Stat415-homework4

2.

(a) Create a binary variable, mpg01, that is equal to 1 if the value of mpg for that car is above the median mpg, and 0 otherwise.

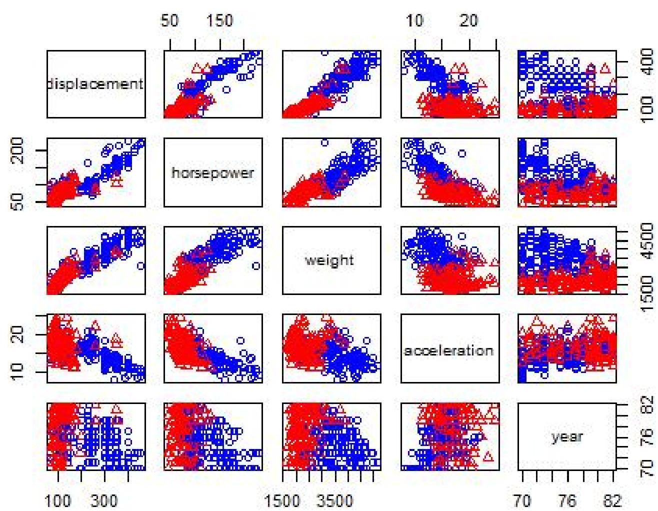
```
library(ISLR)

median_mpg = median(Auto$mpg)
mpg01 = rep("0", nrow(Auto))
for (i in 1:nrow(Auto)){
  if (Auto$mpg[i] > median_mpg){
    mpg01[i] = "1"
  }
}
# add mpg01 to original data
data_new = cbind(Auto, mpg01)
# treat data_new$mpg01 as a categorical variable
```

Comment: the variable mpg01 has been added to the new data set.

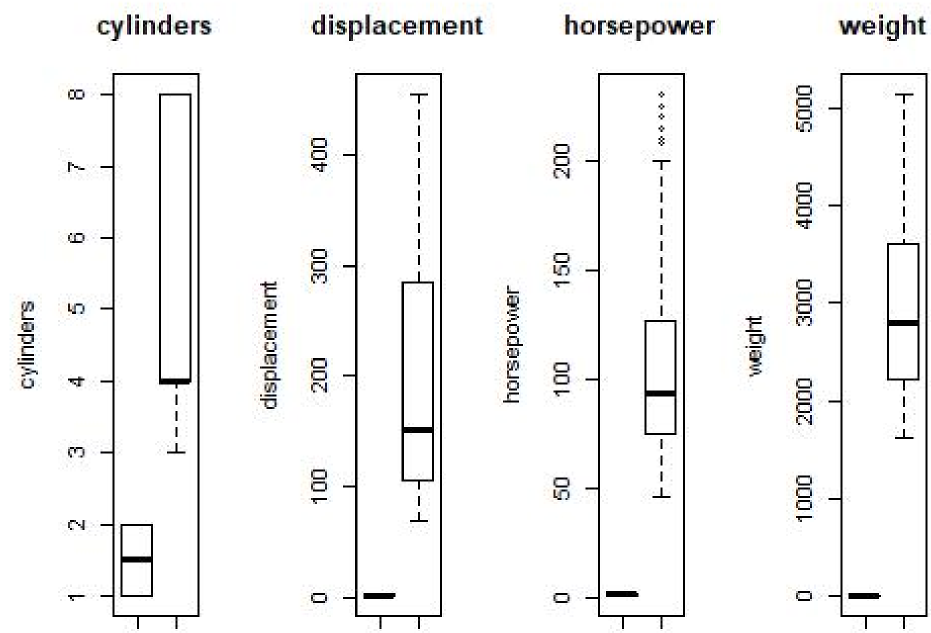
(b) Make some exploratory plots to investigate the association between mpg01 and other variables. Describe your findings.

```
# scatter plots
# round mark represents mpg01 = 0
pairs(data_new[3:7], col=c("blue","red")[data_new$mpg01], pch=c(1,2)[da
ta_new$mpg01])
```

