Math 189 Homework 5: Data Analysis of Auto Mileage

Yunchun Pan, Siqing Lyu, Nathan Ng, Pudan Xu

2/16/2021

Introduction

In this homework, we examine the Auto dataset[1] from the ISLR package[2]. We develop a model to predict whether a given car gets high or low gas mileage based on the features of this car. The Auto dataset has 392 observations of automobiles and each observation has the following variables:

- **cylinders**: Number of cylinders between 4 and 8
- **displacement**: Engine displacement(cubic inches)
- **horsepower**: Engine horsepower
- **weight**: Vehicle weight(lbs.)
- **acceleration**: Time to accelerate from 0 to 60 mph(sec.)
- **year**: Model year(modulo 100)
- **origin**: Origin of car(1 = American, 2 = European, 3 = Japanese)
- **name**: Vehicle name
- **mpg**: Miles per gallon

In order to determine whether a car has high or low gas mileage, we use the median mpg of all cars to create a binary variable based on whether the car has higher or lower mpg than the median. Then, we explore the data using boxplots and chi-square test of independence to investigate the association between the high and low mpg and the other variables in the Auto dataset. Using the variables that are strongly associated with high and low mpg, we split the data into a training and test set and perform Linear Discriminant Analysis on the training set. In the end, we use the resulting linear discriminant function to predict whether a car has good or bad mileage for all test data and discuss the results in terms of the proportion of correctly and incorrectly classified records.

Our Work

We first load the Auto dataset from ISLR package into R to get the data we use throughout this assignment and save it into **auto**. Then, we display its top 6 rows.

```
library(ISLR)
```

```
## Warning: package 'ISLR' was built under R version 3.6.3
```

[1]Source: This dataset was taken from the StatLib library which is maintained at Carnegie Mellon University. The dataset was used in the 1983 American Statistical Association Exposition.

[2]References: James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013) An Introduction to Statistical Learning with applications in R, www.StatLearning.com, Springer-Verlag, New York

```
data(Auto)
auto <- Auto
head(Auto)
```

```
##   mpg cylinders displacement horsepower weight acceleration year origin
## 1  18         8          307        130   3504         12.0   70      1
## 2  15         8          350        165   3693         11.5   70      1
## 3  18         8          318        150   3436         11.0   70      1
## 4  16         8          304        150   3433         12.0   70      1
## 5  17         8          302        140   3449         10.5   70      1
## 6  15         8          429        198   4341         10.0   70      1
##                       name
## 1 chevrolet chevelle malibu
## 2         buick skylark 320
## 3        plymouth satellite
## 4             amc rebel sst
## 5               ford torino
## 6          ford galaxie 500
```

After we load in the Auto dataset, we find the median of all cars' mileage, **median_mpg**. Comparing mpg values with **median_mpg**, we create a binary variable, **mpg01**, that contains 1 if mpg has a value above the median **median_mpg**, and 0 if mpg contains a value below **median_mpg**. As shown by the first 6 rows of the updated Auto dataset, mpg values are all below the median 22.75, hence mpg01 values are all 0.

```
median_mpg = median(Auto$mpg)
Auto$mpg01 <- as.numeric(Auto$mpg > median_mpg)
cat("Median value of mpg:", median_mpg)
```

```
## Median value of mpg: 22.75
```

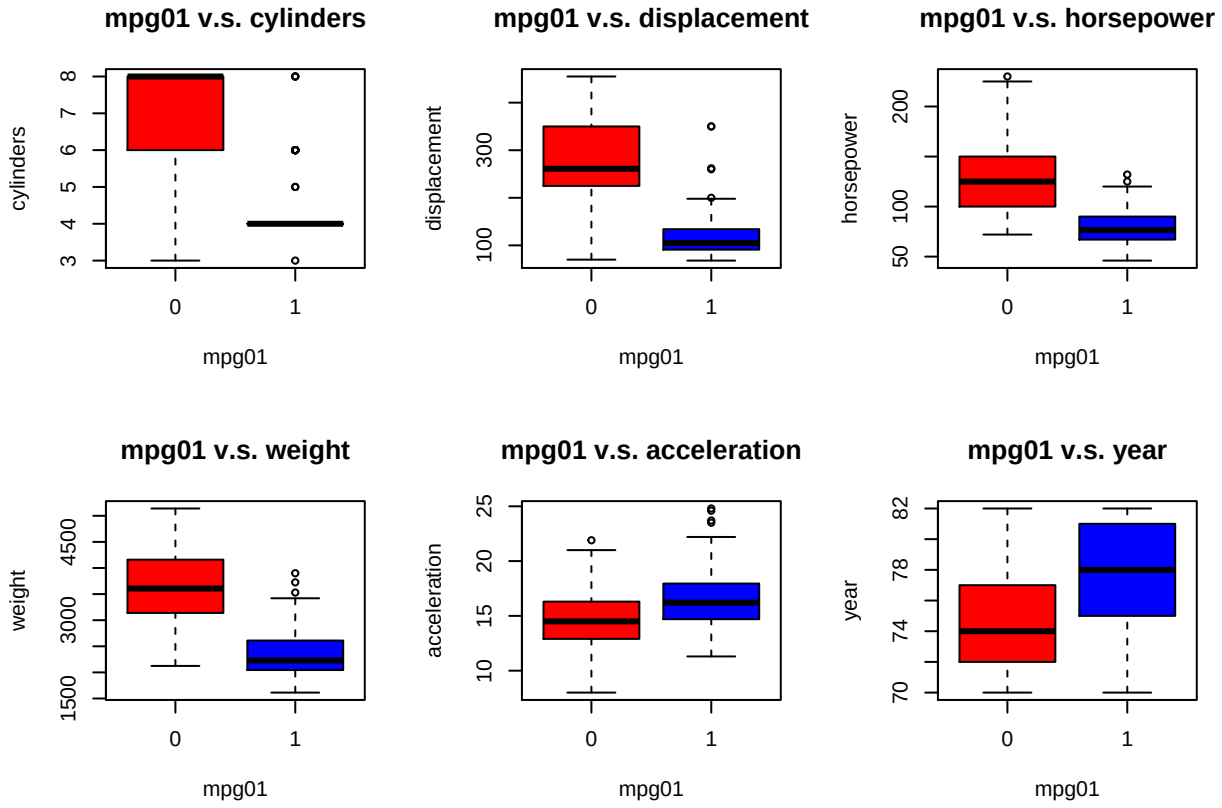
```
head(Auto[, c('mpg', 'mpg01')])
```

```
##   mpg mpg01
## 1  18     0
## 2  15     0
## 3  18     0
## 4  16     0
## 5  17     0
## 6  15     0
```

