c(sum(LDA_predicted_price5), su
                                       Sum_of_Squared_Errors    = round(c(SSE_LDA5, SSE_QDA, SSE_KNN, S

kable(Test_Model_Metrics , caption = "Model Results Analysis Table")
```

Table 2: Model Results Analysis Table

Models	Accuracy	True_Price	Predicted_Price	Sum_of_Squared_Errors
LDA	0.8767	6.4135	6.4326	0.00121
QDA	0.9078	6.4135	6.4347	0.00069
kNN	0.88	6.4135	6.6111	0.00101

Models	Accuracy	True_Price	Predicted_Price	Sum_of_Squared_Errors
MClustDA	0.9044	6.4135	6.4571	0.00073

According to the above models comparison results, QDA model has scored the lowest sum of squared cost value which selects it as the best model for this analysis.

Display Model Properties

```
QDAmodel
```

```
## Call:
## qda(Class ~ Eccentricity + Extent + Area, data = bean_data_updated_train)
##
## Prior probabilities of groups:
##    BOMBAY      CALI  DERMASON     HOROZ     SEKER      SIRA
## 0.1666667 0.1680952 0.1647619 0.1595238 0.1661905 0.1747619
##
## Group means:
##          Eccentricity    Extent      Area
## BOMBAY      0.7676137 0.7782907 174395.20
## CALI        0.8124715 0.7595017  76036.54
## DERMASON    0.7369795 0.7549897  31961.56
## HOROZ       0.8677900 0.7041005  53736.06
## SEKER       0.5864337 0.7704016  39665.08
## SIRA        0.7654264 0.7524302  44804.72
```

Prior probabilities of groups illustrates that the percentage of each beans type in the training data set is about the same with a small increase for SIRA bean type.

Calculating price error by each bean type using the selected QDA model

```
# Average Weight Per Bean
avg.weight.per.bean <- 0.7

# Price Per Bean
price.per.bean
```

```
##      BOMBAY        CALI    DERMASON       HOROZ       SEKER        SIRA
## 0.023534807 0.008095822 0.001222244 0.002785763 0.002938323 0.004523889
```

```
# price per bean error using the QDA model
# (Weight difference/ average weight per bean)* price per bean

# Bombay
Bombay.QDA.price.error   <- (QDA_Weight_diff/ avg.weight.per.bean) * 0.023534807
# CALI
CALI.QDA.price.error     <- (QDA_Weight_diff/ avg.weight.per.bean) * 0.008095822
# DERMASON
DERMASON.QDA.price.error <- (QDA_Weight_diff/ avg.weight.per.bean) * 0.001222244
# HOROZ
HOROZ.QDA.price.error    <- (QDA_Weight_diff/ avg.weight.per.bean) * 0.002785763
# SEKER
SEKER.QDA.price.error    <- (QDA_Weight_diff/ avg.weight.per.bean) * 0.002938323
```

```r
# SIRA
SIRA.QDA.price.error     <- (QDA_Weight_diff/ avg.weight.per.bean) * 0.004523889

price.error <- data.frame(cbind(Models = c("BOMBAY", "CALI", "DERMASON", "HOROZ", "SEKER", "SIRA"),
                                Price.error.by.bean.type  = round(c(Bombay.QDA.price.error,CALI.QI

kable(price.error , caption = "QDA Model Price Error By Each Bean Type")
```

Table 3: QDA Model Price Error By Each Bean Type

Models	Price.error.by.bean.type
BOMBAY	0.0057
CALI	0.002
DERMASON	3e-04
HOROZ	7e-04
SEKER	7e-04
SIRA	0.0011

Creating a Automated function to calculate the price with the selected model

```r
# Building a function with two parameters
auto_price_calc <- function(input_df, new_df){

  # Price per seed of six classes of beans
  price.per.bean<- c( "BOMBAY"=((5.56*1.92)/453.592),
                      "CALI"=((6.02*0.61)/453.592),
                      "DERMASON"=((1.98*0.28)/453.592),
                      "HOROZ"=((2.43*0.52)/453.592),
                      "SEKER"=((2.72*0.49)/453.592),
                      "SIRA"=((5.40*0.38)/453.592))

  # Weight per seed of six classes of beans
  weight.per.bean <- c("BOMBAY"=(1.92),
                       "CALI"=(0.61),
                       "DERMASON"=(0.28),
                       "HOROZ"=(0.52),
                       "SEKER"=(0.49),
                       "SIRA"=(0.38))

    set.seed(12345)


#Create a seq column
input_df$seq <- 1:nrow(input_df)

#Set the training data set
input_df_train <- input_df %>% dplyr::sample_frac(0.70)

#Building the model using training data set
qda_input_mdl <- qda(Class ~ Area +Eccentricity + Extent, data = input_df_train)

# Get predictions using the new input data
```

```
AUTO_QDA_pred <- predict(qda_input_mdl,new_df)

#Add predicted_price column to new input data based on class predicted values
new_df$predicted_price <- price.per.bean[AUTO_QDA_pred$class]

# Set subset data from new input data with two variables (Class,predicted_price)
sub_predicted <- new_df[,c("Class","predicted_price")]


# Calculate beans price based on each bean type
bean_price_group <- aggregate(sub_predicted$predicted_price,list(sub_predicted$Class),FUN=sum)

