FE590. Assignment #2.

2019-03-13

I pledge on my honor that I have not given or received any unauthorized assistance on this assignment/examination. I further pledge that I have not copied any material from a book, article, the Internet or any other source except where I have expressly cited the source. Signature ____Yifu He_____ Date:***03/03/2019***_____

Instructions

In this assignment, you should use R markdown to answer the questions below. Simply type your R code into embedded chunks as shown above. When you have completed the assignment, knit the document into a PDF file, and upload both the .pdf and .Rmd files to Canvas.

```
CWID = 10442277 #Place here your Campus wide ID number, this will personalize
#your results, but still maintain the reproduceable nature of using seeds.
#If you ever need to reset the seed in this assignment, use this as your seed
#Papers that use -1 as this CWID variable will earn 0's so make sure you
change
#this value before you submit your work.
personal = CWID %% 10000
set.seed(personal)#You can reset the seed at any time in your code, but
please always set it to this seed.
```

Question 1

Use the Auto data set from the textbook's website. When reading the data, use the options as.is = TRUE and na.strings="?". Remove the unavailable data using the na.omit() function.

```
#insert r code here
url <- "http://www-bcf.usc.edu/~gareth/ISL/Auto.data"
Autodata <- read.table(url,
                       header = TRUE,
                       as.is= TRUE,
                       na.strings = "?",
                       sep = "")
Autodata = na.omit(Autodata)
```

1. List the names of the variables in the data set.

```
#insert r code here
colnames(Autodata)
```

```
## [1] "mpg"          "cylinders"    "displacement" "horsepower"
## [5] "weight"       "acceleration" "year"         "origin"
## [9] "name"
```

2. The columns origin and name are unimportant variables. Create a new data frame called cars that contains none of these unimportant variables

```
#insert r code here
myvars <- names(Autodata) %in% c("origin", "name")
cars <- Autodata[!myvars]
colnames(cars)
```

```
## [1] "mpg"          "cylinders"    "displacement" "horsepower"
## [5] "weight"       "acceleration" "year"
```

3. What is the range of each quantitative variable? Answer this question using the range() function with the sapply() function e.g., sapply(cars, range). Print a simple table of the ranges of the variables. The rows should correspond to the variables. The first column should be the lowest value of the corresponding variable, and the second column should be the maximum value of the variable. The columns should be suitably labeled.

```
#insert r code here
var.range = sapply(cars,range)
minimun = var.range[1,]
maximum = var.range[2,]
table.range = data.frame(minimun,maximum)
table.range
```

```
##              minimun maximum
## mpg                9    46.6
## cylinders          3     8.0
## displacement      68   455.0
## horsepower        46   230.0
## weight          1613  5140.0
## acceleration       8    24.8
## year              70    82.0
```

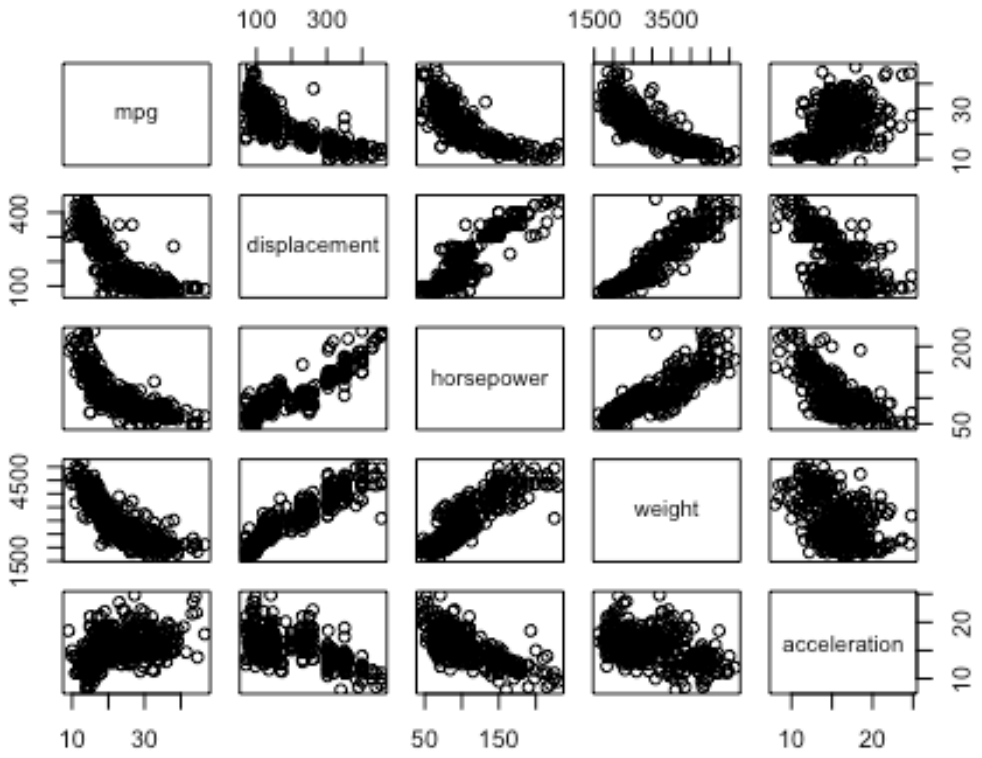
4. What is the mean and standard deviation of each variable? Create a simple table of the means and standard deviations.

```
#insert r code here
mean.value <- sapply(cars,mean)
sd.value <- sapply(cars,sd)
table.statistic <- data.frame(mean.value, sd.value)
table.statistic
```

```
##                mean.value   sd.value
## mpg             23.445918   7.805007
## cylinders        5.471939   1.705783
## displacement   194.411990 104.644004
## horsepower     104.469388  38.491160
## weight        2977.584184 849.402560
## acceleration    15.541327   2.758864
## year            75.979592   3.683737
```