Here, we create boxplots to investigate the association between mpg01 and other features. All of these factors seem to be very useful in predicting mpg01. The boxplot of mpg01 and year shows that fuel-efficient cars whose mileage are above the median were generally made later than fuel-inefficient cars that have mileage below the median. Moreover, fuel-efficient cars generally have fewer cylinders and hence weigh less than fuel-inefficient cars. Fuel-efficient cars also have fewer displacement and horsepower but accelerate faster than fuel-inefficient cars.

```
par(mfrow=c(2, 3))
boxplot(cylinders ~ mpg01, data=Auto,
        col=c("red","blue"), main='mpg01 v.s. cylinders')
```

```
boxplot(displacement ~ mpg01, data=Auto,
        col=c("red","blue"), main='mpg01 v.s. displacement')

boxplot(horsepower ~ mpg01, data=Auto,
        col=c("red","blue"), main='mpg01 v.s. horsepower')

boxplot(weight ~ mpg01, data=Auto,
        col=c("red","blue"), main='mpg01 v.s. weight')

boxplot(acceleration ~ mpg01, data=Auto,
        col=c("red","blue"), main='mpg01 v.s. acceleration')

boxplot(year ~ mpg01, data=Auto,
        col=c("red","blue"), main='mpg01 v.s. year')
```