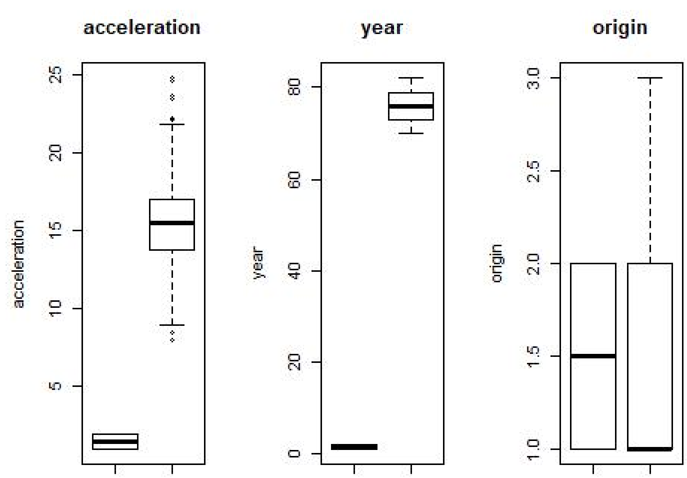



```
# Side-by-side Boxplots
par(mfrow=c(1,4))
boxplot(data_new$mpg01,data_new$cylinders, main = "cylinders",ylab = "c
ylinders")
boxplot(data_new$mpg01,data_new$displacement, main = "displacement",yla
```

```
b = "displacement")
boxplot(data_new$mpg01,data_new$horsepower, main = "horsepower",ylab =
"horsepower")
boxplot(data_new$mpg01,data_new$weight, main = "weight",ylab = "weight")
```




```
par(mfrow=c(1,3))
boxplot(data_new$mpg01,data_new$acceleration, main = "acceleration",yla
b = "acceleration")
boxplot(data_new$mpg01,data_new$year, main = "year",ylab = "year")
boxplot(data_new$mpg01,data_new$origin, main = "origin",ylab = "origin")
```



Comment: Firstly, we need to check the relationship between other variables.According to the scatter plot, we can find that displacement, horsepower and weight are positive correlated and acceleration is almost negative correlated with those three variables. Secondly, in the box plot of horsepower, weight, displacement and acceleration, the range of values for two kinds of classes would be quite different. Also, for these several variables, corresponding scatter plots show that points scattered respectively in the plot of displacement, horsepower, weight and acceleration. It indicates that samples with different mpg01 value have large difference in horsepower, weight and acceleration.Thus we can adopt these three features to predict mpg01.

(c) Split the data into a training set and a test set: fix the random seed to the value 12345, and randomly select 80% of the observations (round down to the nearest integer) from each class to be the training data. Use the rest as test data.

```
set.seed(12345)
table(data_new$mpg01)

##
##   0   1
## 196 196

mpg01_0 = which(data_new$mpg01 == 0)
mpg01_1 = which(data_new$mpg01 == 1)
train_index = c(sample(mpg01_0, size = trunc(0.80 * length(mpg01_0))),
                sample(mpg01_1, size = trunc(0.80 * length(mpg01_1))))
# Divide traing data and test data
Auto_train = data_new[train_index, ]
Auto_test = data_new[-train_index, ]
nrow(Auto_train)

## [1] 312

nrow(Auto_test)

## [1] 80
```

Comment: the data has been divided into training part and test part. The length of training data is 312 and the length of test data is 80.

