Homework #3

Eric Pettengill

4.7.11. Develop a model to predict whether a give car gets high or low gas mileage based on the AUTO dataset

```
library(tidyverse)
library(ISLR)
library(caret)
library(MASS)
```

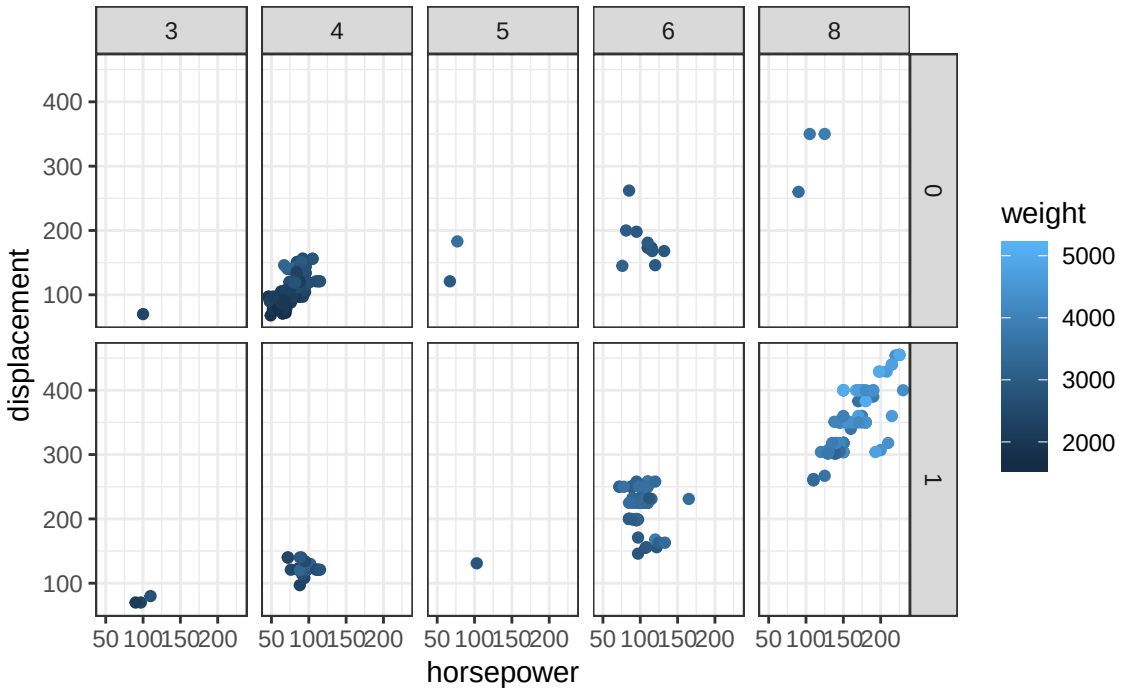
(a) Create a binary variable, `mpg01`, that contains a 1 if `mpg` contains a value below its median.

```
auto_new <- Auto %>%
  mutate(mpg01 = if_else(mpg < median(mpg), 1, 0))

auto_new$mpg01 <- as.factor(auto_new$mpg01)
auto_new$origin <- as.factor(auto_new$origin)

str(auto_new)
```

```
## 'data.frame':    392 obs. of  10 variables:
##  $ mpg         : num  18 15 18 16 17 15 14 14 14 15 ...
##  $ cylinders   : num  8 8 8 8 8 8 8 8 8 8 ...
##  $ displacement: num  307 350 318 304 302 429 454 440 455 390 ...
##  $ horsepower  : num  130 165 150 150 140 198 220 215 225 190 ...
##  $ weight      : num  3504 3693 3436 3433 3449 ...
##  $ acceleration: num  12 11.5 11 12 10.5 10 9 8.5 10 8.5 ...
##  $ year        : num  70 70 70 70 70 70 70 70 70 70 ...
##  $ origin      : Factor w/ 3 levels "1","2","3": 1 1 1 1 1 1 1 1 1 1 ...
##  $ name        : Factor w/ 304 levels "amc ambassador brougham",..: 49 36 231 14 161 141 54 223 241 1
##  $ mpg01       : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
```

(b) Explore the data graphically in order to investigate the association between `mpg01` and other features. Which of the other features seem most likely to be useful in predicting `mpg01`?

```
ggplot(auto_new) +
  geom_point(aes(x = horsepower, y = displacement, color = weight)) +
  facet_grid(mpg01 ~ cylinders) +
  theme_bw()
```



The plot above shows the displacement, horsepower, and weight of each car by the number of cylinders(3,4,5,6,8) and whether the MPG of each is below the median(1) or not(0). As we can see, as displacement, horsepower, weight, and the number of cylinders increase, more cars' MPG fall below the median MPG value(labeled 1 above), most notably for 8 cylinder cars.

