Home_Work_3

Group: Bob [Maryla Wozniak, Shakila Hoque, Elai Shalev, Matthew Perez]

10/5/2019

GENERAL SETUP

```
## Clear all existing variables in global environment
rm(list = ls())
## Clear plot tab and close/save any open files
dev.off()

## Step by step instructions
## Set the working dorectory to the location where the file was saved

## Setp: 1
## The 'rstudioapi' package allows you to access
## system information using the R language
## the 'getSourceEditorContext' function allows you
## to retreive informaton about the source files locaiton
## this includes the file path without you having to
## know where the fiole is. THe computer doest that work for you.

## rstudioapi::getSourceEditorContext()


## Step: 2
## The line above is of type 'list' meaning there are multiple
## peices of information contained within. In order to get the
## information we need we need to acess the file path
## under the 'path' acesor in the return of the functioncall
## to do that we append '$path' this will print the file
## path including the name of the file.

## rstudioapi::getSourceEditorContext()$path


## Step: 3
## In order to omit the name of the file from printing
## we can use the 'dirname' function that will ignore
## file names at the end of a file path and print
## only the path leading up to where the file is saved.

## dirname(rstudioapi::getSourceEditorContext()$path)


## Step: 4
## Lastly we use the 'setwd' comand to set the
## directory to the returned value of the entire function
## call in order to set the working directory to the
## locaiton of the file without ever having to look for where
```

```
## the file is actually stored.

## setwd(dirname(rstudioapi::getSourceEditorContext()$path))

## NOTE: THE LINES ABOVE ARE COMMENTED OUT IN ORDER TO
## PREVENT COMPILATION ERRORS IN 'RMD' AND OTHER 'TEX' BASED
## ENGINES. THIS IS A COMMAND IN ORDER TO SET YOUR WORKING
## DIRECTORY AND SHOULD BE COMMENTED OUT OR COMPLETELY
## OMITED FROM THE FINAL PRODUCT/PRESENTATION.
```

IMPORT CLEAN DATA

Instead of importing and cleaning the data ourselves we are using the clean data contained in the ISLR packge from the CRAN website

```
# Install the ISLR library so we can access the cleaned data set
# for the following questions

# install.packages("ISLR")

# Save local copy of ISLR's Auto data
Auto_local = ISLR::Auto

# Install and load "MASS" packge for aceess to the analytical functions later in the assignment
# install.packages("MASS")
library("MASS")

# Attach reference varables from data set
attach(Auto_local)
```

Question 11: Develop a model to predict whether a car has high/low mpg.

(A) DEFINE BINARY VARIABLE 'mpg01' MPG01 = 1 IF MPG > MEDIAN(MPG) ELSE 0

```
Auto_local[["mpg01"]] = as.integer(Auto_local$mpg > median(Auto_local$mpg))
summary(Auto_local)
```

