# Math 189 Homework 5

Hien Bui, Cole Clisby, Cheolmin Hwang, Andrew Li, James Yu

## Intro

In this report, we utilize an automobile dataset to create a predictive model that determines if a given car's gas mileage is high or low, relative to the sample median. We accomplish this by first creating a new binary feature, then by visualizing the relationship it has with other variables. Lastly, we split the dataset into training and test data and perform Linear Discriminant Analysis (LDA) to create our model and test its accuracy.

## Body

### Data

```
library(ISLR)
auto <- ISLR::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
```

Source: The Auto dataset is presented in An Introduction to Statistical Learning with applications in R by James, Witten, Hastie, and Tibshirani (2013).

URL: https://rdrr.io/cran/ISLR/man/Auto.html (https://rdrr.io/cran/ISLR/man/Auto.html)

Description: Contains 392 observations of automobiles. There are 9 variables containing information on the specifications and name of each car.

1. mpg: in miles per gallon
2. cylinders: in number of cylinders the engine has
3. displacement: engine displacement in cubic inches
4. horsepower: in horsepower
5. weight: in pounds
6. acceleration: in number of seconds to accelerate from 0 to 60mph

7. year: in last two digits of model year (e.g. 1970 -> 70)
8. origin: 1 if American, 2 if European, 3 if Japanese
9. name: vehicle name

## Task 1

```r
mpg_med <- median(auto$mpg)
mpg_med
```

```
## [1] 22.75
```

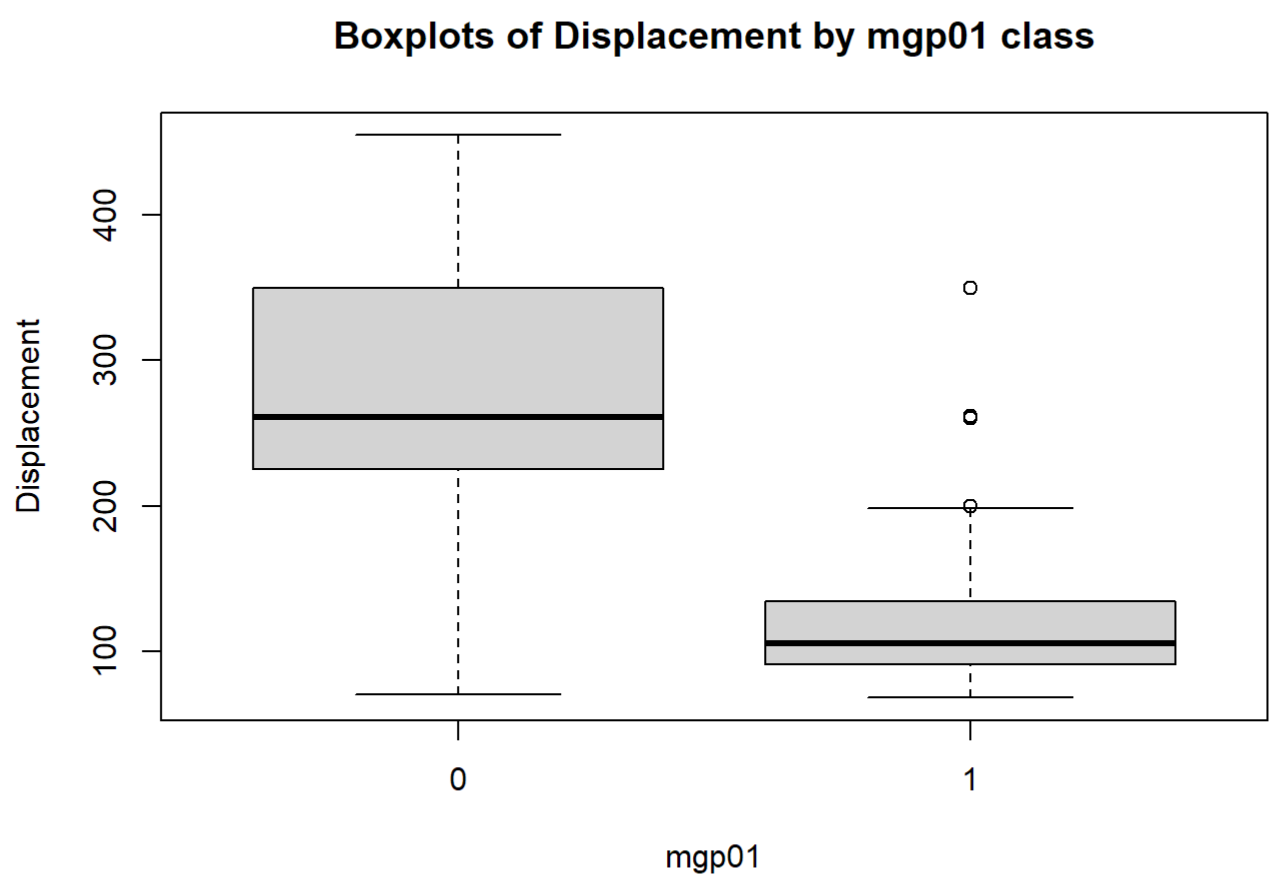
```r
mgp01 <- c()
for (val in auto$mpg) {
  if (val < mpg_med) {
    mgp01 <- append(mgp01, 0)
  } else if (val > mpg_med) {
    mgp01 <- append(mgp01, 1)
  }
}
auto <- cbind(auto, mgp01)
head(auto)
```

```
##   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 mgp01
## 1 chevrolet chevelle malibu     0
## 2         buick skylark 320     0
## 3        plymouth satellite     0
## 4             amc rebel sst     0
## 5               ford torino     0
## 6          ford galaxie 500     0
```