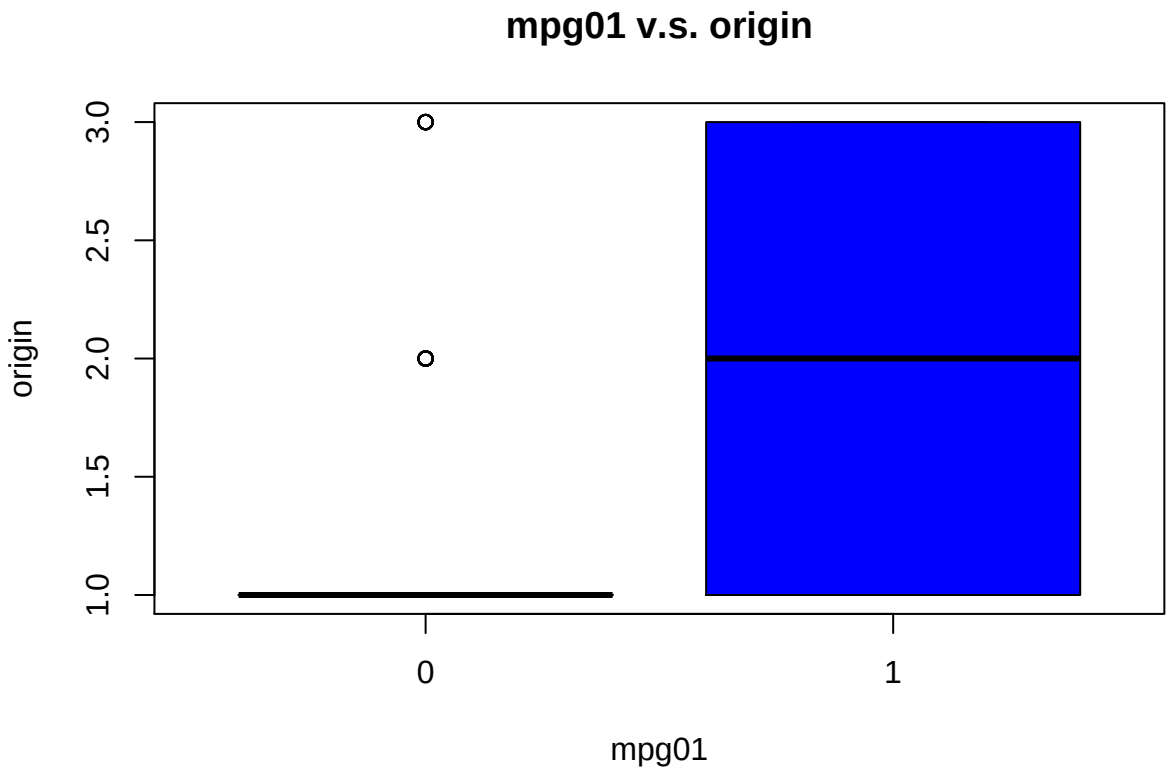


Here, we create the boxplot of **mpg01** and **origin** and count the number of fuel-efficient and fuel-inefficient cars that were made in America, Europe, and Japan. As the dataframe and the boxplot below show, most fuel-inefficient cars were made in America and only 23 cars were made in Europe and Japan. Similarly, most fuel-efficient cars were made in America, but approximately same number of such cars were made in Japan and slightly fewer cars were made in Europe.

```
library(plyr)
data.frame("origin" =c(1,2,3),
           "mpg01_1_freq"=count(Auto[Auto$mpg01==1,],vars = "origin")[,2],
           "mpg01_0_freq"=count(Auto[Auto$mpg01==0,],vars = "origin")[,2])
```

```
##   origin mpg01_1_freq mpg01_0_freq
## 1      1           72          173
## 2      2           54           14
## 3      3           70            9
```

```
boxplot(origin ~ mpg01, data=Auto,
        col=c("red","blue"), main='mpg01 v.s. origin')
```



Moreover, we perform a chi-square test of independence to test whether there is a an association between the categories of **origin** and **mpg01**. The null hypothesis is that the categories of **origin** and **mpg01** are independent, while the alternative hypothesis is that there is an association between the categories of **origin** and **mpg01**. At the significance level of 0.05, if the p value is larger than or equal to 0.05, we fail to reject the null hypothesis that these two categorical variables are independent and hence **origin** is not useful for predicting **mpg01**. Otherwise, we reject the null hypothesis and **origin** is helpful for predicting **mpg01**.

```
cat("The significance level:", 0.05)
```

```
## The significance level: 0.05
```

```
chisq.test(Auto$mpg01, Auto$origin)['p.value']
```

```
## $p.value
## [1] 4.18255e-25
```

Since the p value of this test is 4.18255e-25 and it is significantly less than 0.05, we reject the null hypothesis that the categorical variables **origin** and **mpg01** are independent, and there are clear evidences for us to conclude that there is a significant association between the categories of **origin** and **mpg01**. Hence, **origin** is very helpful for predicting **mpg01**.

Here, we split the Auto dataset into fuel-efficient car subset **mpg.good** and fuel-inefficient car subset **mpg.bad**. After exploring dimensions of these two subsets, we find that they both have 192 observations. Then, we take the first 150 observations from each subset and combine them into a training dataset **auto.train** of 300 observations, and we combine the remaining 92 observations in both subsets into a test set **auto.test**. **n_good** and **n_bad** store the number of fuel-efficient and fuel-inefficient cars in the training set.

```
mpg.good <- Auto[Auto$mpg01 == 1,]
mpg.bad <- Auto[Auto$mpg01 == 0,]
cat("Number of fuel_efficient cars:", dim(mpg.good)[1])
```

```
## Number of fuel_efficient cars: 196
```

```
cat("Number of fuel_inefficient cars:", dim(mpg.bad)[1])
```

```
## Number of fuel_inefficient cars: 196
```

```
auto.train <- rbind(mpg.good[1:150,],mpg.bad[1:150,])
auto.test <- rbind(mpg.good[151:dim(mpg.good)[1],],
                   mpg.bad[151:dim(mpg.bad)[1],])
n_good <- 150
n_bad <- 150

#Prior: relative sample size in train data
p_good <- n_good/300
p_bad <- n_bad/300
```

We proceed to perform Linear Discriminant Analysis on the training data in order to predict **mgp01** using 7 variables that seem most associated with **mgp01**, which are **cylinders**, **displacement**, **horsepower**, **weight**, **acceleration**, **year**, and **origin**. We start by calculating sample mean vectors **Mean_good** and **Mean_bad** that contain means of 7 factors of fuel-efficient and fuel-inefficient cars.

```
Mean_good <- colMeans(auto.train[auto.train$mpg01 == 1,2:8])
Mean_bad <- colMeans(auto.train[auto.train$mpg01 == 0,2:8])

rbind(Mean_good,Mean_bad)
```

```
##           cylinders displacement horsepower   weight acceleration     year
## Mean_good      4.14     112.5767   78.45333 2298.427     16.44400 76.31333
## Mean_bad       6.84     281.3067  133.16667 3679.427     14.42067 73.15333
##             origin
## Mean_good 2.006667
## Mean_bad  1.140000
```

