11. In this problem, you will develop a model to predict whether a given car gets high or low gas mileage based on the Auto data set. *To get the Auto data set, first install ISLR package and load the package (I know this is the same data set we used for Homework 1 but this time use the data set inside the ISLR package. Missing data are already cleaned in this data set).*

(a) Create a binary variable, mpg01, that contains a 1 if mpg contains a value above its median, and a 0 if mpg contains a value below its median. You can compute the median using the median() function. Note you may find it helpful to use the data.frame() function to create a single data set containing both mpg01 and the other Auto variables.

#part a)

```
head(Auto)
summary(Auto)
attach(Auto)
set.seed(1)
median(Auto$mpg) #median of mpg is 22.75
df.auto <- data.frame(Auto)
mpg.median <- median(df.auto$mpg)
df.auto$mpg01 <- ifelse(df.auto$mpg > mpg_median, 1, 0)
summary(df.auto)
```

```
#part a)

head(Auto)
summary(Auto)
attach(Auto)
set.seed(1)
median(Auto$mpg) #median of mpg is 22.75
df.auto <- data.frame(Auto)
mpg.median <- median(df.auto$mpg)
df.auto$mpg01 <- ifelse(df.auto$mpg > mpg_median, 1, 0)
summary(df.auto)
```

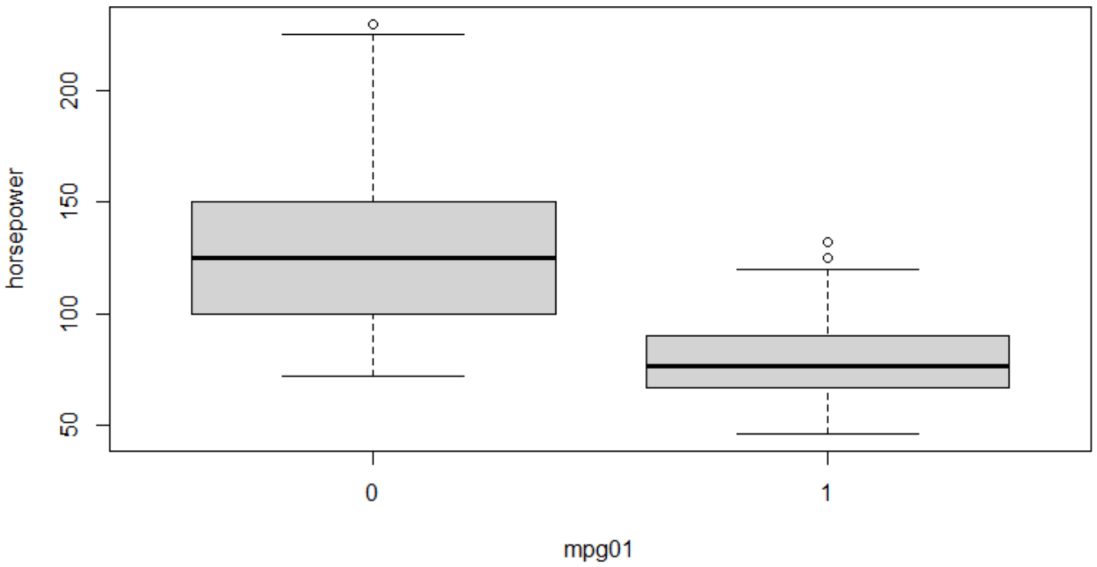
(b) Explore the data graphically in order to investigate the association between mpg01 and the other features. Which of the other features seem most likely to be useful in predicting mpg01? Scatterplots and boxplots may be useful tools to answer this question. Describe your findings.

```
#library(corrplot)
#cor(Auto[, -9])
#corrplot::corrplot.mixed(cor(Auto[, -9]), upper="circle")

fac.cylinders <- as.factor(cylinders)
cylinders <- as.factor(cylinders)
mpg.df$cylinders <- as.factor(cylinders)
summary(mpg.df)

fac.year <- as.factor(year)
year <- as.factor(year)
mpg.df$year <- as.factor(year)
summary(mpg.df)
```