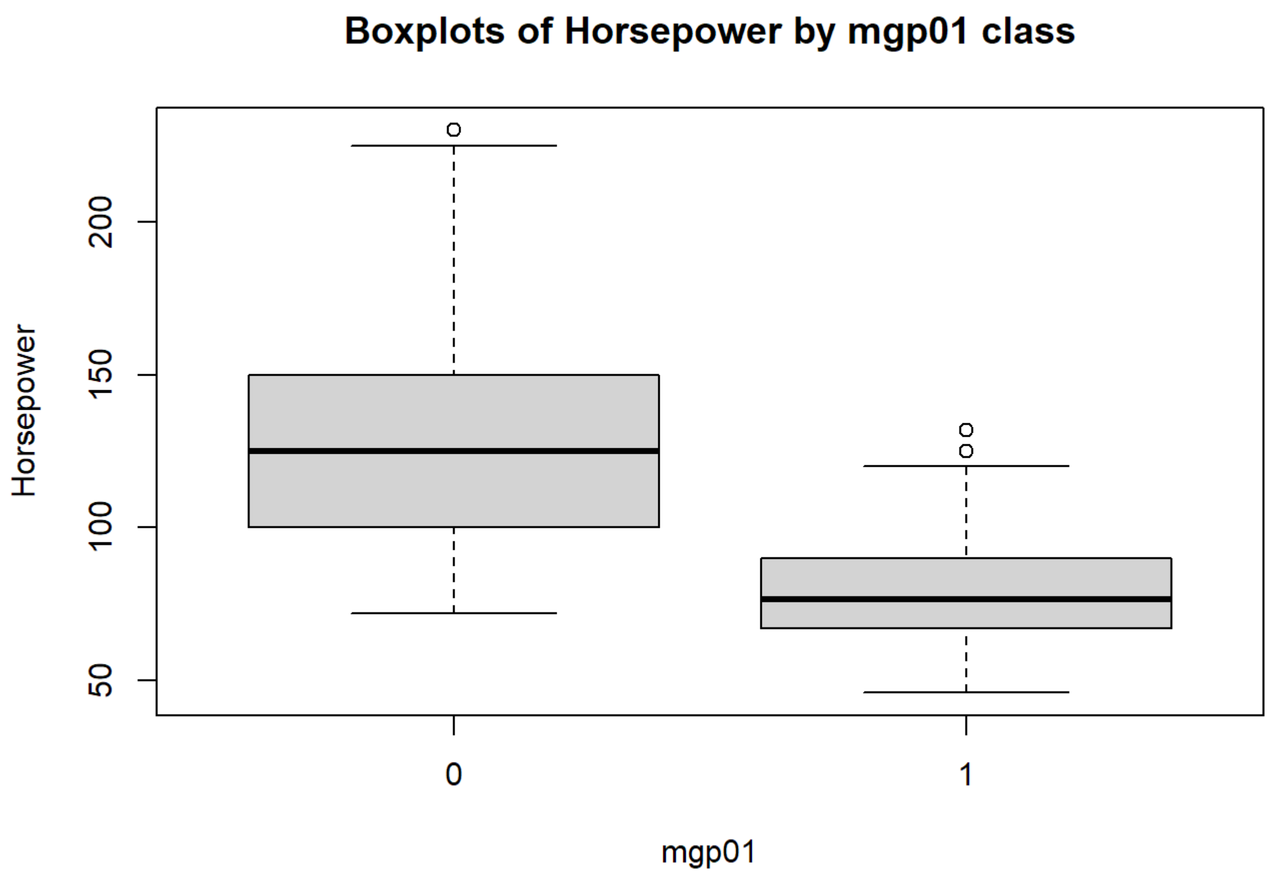
The median is an average of the two middlemost elements because we have an even number of observations. So no values will be exactly equal to the median.

