Filippo Riva
MT4420

Assignment 4

1. Predicting Acceleration.

a. For the Auto data set, use the lm command to find a least squares linear model for acceleration as a function of horsepower and weight. Include your commands and output from the lm summary command, and state your model in the form

$$a = f(h,w) = \hat{\beta}_0 + \hat{\beta}_1 h + \hat{\beta}_2 w .$$

CODE:

```
Auto = read.csv(file="Auto.csv",head=TRUE,sep=",",stringsAsFactors=FALSE, na.strings="?")
Auto = na.omit(Auto)
acceleration=Auto$acceleration
horsepower=Auto$horsepower
weight=Auto$weight
accelerationAFhorsepowerANDweight=lm(acceleration~horsepower+weight)
summary(accelerationAFhorsepowerANDweight)
```

```
Call:
lm(formula = acceleration ~ horsepower + weight)

Residuals:
    Min      1Q  Median      3Q     Max
-4.2802 -1.1236 -0.2544  0.9128  7.1814

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) 18.4357912  0.3264888   56.47   <2e-16 ***
horsepower  -0.0933128  0.0045628  -20.45   <2e-16 ***
weight       0.0023018  0.0002068   11.13   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.745 on 389 degrees of freedom
Multiple R-squared:  0.6018,    Adjusted R-squared:  0.5998
F-statistic:   294 on 2 and 389 DF,  p-value: < 2.2e-16
```

LINE : a = f (h,w) = 18.436 – 0.093 h + 0.002 w

b. Discuss the coefficients βˆ1 and βˆ2. Does the sign (pos or neg) of β^1 make sense in terms of horsepower and acceleration? Does the sign of β^2 make sense in terms of weight and acceleration? Explain/justify your answers.

Analyzing βˆ1 the negative sign makes sense in the analysis showing cars with high acceleration will have smaller horsepower than cars with higher horsepower.

Analyzing βˆ2, The positive sign makes sense and it tells us that cars with higher weights will reach higher accelerations than cars with low weights

c. What does the p-value for the F-statistic tell you about your model? It may help to review pages 75 and 76.
The F-statistic test has a null hypothesis stating that neither one of the predictors has a relationship with the response. However, since the p-value of the F-static is a really small number is possible to reject the null hypothesis stating that there is some relationships between the predictors and the response in our dataset

d. Use the predict() function (page 113) to predict the acceleration for the following cars. For each, give its predicted acceleration, its actual acceleration, and the error between the two (perhaps make a table).
- **Ford Torino**

Code:
```
nam=Auto$name
print(nam[5])
p=predict(accelerationAFhorsepowerANDweight,data.frame(horsepower=horsepower[5],weight=weight[5]))
print(p)
print(acceleration[5])
print(abs(acceleration[5]-p))
```

```
Out[4]: [1] "ford torino"
                1
        13.31098
        [1] 10.5
                1
        2.810981
```

Predicted Acceleration: 13.31098
Actual Acceleration: 10.5
Error: 2.810981

- **Saab 99e**

Code:
```
nam=Auto$name

print(nam[23])
p=predict(accelerationAFhorsepowerANDweight,data.frame(horsepower=horsepower[23],weight=weight[23]))
print(p)
print(acceleration[23])
print(abs(acceleration[23]-p))
```

```
Out[5]: [1] "saab 99e"
             1
        15.0379
        [1] 17.5
             1
        2.462099
```

Predicted Acceleration: 15.0379
Actual Acceleration: 17.5
Error: 2.462099

2. Using the Boston data set, we will find some models for medv.

a. **Use all the variables as ISL does on page 115. Call this Model 1. View the output of the summary command, and make sure you get the same output. To read in the data and get rid of the variable X, I used these commands:**
Boston=read.csv("Boston.csv",header=T,na.strings="?",stringsAsFactors=FALSE)
Boston=Boston[,-1] # Keep all the rows, but remove first column containing X.