(d) Perform LDA on the training data in order to predict mpg01 using four quantitative variables that seem most associated with mpg01 based on (b). Report the training and test errors. Make a plot of the training data points, using two variables which appear to be most associated with the class as your axes. Using different colors to show the true values of mpg01, and different plotting symbols to show predicted values.

```
# The most associated quantitative variables are horsepower, weight, di
splacement and acceleration
library(MASS)

mpg01 = as.numeric(mpg01)
data_new2 = cbind(Auto, mpg01) # numeric mpg01
Auto_train2 = data_new2[train_index, ]
```

```r
Auto_test2 = data_new2[-train_index, ]
lda.fit = lda(mpg01 ~ horsepower + weight + acceleration + displacement,
 data=Auto_train2)
lda.fit
```

```
## Call:
## lda(mpg01 ~ horsepower + weight + acceleration + displacement,
##     data = Auto_train2)
##
## Prior probabilities of groups:
##   0   1
## 0.5 0.5
##
## Group means:
##   horsepower   weight acceleration displacement
## 0  129.37179 3621.090     14.69744     270.5385
## 1   79.10256 2336.314     16.34295     116.2212
##
## Coefficients of linear discriminants:
##                        LD1
## horsepower    0.0011054061
## weight       -0.0009818156
## acceleration -0.0265384951
## displacement -0.0077156357
```

```r
names(predict(lda.fit, Auto_train2))
```

```
## [1] "class"     "posterior" "x"
```

```r
# show the predicted class for each sample
head(predict(lda.fit, Auto_train2)$class, n = 5)
```

```
## [1] 0 0 0 0 1
## Levels: 0 1
```

```r
# show the prob in each class
head(predict(lda.fit, Auto_train2)$posterior, n = 5)
```

```
##              0          1
## 216 0.97899252 0.02100748
## 264 0.81301053 0.18698947
## 227 0.85791542 0.14208458
## 265 0.89420557 0.10579443
## 120 0.08181619 0.91818381
```

```r
lda_train_pred = predict(lda.fit, Auto_train2)$class
lda_train_pred_data = cbind(Auto_train2[,-1],lda_train_pred)
lda_test_pred = predict(lda.fit, Auto_test2)$class
# define the error function
calc_class_err = function(actual, predicted){
  mean(actual != predicted)
}
# training error
calc_class_err(predicted = lda_train_pred, actual = Auto_train2$mpg01)
```

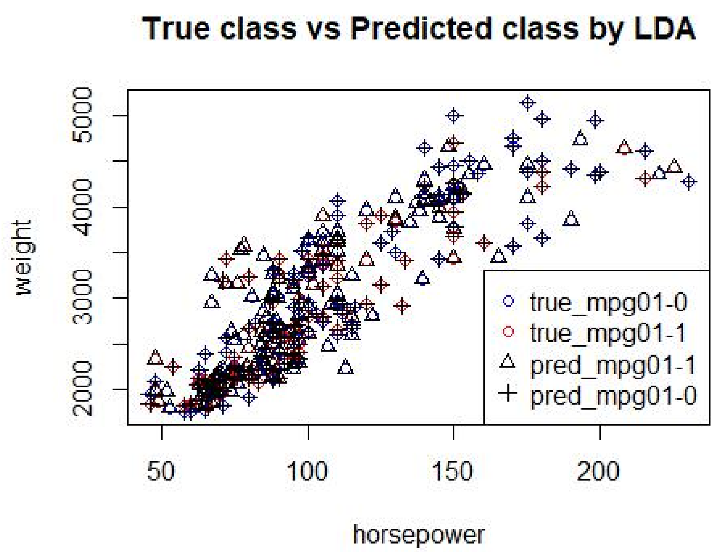
```
## [1] 0.1185897
```

```r
# test error
calc_class_err(predicted = lda_test_pred, actual = Auto_test2$mpg01)
```

```
## [1] 0.075
```

Comment: According to the result, the training error is 0.1185897 and the test error is 0.075.

```r
mpg01 = as.factor(mpg01)
plot(Auto_train2$horsepower,Auto_train2$weight, col = c("blue","red")[m
pg01], xlab = "horsepower", ylab = "weight", main = "True class vs Pred
icted class by LDA")
points(lda_train_pred_data$horsepower,lda_train_pred_data$weight, pch =
 c(2,3)[lda_test_pred])
legend("bottomright", c("true_mpg01-0","true_mpg01-1","pred_mpg01-1","p
red_mpg01-0"), col=c("blue", "red", "black", "black"),pch=c(1,1,2,3))
```



Comment: According to the analysis, we adopt horsepower and weight to plot. The meaning of corresponding colors and shapes have been explained in the label of the plot.

