Assignment #1: Statistical Graphics and Exploratory Data Analysis

Steven Futter: PRED 454-SEC 55

(1) Data Quality Check

The wine data set consists of 178 observations and 14 variables. Of the 14 variables there is one class variable and 13 continuous variables that describe the profile of the wine. In this assignment we are dealing with a classification problem since the goal of the assignment is to predict which class a wine falls into based upon the continuous variables, such as alcohol level, magnesium content, and color intensity.

Missing Data, Data Ranges and Distributions

Of the 178 observations the larger portion of wines belong to class 2 (40%), followed by class 1 (33%), and then class 3 (27%). The my.summary() function below confirms that there are no missing observations in the wine data. Based upon the mean and median values the proline variable appears to have a non-normal distribution sine mean and median values differ relative to other variables in the data set. The mean and median values of the proline variable are noticeable different. The mean value of proline is 746.89 whereas the median value is 673.50. Other variables in the data set appear to have extreme observations appearing as minimum and maximum values. There are no negative minimum values which makes sense and although there are outliers the values appear to be valid at first glance.

```
my.summary(wineData[,c(2:14)])
```

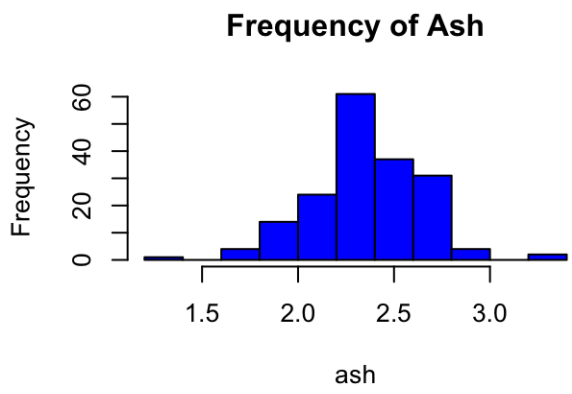
```
##           alcohol malicAcid  ash alcalinityOfAsh magnesium totalPhenols
## max         14.83      5.80 3.23           30.00    162.00         3.88
## min         11.03      0.74 1.36           10.60     70.00         0.98
## 0.01        11.44      0.90 1.70           11.35     78.00         1.14
## 0.05        11.66      1.06 1.92           14.77     80.85         1.38
## 0.25        12.36      1.60 2.21           17.20     88.00         1.74
## 0.5         13.05      1.87 2.36           19.50     98.00         2.36
## 0.75        13.68      3.08 2.56           21.50    107.00         2.80
## 0.95        14.22      4.46 2.74           25.00    124.30         3.27
## 0.99        14.47      5.54 2.99           28.50    141.76         3.60
## mean        13.00      2.34 2.37           19.49     99.74         2.30
## variance     0.66      1.25 0.08           11.15    203.99         0.39
## % missing    0.00      0.00 0.00            0.00      0.00         0.00
##           flavanoids nonflavanoidPhenols proanthocyanins colorIntensity
## max             5.08                0.66            3.58          13.00
## min             0.34                0.13            0.41           1.28
## 0.01            0.47                0.14            0.42           1.86
## 0.05            0.55                0.19            0.73           2.11
## 0.25            1.20                0.27            1.25           3.22
## 0.5             2.13                0.34            1.56           4.69
## 0.75            2.88                0.44            1.95           6.20
## 0.95            3.50                0.60            2.71           9.60
## 0.99            3.79                0.63            3.03          11.02
## mean            2.03                0.36            1.59           5.06
## variance        1.00                0.02            0.33           5.37
## % missing       0.00                0.00            0.00           0.00
##            hue OD280_OD315OfDilutedWines  proline
## max       1.71                      4.00  1680.00
## min       0.48                      1.27   278.00
## 0.01      0.55                      1.29   306.94
## 0.05      0.57                      1.46   354.55
## 0.25      0.78                      1.94   500.50
## 0.5       0.96                      2.78   673.50
## 0.75      1.12                      3.17   985.00
## 0.95      1.28                      3.58  1297.25
## 0.99      1.43                      3.84  1522.36
## mean      0.96                      2.61   746.89
## variance  0.05                      0.50 99166.72
## % missing 0.00                      0.00     0.00
```