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

```
set.seed(1011)
train <- createDataPartition(auto_new$mpg01,
                                    p = 0.5,
                                    list = FALSE)

auto.train <- auto_new[train, ]
auto.test <- auto_new[-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?

```
lda.auto.fit <- lda(
  mpg01 ~ cylinders + displacement + horsepower + weight + acceleration,
  data = auto.train
)

lda.auto.pred <- predict(lda.auto.fit, auto.test)

(lda.table <- confusionMatrix(lda.auto.pred$class, auto.test$mpg01)$table)
```

```
##            Reference
## Prediction  0  1
##          0 90 11
##          1  8 87
```

```
(lda.test.error <- (lda.table[1,2]+lda.table[2,1])/sum(lda.table))
```

```
## [1] 0.09693878
```

(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.auto.fit <- qda(
  mpg01 ~ cylinders + displacement + horsepower + weight + acceleration,
  data = auto.train
)

qda.auto.pred <- predict(qda.auto.fit, auto.test)

(qda.table <- confusionMatrix(qda.auto.pred$class, auto.test$mpg01)$table)
```

```
##            Reference
## Prediction  0  1
##          0 88  8
##          1 10 90
```

```
(qda.test.error <- (qda.table[1,2]+qda.table[2,1])/sum(qda.table))
```

```
## [1] 0.09183673
```

(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?

```
logistic.auto.fit <- glm(
  mpg01 ~ cylinders + displacement + horsepower + weight + acceleration,
  data = auto.train,
  family = binomial
)

logistic.auto.pred <- predict(logistic.auto.fit, auto.test, type = "response")
logistic.auto.pred <- as.factor(if_else(logistic.auto.pred > 0.5, 1, 0))

(logistic.table <- confusionMatrix(logistic.auto.pred, auto.test$mpg01)$table)
```

```
##            Reference
## Prediction  0  1
##          0 88  9
##          1 10 89
```

```
(logistic.test.error <- (logistic.table[1,2] + logistic.table[2,1])/sum(logistic.table))
```

```
## [1] 0.09693878
```

(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?

```
set.seed(1011)
x <- dplyr::select(auto_new, cylinders, displacement, horsepower, weight, acceleration)
y <- auto_new$mpg01

xtrain <- x[train, ]
xtest <- x[-train, ]
ytrain <- y[train]
ytest <- y[-train]

knn.k5.pred <- class::knn(xtrain, xtest, ytrain, k = 5)
```

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

knn.k10.pred <- class::knn(xtrain, xtest, ytrain, k = 10)
knn.k10.tbl <- table(knn.k10.pred, ytest)
knn.k10.test.error <- (knn.k10.tbl[1,2] + knn.k10.tbl[2,1])/sum(knn.k10.tbl)

knn.k20.pred <- class::knn(xtrain, xtest, ytrain, k = 20)
knn.k20.tbl <- table(knn.k20.pred, ytest)
knn.k20.test.error <- (knn.k20.tbl[1,2] + knn.k20.tbl[2,1])/sum(knn.k20.tbl)

knn.k50.pred <- class::knn(xtrain, xtest, ytrain, k = 50)
knn.k50.tbl <- table(knn.k50.pred, ytest)
knn.k50.test.error <- (knn.k50.tbl[1,2] + knn.k50.tbl[2,1])/sum(knn.k50.tbl)

knn.k100.pred <- class::knn(xtrain, xtest, ytrain, k = 100)
knn.k100.tbl <- table(knn.k100.pred, ytest)
knn.k100.test.error <- (knn.k100.tbl[1,2] + knn.k100.tbl[2,1])/sum(knn.k100.tbl)

rbind(knn.k5.test.error,
      knn.k10.test.error,
      knn.k20.test.error,
      knn.k50.test.error,
      knn.k100.test.error)
```

```
##                          [,1]
## knn.k5.test.error   0.1173469
## knn.k10.test.error  0.1122449
## knn.k20.test.error  0.1122449
## knn.k50.test.error  0.1173469
## knn.k100.test.error 0.1071429
```

The larger the value of K, the lower the test error. This may be prone to overfitting, however.