## Task 2

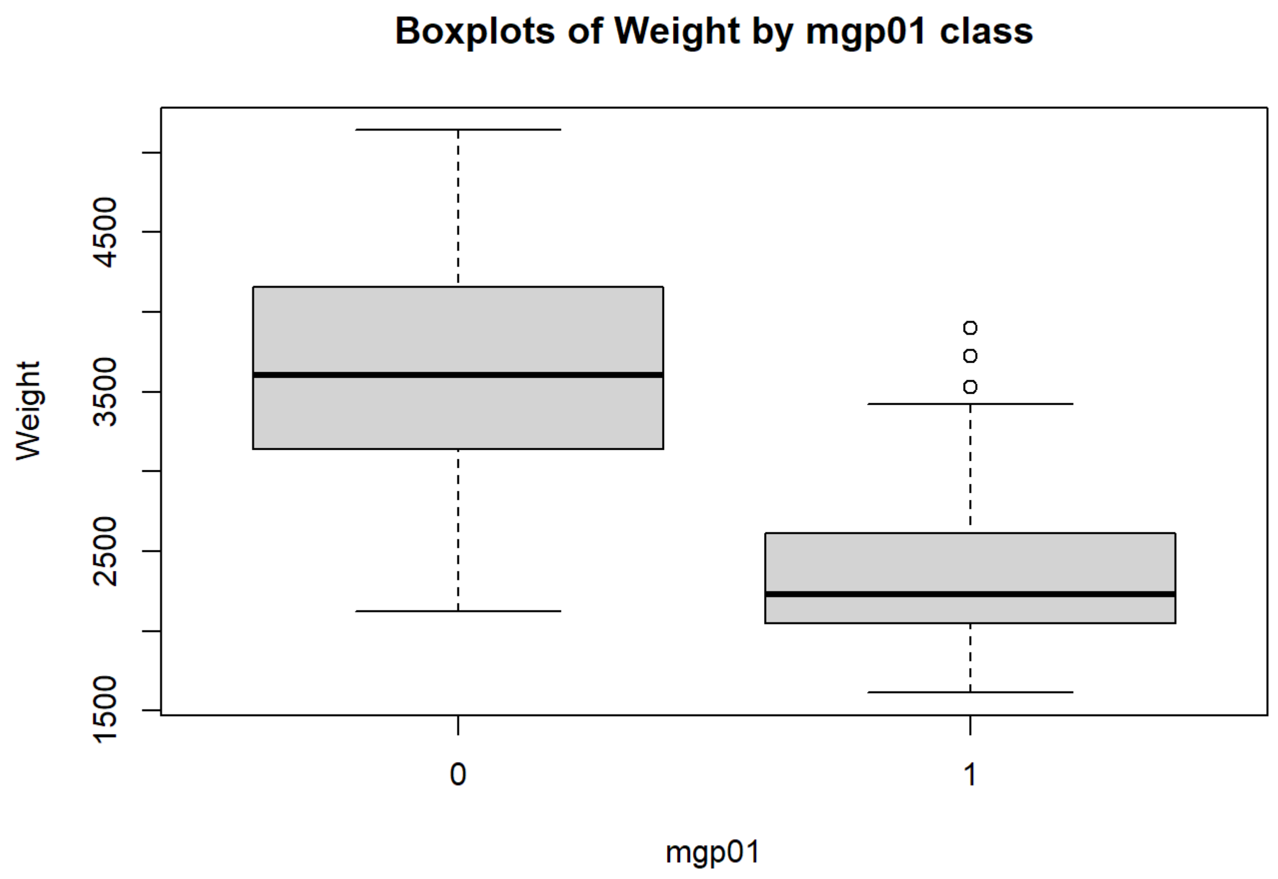
```r
boxplot(auto$displacement ~ auto$mgp01, data=auto, xlab='mgp01', ylab='Displacement')
title('Boxplots of Displacement by mgp01 class')
```