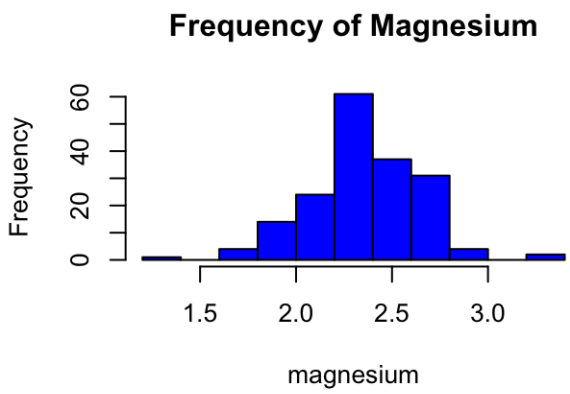
By plotting a matrix of histograms we can visually identify clear outliers in the data set that are described above by the my.summary() function. Outliers can be seen in the following variables: ash, magnesium, flavanoids, hue.

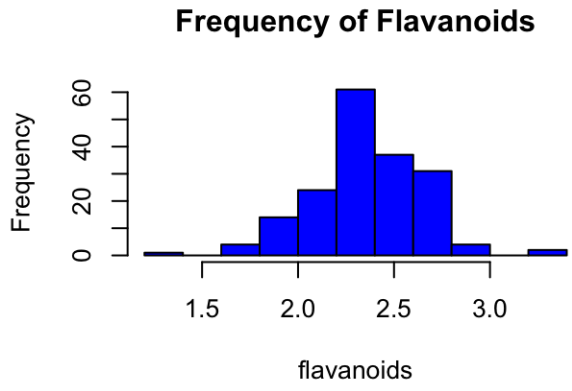
Outliers: i. left tail of ash ii. right tail of ash iii. right tail of magnesium iv. right tail of flavanoids v. right tail of hue

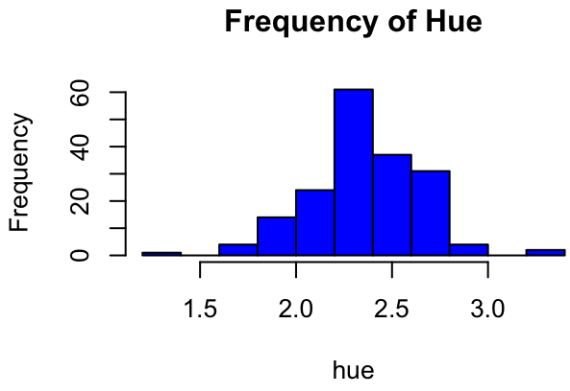
Outliers

```
par(mfrow=c(2,2))
hist(ash,col="blue",main='Frequency of Ash', xlab='ash')
hist(ash,col="blue",main='Frequency of Magnesium', xlab='magnesium')
hist(ash,col="blue",main='Frequency of Flavanoids', xlab='flavanoids')
hist(ash,col="blue",main='Frequency of Hue', xlab='hue')
```