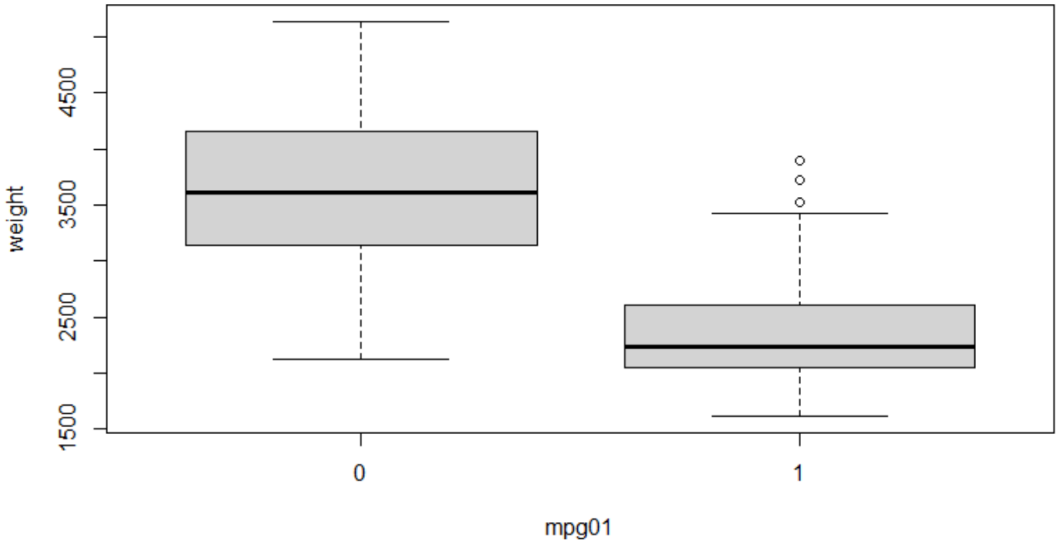
```
fac.origin <- as.factor(origin)
origin <- as.factor(origin)
mpg.df$origin <- as.factor(origin)
summary(mpg.df)
mpg.df

#boxplots:
boxplot(horsepower ~ mpg01, data = mpg.df)
boxplot(weight ~ mpg01, data = mpg.df)
boxplot(displacement ~ mpg01, data = mpg.df)
boxplot(acceleration ~ mpg01, data = mpg.df)

#unique(mpg.df$cylinders)
brplt.cylinder <- table(mpg.df$cylinder, mpg.df$mpg01)
barplot(brplt.cylinder, col = c("lightblue", "orange", "lightgreen", "violet","gold"),
      main="MPG by Number of Cylinders",
      beside=TRUE, legend = rownames(brplt.cylinder))
#names=("3 cyl", "4 cyl", "5 cyl", "6 cyl", "8 cyl")

#unique(mpg.df$year)
brplt.year <- table(mpg.df$year, mpg.df$mpg01)
barplot(brplt.year, col = c("lightblue", "orange", "lightgreen", "violet","gold"),
      main="MPG by Year",
      beside=TRUE, legend = rownames(brplt.year))

brplt.origin <- table(mpg.df$origin, mpg.df$mpg01)
barplot(brplt.origin, col = c("lightblue", "orange", "lightgreen"),
      main="MPG by Country of Origin",
      beside=TRUE, legend = rownames(brplt.origin))
#names=("United States","Europe","Japan")
```

Horsepower, weight, displacement (the three of which are highly correlated), cylinders, year, and country of origin are predictors of MPG.