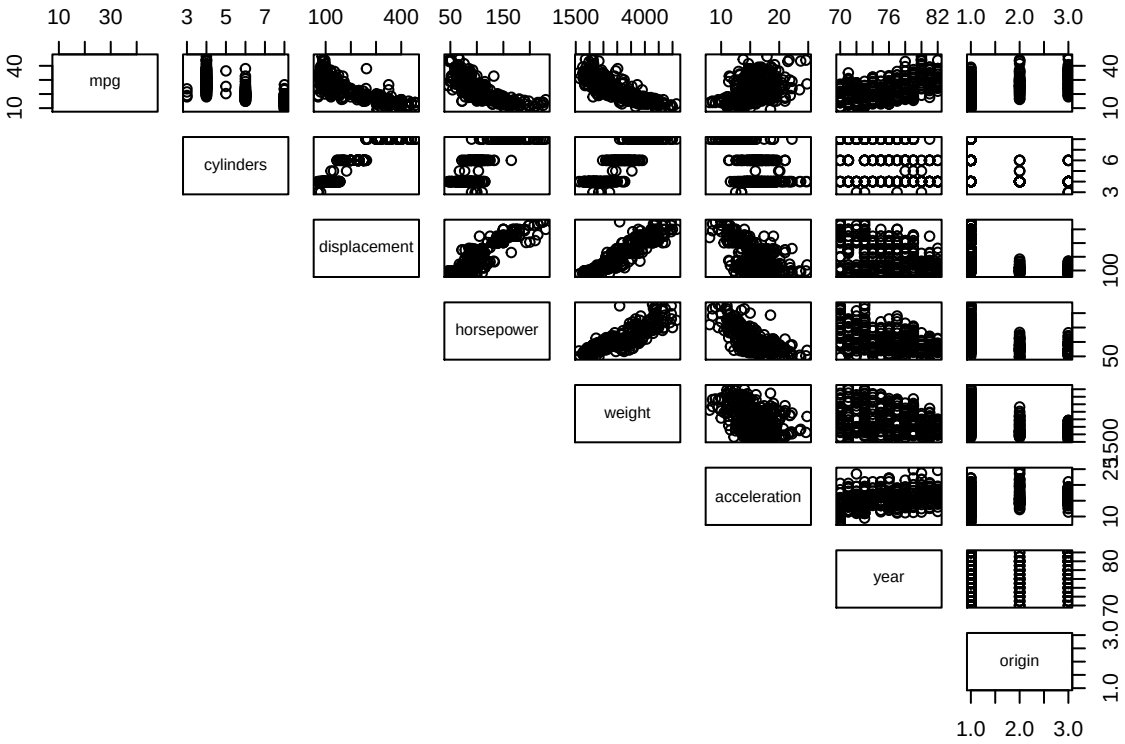
```
##       mpg          cylinders      displacement     horsepower
##  Min.   : 9.00   Min.   :3.000   Min.   : 68.0   Min.   : 46.0
##  1st Qu.:17.00   1st Qu.:4.000   1st Qu.:105.0   1st Qu.: 75.0
##  Median :22.75   Median :4.000   Median :151.0   Median : 93.5
##  Mean   :23.45   Mean   :5.472   Mean   :194.4   Mean   :104.5
##  3rd Qu.:29.00   3rd Qu.:8.000   3rd Qu.:275.8   3rd Qu.:126.0
##  Max.   :46.60   Max.   :8.000   Max.   :455.0   Max.   :230.0
##
##      weight      acceleration        year           origin
##  Min.   :1613   Min.   : 8.00   Min.   :70.00   Min.   :1.000
##  1st Qu.:2225   1st Qu.:13.78   1st Qu.:73.00   1st Qu.:1.000
##  Median :2804   Median :15.50   Median :76.00   Median :1.000
##  Mean   :2978   Mean   :15.54   Mean   :75.98   Mean   :1.577
##  3rd Qu.:3615   3rd Qu.:17.02   3rd Qu.:79.00   3rd Qu.:2.000
##  Max.   :5140   Max.   :24.80   Max.   :82.00   Max.   :3.000
##
##                  name         mpg01
##  amc matador       :  5   Min.   :0.0
##  ford pinto        :  5   1st Qu.:0.0
##  toyota corolla    :  5   Median :0.5
##  amc gremlin       :  4   Mean   :0.5
##  amc hornet        :  4   3rd Qu.:1.0
##  chevrolet chevette:  4   Max.   :1.0
##  (Other)           :365
```