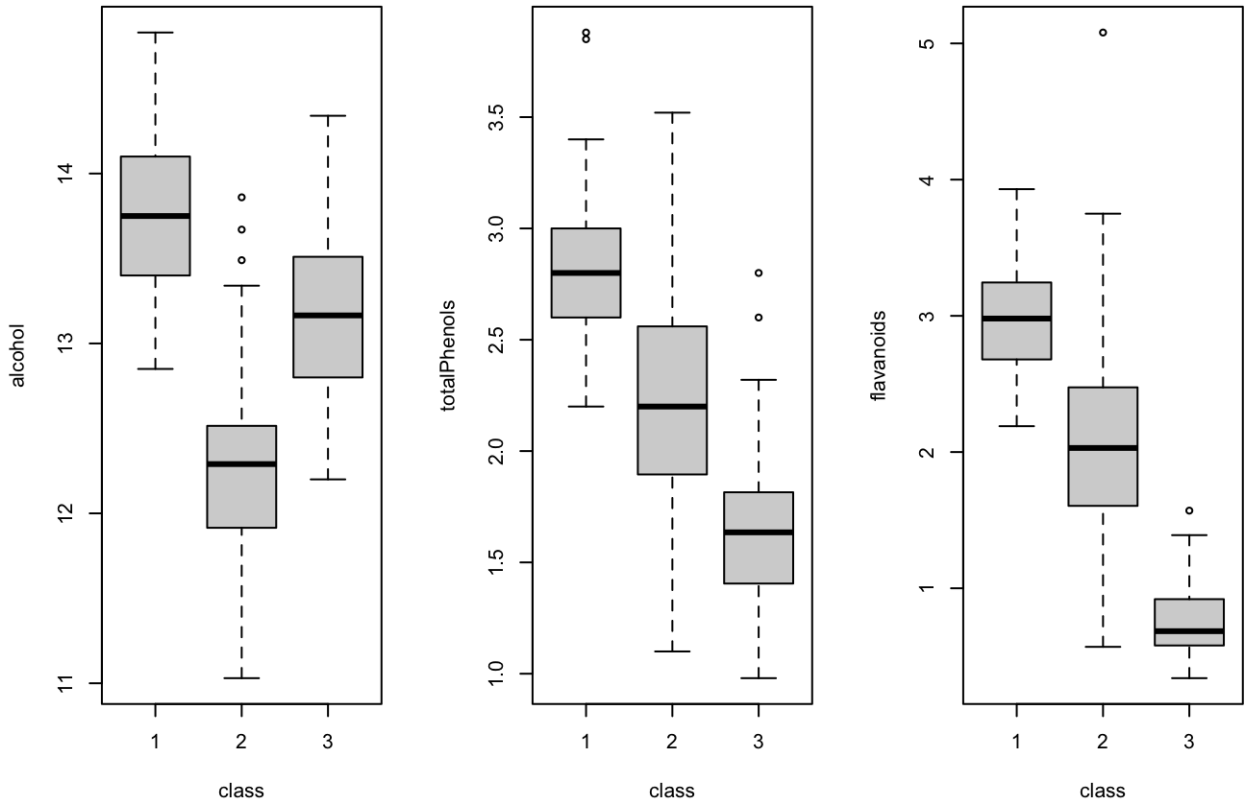




(2) Exploratory data analysis

With the wine data set we have a classification problem which can benefit from viewing the data distributions with boxplots. From the boxplots below we can see that alcohol, totalPhenols, and flavanoids appear to be particularly predictive with the middle 75% (grey shaded bar) of values separating the three wine classes with minimal overlap. This looks promising since the wine classes are differentiated quite nicely by these three variables alone.

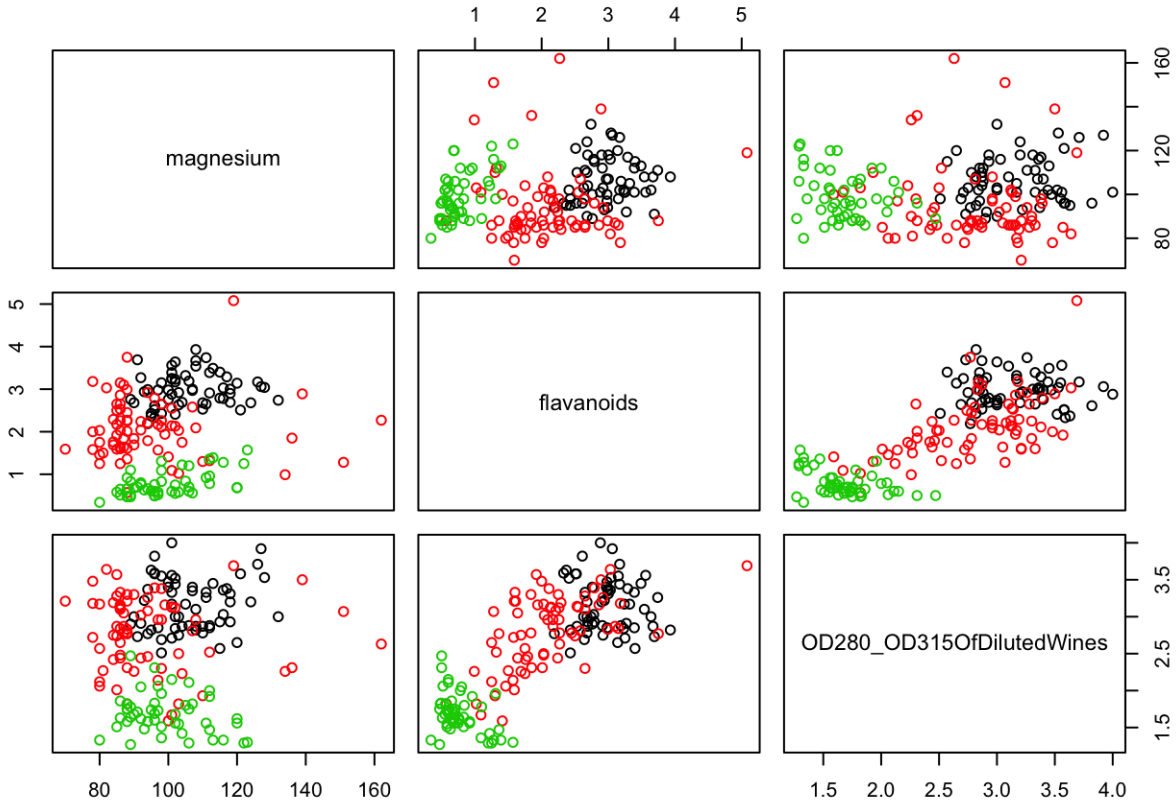
```
par(mfrow=c(1,3))
boxplot(alcohol~class, data=wineData, col='lightgray',xlab="class", ylab='alcohol')
boxplot(totalPhenols~class, data=wineData, col='lightgray',xlab="class", ylab='tota
lPhenols')
boxplot(flavanoids~class, data=wineData, col='lightgray',xlab="class", ylab='flavan
oids')
```



