Homework 3

Anthony Anderson

March 8, 2019

1) a) In the interval (.05, .95), the observations we would use to predict a response are in the interval (x-.05, x+.05) and will be 10% of the total. For x < .05, we would use the interval (0, x+.05), which will be (100x+5)% of the total. For x > .95, the fraction of observations used is (105-100x)%, so to get the average overall, we calculate:

$$\int_{.05}^{.95} 10dx + \int_{0}^{.05} (100x + 5)dx + \int_{.95}^{1} (105 - 100x)dx = 9.75$$

So 9.75% of total observations are used to make a prediction.

b) Assuming X_1 and X_2 are indepedent, the fraction of observations used to make a prediction is simply $.0975 * .0975 = .00950625 = .95\%$. So not even 1% of observations are used.

c) We would simply calculate $.0975^{100}$ which is roughly 0.

d) As p increases, the calculation of fraction of observations used in prediction is $(9.75\%)^p$. So as p approaches infinity, the fraction quickly approaches 0, since even $p = 2$ is below 1%.

e) For $p = 1$, the length of the "cube" is $l = .1$, for $p = 2$, $l = .1^{1/2}$ and for $p = 100$, $l = .1^{1/100}$

2) a) If the boundary is linear, QDA will perform better on the training set because it will fit closer since its more complex. On the test set, the LDA will perform better because it is less likely to overfit.

b) If the boundary is non-linear, the QDA will fit better for both the training and test sets.

c) If n increases enough, the variance will be less of an issue in overfitting, so the QDA will have better accuracy.

d) False. If there are less sample points, the variance will lead to overfitting with QDA. It will have a higher test error rate, so LDA is better when n is small.

3) a) Plugging $X_1 = 40, X_2 = 3.5$ into the equation

$$\hat{p}(X) = \frac{e^{-6+.05X_1+X_2}}{1 + e^{-6+.05X_1+X_2}} = .3775$$

b) The student still has a 3.5GPA, so we need to solve for X_1 in

$$\hat{p}(X) = \frac{e^{-6+.05X_1+3.5}}{1 + e^{-6+.05X_1+3.5}} = .5 \Rightarrow X_1 = 50$$

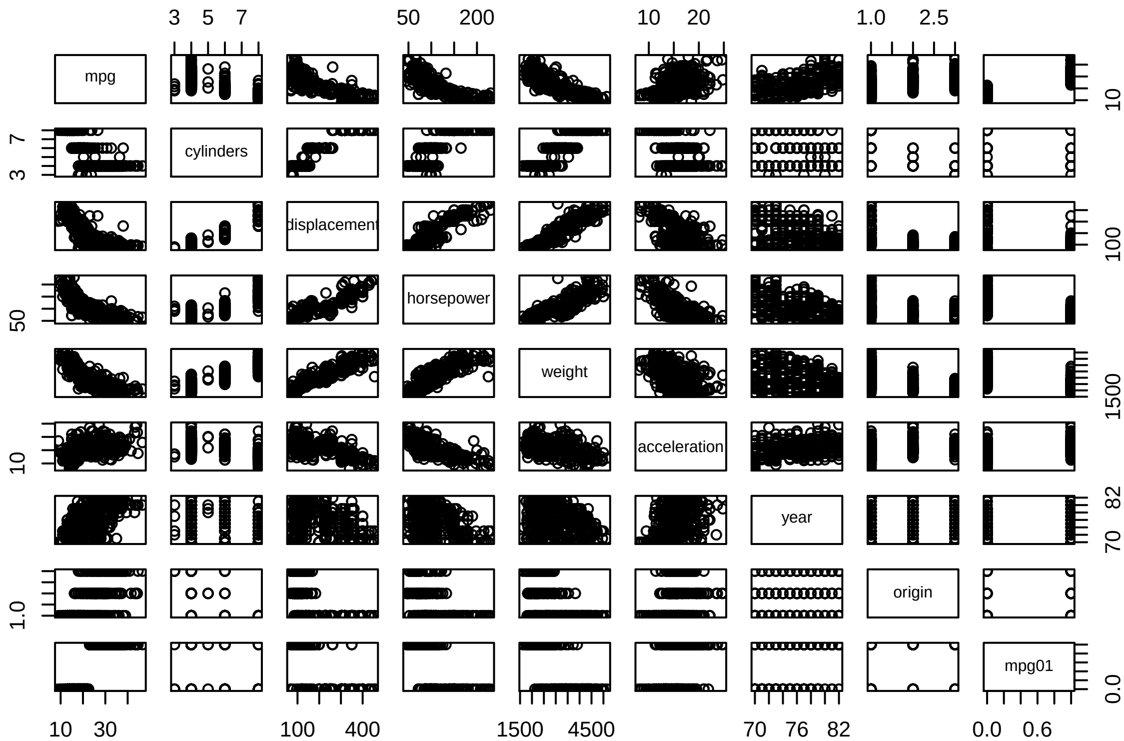
.

