iii.Exercise 4.7.11

Kaarthik Sundaramoorthy, Sahil Shah and Vidhi Shah

6/13/2020

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.

```
require(ISLR)
```

```
## Loading required package: ISLR
```

```
data(Auto)
attach(Auto)
```

(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.

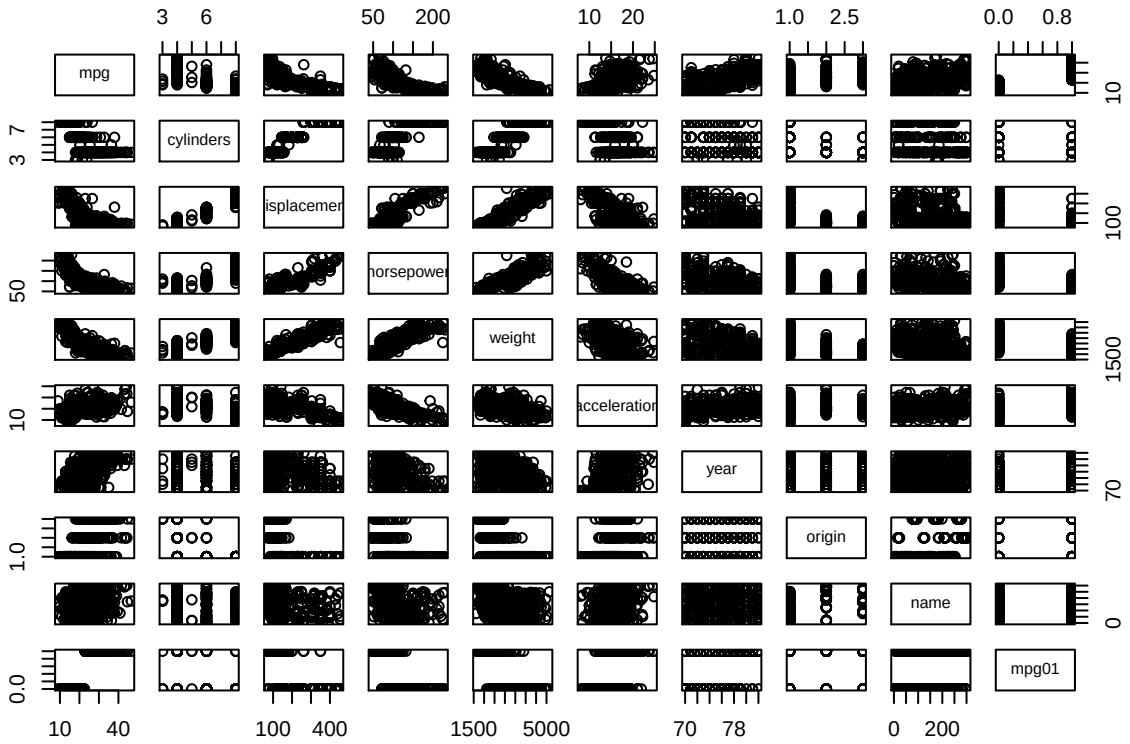
```
mpg01 = rep(0, length(mpg))
mpg01[mpg > median(mpg)] = 1
myAuto = data.frame(Auto, mpg01)
summary(myAuto)
```

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

```
##  Median :0.5
##  Mean   :0.5
##  3rd Qu.:1.0
##  Max.   :1.0
##
```

(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.

```
pairs(myAuto)
```




```
sapply(myAuto, class)
```