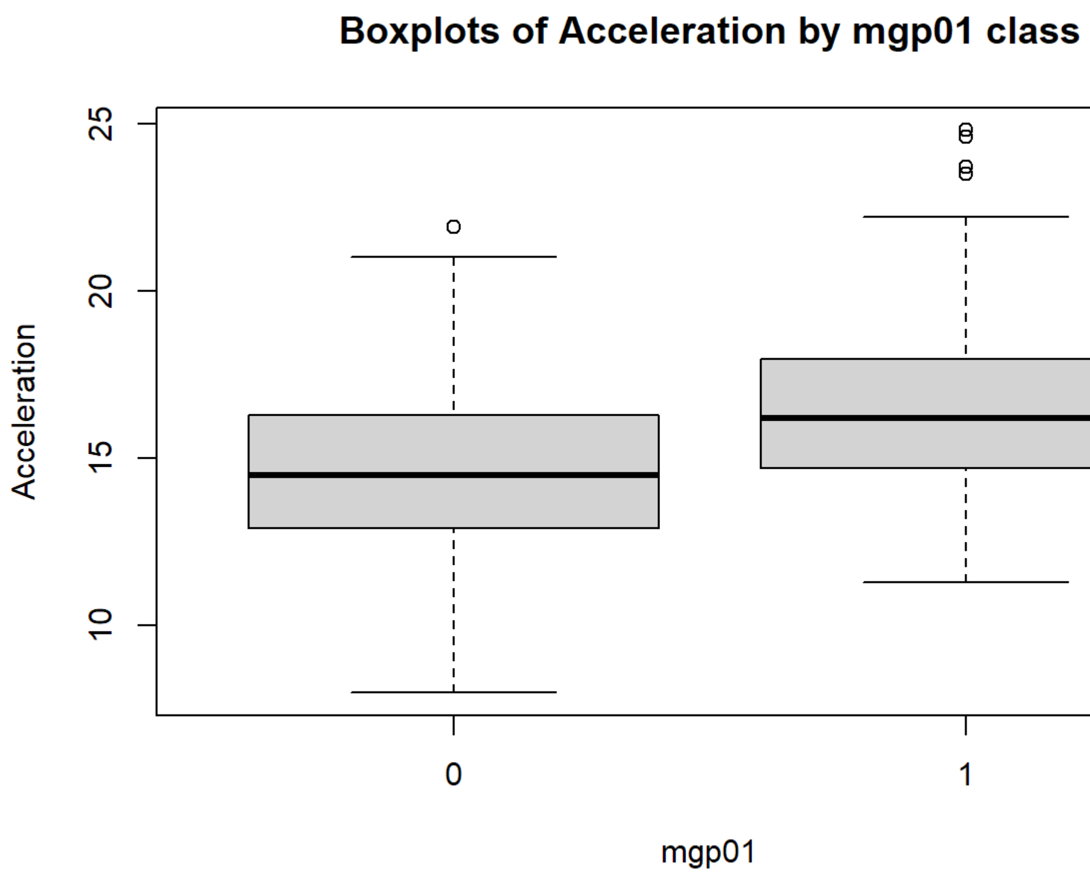
```
boxplot(auto$horsepower ~ auto$mgp01, data=auto, xlab='mgp01', ylab='Horsepower')
title('Boxplots of Horsepower by mgp01 class')
```



