Discriminant Analysis

Daniel Kim

2/27/2020

```
library(MASS)
library(heplots)
```

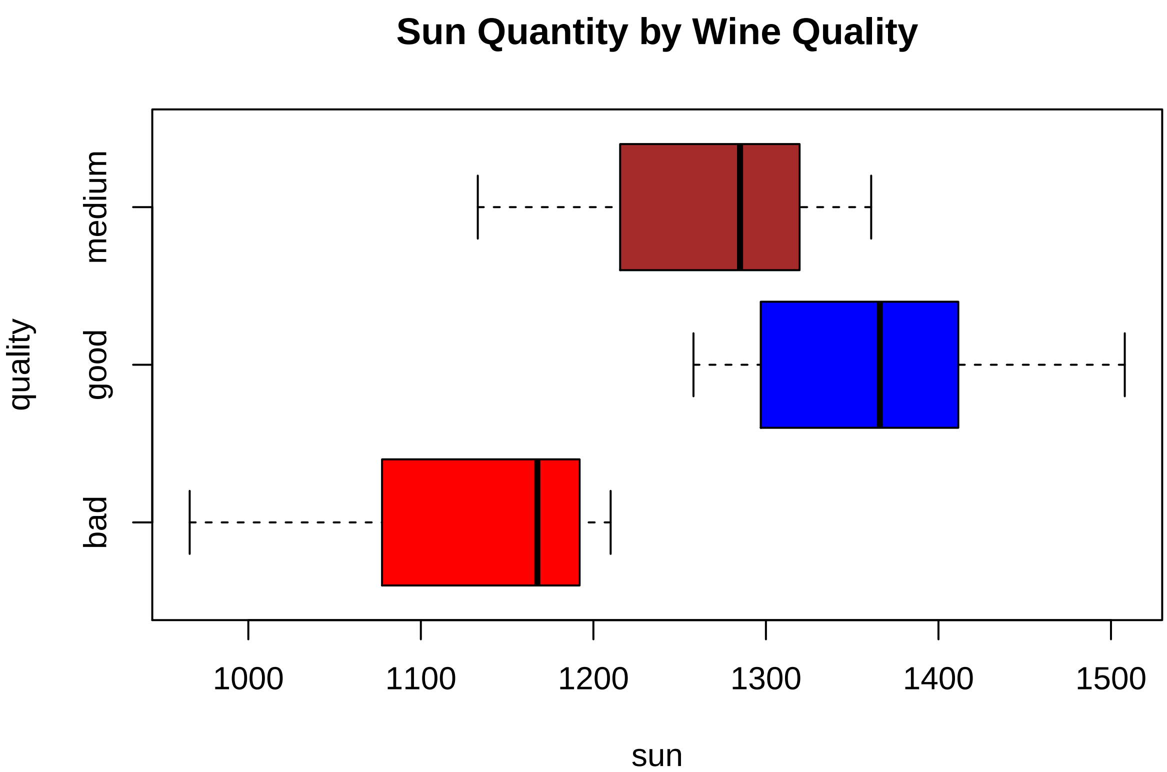
```
## Loading required package: car
```

```
## Loading required package: carData
```

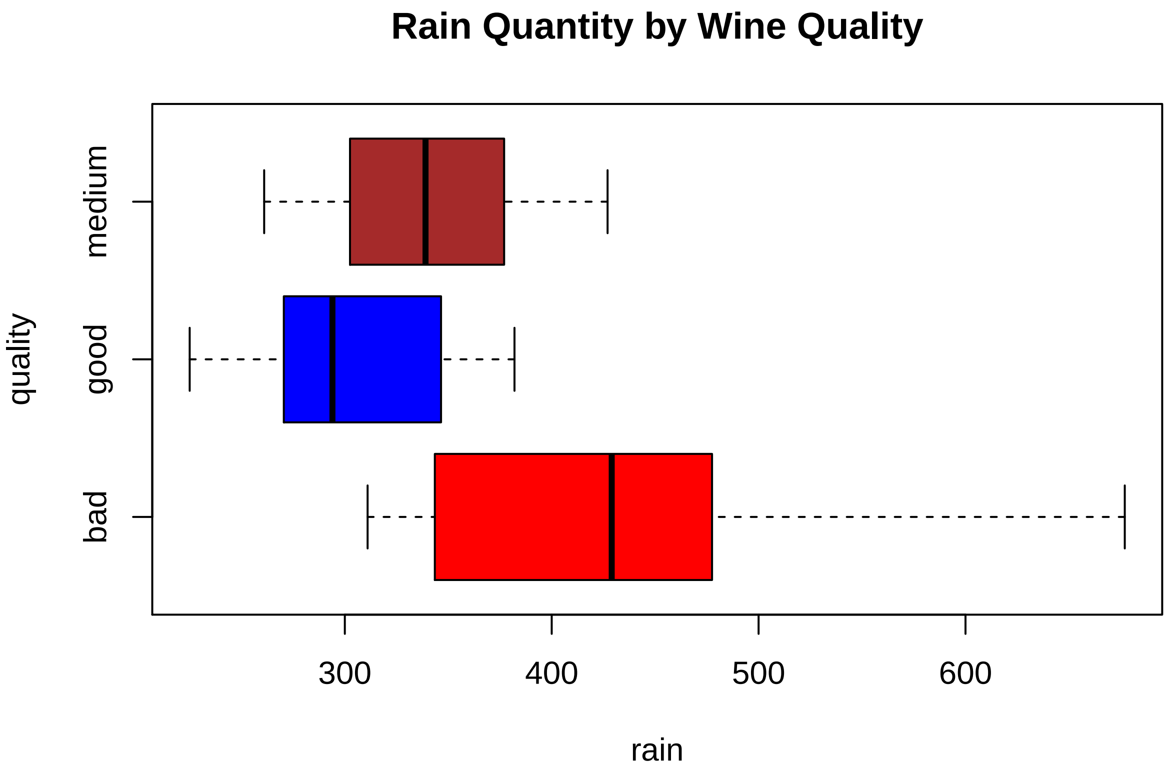
```
library(DiscriMiner)
library(klaR)
```