Then, we calculate the pooled sample covariance of those 7 variables in the training set **auto.train** and save the result into **s_pooled**.

```
S_good <- cov(auto.train[auto.train$mpg01 == 1,2:8])
S_bad <- cov(auto.train[auto.train$mpg01 == 0,2:8])

S_pooled <- ((n_good-1)*S_good+(n_bad-1)*S_bad)/(n_good+n_bad-2)
S_pooled
```

```
##                cylinders displacement   horsepower      weight
## cylinders      1.2087919    67.212248    23.800268    482.4875
## displacement  67.2122483  4779.400207  1727.842685  33406.9009
## horsepower    23.8002685  1727.842685   915.442975  12862.1118
## weight       482.4875168 33406.900895 12862.111812 332214.7698
## acceleration  -1.2913691   -92.937103   -54.398687   -429.8233
## year          -0.0466443    -9.470324   -15.446779    139.1649
## origin        -0.2375168   -17.998378    -3.600515   -125.1187
##              acceleration        year       origin
## cylinders      -1.2913691  -0.0466443   -0.2375168
## displacement  -92.9371029  -9.4703244  -17.9983781
## horsepower    -54.3986868 -15.4467785   -3.6005145
## weight       -429.8232841 139.1648770 -125.1187472
## acceleration    7.1809246   1.1854251    0.1692013
## year            1.1854251   7.6233110   -0.1125280
## origin          0.1692013  -0.1125280    0.4397763
```

Here, we calculate the intercepts of the Linear Discriminant Function.

```
S_inv <- solve(S_pooled)
alpha_good <- -0.5* t(Mean_good) %*% S_inv %*% Mean_good + log(p_good)
alpha_bad <- -0.5* t(Mean_bad) %*% S_inv %*% Mean_bad + log(p_bad)

alpha_auto <- c(alpha_good,alpha_bad)
alpha_auto
```

```
## [1] -548.1675 -518.6668
```

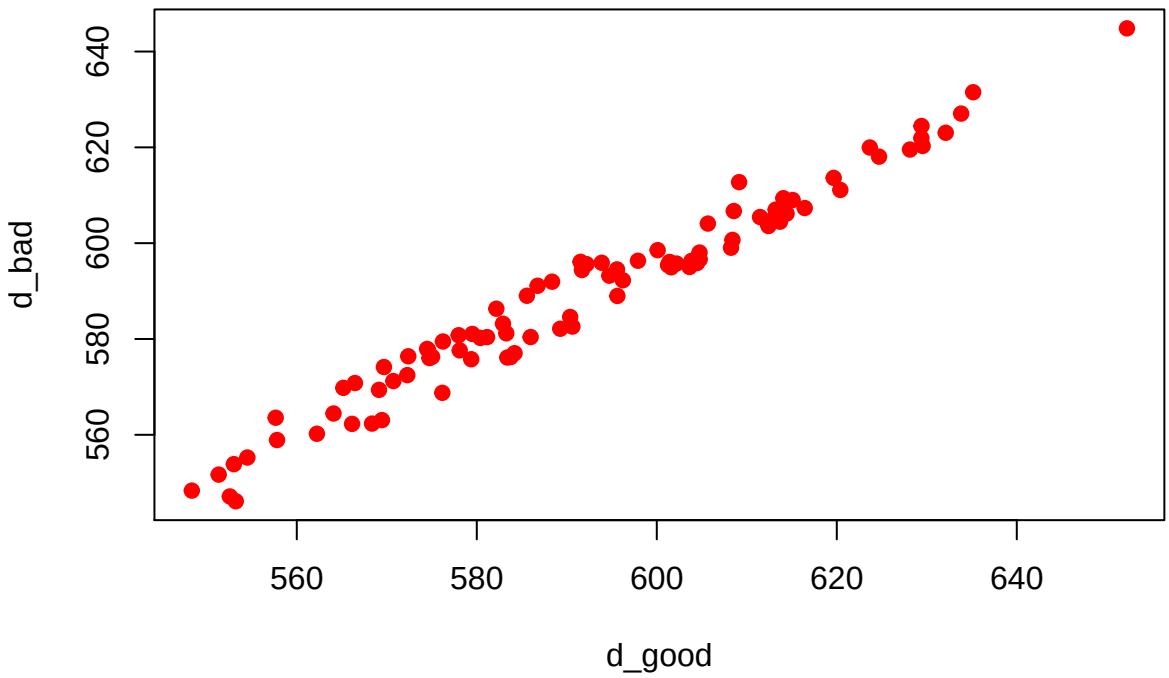
Next, we calculate the slope coefficients of each variable that seem most associated with **mpg01** for both good mpg vehicles and bad mpg vehicles.

```
beta_good <- S_inv %*% Mean_good
beta_bad <- S_inv %*% Mean_bad

beta_auto <- cbind(beta_good,beta_bad)
beta_auto
```

```
##                     [,1]        [,2]
## cylinders     9.19953837 10.40200945
## displacement -0.09750716 -0.08943426
## horsepower    1.25357824  1.20629915
## weight       -0.03826461 -0.03481064
## acceleration  7.78154223  7.77464075
## year         12.04814939 11.48531725
## origin        5.00653203  4.46989416
```