Several of the variables appear to be good candidates for a descriminative classifier such as a Support Vector Machine (SVM). An SVM uses an algorithm to split the data classes into optimal hyperplanes. Below I show that various pairs of variables cluster the wine classes into more or less separate wine classes. From initial EDA it appears that a clustered algorithm approach to modeling may also have a low misclassification error percentage.

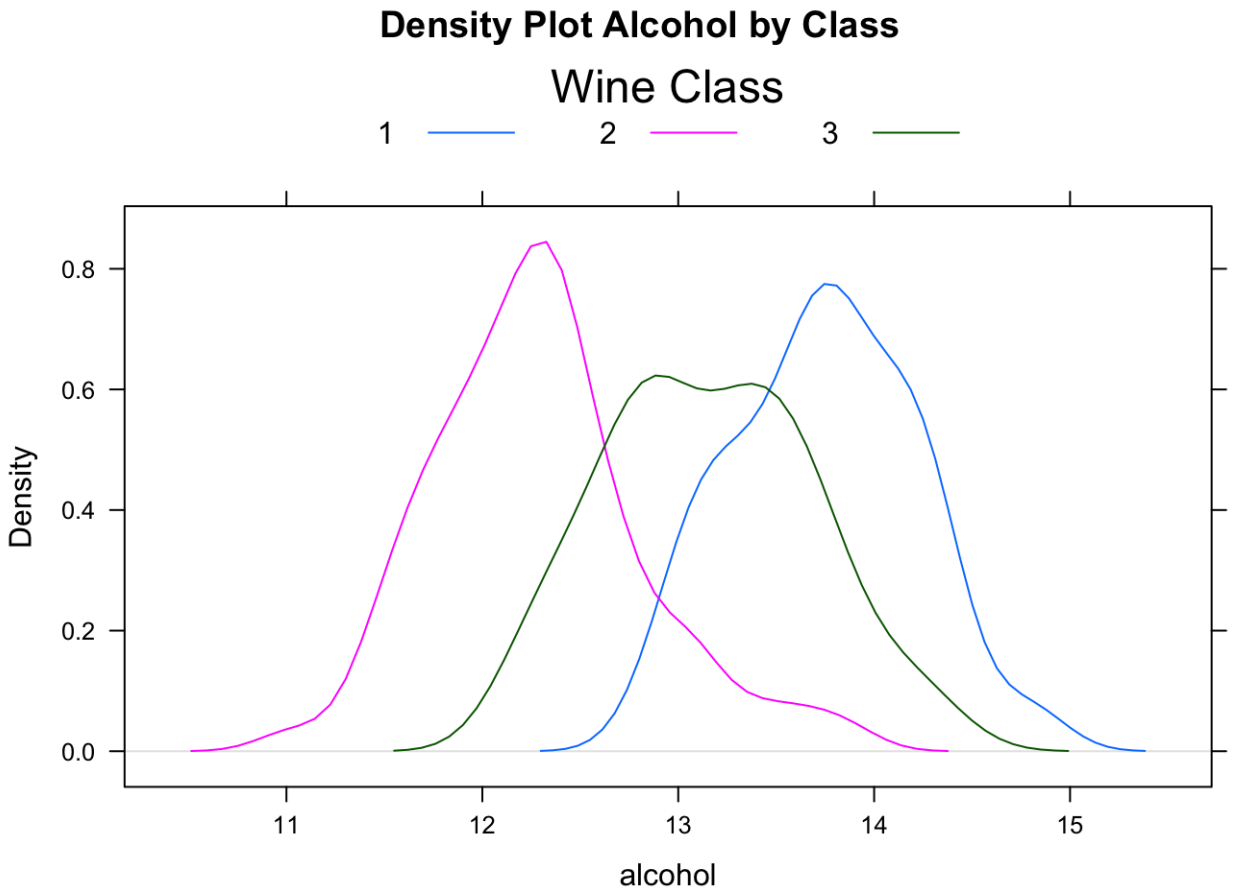
Below we can see how magnesium, flavanoids, and OD280_OD315OfDilutedWines pairs segregate the three wine classes into separate clusters.

```
pairs(wineData[,c('magnesium','flavanoids','OD280_OD315OfDilutedWines')], col=wineD
ata$class)
```