1) Let's look at boxplots for both sun and rain quantity between wine quality groups to see if there appears to be differences.

```
boxplot(sun~ quality, data=bordeaux, col = c("red", "blue", "brown"), horizontal = T, main = "Sun Quant
```

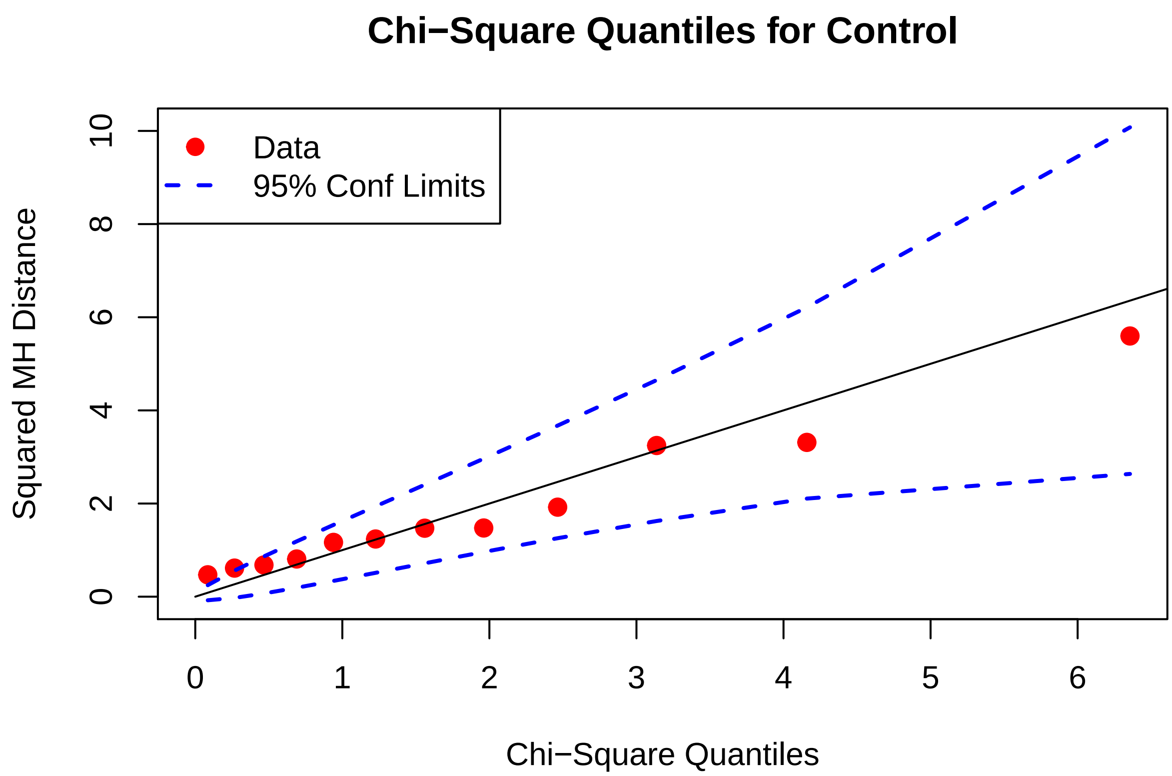


```r
boxplot(rain ~ quality, data=bordeaux, col = c("red", "blue", "brown"), horizontal = T, main = "Rain Qua
```