MODEL 1
Code:

```
Boston=read.csv("Boston.csv",header=T,na.strings="?",stringsAsFactors=FALSE)
Boston=Boston[,-1]
lm.fit = lm(medv ~ ., data = Boston)
summary(lm.fit)
```

```
Call:
lm(formula = medv ~ ., data = Boston)

Residuals:
     Min       1Q   Median       3Q      Max
-15.1304  -2.7673  -0.5814   1.9414  26.2526

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  41.617270   4.936039   8.431 3.79e-16 ***
crim         -0.121389   0.033000  -3.678 0.000261 ***
zn            0.046963   0.013879   3.384 0.000772 ***
indus         0.013468   0.062145   0.217 0.828520
chas          2.839993   0.870007   3.264 0.001173 **
nox         -18.758022   3.851355  -4.870 1.50e-06 ***
rm            3.658119   0.420246   8.705  < 2e-16 ***
age           0.003611   0.013329   0.271 0.786595
dis          -1.490754   0.201623  -7.394 6.17e-13 ***
rad           0.289405   0.066908   4.325 1.84e-05 ***
tax          -0.012682   0.003801  -3.337 0.000912 ***
```

```
ptratio      -0.937533   0.132206  -7.091 4.63e-12 ***
lstat        -0.552019   0.050659 -10.897  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.798 on 493 degrees of freedom
Multiple R-squared:  0.7343,    Adjusted R-squared:  0.7278
F-statistic: 113.5 on 12 and 493 DF,  p-value: < 2.2e-16
```

b. Let’s do some backward selection (p. 79) to revise Model 1. You can see how to remove variables from this model on page 116.
From the output for Model 1, we see that predictors indus, chas, and age do not have 3 stars (thus they have the largest p-values). Consider two more models:

• Model 2: Remove indus and age from Model 1.
CODE:
lm.fit2 = lm(medv ~ . - age - indus, data = Boston)
summary(lm.fit2)

```
Call:
lm(formula = medv ~ . - age - indus, data = Boston)

Residuals:
     Min       1Q   Median       3Q      Max
-15.1814  -2.7625  -0.6243   1.8448  26.3920

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  41.451747   4.903283   8.454 3.18e-16 ***
crim         -0.121665   0.032919  -3.696 0.000244 ***
zn            0.046191   0.013673   3.378 0.000787 ***
chas          2.871873   0.862591   3.329 0.000935 ***
nox         -18.262427   3.565247  -5.122 4.33e-07 ***
rm            3.672957   0.409127   8.978  < 2e-16 ***
dis          -1.515951   0.187675  -8.078 5.08e-15 ***
rad           0.283932   0.063945   4.440 1.11e-05 ***
tax          -0.012292   0.003407  -3.608 0.000340 ***
ptratio      -0.930961   0.130423  -7.138 3.39e-12 ***
lstat        -0.546509   0.047442 -11.519  < 2e-16 ***
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.789 on 495 degrees of freedom
Multiple R-squared:  0.7342,    Adjusted R-squared:  0.7289
F-statistic: 136.8 on 10 and 495 DF,  p-value: < 2.2e-16
```

• Model 3: Remove chas from Model 2.

CODE:

```
lm.fit3 = lm(medv ~ . - age - indus - chas, data = Boston)
summary(lm.fit3)
```

```
Call:
lm(formula = medv ~ . - age - indus - chas, data = Boston)

Residuals:
     Min       1Q   Median       3Q      Max
-12.8021  -2.7866  -0.6165   2.0405  26.6133

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  42.003257   4.950052   8.485 2.50e-16 ***
crim         -0.128304   0.033191  -3.866 0.000126 ***
zn            0.046100   0.013811   3.338 0.000908 ***
nox         -17.346284   3.590566  -4.831 1.81e-06 ***
rm            3.712126   0.413094   8.986  < 2e-16 ***
dis          -1.552476   0.189249  -8.203 2.02e-15 ***
rad           0.300012   0.064407   4.658 4.11e-06 ***
tax          -0.013267   0.003428  -3.870 0.000124 ***
ptratio      -0.963988   0.131360  -7.339 8.89e-13 ***
lstat        -0.553587   0.047874 -11.563  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.837 on 496 degrees of freedom
Multiple R-squared:  0.7283,    Adjusted R-squared:  0.7234
F-statistic: 147.7 on 9 and 496 DF,  p-value: < 2.2e-16
```

Complete the table below.

MODEL	RSE	Adjusted R-squared
1	4.798	0.7278
2	4.789	0.7289
3	4.837	0.7234

Of the 3 models considered, state which model you think is best.
Include your code and output for each model.

Out of the three models, model 2 is the best because it has the lowest Residual Standard Error and the highest Adjusted R-squared.