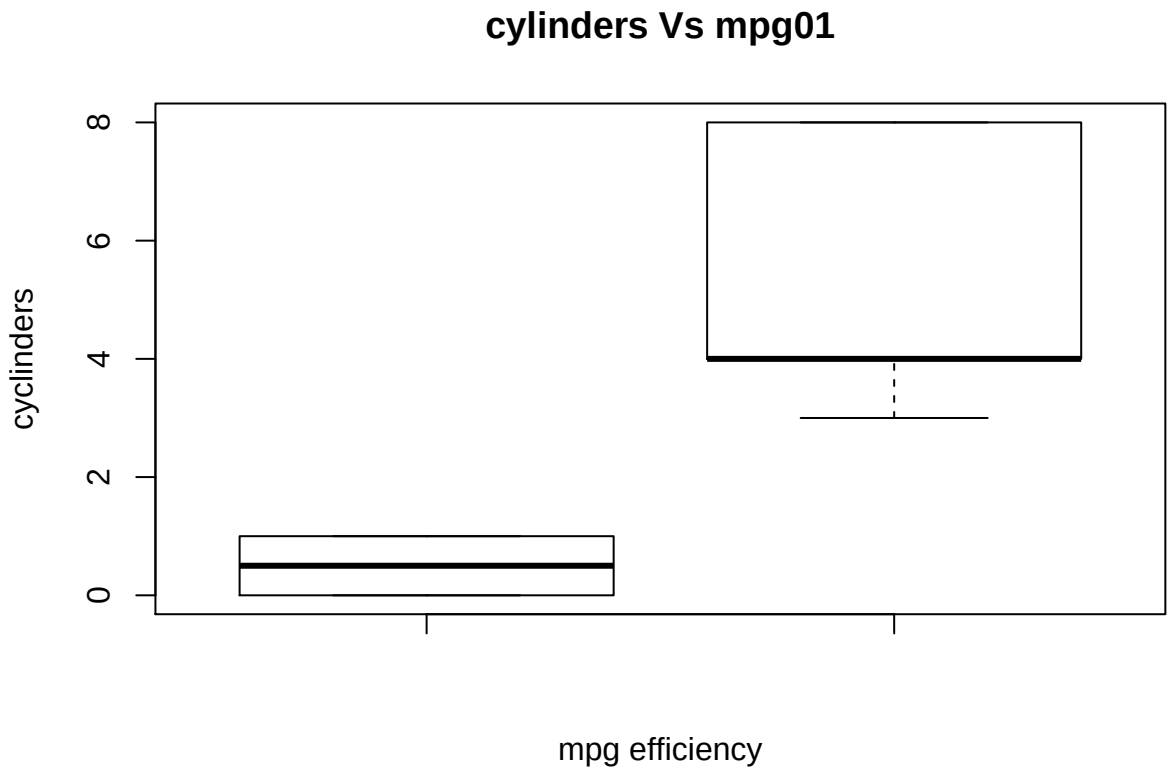
```
##          mpg    cylinders displacement   horsepower       weight acceleration
##    "numeric"    "numeric"    "numeric"    "numeric"    "numeric"    "numeric"
##         year       origin         name        mpg01
##    "numeric"    "numeric"     "factor"    "numeric"
```

```
cor(myAuto[-9])
```