(B) VISUAL DATA EXPLORATION

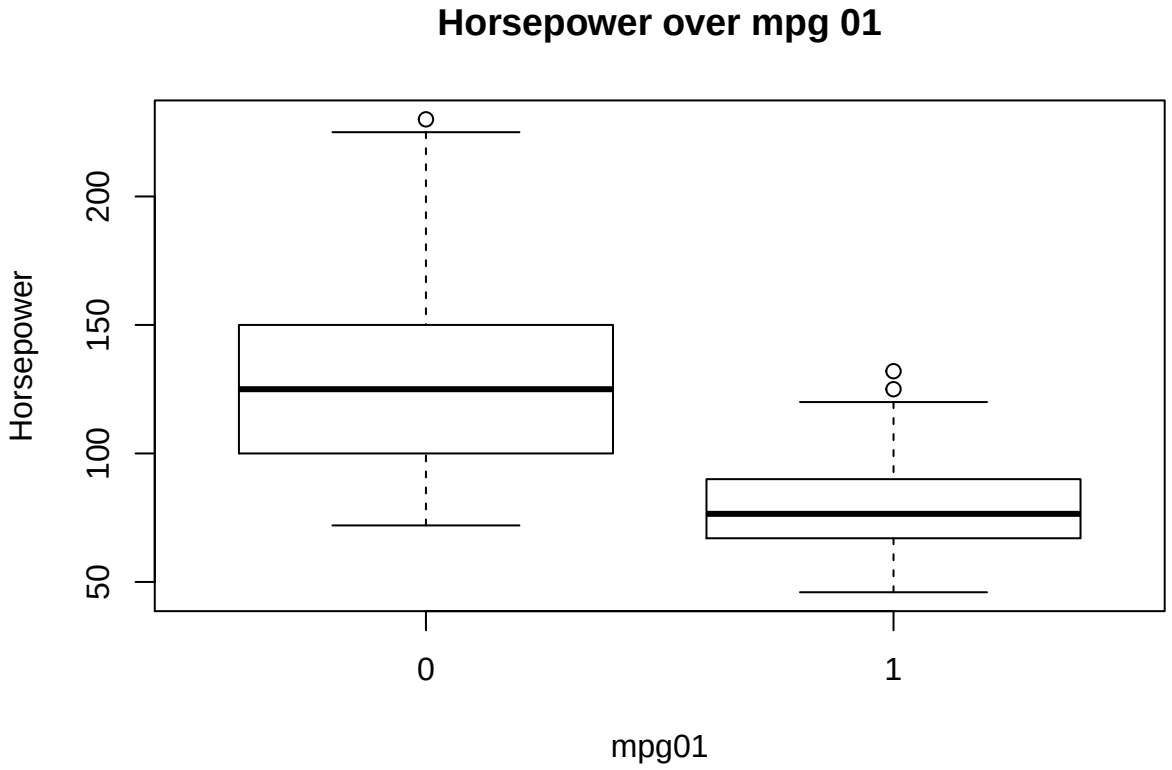
WHICH FEATURES ARE LIKELY TO HAVE THE GREATEST IMPACT?