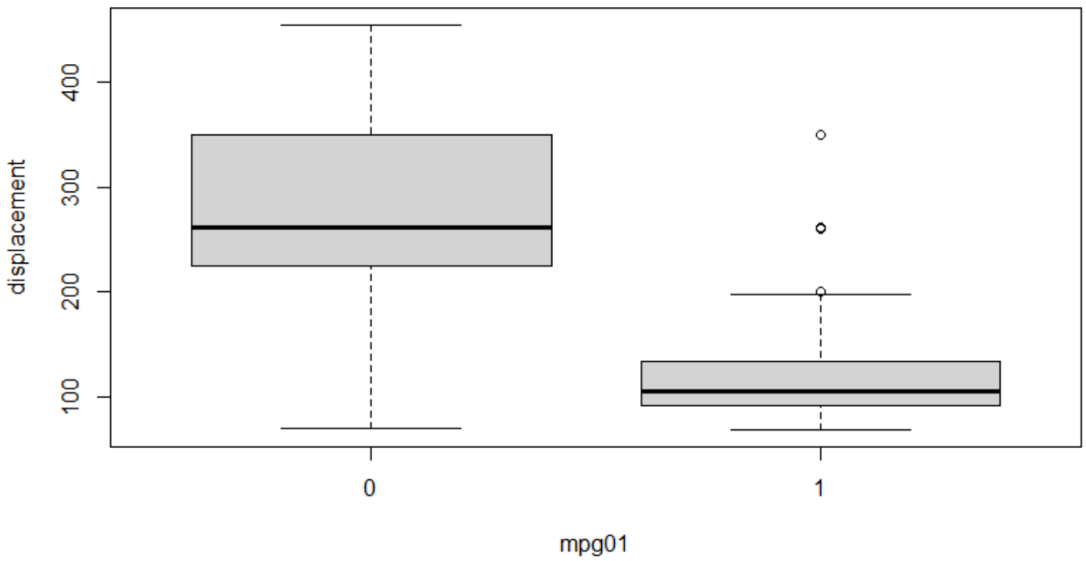
The variables of displacement, horsepower and weight display multicollinearity; these three variables of displacement, horsepower and weight have a direct relationship with each other. Horsepower is directly related to weight (the higher one goes, the higher the other goes). Displacement and weight are directly related and displacement and horsepower are directly related. The greater the weight, the more displacement and/or the higher the horsepower, the less MPG the car will have.



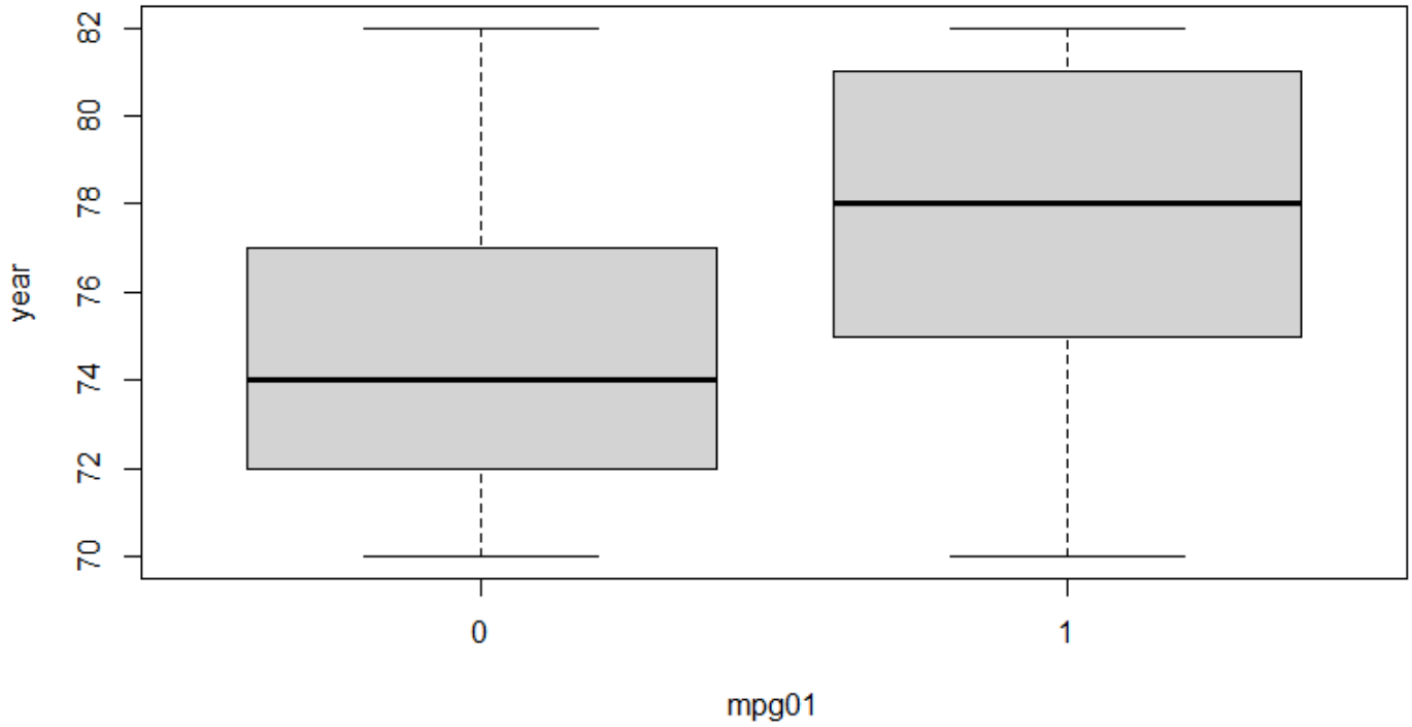
Horsepower: Cars that get less than or equal to the median mpg tend to have higher horsepower.