(e) Perform QDA on the training data in order to predict mpg01 using the same variables you used for LDA. Report the training and test errors. Make a plot analogous to the one you made for LDA.

```r
qda.fit = qda(mpg01 ~ horsepower + weight + acceleration + displacement,
 data=Auto_train2)
qda.fit
```

```
## Call:
## qda(mpg01 ~ horsepower + weight + acceleration + displacement,
##     data = Auto_train2)
##
```

```
## Prior probabilities of groups:
##   0   1
## 0.5 0.5
##
## Group means:
##   horsepower   weight acceleration displacement
## 0  129.37179 3621.090     14.69744     270.5385
## 1   79.10256 2336.314     16.34295     116.2212
```

```
names(predict(qda.fit, Auto_train2))
```

```
## [1] "class"     "posterior"
```

```
# show the predicted class for each sample
head(predict(qda.fit, Auto_train2)$class, n = 5)
```

```
## [1] 0 0 0 0 1
## Levels: 0 1
```

```
# show the prob in each class
head(predict(qda.fit, Auto_train2)$posterior, n = 5)
```

```
##             0            1
## 216 0.9999998 2.154103e-07
## 264 0.9999980 1.952750e-06
## 227 0.9177769 8.222307e-02
## 265 0.9999998 1.689040e-07
## 120 0.0296050 9.703950e-01
```

```
qda_train_pred = predict(qda.fit, Auto_train2)$class
qda_test_pred = predict(qda.fit, Auto_test2)$class
qda_train_pred_data = cbind(Auto_train2[,-1],qda_train_pred)
# training error
calc_class_err(predicted = qda_train_pred, actual = Auto_train2$mpg01)
```

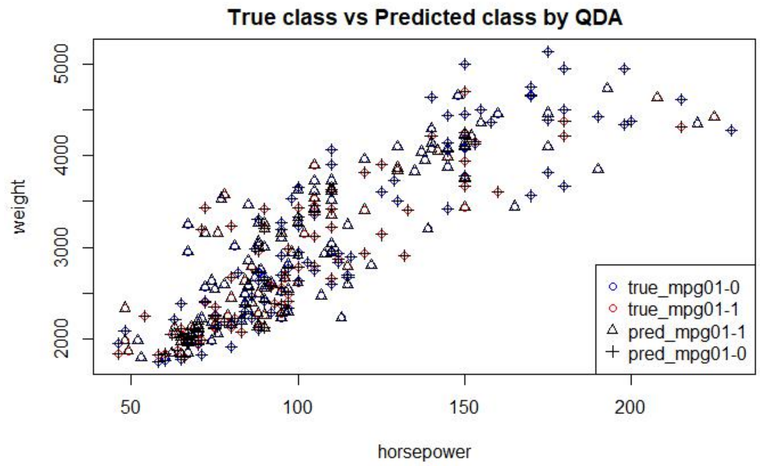
```
## [1] 0.1025641
```

```
# test error
calc_class_err(predicted = qda_test_pred, actual = Auto_test2$mpg01)
```

```
## [1] 0.0625
```

Comment: According to the result, the training error is 0.1025641 and the test error is 0.0625.

```
mpg01 = as.factor(mpg01)
plot(Auto_train2$horsepower,Auto_train2$weight, col = c("blue","red")[m
pg01], xlab = "horsepower", ylab = "weight", main = "True class vs Pred
icted class by QDA")
points(qda_train_pred_data$horsepower,qda_train_pred_data$weight, pch =
 c(2,3)[qda_test_pred])
legend("bottomright", c("true_mpg01-0","true_mpg01-1","pred_mpg01-1","p
red_mpg01-0"), col=c("blue", "red", "black", "black"),pch=c(1,1,2,3))
```



Comment: According to the analysis, we adopt horsepower and weight to plot. The meaning of corresponding colors and shapes have been explained in the label of plot.

(f) Compare and contrast the performance of LDA and QDA. What do your results suggest about the class-specific co-variances? Comment: According to the result shown above, the training error and test error of LDA are 0.1185897 and 0.075. The training error and test error of QDA are 0.1025641 and 0.0625. Both training error and test error of QDA are smaller than those of LDA. Thus QDA is more suitable here to fit the data. Since QDA assumes that the co-variance in each class should be different, then we should also assume the class specific co-variances to be different in the data set.