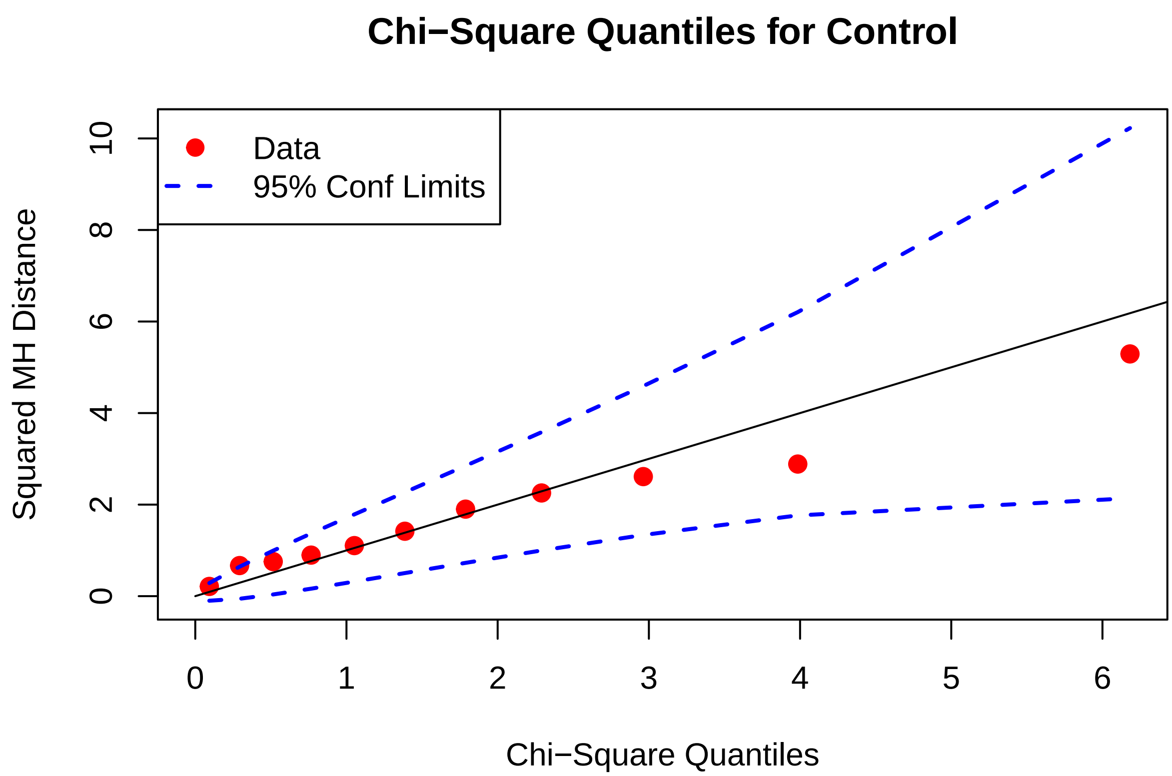


There in fact appears to be differences between groups. Now let's look at CSQ plots for each group to see if multivariate normality within each group holds

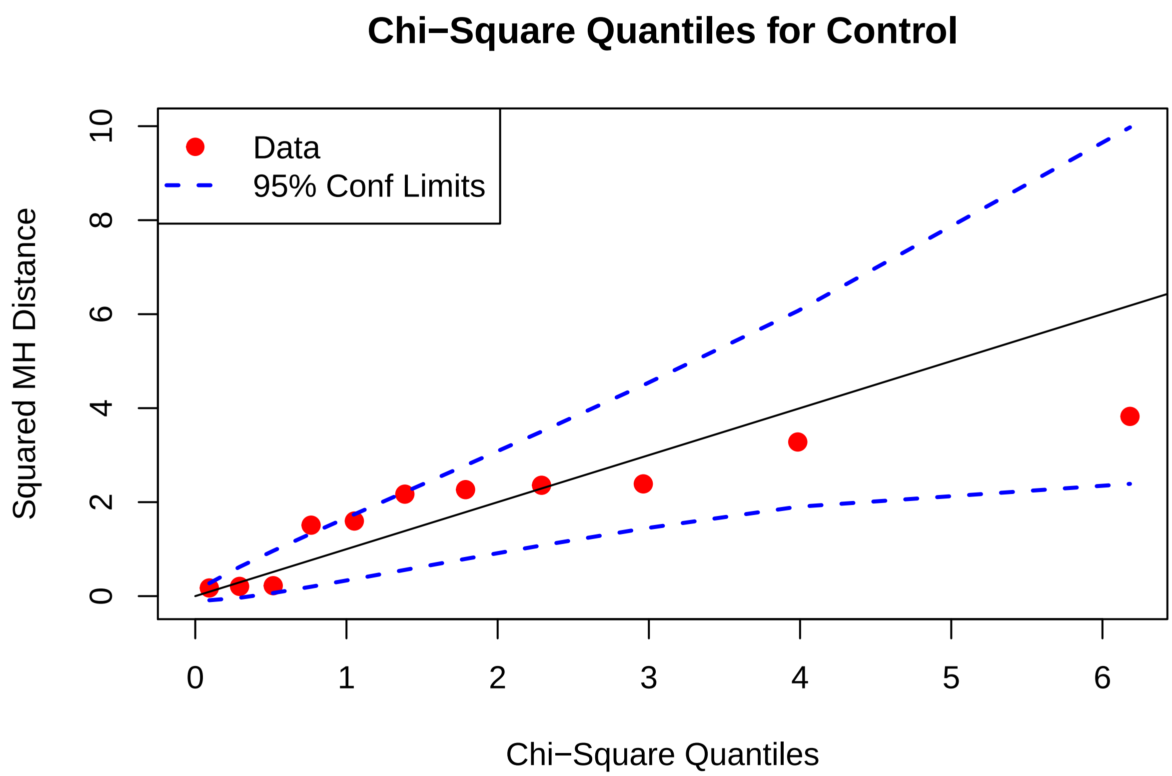
```r
#see if data is multivariate normal in EACH GROUP
#get online function
source("http://www.reuningscherer.net/STAT660/R/CSQPlot.r.txt")

#examine multivariate normality within each belly group
CSQPlot(bordeaux[bordeaux$quality == "bad", c("sun","rain")], label = "Control")
```




```r
CSQPlot(bordeaux[bordeaux$quality == "medium", c("sun","rain")], label = "Control")
```