5. Create a scatterplot matrix that includes the variables mpg, displacement, horsepower, weight, and acceleration using the pairs() function.

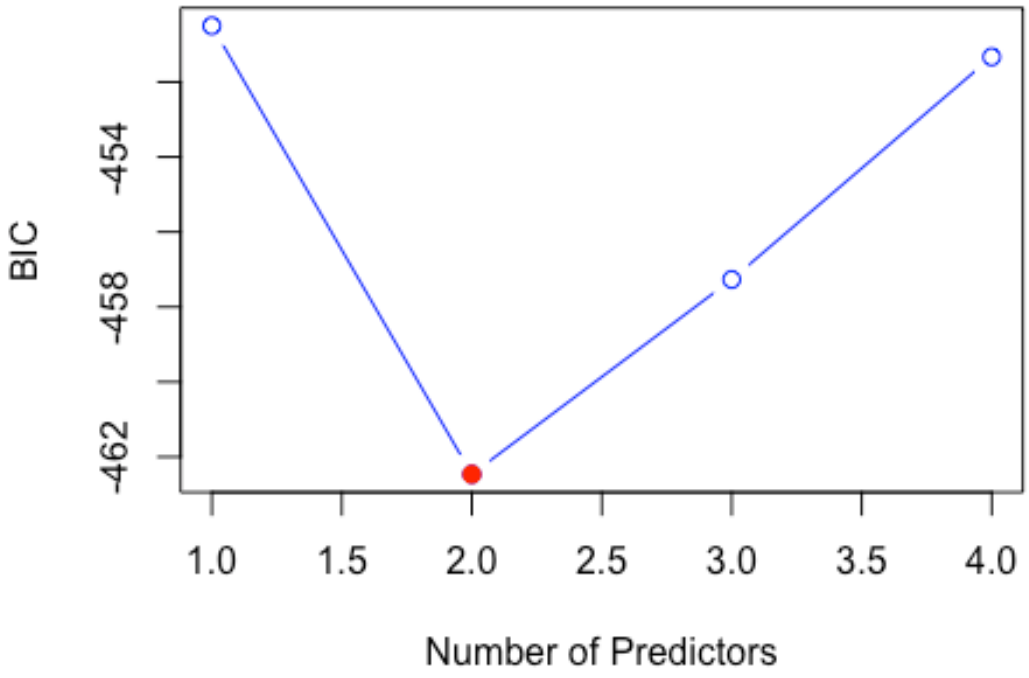
```
#insert r code here
pairs(~mpg+displacement+horsepower+weight+acceleration,cars)
```



6. Using the function in the leaps library, regress mpg onto

```
#insert r code here
library(leaps)
reg <- regsubsets(mpg~ displacement+ horsepower + weight + acceleration,
                          data = cars,
                          method="exhaustive")
```

```
bic = summary(reg)$bic
i = which.min(summary(reg)$bic)
plot(bic,type='b',col="blue",xlab="Number of
Predictors",ylab=expression("BIC"))
points(i,bic[i],pch = 19,col = "red")
```




```
t(summary(reg)$outmat)
```

```
##              1  ( 1 ) 2  ( 1 ) 3  ( 1 ) 4  ( 1 )
## displacement " "      " "      "*"      "*"
## horsepower   " "      "*"      "*"      "*"
## weight       "*"      "*"      "*"      "*"
## acceleration " "      " "      " "      "*"
```

```
summary(reg)$adjr2[i]
```

```
## [1] 0.7048656
```

According to the outcome of subset selection, the best model has 2 predictor according to the criterion of BIC, which are horsepower and weight.It's adjusted R-square is 0.7048656, which means the multi-linear model can explain 70.49% of the data.