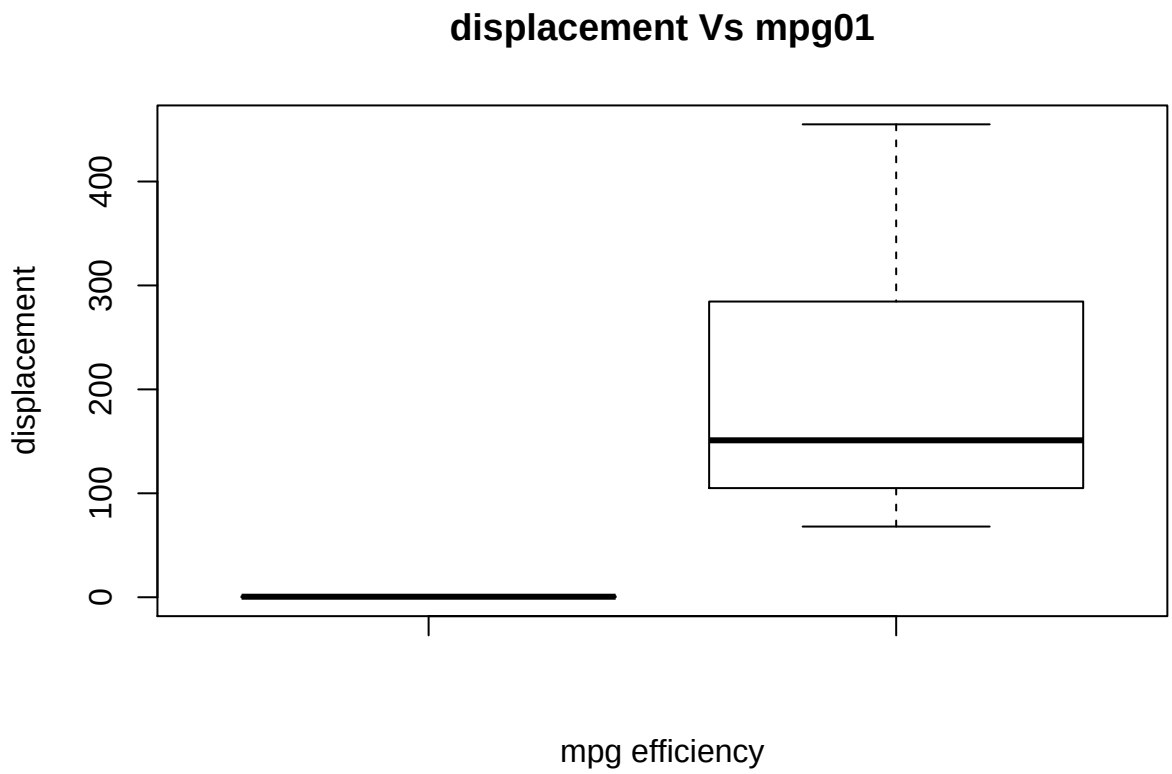
```
##                     mpg  cylinders displacement horsepower     weight
## mpg           1.0000000 -0.7776175   -0.8051269 -0.7784268 -0.8322442
## cylinders    -0.7776175  1.0000000    0.9508233  0.8429834  0.8975273
```

```
## displacement -0.8051269  0.9508233    1.0000000  0.8972570  0.9329944
## horsepower   -0.7784268  0.8429834    0.8972570  1.0000000  0.8645377
## weight       -0.8322442  0.8975273    0.9329944  0.8645377  1.0000000
## acceleration  0.4233285 -0.5046834   -0.5438005 -0.6891955 -0.4168392
## year          0.5805410 -0.3456474   -0.3698552 -0.4163615 -0.3091199
## origin        0.5652088 -0.5689316   -0.6145351 -0.4551715 -0.5850054
## mpg01         0.8369392 -0.7591939   -0.7534766 -0.6670526 -0.7577566
##              acceleration       year     origin      mpg01
## mpg             0.4233285  0.5805410  0.5652088  0.8369392
## cylinders      -0.5046834 -0.3456474 -0.5689316 -0.7591939
## displacement   -0.5438005 -0.3698552 -0.6145351 -0.7534766
## horsepower     -0.6891955 -0.4163615 -0.4551715 -0.6670526
## weight         -0.4168392 -0.3091199 -0.5850054 -0.7577566
## acceleration    1.0000000  0.2903161  0.2127458  0.3468215
## year            0.2903161  1.0000000  0.1815277  0.4299042
## origin          0.2127458  0.1815277  1.0000000  0.5136984
## mpg01           0.3468215  0.4299042  0.5136984  1.0000000
```