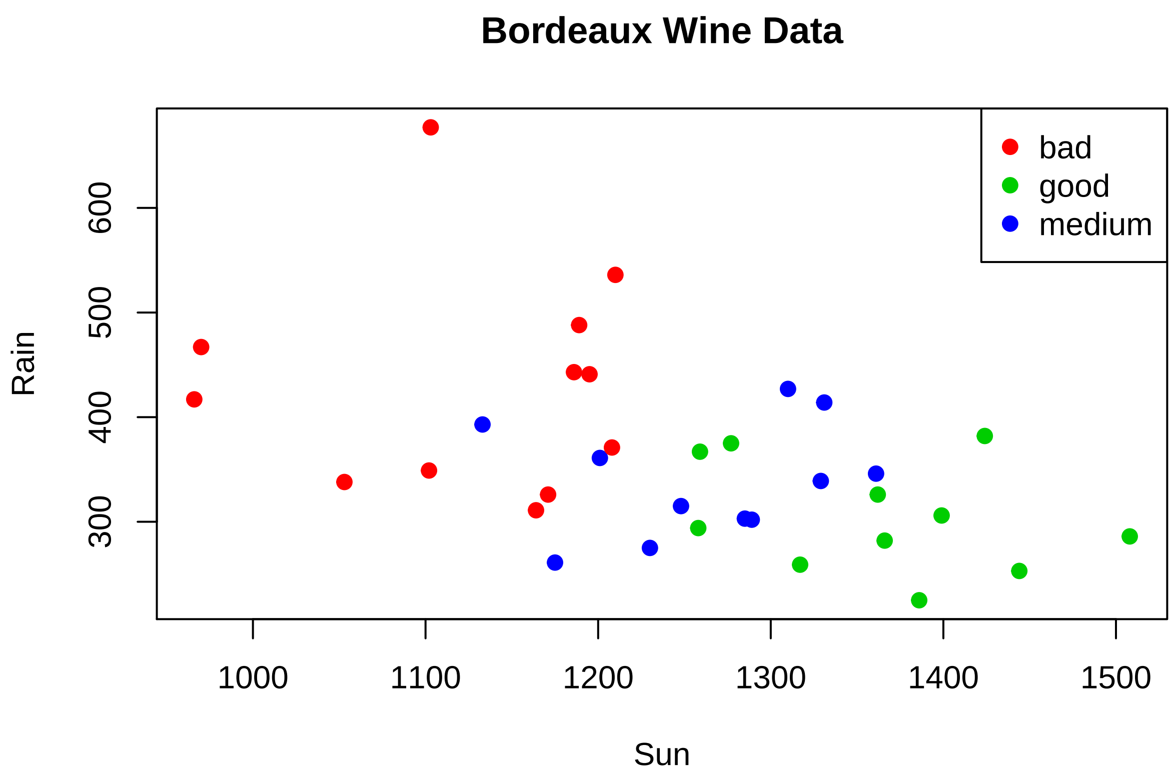

```r
CSQPlot(bordeaux[bordeaux$quality == "good", c("sun","rain")], label = "Control")
```



For all three chi-square quantile plots, multivariate normality does seem to hold. We don't need to apply any transformations to the data.

Let's plot the data together to see visually if we can assume equality of covariance matrices.

```
plot(bordeaux$sun, bordeaux$rain, col = as.numeric(bordeaux$quality)+1, pch = 19, main = "Bordeaux Wine
legend("topright", col = c(2:4), legend = levels(bordeaux$quality), pch = 19)
```



We can also look at the Covariance matrices themselves

```r
print("Covariance Matrix for Bad Wine")
```

```
## [1] "Covariance Matrix for Bad Wine"
```

```r
cov((bordeaux[bordeaux$quality == "bad", c("sun","rain")]))
```

```
##              sun        rain
## sun   7813.35606   -59.78788
## rain   -59.78788 10992.60606
```

```r
print("Covariance Matrix for Medium Wine")
```

```
## [1] "Covariance Matrix for Medium Wine"
```

```r
cov((bordeaux[bordeaux$quality == "medium", c("sun","rain")]))
```

```
##             sun      rain
## sun   5175.4909  912.0636
## rain   912.0636 3023.4545
```

```r
print("Covariance Matrix for Good Wine")
```

```
## [1] "Covariance Matrix for Good Wine"
```

```r
cov((bordeaux[bordeaux$quality == "good", c("sun","rain")]))
```

```
##             sun    rain
## sun    6449.055 -1336.1
## rain  -1336.100  2734.6
```

```r
#calculate Box's M statistic
boxM(bordeaux[,c("sun","rain")], bordeaux$quality)
```

```
##
##  Box's M-test for Homogeneity of Covariance Matrices
##
## data:  bordeaux[, c("sun", "rain")]
## Chi-Sq (approx.) = 7.9852, df = 6, p-value = 0.2392
```