DESCRIBE YOUR FINDINGS

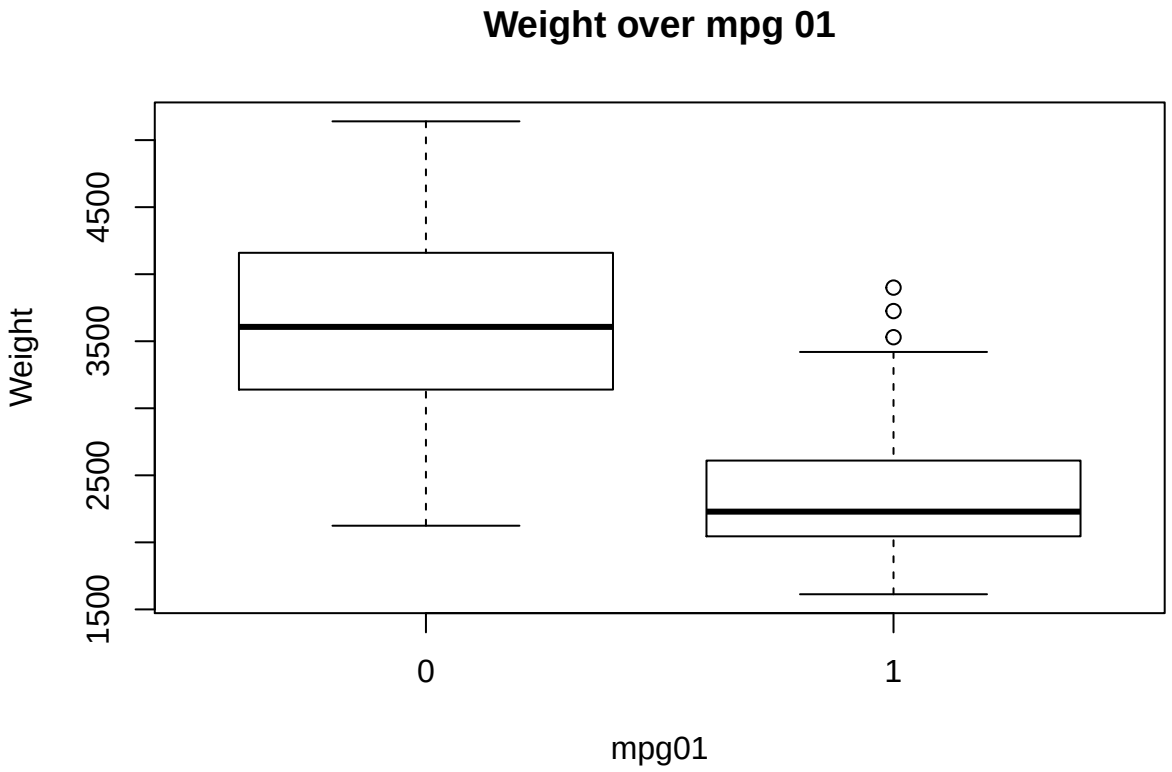
```
## INITIAL PLOT
## The 'names' and 'mpg01' columns are not used in the following
## matrix plot because 'names' won't provide anymeaningful data in
## the visual analysis while 'mpg01' would be best analyzed using box plots

pairs(Auto_local[,c(-9, -10)], lower.panel = NULL)
```



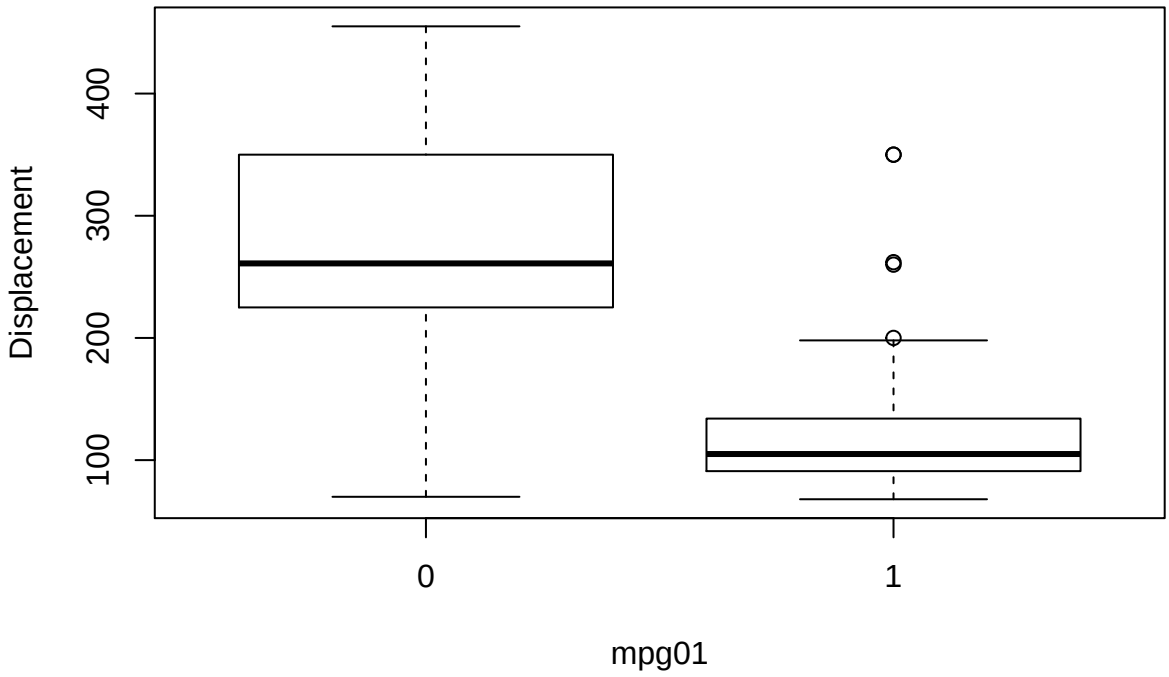

```
## COMMENTS:
## mpg01 is a direct product of mpg.
## Since horsepower, weight, displacement, and acceleration
## features affect mpg, they affect mpg01 as well. We boxplot
## mpg and each of these features to investigate how they might affect mpg01


## COMMENTS:
## The boxplot below tells us that we can predict that cars with high
## horsepower would have mpg01=0 (mpg below median). The highest
## observation of the mpg01=1 category (neglecting outliers) is
## about equal to the median observation of mpg01=0 category.
boxplot(horsepower~mpg01,
        data = Auto_local,
        ylab = "Horsepower",
        main = "Horsepower over mpg 01")
```