4)

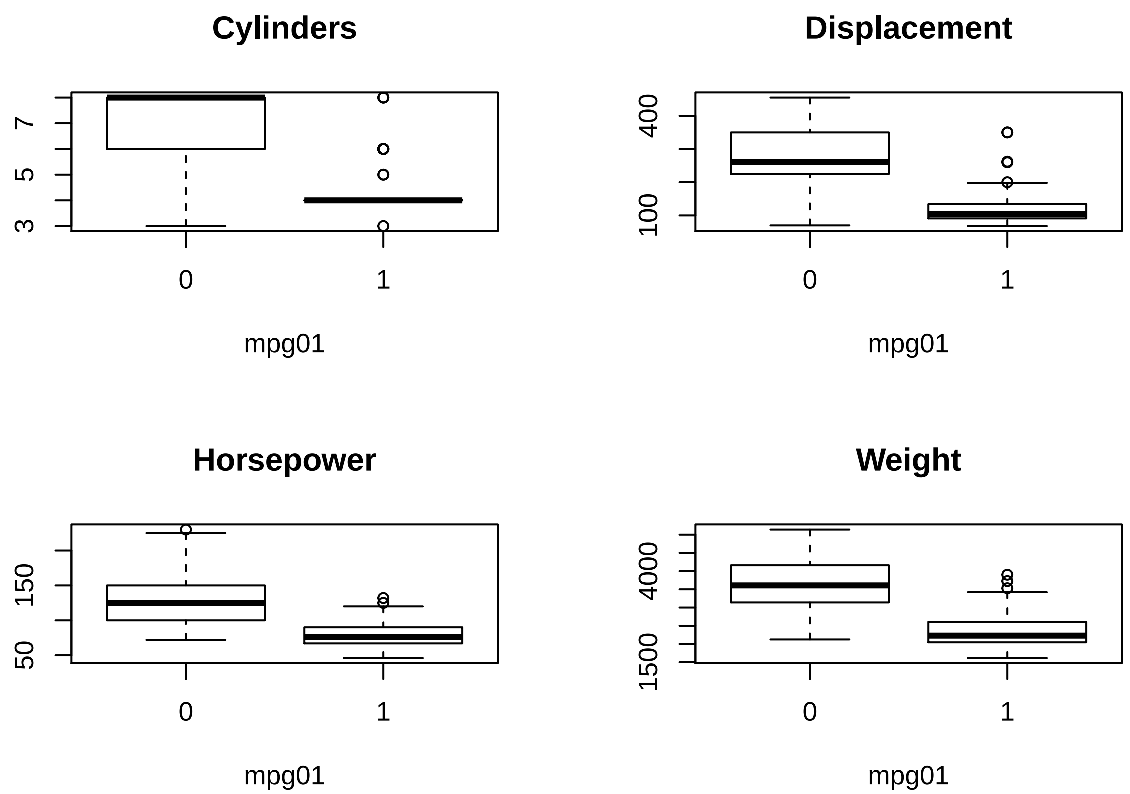
```r
#a)
library("ISLR")
summary(Auto)
```

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

```r
Dataset <- data.frame(Auto)
mpg01 <-rep(0,length(Dataset$mpg))
mpg01[Dataset$mpg > median(Dataset$mpg)]<- 1
Dataset <- data.frame(Auto,mpg01)
#b)
pairs(Dataset[,-9])
```




```
opar <- par(no.readonly = T)
par(mfrow=c(2,2))
boxplot(cylinders~mpg01, data=Dataset, main="Cylinders", xlab="mpg01")
boxplot(displacement~mpg01, data=Dataset, main="Displacement", xlab="mpg01")
boxplot(horsepower~mpg01, data=Dataset, main="Horsepower", xlab="mpg01")
boxplot(weight~mpg01, data=Dataset, main="Weight", xlab="mpg01")
```