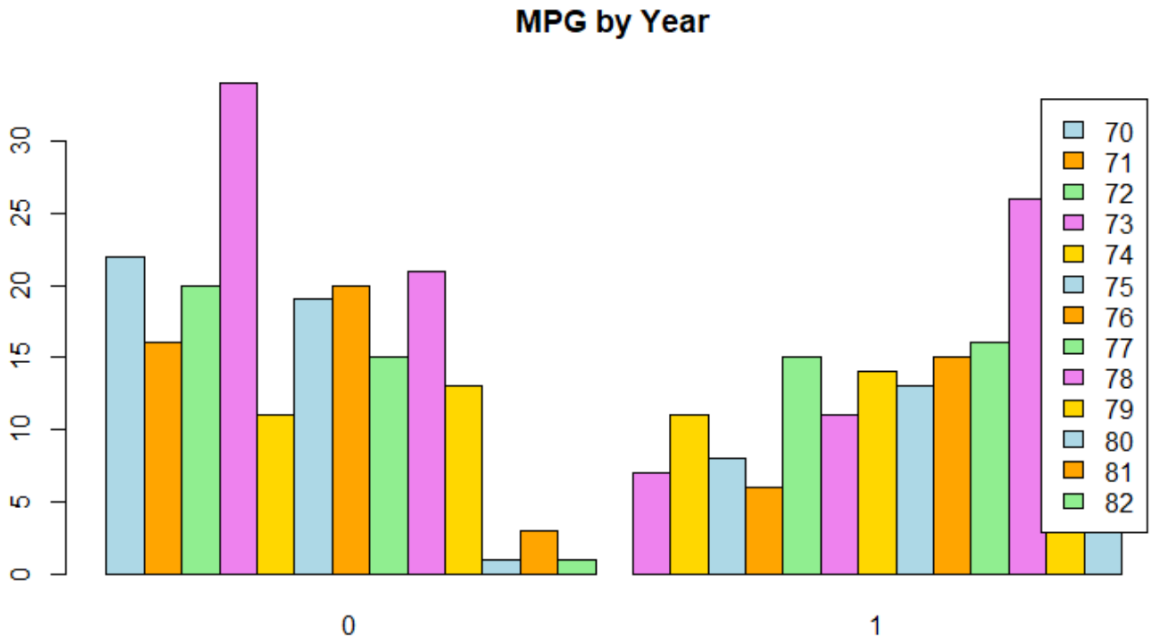


Weight: Cars that get less than or equal to the median mpg tend to have greater weight.