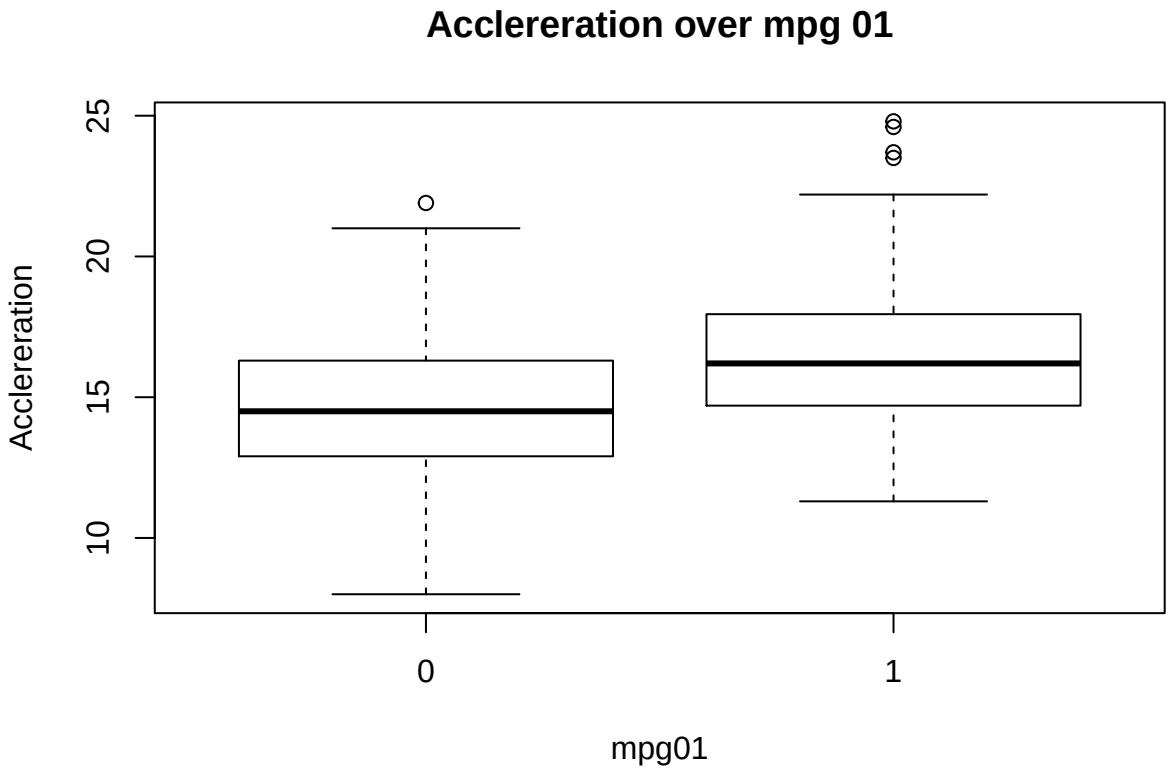

```
## COMMENTS:
## The boxplot below tells us that we can predict that heavy
## cars would belong to the mpg01=0 category (mpg below median).
## The median observation of the mpg01=1 category is about
## equal to the smallest observation of mpg01=0! If a car
## weighs over 3500 we can certainly predict it will belong
## to the mpg01=0 category, and half of the cars in the
## Auto dataset weigh more than 3500.
boxplot(weight~mpg01,
        data = Auto_local,
        ylab = "Weight",
        main = "Weight over mpg 01")
```




```
## COMMENTS:
## The boxplot below tells us that we can predict that cars
## with high displacement would belong to the mpg01=0
## category (mpg below median). All of the observations
## of the mpg01=1 category fit in the first quartile of
## the mpg01=0 category. Using the displacement feature,
## we can predict over 75% of the cars would belong in
## the mpg01=0 category. This feature seems to be the
## most useful in trying to predict mpg01.
boxplot(displacement~mpg01,
        data = Auto_local,
        ylab = "Displacement",
        main = "Displacement over mpg 01")
```




```
## COMMENTS:
## The boxplot below tells us that it would be very hard
## to use the acceleration feature to predict mpg01,
## since the two categories distribute very similarly.

boxplot(acceleration~mpg01,
        data = Auto_local,
        ylab = "Acclereration",
        main = "Acclereration over mpg 01")
```