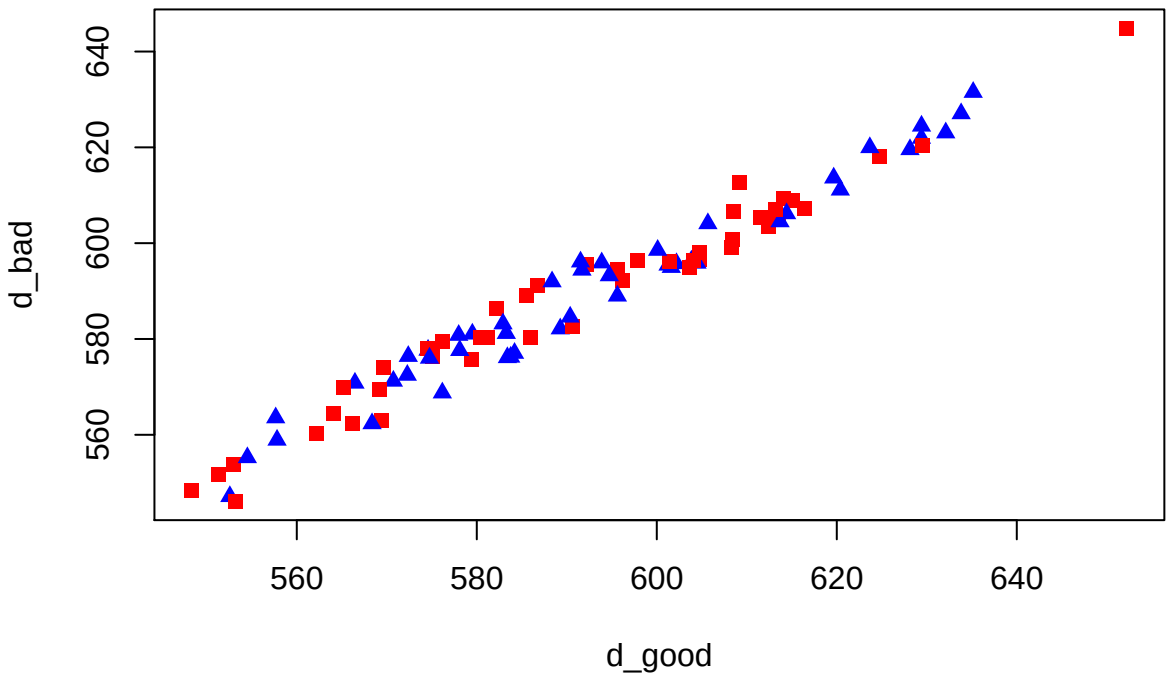
After calculating the intercepts and slopes necessary for our linear discriminant function, we plot out the function values for each observation in our test dataset. The following scatterplots are our results. For each of the 92 test records $\underline{x}$, we plot $\widehat{d}_k^L(\underline{x})$ for $k = 1, 2$. These are plotted on axes for good mpg ($k = 1$) and bad mpg ($k = 2$). There is a very strong positive linear relationship between the two values and there does not seem to be any clear clusters of data in the first scatterplot.

```
prediction <- c()
d_good_vec <- c()
d_bad_vec <- c()
d_virginica_vec <- c()
label <- c(1, 0)

for(i in 1:nrow(auto.test)){
    #Read an observation in test data
    x <- t(auto.test[i,2:8])

    #Calculate linear discriminant functions for each
    d_good <- alpha_good + t(beta_good) %*% x
    d_bad <- alpha_bad + t(beta_bad) %*% x

    #Classify the observation to the class with highest function value
    d_vec <- c(d_good, d_bad)
    prediction <- append(prediction, label[which.max( d_vec )])

    d_good_vec <- append(d_good_vec, d_good)
    d_bad_vec <- append(d_bad_vec, d_bad)
}

#Combine the predicted results to the test dataset.
auto.test$prediction <- prediction

plot(x = d_good_vec, y = d_bad_vec, xlab = "d_good", ylab = "d_bad",
              col="red", pch=19)
```



In the second scatterplot, we color the records according to their true labels: red square for good mpg cars and blue triangle for bad mpg cars. By differentiating two different groups with different shapes and colors, we can observe how the two different groups might cluster together. There does not seem to be any clear clusters between the two true groups of good and bad mpg cars. Both observations with true good mpg and true bad mpg all seem to overlap one another in a positive linear relationship. It is difficult to determine which observations are in the good mpg group and which are in the bad mpg group without differentiating the points.

```
plot(x = d_good_vec, y = d_bad_vec,
             xlab = "d_good", ylab = "d_bad",
             col=c("red","blue"), pch=c(15,17))
```