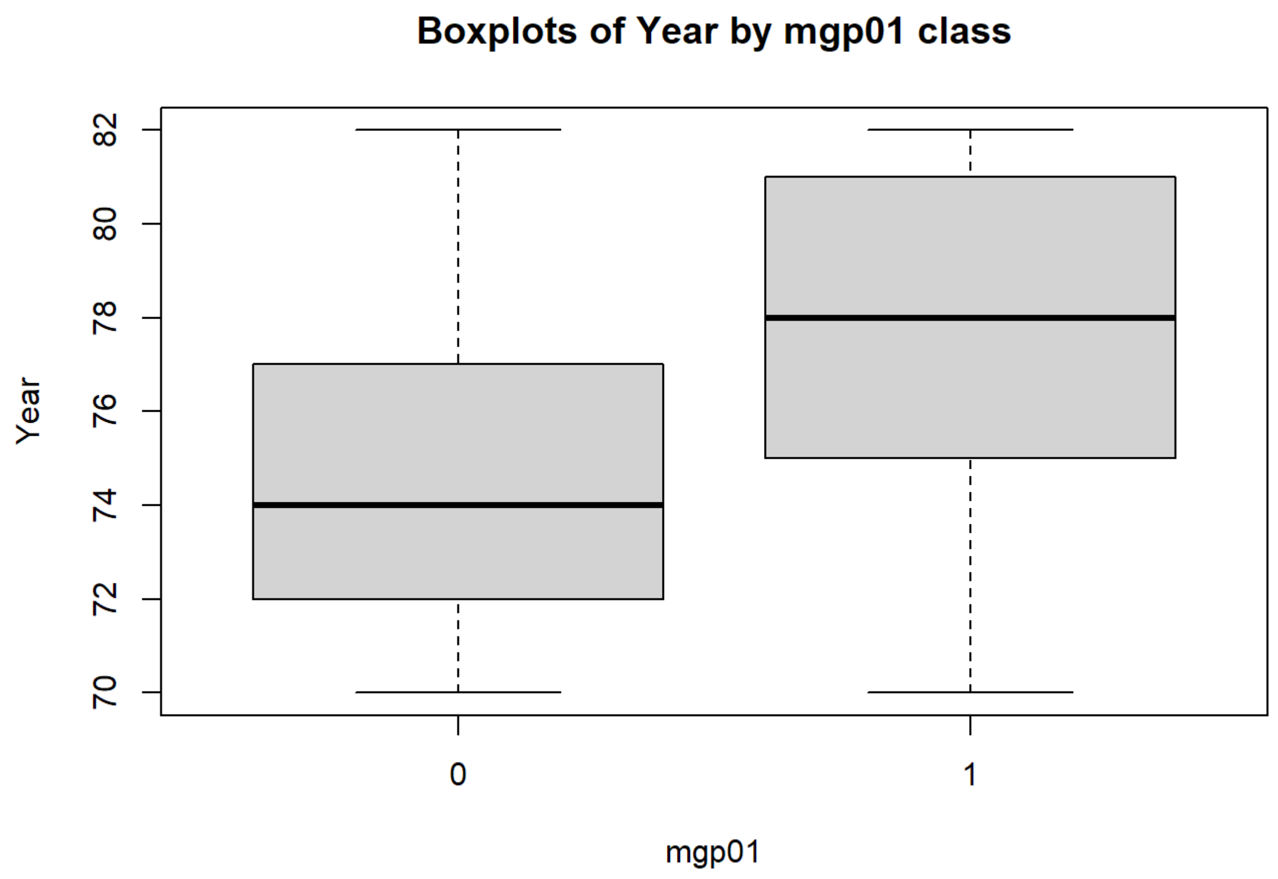
```
boxplot(auto$weight ~ auto$mgp01, data=auto, xlab='mgp01', ylab='Weight')
title('Boxplots of Weight by mgp01 class')
```



```
boxplot(auto$acceleration ~ auto$mgp01, data=auto, xlab='mgp01', ylab='Acceleration')
title('Boxplots of Acceleration by mgp01 class')
```



```
boxplot(auto$year ~ auto$mgp01, data=auto, xlab='mgp01', ylab='Year')
title('Boxplots of Year by mgp01 class')
```



