Homework 3

Rohan Krishnan

2024-02-05

Problem 1

Not required to turn in anything for this problem.

Problem 2

(A) ISLR Conceptual Exercise 4

(a) For observations in the range [0,0.05] and [0.95, 1], we use less than 10% of the observations since we hit the upper and lower bounds of the uniform distribution. For observations in the range [0.05,0.95], we use 10% of the observations (5% on either side). To calculate the average fraction of observations used, we can take a weighted average. Assuming the special cases use around 7.5% of the observations (somewhere between 5% and 10%), we can say that 10% of cases will only use 7.5% of observations. Given this assumption, we can calculate $(0.075 \times 0.10) + (0.10 \times 0.90) \times 100 = 9.75\%$

(b) Since each observation needs to be in the range of **both** X_1 and X_2, we have to multiply their respective fractions of observations. Thus, we get $0.0975^2 \times 100 = 0.95\%$.

(c) As the number of features p increases, the more constrained each reference point becomes (as it must be within 10% of each of the p features). So, for $p = 100$ features, the fraction of available observations would be $0.0975^100 \times 100 = 7.951729e - 100 \approx 0\%$

(d) Since KNN uses nearby points' values to assign a value to the point of interest, it encounters the same dimensionality problem that we observe above. As the number of features p increases, the fraction of observations that are "nearby" rapidly becomes smaller (based on choice of k) points for the algorithm to use to infer a value for the point of interest.

(e) The hypercube must contain 10%, on average, of the observations. When $p = 1$, the hypercube is a line segment with length $l = 0.10$. As the dimensions, p, increase, the area of the cube will always contain 10% of the observations. Thus, given p dimensions, we can calculate the length of each side of the hypercube via $l^p = 0.10$. For $p = 2$, the length of each side would be $l = \sqrt{0.10} = 0.32$. For $p = 100$, the length of each side would be $l = 0.10^{1/100} = 0.98$. It appears that as p increases, the length of each side of the hypercube approaches 1, which is the entire space of potential observations. This means that as we consider more features, we would need a larger and larger search area to cover 10% of observations.

(B) Argue that the line "non-parametric approaches often perform poorly when p is large" is actually not quite the whole picture.

As seen in part (A), the amount of observations being used is important when considering the effectiveness of a non-parametric method. If the number of observations being considered is large enough, a non-parametric method could perform well on the data (a.k.a. the hypercube encompasses most of the data).

(C) The point that ISLR Ch. 4 Exercise 4 is trying to illustrate, is that in high dimensional

space, you are often forced to _____.

Overfit (use more and more of the feature space to effectively "train" the model).

(D) How would you respond to the notion that "more data is never a bad thing"

One way to respond to the above notion given the previous parts of this problem is to note that, given more data, you can reference more points when using non-parametric methods, which allows for better performance. While there are concern surrounding dimensionality, there are methods of selecting important features and reducing dimensionality that make it more desirable to have more data rather than less.

Problem 3

ISLR Chapter 4 Conceptual Exercise 5

(a) Even if the decision boundary is linear, we expect the more flexible QDA to perform better on the training set. However, the LDA would more closely model the linear decision boundaries on new data and would be expected to perform better on the testing set.

(b) If the decision boundary is non-linear, we expect the QDA to pick up on more of the complex relationship than the LDA and to perform better on both the training and testing set.

(c) In general, we would expect the test prediction accuracy of QDA to increase relative to LDA as n increases because there is more data to extract the more complex relationships between observations.

(d) False, QDA could over fit the training data and perform worse on the test set.

ISLR Chapter 4 Conceptual Exercise 8

In k=1 KNN, the training error is 0 because each point is perfectly matched. This means that the test error is 36%. Given this information, we should use the logistic regression as it only has a test error of 30%.

ISLR Chapter 4 Conceptual Exercise 12

(a) The log odds of orange versus apple is $\hat{\beta}_0 + \hat{\beta}_1 x$.

(b) The log odds of our friend's model is $(\hat{\alpha}_{orange0} - \hat{\alpha}_{apple0}) + (\hat{\alpha}_{orange1} - \hat{\alpha}_{apple1})x$

(c) There is no way to determine the specific value of each parameter. However,given the above answers, our friends' model would have $(\hat{\alpha}_{orange0} - \hat{\alpha}_{apple0}) = 2$ and $(\hat{\alpha}_{orange1} - \hat{\alpha}_{apple1}) = -1$.

(d) My estimates would be $\hat{\beta}_0 = (\hat{\alpha}_{orange0} - \hat{\alpha}_{apple0}) = 1.2 - 3 = -1.8$ and $\hat{\beta}_1 = (\hat{\alpha}_{orange1} - \hat{\alpha}_{apple1}) = -2 - 0.6 = -2.6$.