It does appear that we can assume equality of covariance matrices since the p value is well above a rejection threshold of around 0.05. We seem to fit the assumptions of discriminant analysis and because of the homogeneity among covariance matrices, we use linear discriminant analysis.

2) Linear Discriminant Analysis

Because of the homogeneity among the covariance matrices, we would run linear discriminant analysis as the best model

```
bordeaux.disc <- lda(bordeaux[,c(3,5)], grouping = bordeaux$quality)
names(bordeaux.disc)
```

```
## [1] "prior"   "counts"  "means"   "scaling" "lev"     "svd"     "N"
## [8] "call"
```

```
(step1 <- stepclass(quality ~ sun + rain, data = bordeaux, method = "lda", direction = 'both'))
```

```
##  `stepwise classification', using 10-fold cross-validated correctness rate of method lda'.

## 34 observations of 2 variables in 3 classes; direction: both

## stop criterion: improvement less than 5%.

## correctness rate: 0.6;  in: "sun";  variables (1): sun
## correctness rate: 0.7;  in: "rain";  variables (2): sun, rain
##
##  hr.elapsed min.elapsed sec.elapsed
##       0.000       0.000       0.194

## method      : lda
## final model : quality ~ sun + rain
## <environment: 0x7fd7c272e1a0>
##
## correctness rate = 0.7
```

```
step1
```

```
## method      : lda
## final model : quality ~ sun + rain
## <environment: 0x7fd7c272e1a0>
##
## correctness rate = 0.7
```

```
step1$model
```

```
##   nr name
## 1  1  sun
## 2  2 rain
```

The model includes the variables both sun and rain indicating that they are two significant discriminating variables in sun and rain

3) Let's Run the Multivariate Wilk's Lambda test

```
bordeaux.manova <- manova(as.matrix(bordeaux[,c(3,5)]) ~ bordeaux$quality)
summary.manova(bordeaux.manova, test="Wilks")
```

```
##                   Df   Wilks approx F num Df den Df    Pr(>F)
## bordeaux$quality  2 0.31868   11.572      4     60 4.993e-07 ***
## Residuals        31
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

There is statistical evidence that the multivariate means are different. We reject the null meaning that it is possible to discriminate between the bad, medium, and good quality.

4)

```r
summary.aov(bordeaux.manova)
```

```
##  Response sun :
##                   Df Sum Sq Mean Sq F value    Pr(>F)
## bordeaux$quality  2 326909  163455  25.061 3.346e-07 ***
## Residuals        31 202192    6522
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response rain :
##                   Df Sum Sq Mean Sq F value   Pr(>F)
## bordeaux$quality  2  97191   48596  8.4396 0.001185 **
## Residuals        31 178499    5758
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

There exists at least one function that is significant in discriminating between groups.

```r
bordeaux.disc
```

```
## Call:
## lda(bordeaux[, c(3, 5)], grouping = bordeaux$quality)
##
## Prior probabilities of groups:
##       bad      good    medium
## 0.3529412 0.3235294 0.3235294
##
## Group means:
##             sun     rain
## bad    1126.417 430.3333
## good   1363.636 305.0000
## medium 1262.909 339.6364
##
## Coefficients of linear discriminants:
##               LD1          LD2
## sun   0.010691702 -0.006253706
## rain -0.006360228 -0.011547002
##
## Proportion of trace:
##    LD1    LD2
## 0.9948 0.0052
```

Looking at the proportion of trace, there are two discriminating functions but LD1 holds more importance than LD2. LD1 holds much more discriminating power relative to LD2

5)

```r
# raw results - use the 'predict' function

ctraw <- table(bordeaux$quality, predict(bordeaux.disc)$class)
ctraw
```

```
##
##          bad good medium
##   bad      9    0      3
##   good     0    8      3
##   medium   1    2      8
```

```r
# total percent correct
round(sum(diag(prop.table(ctraw))),2)
```

```
## [1] 0.74
```

```r
#cross-validated results
bordeaux.discCV <- lda(bordeaux$quality ~ bordeaux$rain + bordeaux$sun, CV = TRUE)
ctCV <- table(bordeaux$quality, bordeaux.discCV$class)
ctCV
```

```
##
##          bad good medium
##   bad      9    0      3
##   good     0    8      3
##   medium   1    2      8
```

```r
# total percent correct
round(sum(diag(prop.table(ctCV))), 2)
```

```
## [1] 0.74
```

Both percentages for classification with and without cross validation are the same at 74% correct

6)