When mgp01 is 1 (mpg above the median), the boxplots for displacement, horsepower, and weight are lower than the boxplots for those variables when mgp01 is 0. In contrast, the year and acceleration are higher for when mgp01 is 1, compared to when mgp01 is 0. This makes sense since cars with a smaller engine displacement, less horsepower and weight will be more efficient and have a higher miles per gallon, since the car is less powerful and weighs less. So we can get more miles per gallon as the car doesn't have to drive around the extra components that make it faster.

It also makes sense that the year on average would be higher (i.e. manufactured later) since cars can get more efficient over time as new manufacturing processes and optimizations can be made.

The boxplots for acceleration between the two groups of mgp01 are pretty similar, but the 1 group is slightly higher and has some high value outliers, meaning that it takes longer to accelerate from 0 to 60mph, which matches with the other information we visualized and discussed above.

We can still visualize the relationship between mgp01 and some of the categorical variables like cylinders and origin, even though we won't be able to use them in our LDA.

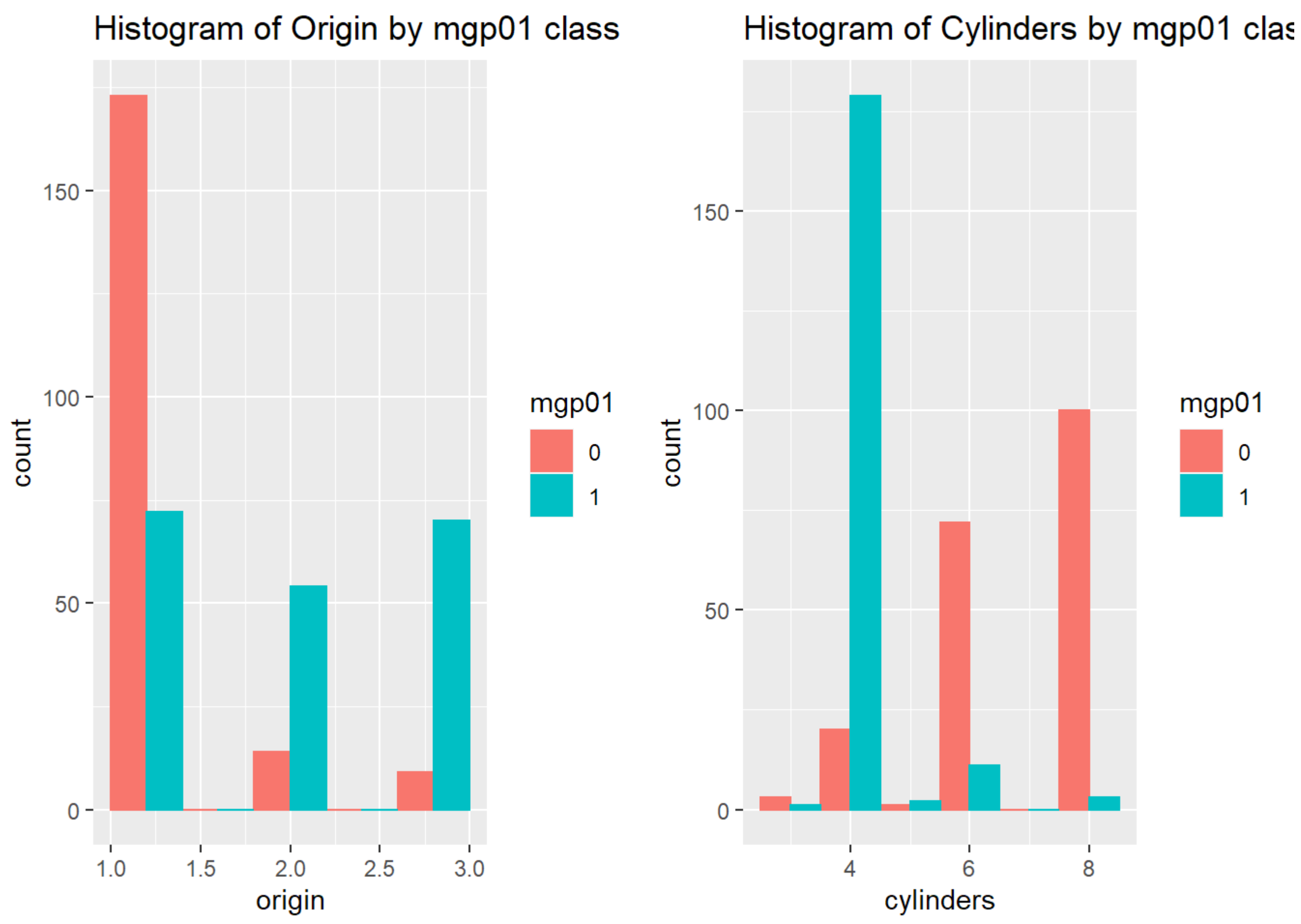
```
require(gridExtra)
```

```
## Loading required package: gridExtra
```

```r
library(ggplot2)

temp_auto <- auto
temp_auto$mgp01 <- as.factor(temp_auto$mgp01)
gp1 <- ggplot(temp_auto, aes(x=origin, fill=mgp01, color=mgp01)) + geom_histogram(position='dodg
e', bins=6) + ggtitle('Histogram of Origin by mgp01 class')
gp2 <- ggplot(temp_auto, aes(x=cylinders, fill=mgp01, color=mgp01)) + geom_histogram(position='d
odge', bins=6) + ggtitle('Histogram of Cylinders by mgp01 class')

grid.arrange(gp1, gp2, ncol=2)
```