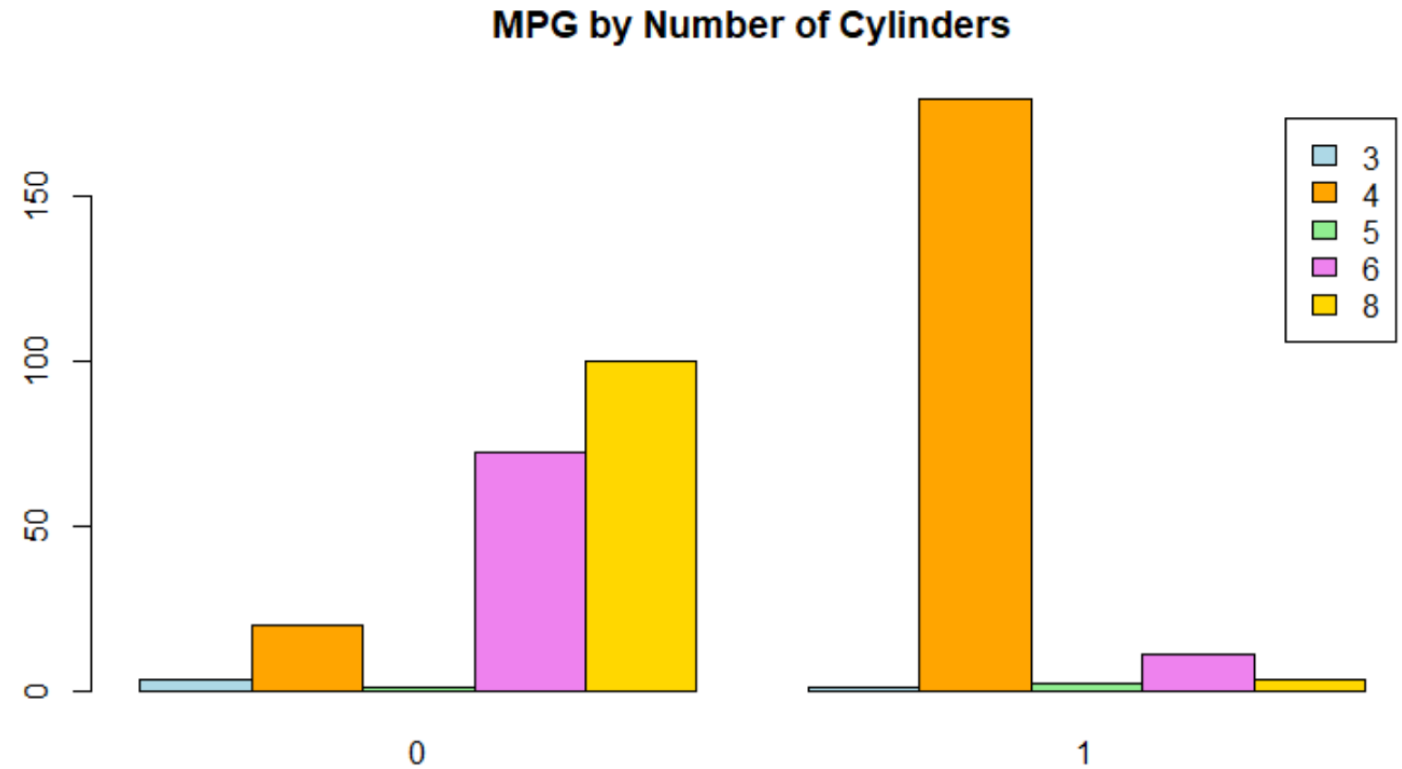


Displacement: Cars that get less than or equal to the median mpg tend to have greater displacement.