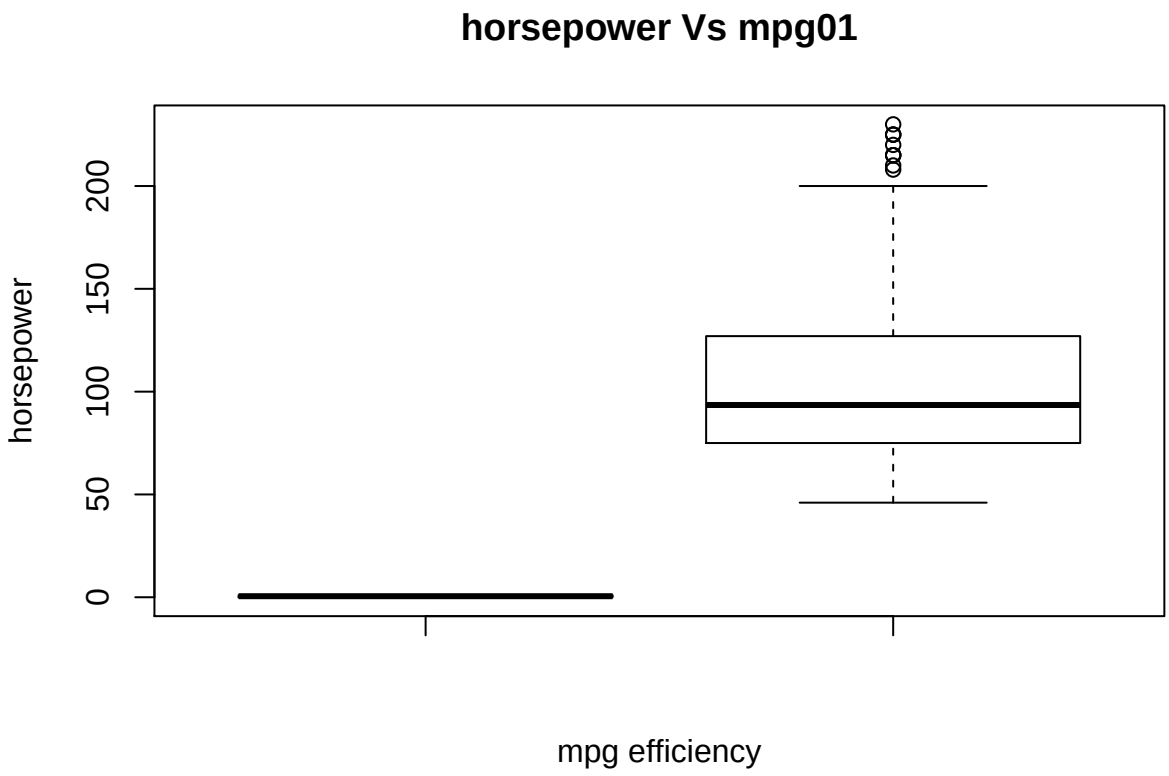
```r
#par(mfrow = c(2,2))
boxplot(mpg01, cylinders, main = "cylinders Vs mpg01",
        xlab = "mpg efficiency", ylab = "cyclinders")
```



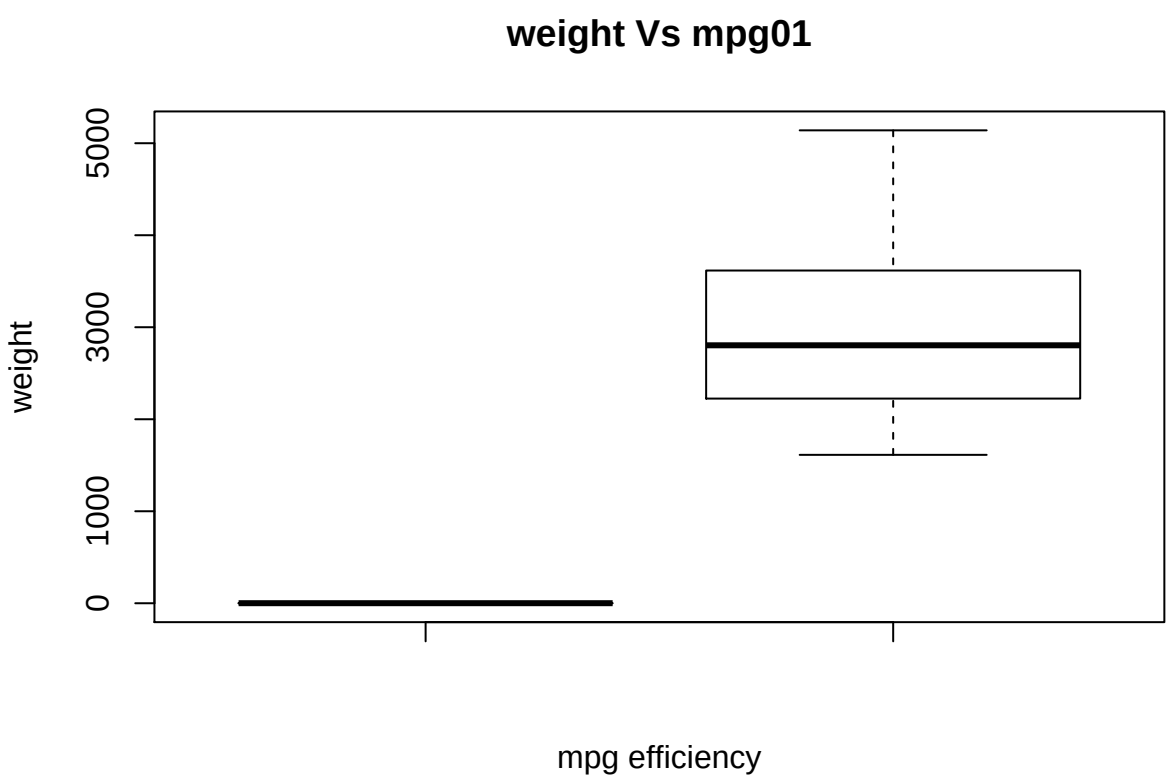

```r
boxplot(mpg01, displacement, main = "displacement Vs mpg01",
        xlab = "mpg efficiency", ylab = "displacement")
```




```
boxplot(mpg01, horsepower, main = "horsepower Vs mpg01",
        xlab = "mpg efficiency", ylab = "horsepower")
```




```
boxplot(mpg01, weight, main = "weight Vs mpg01",
        xlab = "mpg efficiency", ylab = "weight")
```




```
#par(mfrow = c(1,1))
```