Using the Lattice library we can see the density function of each wine class against alcohol. We can see that the peak densities of each wine class vary as alcohol concentration increases. Generally speaking we can see that class 2 wines have the least alcohol content and class 1 wines have the most.

```
library('lattice')
densityplot(~ alcohol, data = wineData, groups = class,
plot.points = FALSE, ref = TRUE,
auto.key = list(columns = 3, title='Wine Class'), main='Density Plot Alcohol by Cla
ss')
```



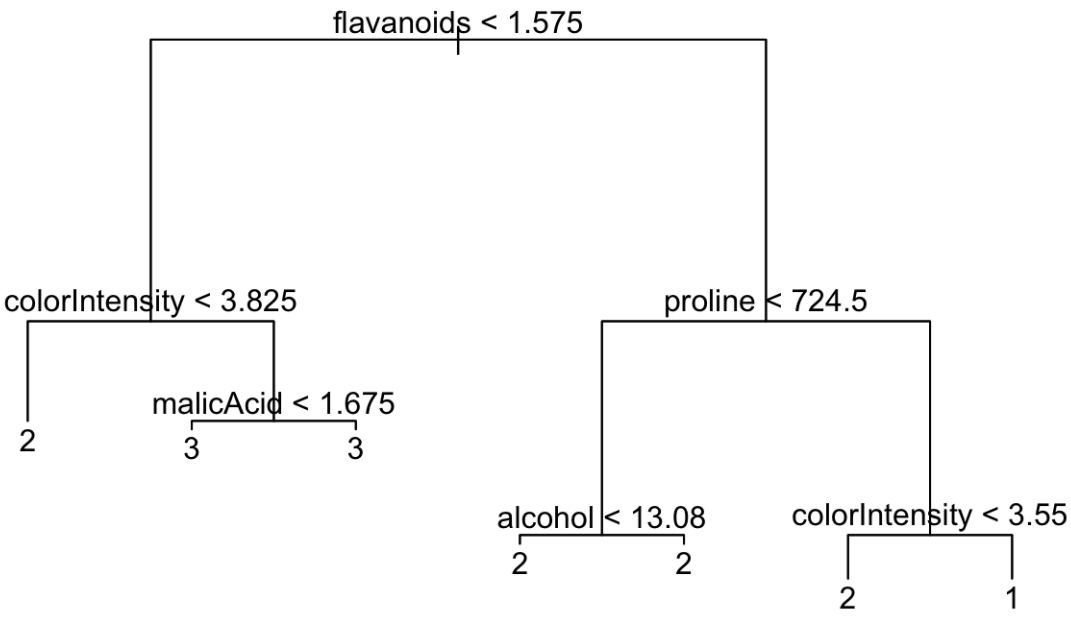
I now use a tree model to explore the data further and evaluate missclassification error.

Tree Model

```
summary(tree.wineData)
```

```
##
## Classification tree:
## tree(formula = class ~ ., data = wineData)
## Variables actually used in tree construction:
## [1] "flavanoids"     "colorIntensity" "malicAcid"      "proline"
## [5] "alcohol"
## Number of terminal nodes:  7
## Residual mean deviance:  0.08779 = 15.01 / 171
## Misclassification error rate: 0.01685 = 3 / 178
```

```
plot(tree.wineData)
text(tree.wineData, pretty=0) # terminal nodes are 1,2,and 3.
```



The tree model above has a misclassification of 1.7%.

(3) Model Based Exploratory Data Analysis

In this next section I fit some naive models to help understand the relationships in the data. As can be seen from the tree model above the data can be segregated quite accurately using a tree model. The tree model above has a misclassification error of 1.7% by evaluating the values of just five (flavanoids, colorIntensity, malicAcid, proline, and alcohol) of the 13 variables. Notably the flavanoids variable is the root of the tree with a split value of 1.575.