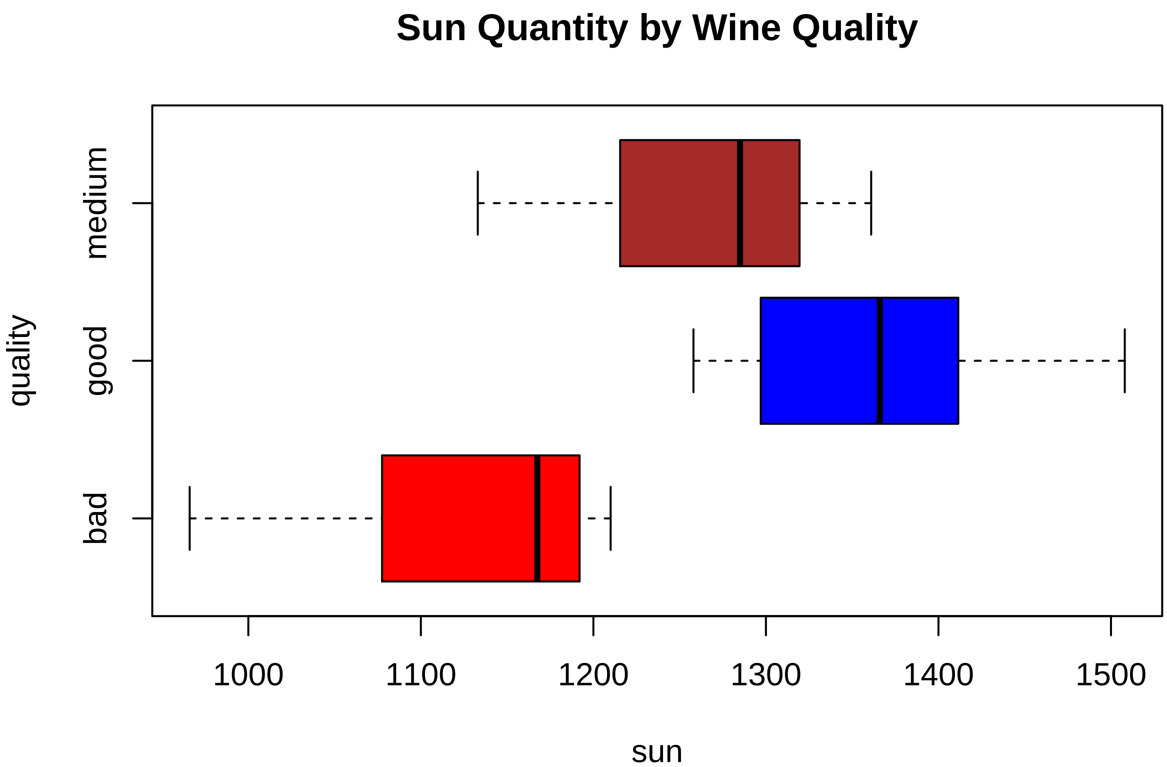
```r
bordeaux.disc
```

```
## Call:
## lda(bordeaux[, c(3, 5)], grouping = bordeaux$quality)
##
## Prior probabilities of groups:
##       bad      good    medium
## 0.3529412 0.3235294 0.3235294
##
## Group means:
##             sun     rain
## bad    1126.417 430.3333
## good   1363.636 305.0000
## medium 1262.909 339.6364
##
## Coefficients of linear discriminants:
##               LD1          LD2
## sun   0.010691702 -0.006253706
## rain -0.006360228 -0.011547002
##
## Proportion of trace:
##    LD1    LD2
```