Using this we notice that the absolute correlation value of mpg01 with the cylinder, displacement, horsepower and weight is closer to 1 and hence these features are most likely to be useful in predicting mpg01. If the displacement is above 200, the mpg01 will be less than the median and if the displacement is greater than 200 then there is a higher possibility of mpg01 being greater than median. if the weight is above 3500, its mpg is less than the median. Likewise, if the weight is below 2700, the mpg is greater than the median. if the horsepower is below 100 then mpg is greater than median. And if horsepower is greater than 100, then mpg is less than median. We can see that cylinders also have some relationship with mpg01. It is very common that for a vehicle with 4 cylinders to have higher mpg than lower mpg, but it is common for 6 or 8 cylinder vehicles to have lower mpg than higher mpg.

(c) Split the data into a training set and a test set.

```
set.seed(123)
splitratio <- sample(1:nrow(myAuto), nrow(myAuto)*0.75 , replace=F) # 75% train, 25% test
Auto.train <- myAuto[splitratio,]
Auto.test <- myAuto[-splitratio,]
mpg01.test <- mpg01[-splitratio]
dim(Auto.train)
```

```
## [1] 294  10
```

```
dim(Auto.test)
```

```
## [1] 98 10
```

```
dim(mpg01.test)
```

```
## NULL
```

(d) Perform LDA on the training data in order to predict **mpg01** using the variables that seemed most associated with **mpg01** in

```
library(MASS)
lda.fit <- lda(mpg01 ~ cylinders + weight + displacement + horsepower, data = Auto.train)
lda.fit
```

```
## Call:
## lda(mpg01 ~ cylinders + weight + displacement + horsepower, data = Auto.train)
##
## Prior probabilities of groups:
##         0         1
## 0.4931973 0.5068027
##
## Group means:
##   cylinders   weight displacement horsepower
## 0  6.779310 3623.538     273.4207  130.03448
## 1  4.174497 2318.987     114.2752   78.16107
##
## Coefficients of linear discriminants:
##                         LD1
## cylinders    -0.4467367673
## weight       -0.0009502462
## displacement -0.0024402207
## horsepower    0.0045406476
```

```
lda.fit.pred <- predict(lda.fit, Auto.test)$class
table(lda.fit.pred, Auto.test$mpg01)
```

```
##
## lda.fit.pred  0  1
##            0 43  3
##            1  8 44
```

(d).(b). What is the test error of the model obtained?

```
mean(lda.fit.pred != Auto.test$mpg01)
```

```
## [1] 0.1122449
```

The test error rate of the model based on the 75:25 split with respect to 75% as in training set and 25% as in test set is 11.2%.