# Display beans price
bean_price_group

}

# Testing the function using the original beans data set split (70/30) between training and testing

# Set training data set as original data set
tr_df <- bean_data_updated[,-9]

# Set testing data as new input data
ts_df <- bean_data_updated_test[,-9]

# Calling the function with the two input parameters.
pred_price <- auto_price_calc(tr_df,ts_df)

# Rename columns
colnames(pred_price) <- c("Beans Type", "Total Predicted Price by Bean Type")

# Display predicted Total cost by bean type
pred_price
```

```
##   Beans Type Total Predicted Price by Bean Type
## 1     BOMBAY                          3.5302210
## 2       CALI                          1.1546535
## 3   DERMASON                          0.2529119
## 4      HOROZ                          0.5003598
## 5      SEKER                          0.4479215
## 6       SIRA                          0.5503779
```

The results shows the estimated price for each bean type for the new submitted data set.

Reference:

1. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). An Introduction to Statistical Learning with Applications in R (2nd ed., p. 612). Springer. https://hastie.su.domains/ISLR2/ISLRv2_website.pdf
2. Wickham, H., Chang, W., & Henry, L. (n.d.). A box and whiskers plot (in the style of Tukey). ggplot2.tidyverse. Retrieved 10 March 2023, from https://ggplot2.tidyverse.org/reference/geom_boxplot.html
3. Statistics Globe (n.d.). Draw multiple Boxplots in one graph.statisticsglobe .Retrieved 10 March 2023, from https://statisticsglobe.com/draw-multiple-boxplots-in-one-graph-in-r
4. Wei, T., & Simko, V. (2021, November 18). An Introduction to corrplot Package. Cran.R-Project. Retrieved March 11, 2023, from https://cran.r-project.org/web/packages/corrplot/vignettes/corrplot-

intro.html
5. Z. (2022, April 12). How to Split Data into Training & Test Sets in R. Statology. Retrieved March 11, 2023, from https://www.statology.org/train-test-split-r/
6. (n.d.). Associations between Variables. Codecademy. Retrieved March 11, 2023, from https://www.codecademy.com/learn/stats-associations-between-variables/modules/stats-associations-between-variables/cheatsheet
7. Z. (2020, October 30). Linear Discriminant Analysis in R (Step-by-Step). Statology. Retrieved March 11, 2023, from https://www.statology.org/linear-discriminant-analysis-in-r/
8. Sarkar, Priyankur. "What Is LDA: Linear Discriminant Analysis for Machine Learning." What Is Linear Discriminant Analysis (LDA)?, Knowledgehut, 27 Dec. 2022, https://www.knowledgehut.com/blog/data-science/linear-discriminant-analysis-for-machine-learning.
9. Z. (2020, November 2). Quadratic Discriminant Analysis in R (Step-by-Step). Statology. Retrieved March 12, 2023, from https://www.statology.org/quadratic-discriminant-analysis-in-r/
10. Saunders, C. (2023, February 10). Classification Part 2 LDA and QDA [Lecture]. D2l.Sdbor.edu. https://d2l.sdbor.edu/d2l/le/content/1781558/viewContent/11116132/View
11. D. (2020, June 22). K-NN Classifier in R Programming. Geeksforgeeks. Retrieved March 12, 2023, from https://www.geeksforgeeks.org/k-nn-classifier-in-r-programming/
12. "What Is the K-Nearest Neighbors Algorithm?" IBM, https://www.ibm.com/topics/knn.
13. Saunders, C. (2023, February 10). Chapter 2 Section 2 Part 2 [Lecture]. D2l.Sdbor.edu. https://d2l.sdbor.edu/d2l/le/content/1781558/viewContent/11116058/View
14. Fraley, C., Raftery, A. E., & Scrucca, L. (n.d.). MclustDA discriminant analysis. Mclust-Org.Github. Retrieved March 13, 2023, from https://mclust-org.github.io/mclust/reference/MclustDA.html
15. shanem@mtu.edu, Shane T. Mueller. Model-Based Clustering and Mclust, 28 Mar. 2021, https://pages.mtu.edu/~shanem/psy5220/daily/Day19/modelbasedclustering.html#content.
16. Saunders, C. (2023, February 24). MclustDA part1 [Lecture]. D2l.Sdbor.edu. https://d2l.sdbor.edu/d2l/le/content/1781558/viewContent/11116023/View
17. Saunders, C. (2023, February 24). MclustDA part2 [Lecture]. D2l.Sdbor.edu. https://d2l.sdbor.edu/d2l/le/content/1781558/viewContent/11116022/View
18. Saunders, C. (2023, February 24). MclustDA part3 Cross validation [Lecture]. D2l.Sdbor.edu. https://d2l.sdbor.edu/d2l/le/content/1781558/viewContent/11116024/View
19. Saunders, C. (2023, February 24). Mclust Play Part2 R file [Lecture]. D2l.Sdbor.edu. https://d2l.sdbor.edu/d2l/le/content/1781558/viewContent/11116026/View
20. Saunders, C. (2023, February 24). Mclust Play Part3 R file [Lecture]. D2l.Sdbor.edu. https://d2l.sdbor.edu/d2l/le/content/1781558/viewContent/11116025/View
21. The Carpentries (n.d.). Programming with R Creating Functions. Swcarpentry.Github. Retrieved March 14, 2023, from https://swcarpentry.github.io/r-novice-inflammation/02-func-R/
22. Chat.openai.com, https://chat.openai.com/.