The table below shows the results of our linear discriminant analysis. It includes all data of the original Auto test set and the predictions that our linear discriminant analysis make. A prediction of 1 indicates that the vehicle has good mpg, while a prediction of 0 indicates that the vehicle has bad mpg. We also find the number of correct predictions of good mpg vehicles and bad mpg vehicles and store them into **good_true** and **bad_true**.

```
good_true <- sum((auto.test$mpg01 == 1) & (auto.test$prediction == 1))
bad_true <- sum((auto.test$mpg01 == 0) & (auto.test$prediction == 0))
auto.test
```

```
##      mpg cylinders displacement horsepower weight acceleration year origin
## 347 32.3         4           97         67   2065         17.8   81      3
## 348 37.0         4           85         65   1975         19.4   81      3
## 349 37.7         4           89         62   2050         17.3   81      3
## 350 34.1         4           91         68   1985         16.0   81      3
## 351 34.7         4          105         63   2215         14.9   81      1
## 352 34.4         4           98         65   2045         16.2   81      1
## 353 29.9         4           98         65   2380         20.7   81      1
## 354 33.0         4          105         74   2190         14.2   81      2
## 356 33.7         4          107         75   2210         14.4   81      3
## 357 32.4         4          108         75   2350         16.8   81      3
## 358 32.9         4          119        100   2615         14.8   81      3
## 359 31.6         4          120         74   2635         18.3   81      3
## 360 28.1         4          141         80   3230         20.4   81      2
## 361 30.7         6          145         76   3160         19.6   81      2
## 362 25.4         6          168        116   2900         12.6   81      3
```

```
## 363 24.2        6         146         120   2930         13.8   81      3
## 365 26.6        8         350         105   3725         19.0   81      1
## 368 28.0        4         112          88   2605         19.6   82      1
## 369 27.0        4         112          88   2640         18.6   82      1
## 370 34.0        4         112          88   2395         18.0   82      1
## 371 31.0        4         112          85   2575         16.2   82      1
## 372 29.0        4         135          84   2525         16.0   82      1
## 373 27.0        4         151          90   2735         18.0   82      1
## 374 24.0        4         140          92   2865         16.4   82      1
## 375 36.0        4         105          74   1980         15.3   82      2
## 376 37.0        4          91          68   2025         18.2   82      3
## 377 31.0        4          91          68   1970         17.6   82      3
## 378 38.0        4         105          63   2125         14.7   82      1
## 379 36.0        4          98          70   2125         17.3   82      1
## 380 36.0        4         120          88   2160         14.5   82      3
## 381 36.0        4         107          75   2205         14.5   82      3
## 382 34.0        4         108          70   2245         16.9   82      3
## 383 38.0        4          91          67   1965         15.0   82      3
## 384 32.0        4          91          67   1965         15.7   82      3
## 385 38.0        4          91          67   1995         16.2   82      3
## 386 25.0        6         181         110   2945         16.4   82      1
## 387 38.0        6         262          85   3015         17.0   82      1
## 388 26.0        4         156          92   2585         14.5   82      1
## 390 32.0        4         144          96   2665         13.9   82      3
## 391 36.0        4         135          84   2370         13.0   82      1
## 392 27.0        4         151          90   2950         17.3   82      1
## 393 27.0        4         140          86   2790         15.6   82      1
## 394 44.0        4          97          52   2130         24.6   82      2
## 395 32.0        4         135          84   2295         11.6   82      1
## 396 28.0        4         120          79   2625         18.6   82      1
## 397 31.0        4         119          82   2720         19.4   82      1
## 230 16.0        8         400         180   4220         11.1   77      1
## 231 15.5        8         350         170   4165         11.4   77      1
## 232 15.5        8         400         190   4325         12.2   77      1
## 233 16.0        8         351         149   4335         14.5   77      1
## 242 22.0        6         146          97   2815         14.5   77      3
## 243 21.5        4         121         110   2600         12.8   77      2
## 244 21.5        3          80         110   2720         13.5   77      3
## 250 19.9        8         260         110   3365         15.5   78      1
## 251 19.4        8         318         140   3735         13.2   78      1
## 252 20.2        8         302         139   3570         12.8   78      1
## 253 19.2        6         231         105   3535         19.2   78      1
## 254 20.5        6         200          95   3155         18.2   78      1
## 255 20.2        6         200          85   2965         15.8   78      1
## 257 20.5        6         225         100   3430         17.2   78      1
## 258 19.4        6         232          90   3210         17.2   78      1
## 259 20.6        6         231         105   3380         15.8   78      1
## 260 20.8        6         200          85   3070         16.7   78      1
## 261 18.6        6         225         110   3620         18.7   78      1
## 262 18.1        6         258         120   3410         15.1   78      1
## 263 19.2        8         305         145   3425         13.2   78      1
## 264 17.7        6         231         165   3445         13.4   78      1
## 265 18.1        8         302         139   3205         11.2   78      1
## 266 17.5        8         318         140   4080         13.7   78      1
```

```