(C) SPLIT DATA INTO TEST AND TRAINING SETS

```
## not true random but good enough
set.seed(1)
num_train <- nrow(Auto_local) * 0.80

inTrain <- sample(nrow(Auto_local), size = num_train)

training <- Auto_local[inTrain,]
testing <- Auto_local[-inTrain,]
```

(D) PERFORM LDA ON THE TRAINGING SET TO PREDICT 'mpg01'

WHAT IS THE TEST ERROR?

```
## The following formula will be used for each
## of the following models

fmla <- as.formula('mpg01 ~ horsepower + weight + displacement')

lda_model <- lda(fmla, data = training)

lda_pred <- predict(lda_model, testing)

table(lda_pred$class, testing$mpg01)
```

```
##
##      0  1
##   0 33  1
##   1  9 36
```

```r
1 - mean(lda_pred$class == testing$mpg01)
```

```
## [1] 0.1265823
```

```r
## COMMENTS:
#The LDA model works great!
#The test error is 0.1265823
```

(E) PERFORM QDA ON THE TRAINGING SET TO PREDICT 'mpg01'

WHAT IS THE TEST ERROR?

```r
qda_model = qda(fmla, data = training)

qda_pred <- predict(qda_model, testing)

table(qda_pred$class, testing$mpg01)
```

```
##
##      0  1
##   0 36  2
##   1  6 35
```

```r
1 - mean(qda_pred$class == testing$mpg01)
```

```
## [1] 0.1012658
```

```r
## COMMENTS:
## QDA modal test error is lower then the LDA,
## this is likely because QDA is a quadratic line
## meaning the raltionship between the variables
## that we ploted dont have a linear relationship.
## The error rate is 0.1012658
```

(F) PERFORM LOGISTIC REGRESSION ON THE TRAINING DATA IN ORDER TO PREDICT 'mpg01'

WHAT IS THE TEST ERROR?

```r
log_reg <- glm(fmla, data = training, family = binomial)

pred <- predict(log_reg, testing, type = 'response')
pred_values <- round(pred)
table(pred_values, testing$mpg01)
```

```
##
## pred_values  0  1
##           0 38  1
##           1  4 36
```

```r
1- mean(pred_values == testing$mpg01)
```

```
## [1] 0.06329114
```

```r
## COMMENTS:
## The Legistic Regression model performed
## considerably better then both the LDA
## and the QDA models. Our asssumption is that
## because the Logistic regression is designed to
```

```
## handle non-numeric data it's better suited to conduct
## the analysis for this exercise.
## The error rate is 0.06329114
```