(e) The models are identical but just expressed differently so they should agree every time.

Problem 4

ISLR Chapter 4 Applied Exercise 14

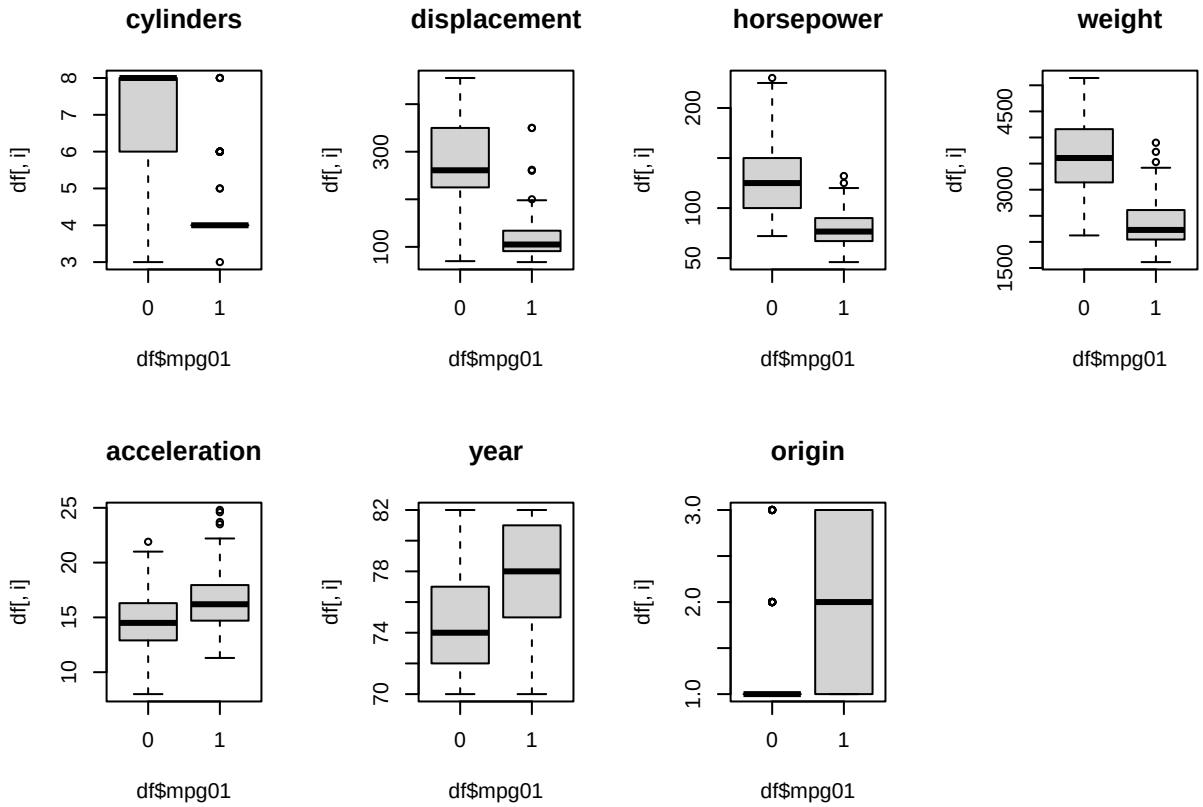
(a)

```
#Load ISLR 2 Library
library(ISLR2)
#Create mpg01
mpg01<- rep(0,392)
mpg01[Auto$mpg > median(Auto$mpg)] <- 1
#Create data frame with Auto and mpg01
df <- data.frame(Auto[,-1], mpg01)
```

(b)