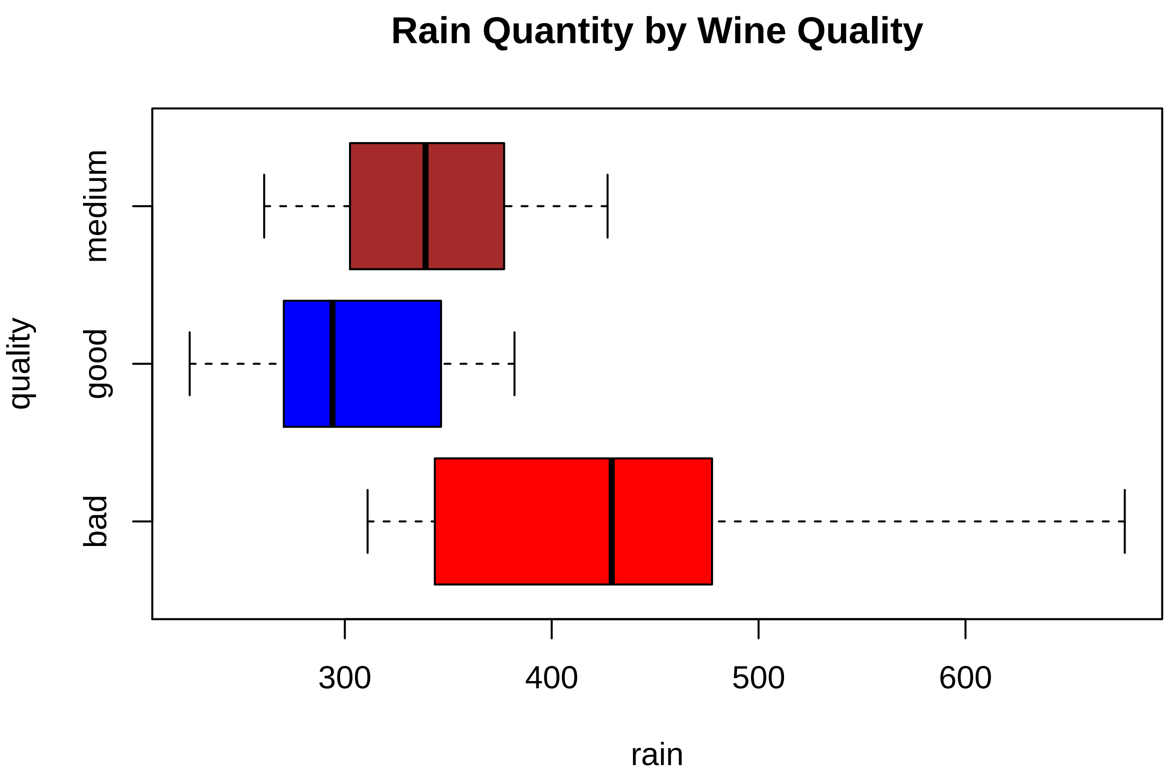
```
## 0.9948 0.0052
```

Looking at the Coefficients for LD1, sun is a better discriminator between groups than rain since its coefficient is of larger magnitude for the stronger discriminant function. This is supplanted by looking at our boxplots from earlier since visually the 3 groups seem to differ more according to the sun variable.

```r
boxplot(sun~ quality, data=bordeaux, col = c("red", "blue", "brown"), horizontal = T, main = "Sun Quanti
```

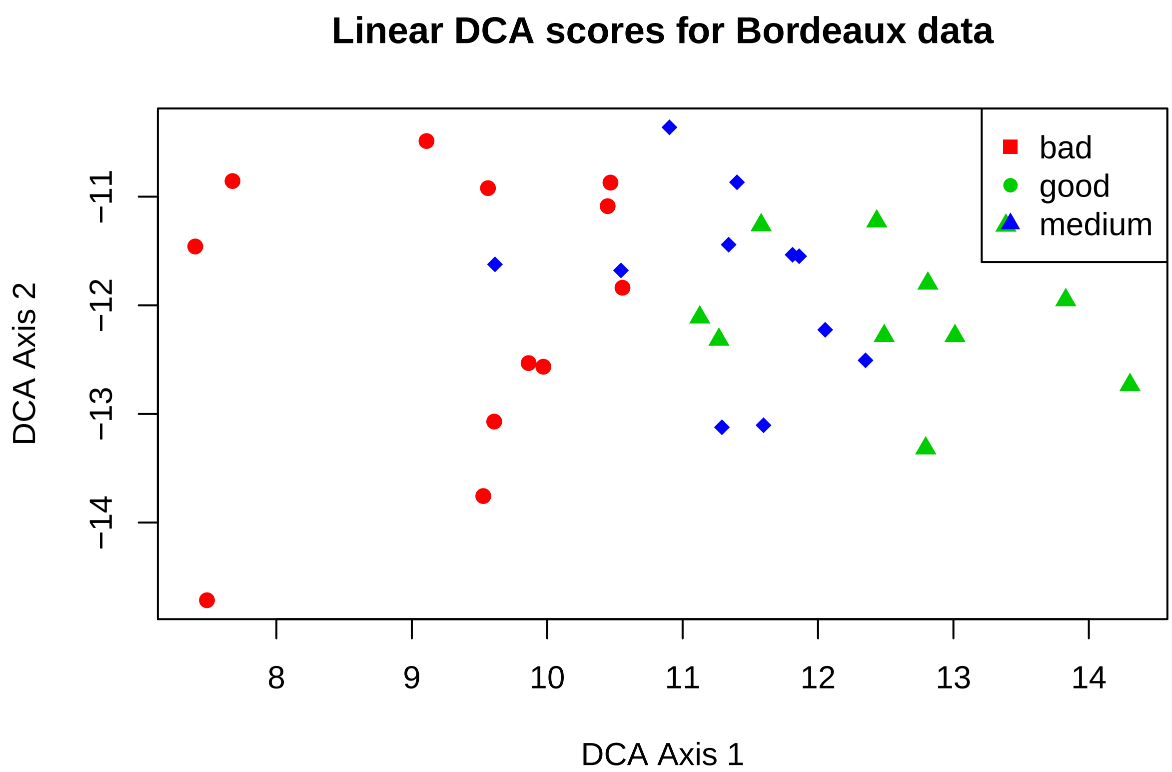



```r
boxplot(rain ~ quality, data=bordeaux, col = c("red", "blue", "brown"), horizontal = T, main = "Rain Qua
```



7)