There seems to be significantly different values for mpg01, suggesting there is association between mpg01 and Cylinders, Displacement, Horsepower, and Weight.

c)

```
training <- (Dataset$year %% 2==0)
Dataset.training <- Dataset[training,]
Dataset.test <- Dataset[!training,]
mpg01.test <- mpg01[!training]
```

d)

```
library(MASS)
ldafit <- lda(mpg01~cylinders+weight+displacement+horsepower, data=Dataset, subset=training)
ldafit
```

```
## Call:
## lda(mpg01 ~ cylinders + weight + displacement + horsepower, data = Dataset,
##     subset = training)
##
## Prior probabilities of groups:
##         0         1
## 0.4571429 0.5428571
##
## Group means:
##   cylinders   weight displacement horsepower
## 0  6.812500 3604.823     271.7396  133.14583
## 1  4.070175 2314.763     111.6623   77.92105
##
```

```
## Coefficients of linear discriminants:
##                       LD1
## cylinders    -0.6741402638
## weight       -0.0011465750
## displacement  0.0004481325
## horsepower    0.0059035377
```

```r
ldapredict <- predict(ldafit, Dataset.test)
mean(ldapredict$class !=mpg01.test)
```

```
## [1] 0.1263736
```

The LDA test error rate is 12.64%
e)

```r
qdafit <- qda(mpg01~cylinders+weight+displacement+horsepower, data=Dataset, subset=training)
qdafit
```

```
## Call:
## qda(mpg01 ~ cylinders + weight + displacement + horsepower, data = Dataset,
##     subset = training)
##
## Prior probabilities of groups:
##         0         1
## 0.4571429 0.5428571
##
## Group means:
##   cylinders   weight displacement horsepower
## 0  6.812500 3604.823     271.7396  133.14583
## 1  4.070175 2314.763     111.6623   77.92105
```

```r
qdapredict <- predict(qdafit, Dataset.test)
mean(qdapredict$class!=mpg01.test)
```

```
## [1] 0.1318681
```

The QDA test error rate is 13.19%
f)

```r
glmfit<- glm(mpg01~cylinders+weight+displacement+horsepower, data=Dataset, family=binomial, subset=trai
glmfit
```

```
##
## Call:  glm(formula = mpg01 ~ cylinders + weight + displacement + horsepower,
##     family = binomial, data = Dataset, subset = training)
##
## Coefficients:
##  (Intercept)     cylinders        weight  displacement    horsepower
##    17.658730     -1.028032     -0.002922      0.002462     -0.050611
##
## Degrees of Freedom: 209 Total (i.e. Null);  205 Residual
## Null Deviance:       289.6
## Residual Deviance: 83.24     AIC: 93.24
```

```r
logprobs <- predict(glmfit, Dataset.test, type="response")
logpredict <- rep(0,length(logprobs))
logpredict[logprobs >.5]<-1
mean(logpredict!=mpg01.test)
```

```
## [1] 0.1208791
```

Test error rate is 12.09%
g)

```r
library(class)
attach(Dataset)
```

```
## The following object is masked _by_ .GlobalEnv:
##
##     mpg01
```

```r
train.X <- cbind(cylinders,weight,displacement,horsepower)[training,]
test.X <- cbind(cylinders,weight,displacement,horsepower)[!training,]
trainingmpg01<-mpg01[training]
detach(Dataset)
set.seed(42)
knnpredict1 <-knn(train.X, test.X, trainingmpg01, k=1)
mean(knnpredict1!=mpg01.test)
```

```
## [1] 0.1538462
```

```r
set.seed(42)
knnpredict10 <-knn(train.X, test.X, trainingmpg01, k=10)
mean(knnpredict10!=mpg01.test)
```

```
## [1] 0.1593407
```

```r
set.seed(42)
knnpredict100 <-knn(train.X, test.X, trainingmpg01, k=100)
mean(knnpredict100!=mpg01.test)
```

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

The KNN test error rate for K=1 is 15.38%, for K=10 it's 15.93%, and when K=100 it's 14.29%, so K=100 is the best performing.