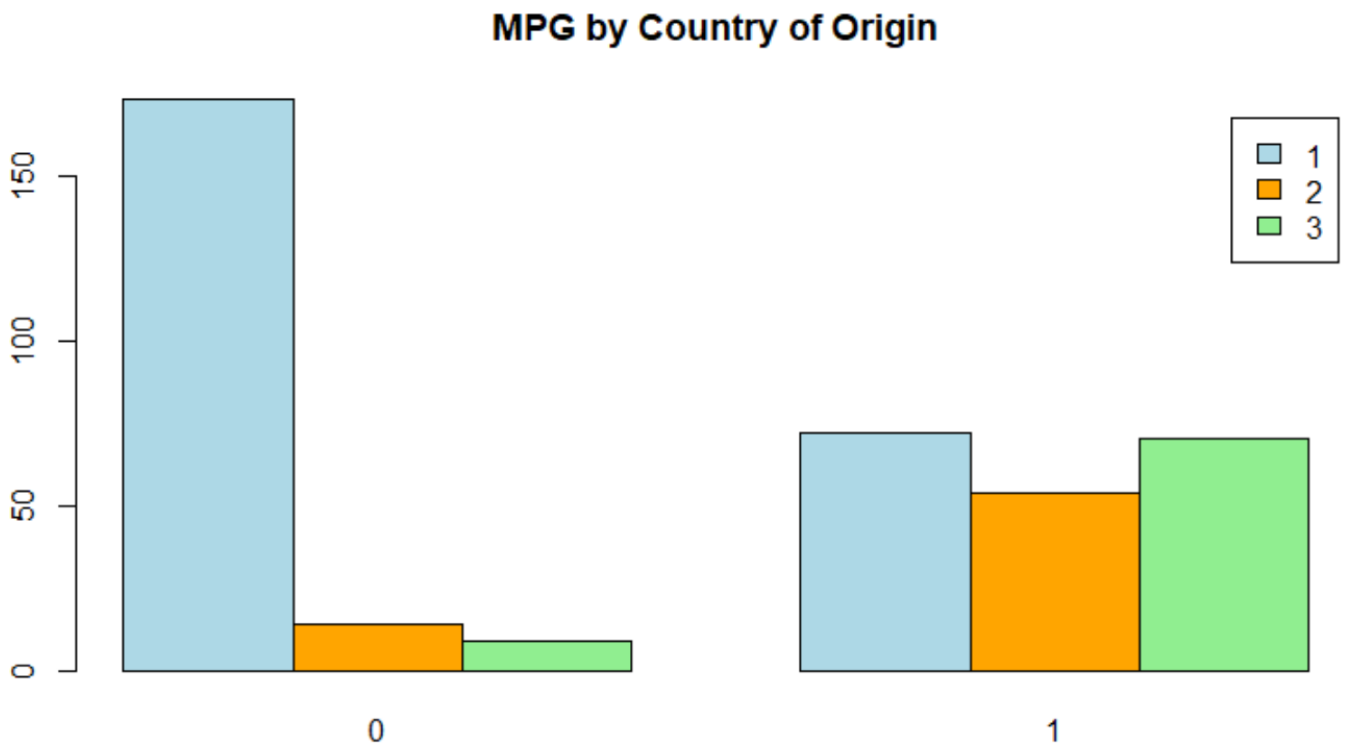




Year: Cars that are newer tend to have mpg greater than the median mpg.



Cylinders: 4-cylinder cars comprise the greatest proportion of cards that have a greater-than-median mpg, and they also have the highest mpg. In the subset of cars that have equal to or lower than the median mpg, the mpg tends to increase with the number of cylinders (with the exclusion of five cylinders, which is uncommon).



Country of Origin: In the subset of cars that have equal to or lower than the median mpg, the United States (Country #1) produces the car with the least best mpg. In the subset of cars that has higher than the median mpg, the three countries of origin are close to equal. (This may have a relationship with year of production.)

(c) Split the data into a training set and a test set. *For problem (c), split the data into a training set and a test set using following R code.*

```
set.seed(1011)
train <- sort(sample(1:dim(mpg.df)[1], 196))
Auto.train <- mpg.df[train, ]
Auto.test <- mpg.df[-train, ]
```

(d) Perform LDA on the training data in order to predict mpg01 using the variables that seemed most associated with mpg01 in (b). What is the test error of the model obtained?

```
library(MASS)
lda.fit.mpg01 <- lda(mpg01 ~ horsepower + weight + displacement + cylinders + year + origin, data = Auto.train)
lda.pred.mpg01 <- predict(lda.fit.mpg01, Auto.test)
lda.table <- table(lda.pred.mpg01$class, Auto.test$mpg01)
#mean(lda.pred.mpg01$class == Auto.test$mpg01)
#one.minus.mean <-1 - mean(lda.pred.mpg01$class == Auto.test$mpg01)
#one.minus.mean
lda.test.error <- (lda.table[1,2] + lda.table[2,1])/sum(lda.table)
lda.test.error
```

```
> lda.test.error
[1] 0.1122449
```

(e) Perform QDA on the training data in order to predict mpg01 using the variables that seemed most associated with mpg01 in (b). What is the test error of the model obtained?

```
qda.fit.mpg01 = qda(mpg01 ~ horsepower + weight + displacement + cylinders + year + origin, data = Auto.train)
```

```
qda.pred.mpg01 <- predict(qda.fit.mpg01, Auto.test)
names(qda.pred.mpg01)
qda.table <- table(qda.pred.mpg01$class, Auto.test$mpg01)
#mean(qda.pred.mpg01$class == Auto.test$mpg01)
#one.minus.mean <-1 - mean(qda.pred.mpg01$class == Auto.test$mpg01)
#one.minus.mean
qda.test.error <- (qda.table[1,2] + qda.table[2,1])/sum(qda.table)
qda.test.error
```

```
> qda.test.error
[1] 0.09693878
```