We can see that the 0 class is more likely to have origin 1, which is American made. The 1 class is more likely to have origins 2 or 3, which are European or Japanese made. This shows how the American made cars in the sample are less fuel efficient, since more of them are in the 0 class, meaning their miles per gallon is below the median.

The 1 class is more likely to have 4 cylinders, with the 0 class having a much higher proportion of 6 and 8 cylinder engines. This indicates that the more cylinders the engine has, the less efficient it can be, causing the miles per gallon to decrease.

## Task 3

Ultimately, we chose to use displacement, horsepower, weight, and year since they showed the strongest relationship to mgp01 in our visualizations. We didn't use any of the categorical/ordinal variables like origin, cylinders, or name since we are performing LDA and it assumes that the variables used will be continuous.

```
auto.train <- subset(auto[sample(nrow(auto), 300), ], select=c('displacement', 'horsepower', 'we
ight', 'year', 'mgp01'))
auto.test <- subset(auto[sample(nrow(auto), 92), ], select=c('displacement', 'horsepower', 'weig
ht', 'year', 'mgp01'))
nrow(auto.train)
```

```
## [1] 300
```

```
nrow(auto.test)
```

```
## [1] 92
```

We split the data into a training set of size 300 and test set of size 92, using random sampling.

## Task 4

Assumptions made:

1. The data from each group has a common mean vector that is equal to the population mean vector
2. Homoskedasticity: the data from all groups has a common covariance matrix that does not depend on the group
3. Independence: the observations are independently sampled
4. Normality: the data are multivariate normally distributed

```
# Prior probabilities - based on relative sample size in training data
n_0 <- sum(auto.train$mgp01 == 0)
n_1 <- sum(auto.train$mgp01 == 1)
p_0 <- n_0 / 300
p_1 <- n_1 / 300
p_0
```

```
## [1] 0.51
```

```
p_1
```

```
## [1] 0.49
```