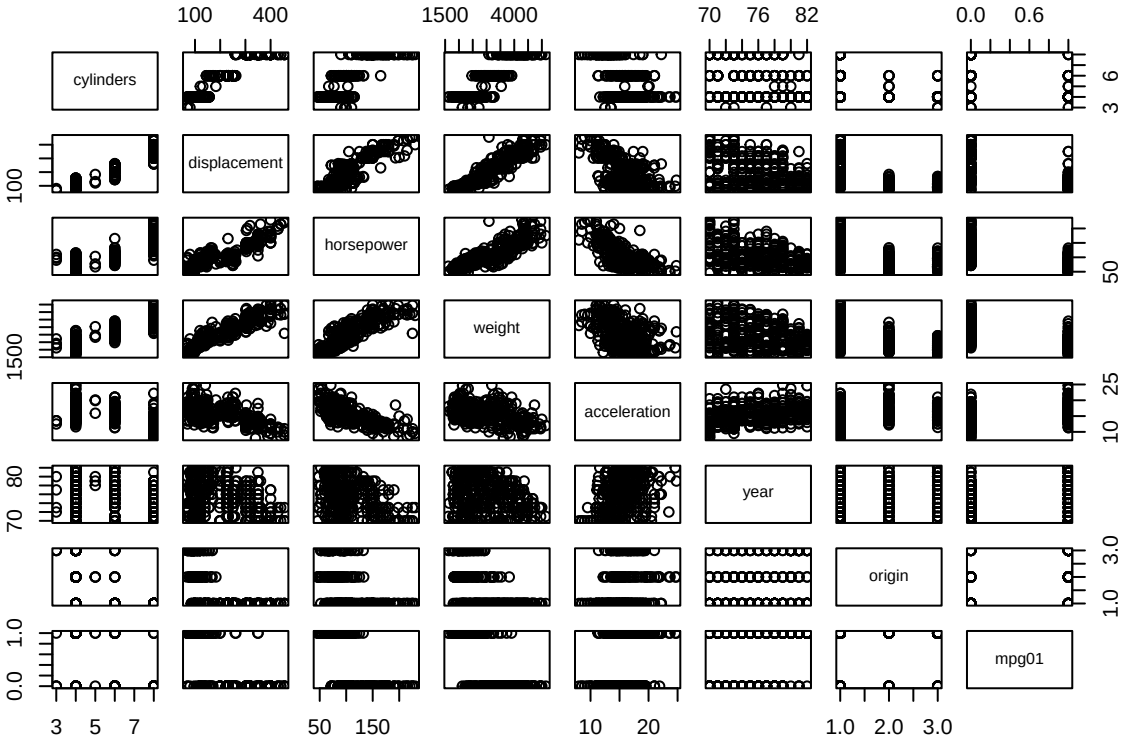
```
#Generate box plots grouped by mpg01
par(mfrow = c(2,4))
for(i in c(1:7)){
```

```
  boxplot(df[,i]~df$mpg01, main = colnames(df)[i])
}
```



It appears that there is a clear difference in feature values when the mpg is greater than the median versus not. This indicates some relationship that should be explored further.

```
#Generate pair plot
pairs(df[,c(1:7,9)])
```



There appears to be clear differences in distribution for several of the variables with *mpg*01. In particular, cylinders, weight, and displacement seem to have some relationship with *mpg*01.

(c)

```
#Set seed and split into train and test sets
set.seed(100)
sample <- sample(c(TRUE, FALSE), nrow(df),
                 replace = TRUE, prob = c(0.70, 0.30))
train <- df[sample,]
test <- df[!sample,]
```

(d)

```
#Load MASS library and run LDA using training and test sets
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:ISLR2':
##
##     Boston
```

```
lda.fit <- lda(mpg01 ~ cylinders + weight + displacement, data = train)
#Predict with LDA and calculate test error rate
lda.class <- predict(lda.fit, test)$class
mean(lda.class != test$mpg01)
```

```
## [1] 0.09565217
```

(e)

```r
#Perform QDA on train and test sets
qda.fit <- qda(mpg01 ~ cylinders + weight + displacement, data = train)
#Generate QDA predictions and calculate test error rate
qda.class <- predict(qda.fit, test)$class
mean(qda.class != test$mpg01)
```

```
## [1] 0.09565217
```

(f)

```r
#Run logistic regression
glm.fit <- glm(mpg01 ~ cylinders + weight + displacement, data = train, family = binomial)
#Generate glm predictions and calculate test error rate
glm.pred <- predict(glm.fit, test, type = "response")>0.50
mean(glm.pred != test$mpg01)
```

```
## [1] 0.1043478
```

(g)

```r
#Load library and perform naive bayes
library(e1071)
nb.fit <- naiveBayes(mpg01 ~ cylinders + weight + displacement, data = train)
#Generate nb predictions and calculate test error rate
nb.pred <- predict(nb.fit, test)
mean(nb.pred != test$mpg01)
```

```
## [1] 0.1043478
```

```r
#Load library and perform KNNs
library(class)
set.seed(500)
knn.fit1 <- knn(train[,c(1,2,4)], test[,c(1,2,4)], train$mpg01, k = 1)
knn.fit2 <- knn(train[,c(1,2,4)], test[,c(1,2,4)], train$mpg01, k = 5)
knn.fit3 <- knn(train[,c(1,2,4)], test[,c(1,2,4)], train$mpg01, k = 10)
knn.fit4 <- knn(train[,c(1,2,4)], test[,c(1,2,4)], train$mpg01, k = 20)
#Calculate test error rates
mean(knn.fit1 != test$mpg01)
```

```
## [1] 0.1391304
```

```r
mean(knn.fit2 != test$mpg01)
```

```
## [1] 0.1130435
```

```r
mean(knn.fit3 != test$mpg01)
```

```
## [1] 0.1217391
```

```r
mean(knn.fit4 != test$mpg01)
```

```
## [1] 0.1217391
```

It appears that a value of k = 5 performs best on the test set with the above parameters.