(e) Perform QDA on the training data in order to predict **mpg01** using the variables that seemed most associated with **mpg01** in

```
fit.qda <- qda(mpg01~cylinders + weight + displacement + horsepower, data=Auto.train)
fit.qda.pred <- predict(fit.qda, Auto.test)$class
table(fit.qda.pred, Auto.test$mpg01)
```

```
##
## fit.qda.pred  0  1
##            0 45  4
##            1  6 43
```

(e). (b). What is the test error of the model obtained?

```
mean(fit.qda.pred != Auto.test$mpg01)
```

```
## [1] 0.1020408
```

The test error rate of the model is 10.2%.

(f) Perform logistic regression on the training data in order to predict **mpg01** using the variables that seemed most associated with **mpg01** in

```
fit.logit <- glm(mpg01~cylinders+displacement+horsepower+weight, data=Auto.train,
                 family=binomial)
logit.prob <- predict(fit.logit, Auto.test, type="response")
logit.pred <- ifelse(logit.prob > 0.5, 1, 0)
table(logit.pred, Auto.test$mpg01)
```

```
##
## logit.pred  0  1
##          0 44  5
##          1  7 42
```

(f).(b). What is the test error of the model obtained?

```
mean(logit.pred != Auto.test$mpg01)
```

```
## [1] 0.122449
```

The test error rate of the model is 12.25%

(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?

```
library(class)
train.X <- cbind(Auto.train$cylinders, Auto.train$displacement, Auto.train$weight,
                 Auto.train$horsepower)
test.X <- cbind(Auto.test$cylinders, Auto.test$displacement, Auto.test$weight,
                Auto.test$horsepower)
knn.pred <- knn(train.X, test.X, Auto.train$mpg01, k=1)
table(knn.pred, Auto.test$mpg01)
```

```
##
## knn.pred  0  1
##        0 43  9
##        1  8 38
```

```
mean(knn.pred != Auto.test$mpg01)
```

```
## [1] 0.1734694
```

```
knn.pred <- knn(train.X, test.X, Auto.train$mpg01, k=10)
table(knn.pred, Auto.test$mpg01)
```

```
##
## knn.pred  0  1
##        0 43  3
##        1  8 44
```

```
mean(knn.pred != Auto.test$mpg01)
```

```
## [1] 0.1122449
```

```
knn.pred <- knn(train.X, test.X, Auto.train$mpg01, k=20)
table(knn.pred, Auto.test$mpg01)
```

```
##
## knn.pred  0  1
##        0 43  3
##        1  8 44
```

```
mean(knn.pred != Auto.test$mpg01)
```

```
## [1] 0.1122449
```

```
knn.pred <- knn(train.X, test.X, Auto.train$mpg01, k=30)
table(knn.pred, Auto.test$mpg01)
```

```
##
## knn.pred  0  1
##        0 44  4
##        1  7 43
```

```
mean(knn.pred != Auto.test$mpg01)
```

```
## [1] 0.1122449
```

```
knn.pred <- knn(train.X, test.X, Auto.train$mpg01, k=50)
table(knn.pred, Auto.test$mpg01)
```

```
##
## knn.pred  0  1
##        0 43  3
##        1  8 44
```

```
mean(knn.pred != Auto.test$mpg01)
```

```
## [1] 0.1122449
```

```
knn.pred <- knn(train.X, test.X, Auto.train$mpg01, k=100)
table(knn.pred, Auto.test$mpg01)
```

```
##
## knn.pred  0  1
##        0 43  4
##        1  8 43
```

```
mean(knn.pred != Auto.test$mpg01)
```

```
## [1] 0.122449
```

```
knn.pred <- knn(train.X, test.X, Auto.train$mpg01, k=200)
table(knn.pred, Auto.test$mpg01)
```

```
##
## knn.pred  0  1
##        0 39  2
##        1 12 45
```

```
mean(knn.pred != Auto.test$mpg01)
```

```
## [1] 0.1428571
```

The K values are as above:

1. K = 1, The error rate is 17.3%
2. K = 10, The error rate is 10.2%
3. K = 20, The error rate is 10.2%
4. K = 30, The error rate is 11.2%
5. K = 50, The error rate is 12.2%
6. K = 100, The error rate is 12.2%
7. K = 200, The error rate is 14.3%

Which value of K seems to perform the best on this data set?

According to the dataset, the best K with respect to the error rate is 10 and 20.