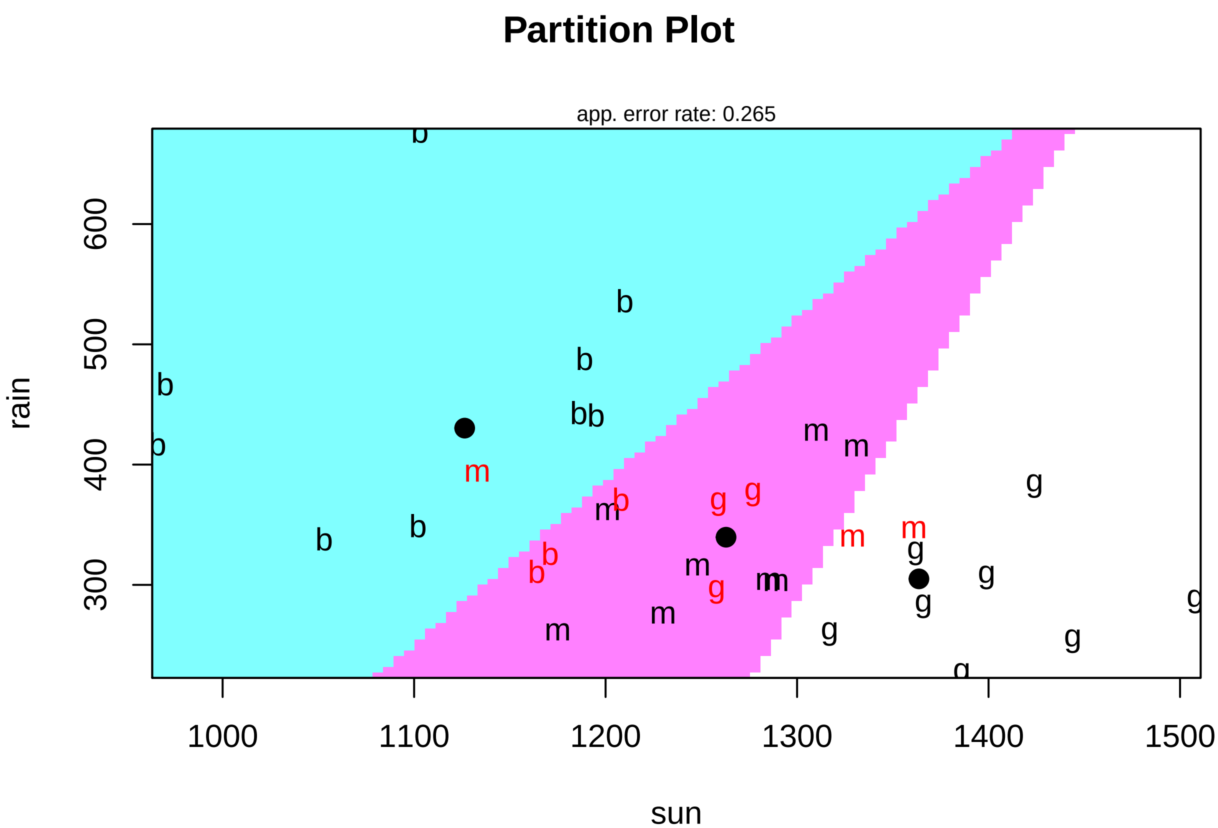
```
#SCORE PLOTS for linear DA
bordeauxlda <- lda(bordeaux[,c("rain","sun")], grouping = bordeaux$quality)
#Calculate scores
scores <- as.matrix(bordeaux[,c("rain","sun")])%*%matrix(bordeauxlda$scaling, ncol = 2)

#NOTE - if use cross-validation option, scores are calculated automatically
plot(scores[,1], scores[,2], type = "n", main = "Linear DCA scores for Bordeaux data",
     xlab = "DCA Axis 1", ylab = "DCA Axis 2")

bordeauxnames <- names(summary(bordeaux[,6]))


for (i in 1:3){
  points(scores[bordeaux$quality == bordeauxnames[i],1],
         scores[bordeaux$quality == bordeauxnames[i],2], col = i+1, pch = 15+i, cex = 1.1)
}
legend("topright", legend = bordeauxnames, col = c(2:4), pch = c(15,16,17))
```



Only two discriminant functions (and second is probably not significant – note that there is not much discrimination in the direction of the second functions). Because we started with two dimensions, this is basically a rotation.

Just as bonus, I included a partition plot as well.

```
partimat(quality ~ rain+sun, data = bordeaux, method = "lda")
```



8)

```
library(class)
##run knn function
bordeaux_train <- bordeaux[,(c("sun", "rain"))]
bordeaux_test <-bordeaux[,(c("sun", "rain"))]
pr <- knn(bordeaux_train, bordeaux_test, cl=bordeaux$quality, k=13)

##create confusion matrix
tab <- table(pr,bordeaux$quality)

##this function divides the correct predictions by total number of predictions that tell us how accurat

accuracy <- function(x){sum(diag(x)/(sum(rowSums(x)))) * 100}
accuracy(tab)
```

```
## [1] 73.52941
```