Problem 5

(a) There appears to be a heavy bias towards admitting male applicants at UCB based on the first graphic. It appears that around 2/3 of the admits are male despite the rejected pool being nearly even between men and women.

```r
#Male admission rate
1198/(1198+1493)
```

```
## [1] 0.4451877
```

```r
#Female admission rate
557/(557+1278)
```

```
## [1] 0.3035422
```

From the above calculations, men nearly had a 15% higher acceptance rate compared to women.

(b) These plots show a different story. Here, it appears that in Departments A and B, there were very few women who even applied while the majority of applicants were men (as shown through acceptances and rejections). The other departments either have a relatively even amount of applicants and acceptances across men and women (D and F) or have more women applicants and acceptances (C and E).

(c) When looking at overall admissions, it appears that UCB is biased towards admitting men. However, when broken down by department that bias seems to disappear.

(d) Women could be applying to more selective departments with smaller sizes whereas men are applying to larger departments with higher acceptance rates. This would cause there to be more men than women overall without biased admissions in any one department at UCB.

(e)

```r
#Create UCB admissions data frame
data(UCBAdmissions)
Adm <- as.integer(UCBAdmissions)[(1:(6*2))*2-1]
Rej <- as.integer(UCBAdmissions)[(1:(6*2))*2]
Dept <- gl(6,2,6*2,labels=c("A","B","C","D","E","F"))
Sex <- gl(2,1,6*2,labels=c("Male","Female"))
Ratio <- Adm/(Rej+Adm)
berk <- data.frame(Adm,Rej,Sex,Dept,Ratio)
head(berk)
```

```
##   Adm Rej    Sex Dept     Ratio
## 1 512 313   Male    A 0.6206061
## 2  89  19 Female    A 0.8240741
## 3 353 207   Male    B 0.6303571
## 4  17   8 Female    B 0.6800000
## 5 120 205   Male    C 0.3692308
## 6 202 391 Female    C 0.3406408
```

```r
#Perform logistic regression on data
LogReg.gender <- glm(cbind(Adm, Rej) ~ Sex, data = berk, family = binomial("logit"))
summary(LogReg.gender)
```

```
##
## Call:
## glm(formula = cbind(Adm, Rej) ~ Sex, family = binomial("logit"),
##     data = berk)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.22013    0.03879  -5.675 1.38e-08 ***
## SexFemale   -0.61035    0.06389  -9.553  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 877.06  on 11  degrees of freedom
## Residual deviance: 783.61  on 10  degrees of freedom
## AIC: 856.55
##
## Number of Fisher Scoring iterations: 4
```

The above regression indicates that being female has a highly statistically significant negative effect on an applicants probability of admission (-0.61).

(f)

```
#Refit logistic regression using Sex and Department
LogReg.genDep <- glm(cbind(Adm, Rej) ~ Sex + Dept, data = berk, family = binomial("logit"))
summary(LogReg.genDep)
```

```
##
## Call:
## glm(formula = cbind(Adm, Rej) ~ Sex + Dept, family = binomial("logit"),
##     data = berk)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.58205    0.06899   8.436   <2e-16 ***
## SexFemale    0.09987    0.08085   1.235    0.217
## DeptB       -0.04340    0.10984  -0.395    0.693
## DeptC       -1.26260    0.10663 -11.841   <2e-16 ***
## DeptD       -1.29461    0.10582 -12.234   <2e-16 ***
## DeptE       -1.73931    0.12611 -13.792   <2e-16 ***
## DeptF       -3.30648    0.16998 -19.452   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 877.056  on 11  degrees of freedom
## Residual deviance:  20.204  on  5  degrees of freedom
## AIC: 103.14
##
## Number of Fisher Scoring iterations: 4
```

The coefficient of *SexFemale* become positive with a large p-value, meaning that it is not statistically significantly different from zero. We also see several of the departments having a statistically significant negative effect on probability of admission, indicating that department selectiveness plays a significant role in the distribution of genders at UCB. This indicates some type of selection across genders into specific departments. Overall, we've shown that by including potential confounding variables into a regression, our variable of interest's coefficient can completely flip and even lose its statistical significance, highlighting the importance of thinking through the research problem completely and having sound logic when designing a regression.