We chose to use the training data prior probabilities since we believe that the relative sample sizes in the training data are close to the relative population sizes.

```
# Sample mean vectors for each group
num_feat <- 4
mean_0 <- colMeans(auto.train[auto.train$mgp01 == 0, 1:num_feat])
mean_1 <- colMeans(auto.train[auto.train$mgp01 == 1, 1:num_feat])
mean_0
```

```
## displacement   horsepower       weight         year
##    267.75163    127.81046   3586.79085     74.35948
```

```
mean_1
```

```
## displacement   horsepower       weight         year
##    116.95578     78.19728   2337.95238     77.53741
```

```
# Pooled sample covariance for each group
S_0 <- cov(auto.train[auto.train$mgp01 == 0, 1:num_feat])
S_1 <- cov(auto.train[auto.train$mgp01 == 1, 1:num_feat])
S_pooled <- ((n_0-1)*S_0 + (n_1-1)*S_1) / (n_0 + n_1 - 2)
S_pooled
```

```
##              displacement  horsepower      weight      year
## displacement   5043.97827  1629.30899  33417.0864 -18.50787
## horsepower     1629.30899   810.88182  11651.6386 -18.07772
## weight        33417.08639 11651.63864 316350.7919  99.21230
## year            -18.50787   -18.07772     99.2123  10.87172
```

```
# Intercepts of Linear Discriminant Function
S_inv <- solve(S_pooled)
alpha_0 <- -0.5 * t(mean_0) %*% S_inv %*% mean_0 + log(p_0)
alpha_1 <- -0.5 * t(mean_1) %*% S_inv %*% mean_1 + log(p_1)
alpha_auto <- c(alpha_0, alpha_1)
alpha_auto
```

```
## [1] -299.9862 -315.3128
```

```
# Slope intercepts of LD function
beta_0 <- S_inv %*% mean_0
beta_1 <- S_inv %*% mean_1
beta_auto <- cbind(beta_0, beta_1)
beta_auto
```

```
##                      [,1]         [,2]
## displacement -0.053713808 -0.065501835
## horsepower    0.475497892  0.497829152
## weight       -0.002873827 -0.006506172
## year          7.565166714  7.907691875
```

## Task 5

```r
prediction <- c()
d_0_vec <- c()
d_1_vec <- c()
label <- c(0, 1)

for (i in 1:nrow(auto.test)) {
  # Gets the features for individual test observation
  x <- t(auto.test[i, 1:num_feat])

  # Calculates LD functions for each group
  d_0 <- alpha_0 + t(beta_0) %*% x
  d_1 <- alpha_1 + t(beta_1) %*% x

  # Classify the observation to the class w/ highest val
  d_vec <- c(d_0, d_1)
  prediction <- append(prediction, label[which.max(d_vec)])
  d_0_vec <- append(d_0_vec, d_0)
  d_1_vec <- append(d_1_vec, d_1)
}

auto.test$prediction <- prediction

# Calculating accuracy for each class
zero_true <- sum((auto.test$mgp01 == 0) & (auto.test$prediction == 0))
one_true <- sum((auto.test$mgp01 == 1) & (auto.test$prediction == 1))

# Creates table of results
n_0_full <- sum(auto$mgp01 == 0)
n_1_full <- sum(auto$mgp01 == 1)
class_tab <- c(sum(auto.test$mgp01==0), sum(auto.test$mgp01==1))
class_tab <- rbind(class_tab, c(zero_true, one_true))
class_tab <- rbind(class_tab, class_tab[1,] - class_tab[2,])
colnames(class_tab) <- c('Class 0', 'Class 1')
rownames(class_tab) <- c('Num Observations', 'Num Correct', 'Num Wrong')
class_tab
```

```
##                  Class 0 Class 1
## Num Observations      46      46
## Num Correct           37      45
## Num Wrong              9       1
```

```r
acc <- (zero_true + one_true) / nrow(auto.test)
acc
```

```
## [1] 0.8913043
```

We get around 0.85 to 0.90 accuracy on the test set. Because our test set size is only 92 and we are using a random sample to get our training and test sets, the accuracy can vary somewhat. However, generally the LDA classification performs well. Since we have two classes, if we randomly guess the class each time we would get roughly 0.5 accuracy. So our LDA classification is much better than chance.

## Conclusion

We were able to perform LDA on the Auto dataset in order to classify if a given car was above or below the median miles per gallon. We created the binary variable mgp01 and graphed its relationship with the other variables in order to determine which of the variables would be useful in our LDA. We chose to use displacement, horsepower, weight, and year since they looked to have the strongest relationship with mgp01. We used random sampling to create a training and test set, and performed LDA on the training set. Using our test set, we got roughly 85-90% accuracy, with some variance due to our sets being random samples.