## 271 21.1         4          134         95   2515         14.8   78      3
## 275 20.3         5          131        103   2830         15.9   78      2
## 276 17.0         6          163        125   3140         13.6   78      2
## 277 21.6         4          121        115   2795         15.7   78      2
## 278 16.2         6          163        133   3410         15.8   78      2
## 281 21.5         6          231        115   3245         15.4   79      1
## 282 19.8         6          200         85   2990         18.2   79      1
## 283 22.3         4          140         88   2890         17.3   79      1
## 284 20.2         6          232         90   3265         18.2   79      1
## 285 20.6         6          225        110   3360         16.6   79      1
## 286 17.0         8          305        130   3840         15.4   79      1
## 287 17.6         8          302        129   3725         13.4   79      1
## 288 16.5         8          351        138   3955         13.2   79      1
## 289 18.2         8          318        135   3830         15.2   79      1
## 290 16.9         8          350        155   4360         14.9   79      1
## 291 15.5         8          351        142   4054         14.3   79      1
## 292 19.2         8          267        125   3605         15.0   79      1
## 293 18.5         8          360        150   3940         13.0   79      1
## 317 19.1         6          225         90   3381         18.7   80      1
## 364 22.4         6          231        110   3415         15.8   81      1
## 366 20.2         6          200         88   3060         17.1   81      1
## 367 17.6         6          225         85   3465         16.6   81      1
## 389 22.0         6          232        112   2835         14.7   82      1
##                                      name mpg01 prediction
## 347                               subaru     1          1
## 348                   datsun 210 mpg     1          1
## 349                    toyota tercel     1          1
## 350                      mazda glc 4     1          1
## 351                plymouth horizon 4     1          1
## 352                    ford escort 4w     1          1
## 353                    ford escort 2h     1          1
## 354                  volkswagen jetta     1          1
## 356                     honda prelude     1          1
## 357                    toyota corolla     1          1
## 358                      datsun 200sx     1          1
## 359                         mazda 626     1          1
## 360           peugeot 505s turbo diesel     1          1
## 361                       volvo diesel     1          1
## 362                  toyota cressida     1          1
## 363                 datsun 810 maxima     1          1
## 365              oldsmobile cutlass ls     1          0
## 368                 chevrolet cavalier     1          1
## 369           chevrolet cavalier wagon     1          1
## 370           chevrolet cavalier 2-door     1          1
## 371         pontiac j2000 se hatchback     1          1
## 372                    dodge aries se     1          1
## 373                   pontiac phoenix     1          1
## 374               ford fairmont futura     1          1
## 375              volkswagen rabbit l     1          1
## 376               mazda glc custom l     1          1
## 377                 mazda glc custom     1          1
## 378            plymouth horizon miser     1          1
## 379                    mercury lynx l     1          1
## 380                 nissan stanza xe     1          1
```

```
## 381                        honda accord     1          1
## 382                      toyota corolla     1          1
## 383                        honda civic     1          1
## 384                   honda civic (auto)     1          1
## 385                        datsun 310 gx     1          1
## 386                buick century limited     1          1
## 387    oldsmobile cutlass ciera (diesel)     1          1
## 388           chrysler lebaron medallion     1          1
## 390                      toyota celica gt     1          1
## 391                     dodge charger 2.2     1          1
## 392                      chevrolet camaro     1          1
## 393                       ford mustang gl     1          1
## 394                             vw pickup     1          1
## 395                         dodge rampage     1          1
## 396                           ford ranger     1          1
## 397                            chevy s-10     1          1
## 230                pontiac grand prix lj     0          0
## 231          chevrolet monte carlo landau     0          0
## 232                      chrysler cordoba     0          0
## 233                      ford thunderbird     0          0
## 242                            datsun 810     0          1
## 243                              bmw 320i     0          1
## 244                            mazda rx-4     0          1
## 250    oldsmobile cutlass salon brougham     0          0
## 251                        dodge diplomat     0          0
## 252                 mercury monarch ghia     0          0
## 253                   pontiac phoenix lj     0          0
## 254                     chevrolet malibu     0          0
## 255                  ford fairmont (auto)     0          0
## 257                       plymouth volare     0          0
## 258                           amc concord     0          0
## 259                buick century special     0          0
## 260                        mercury zephyr     0          0
## 261                           dodge aspen     0          0
## 262                      amc concord d/l     0          0
## 263          chevrolet monte carlo landau     0          0
## 264       buick regal sport coupe (turbo)     0          1
## 265                           ford futura     0          0
## 266                       dodge magnum xe     0          0
## 271            toyota celica gt liftback     0          1
## 275                             audi 5000     0          1
## 276                           volvo 264gl     0          1
## 277                            saab 99gle     0          1
## 278                          peugeot 604sl     0          1
## 281                       pontiac lemans v6     0          1
## 282                       mercury zephyr 6     0          1
## 283                        ford fairmont 4     0          1
## 284                       amc concord dl 6     0          0
## 285                          dodge aspen 6     0          1
## 286            chevrolet caprice classic     0          0
## 287                       ford ltd landau     0          0
## 288                  mercury grand marquis     0          0
## 289                         dodge st. regis     0          0
## 290               buick estate wagon (sw)     0          0
```

```
## 291             ford country squire (sw)     0          0
## 292        chevrolet malibu classic (sw)     0          0
## 293 chrysler lebaron town @ country (sw)     0          0
## 317                           dodge aspen     0          0
## 364                         buick century     0          1
## 366                       ford granada gl     0          1
## 367               chrysler lebaron salon     0          0
## 389                        ford granada l     0          1
```