Discussion of Other Modeling Approaches

As mentioned in the EDA an SVM model should perform well on the wineData.

```
##
## Call:
## svm.default(x = x, y = y, data = wineData)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  1
##       gamma:  0.07692308
##
## Number of Support Vectors:  69
##
##  ( 19 31 19 )
##
##
## Number of Classes:  3
##
## Levels:
##  1 2 3
```

See the confusion matrix result of prediction:

```
table(pred,y)
```

```
##     y
## pred  1  2  3
##    1 59  0  0
##    2  0 71  0
##    3  0  0 48
```

Looking at the results from the confusion matrix for the SVM model every observation has been correctly classified. Further work is needed to ensure that these results can be relied upon and that the SVM model is not overfitting to the data. However, these results are very pleasing. For further investigation I can review other clustered models such as k-nearest neighbor. Although k-nearest neighbor is an unsupervised learning model it could still be used to evaluate the expected class output in the wine data set.

Appendix

Below I present relevant R code related to the results and graphics above.

```
# Read in data into data.frame. check first 5 rows.
library(data.table)
wineData <- fread('http://archive.ics.uci.edu/ml/machine-learning-databases/wine/wine.data')
colnames(wineData) = c('class','alcohol','malicAcid','ash','alcalinityOfAsh','magnesium','totalPhenols',
                       'flavanoids','nonflavanoidPhenols','proanthocyanins','colorIntensity',
                      'hue','OD280_OD315OfDilutedWines','proline')

head(wineData,5)
dim(wineData)      # 178 row x 14 columns (13 continuous variables and 1 class variable)
```

```r
str(wineData)     # evaluate the data type of each column


# update class of data.table to data.frame
class(wineData)
wineData = data.frame(wineData)

# convert class to a factor variable
wineData$class = factor(wineData$class)

# remove need for $ expansion
attach(wineData)



#### (1) Data Quality Check
# DATA QUALITY SECTION
# summary function:
library(plyr)

maxfun <- function(x){round(max(x,na.rm = TRUE),2)}
minfun <- function(x){round(min(x,na.rm = TRUE),2)}
quantilefun <- function(x){round(quantile(x,c(0.01,0.05,0.25,0.5,0.75,0.95,0.99)),2
)}
meanfun <- function(x){round(mean(x),2)}
varfun <- function(x){round(var(x),2)}
percentMissingfun <- function(x){round(100*(sum(is.na(x))/length(x)),2)}

my.summary = function(df){
  #create rows of summarized values using colwise function from plyr package
  max = colwise(maxfun)(df)
  min = colwise(minfun)(df)
  quantile = colwise(quantilefun)(df)
  mean = colwise(meanfun)(df)
  variance = colwise(varfun)(df)
  percentMissing = colwise(percentMissingfun)(df)

  # bind data.frame and rename rows
  mydf = rbind(max, min, quantile, mean, variance, percentMissing)
  rownames(mydf) = c('max','min',0.01,0.05,0.25,0.5,0.75,0.95,0.99,'mean','variance
','% missing')
  return(mydf)


# Loop through all variables to produce a matrix of histograms to identify outliers

par(mfrow=c(2,2))
for (i in 2:14){
  hist(wineData[[i]],col="blue",main='Histogram', xlab=names(wineData)[i])
}

# Loop through all variables to create boxplots of each variable against the wine c
lass.
par(mfrow=c(1,3))
for (i in 2:14){
  boxplot(wineData[[i]]~class, data=wineData, col='lightgray',xlab="class", ylab=n
ames(wineData)[i])
```

```
}

# Loop through all variables to produce scatterplots of each variable with class as
color variable
for (r in 14:14){     # change 14:14 to 2:2, 3:3, for each non-class variable
  for (i in 2:14){    # loop through each non-class variable
    pairs(wineData[,c(r,i)], col=wineData$class)
  }
}


#### (2) Exploratory data analysis
require(tree)
tree.wineData=tree(class~., data=wineData)
summary(tree.wineData)
plot(tree.wineData)
text(tree.wineData, pretty=0) # terminal nodes are 1,2,and 3.


#### (3) Model Based Exploratory Data Analysis
# SVM MODEL
# Divide wineData to x (the variables) and y the classes
x = subset(wineData, select=-class)
y = class

# Create SVM Model and show summary
library(e1071)
svmModel = svm(x,y, data=wineData)
summary(svmModel)

# Run the prediction:
pred = predict(svmModel, x)

#See the confusion matrix result of prediction:
table(pred,y)
```