(f) Perform logistic regression on the training data in order to predict mpg01 using the variables that seemed most associated with mpg01 in (b). What is the test error of the model obtained?

```
auto.lgrgsn <- glm(mpg01 ~ horsepower + weight + displacement + cylinders + year + origin, data = Auto.train, family = binomial)
auto.prob <- predict(auto.lgrgsn, type = "response")
auto.prob[1:10]
auto.lr.pred <- as.factor(ifelse(auto.prob> 0.5, 1, 0))
logit.table <-table(auto.lr.pred,Auto.test$mpg01)
#mean(auto.lr.pred==Auto.test$mpg01)
#one.minus.mean <-1 - mean(auto.lr.pred==Auto.test$mpg01)
#one.minus.mean
logit.test.error <- (logit.table[1,2] + logit.table[2,1])/sum(logit.table)
logit.test.error
```

```
> logit.test.error
[1] 0.3520408
```

(g) Perform KNN on the training data, with several values of K, in order to predict mpg01. Use only the variables that seemed most associated with mpg01 in (b). What test errors do you obtain? Which value of K seems to perform the best on this data set?

```
library(FNN)
x.train <- cbind(mpg.df$horsepower, mpg.df$weight, mpg.df$displacement, mpg.df$cylinders, mpg.df$year, mpg.df$origin)[train, ]
x.test <- cbind(mpg.df$horsepower, mpg.df$weight, mpg.df$displacement, mpg.df$cylinders, mpg.df$year, mpg.df$origin)[-train, ]
y.train <- mpg.df$mpg01[train]
y.test <- mpg.df$mpg01[-train]
knn.k3.pred <- knn(x.train, x.test, y.train, k = 3)
knn.k3.table <- table(knn.k3.pred, y.test)
knn.k3.test.error <- (knn.k3.table[1,2] + knn.k5.table[2,1])/sum(knn.k3.table)
knn.k3.test.error
```

```
> knn.k3.test.error
[1] 0.08163265
```

K value of 3 has the lowest error rate of the tested K values

```
# KNN with k = 5
knn.k5.pred <- knn(x.train, x.test, y.train, k = 5)
```

```
knn.k5.table <- table(knn.k5.pred, y.test)
knn.k5.test.error <- (knn.k5.table[1,2] + knn.k5.table[2,1])/sum(knn.k5.table)
knn.k5.test.error
```

```
> knn.k5.test.error
[1] 0.09183673
```

```
knn.k7.pred <- knn(x.train, x.test, y.train, k = 7)
knn.k7.table <- table(knn.k7.pred, y.test)
knn.k7.test.error <- (knn.k7.table[1,2] + knn.k7.table[2,1])/sum(knn.k7.table)
knn.k7.test.error
```

```
> knn.k7.test.error
[1] 0.1071429
```

```
knn.k9.pred <- knn(x.train, x.test, y.train, k = 9)
knn.k9.table <- table(knn.k9.pred, y.test)
knn.k9.test.error <- (knn.k9.table[1,2] + knn.k9.table[2,1])/sum(knn.k9.table)
knn.k9.test.error
```

```
> knn.k9.test.error
[1] 0.1122449
```

```
knn.k15.pred <- knn(x.train, x.test, y.train, k = 9)
knn.k15.table <- table(knn.k15.pred, y.test)
knn.k15.test.error <- (knn.k15.table[1,2] + knn.k15.table[2,1])/sum(knn.k15.table)
knn.k15.test.error
```

```
> knn.k15.test.error
[1] 0.1071429
```

```
# KNN with k = 25
knn.k25.pred <- knn(x.train, x.test, y.train, k = 25)
knn.k25.table <- table(knn.k25.pred, y.test)
knn.k25.test.error <- (knn.k25.table[1,2] + knn.k25.table[2,1])/sum(knn.k25.table)
knn.k25.test.error
```

```
> knn.k25.test.error
[1] 0.1173469
```

```
knn.k50.pred <- knn(x.train, x.test, y.train, k = 50)
knn.k50.table <- table(knn.k50.pred, y.test)
knn.k50.test.error <- (knn.k50.table[1,2] + knn.k50.table[2,1])/sum(knn.k50.table)
knn.k50.test.error
```

```
> knn.k50.test.error
[1] 0.1122449
```