After predicting whether each vehicle in our test set has good or bad mpg, we sum up the total number of cars in each group and create a summary table shown below. The table includes the total number of vehicles with good mpg and bad mpg, as well as the number of good and bad mpg vehicles that are correctly and incorrectly predicted.

```
class_tab <- c(dim(mpg.good)[1]-n_good,
               dim(mpg.bad)[1]-n_bad)
class_tab <- rbind(class_tab,
       c(good_true,bad_true))
class_tab <- rbind(class_tab,class_tab[1,] - class_tab[2,])
colnames(class_tab) <- c("good","bad")
rownames(class_tab) <- c("Number Observations","Number Correct","Number Wrong")
class_tab
```

```
##                      good bad
## Number Observations   46  46
## Number Correct        45  30
## Number Wrong           1  16
```

As we see from the table above, the linear discriminant analysis we perform on our test set correctly predicts whether the car has good mpg or bad mpg for the majority of test data observations. Using the seven variables we find to be highly associated with mpg01, we correctly classify 75 of the test data observations from a total of 92. This gives our linear discriminant function an accuracy of 81.522%, or 75/92. But, our linear discriminant function predicts the good mpg vehicles more accurately than the bad mpg vehicles. The prediction result has a proportion of 45/46 correctly predicted out of all good mpg vehicles in the test set, while it only has a proportion of 30/46 correctly predicted out of all bad mpg vehicles. In addition, wrong prediction porportion of true bad mpg cars is significantly higher than that of true good mpg cars. The linear discriminant function only incorrectly classifies 1/46 true good mpg vehicles in our test set, while it incorrectly classifies 16/46 true bad mpg vehicles in our test set.

Conclusion:

First we create a binary variable, mgp01, that contains 1 if mpg contains a value above its median, and a 0 if mpg contains a value below its median. Then we create boxplots to investigate the association between mpg01 and cylinders, displacement, horsepower, weight, acceleration, and year respectively. Based on the results, we find that the boxplot of mpg01 and year shows that fuel-efficient cars whose mileage are above the median were generally made later than fuel-inefficient cars that have mileage below the median. Moreover, fuel-efficient cars generally have fewer cylinders and hence weigh less than fuel-inefficient cars. Fuel-efficient cars also have fewer displacement and horsepower but accelerate faster than fuel-inefficient cars. Hence, all of these factors seem to be very useful in predicting mpg01.

Also, we create the boxplot of mpg01 and origin and count the number of fuel-efficient and fuel-inefficient cars that were made in America, Europe, and Japan. The result shows that most fuel-inefficient cars were made in America and only 23 cars were made in Europe and Japan. Similarly, most fuel-efficient cars were made

in America, but approximately same number of such cars were made in Japan and slightly fewer cars were made in Europe. Then we perform a chi-square test of independence to test whether there is a an association between the categories of origin and mpg01. The null hypothesis is that the categories of origin and mpg01 are independent, while the alternative hypothesis is that there is an association between the categories of origin and mpg01. Since p-value is less than 0.05, we reject the null hypothesis that the categorical variables origin and mpg01 are independent, and there are clear evidences for us to conclude that there is a significant association between the categories of origin and mpg01. Hence, origin is very helpful for predicting mpg01.

Next, we split the data into a training set of size 300 and a test set of size 92. We proceed to perform Linear Discriminant Analysis on the training data in order to predict mgp01 using 7 variables that seem most associated with mgp01, which are cylinders, displacement, horsepower, weight, acceleration, year, and origin. In the scatterplots we make for each observation in test set, there does not seem to be any clear clusters between the two true groups of good and bad mpg cars. Both observations with true good mpg and true bad mpg all seem to overlap one another in a positive linear relationship. It is difficult to determine which observations are in the good mpg group and which are in the bad mpg group without differentiating the points.

We then sum up the total number of vehicles with good mpg and bad mpg, as well as the number of good and bad mpg vehicles that are correctly and incorrectly predicted. The linear discriminant analysis we perform on our test set correctly predicts whether the car has good mpg or bad mpg for the majority of test data observations. Using the seven variables we find to be highly associated with mpg01, we correctly classify 75 of the test data observations from a total of 92. This gives our linear discriminant function an accuracy of 81.522%, or 75/92. But, our linear discriminant function predicts the good mpg vehicles more accurately than the bad mpg vehicles. The prediction result has a proportion of 45/46 correctly predicted out of all good mpg vehicles in the test set, while it only has a proportion of 30/46 correctly predicted out of all bad mpg vehicles. In addition, wrong prediction proportion of true bad mpg cars is significantly higher than that of true good mpg cars. The linear discriminant function only incorrectly classifies 1/46 true good mpg vehicles in our test set, while it incorrectly classifies 16/46 true bad mpg vehicles in our test set.