7. Print a table showing what variables would be selected using best subset selection for all predictors (displacement, horsepower, weight, acceleration) up to order 2 (i.e. weight and weight^2).

```
#insert r code here
reg2 <- regsubsets(mpg~ displacement+ horsepower + weight +
acceleration+I(displacement^2)
                   + I(horsepower^2)+ I(weight^2)+I(acceleration^2),
                         data = cars,
                         method="exhaustive")
t(summary(reg2)$which)
```

```
##                      1     2     3     4     5     6     7    8
## (Intercept)       TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE TRUE
## displacement     FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE TRUE
## horsepower       FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE TRUE
## weight            TRUE  TRUE  TRUE FALSE FALSE  TRUE  TRUE TRUE
## acceleration     FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE TRUE
## I(displacement^2) FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE TRUE
## I(horsepower^2)   FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE TRUE
## I(weight^2)       FALSE  TRUE FALSE FALSE FALSE FALSE FALSE TRUE
## I(acceleration^2) FALSE FALSE FALSE FALSE FALSE FALSE  TRUE TRUE
```

a. What is the most important variable affecting fuel consumption?

```
#insert r code here
print("Most important variable: weight")
```

```
## [1] "Most important variable: weight"
```

b. What is the second most important variable affecting fuel consumption?

```
#insert r code here
print("With the use of best subset selection, with weight as the baseline,
weight^2 would be the second most important variable as it provide the most
additional explanation for the regression comparing to others. This choice of
variable would may change based on different method use.")
```

```
## [1] "With the use of best subset selection, with weight as the baseline,
weight^2 would be the second most important variable as it provide the most
additional explanation for the regression comparing to others. This choice of
variable would may change based on different method use."
```

c. What is the third most important variable affecting fuel consumption?

```
#insert r code here
print("Third most important variable: horsepower or horsepower^2")
```

```
## [1] "Third most important variable: horsepower or horsepower^2"
```

Question 2

This exercise involves the Boston housing data set.

1. Load in the Boston data set, which is part of the MASS library in R. The data set is contained in the object Boston. Read about the data set using the command ?Boston. How many rows are in this data set? How many columns? What do the rows and columns represent?

```
#insert r code here
library("MASS")
?Boston
length(rownames(Boston))

## [1] 506

length(colnames(Boston))

## [1] 14

print("The Boston data frame has 506 rows and 14 columns. ")

## [1] "The Boston data frame has 506 rows and 14 columns. "

print("Rows represent different observations which are towns in Boston.")

## [1] "Rows represent different observations which are towns in Boston."

print("Columns represent different variables which are different aspects of
the town.")

## [1] "Columns represent different variables which are different aspects of
the town."
```

The Boston data frame has 506 rows and 14 columns.

2. Do any of the suburbs of Boston appear to have particularly high crime rates?

The method to determines particularly high observation in this question is the method of defining outlier based on Inter Quartile Range (IQR): $Q_1 - 1.5IQR$ or $Q_3 + 1.5IQR$ with $IQR = Q_3 - Q_1$. We will use the upper outlier for these questions

```
#insert r code here
attach(Boston)
upcrim=which(crim>(quantile(crim,0.75)+1.5*IQR(crim)))
length(upcrim)

## [1] 66
```

```
print("There exists 66 suburbs in which their crime rates are particularly higher than those of the rest of Boston")
```

```
## [1] "There exists 66 suburbs in which their crime rates are particularly higher than those of the rest of Boston"
```

```
detach(Boston)
```

Tax rates?

```
#insert r code here
attach(Boston)
uptax=which(tax>(quantile(tax,0.75)+1.5*IQR(tax)))
length(uptax)
```

```
## [1] 0
```

```
print("For tax rates, there is suburbs with higher than average rates in Boston but those rates do not guarantee to be considered as particularly high rates because they do not cross the 1.5 IQR threshold")
```

```
## [1] "For tax rates, there is suburbs with higher than average rates in Boston but those rates do not guarantee to be considered as particularly high rates because they do not cross the 1.5 IQR threshold"
```

```
detach(Boston)
```

Pupil-teacher ratios?

```
#insert r code here
attach(Boston)
upptr=which(ptratio>(quantile(ptratio,0.75)+1.5*IQR(ptratio)))
length(upptr)
```

```
## [1] 0
```

```
print("There is no suburbs with particularly high pupil - teacher ratio as all of the ratios stay inside the 1.5 IQR rule.")
```

```
## [1] "There is no suburbs with particularly high pupil - teacher ratio as all of the ratios stay inside the 1.5 IQR rule."
```

```
detach(Boston)
```

Comment on the range of each predictor.

```
#calculate the mean of crim, tax,ptratio
mean=sapply(Boston[,c(1,10,11)],mean)
#calculate range of crim, tax,ptratio
range=sapply(Boston[,c(1,10,11)],range)
range
```

```
##          crim tax ptratio
## [1,]  0.00632 187    12.6
## [2,] 88.97620 711    22.0
```

```
#taking out the sd of crim, tax,ptratio
sd=sapply(Boston[,c(1,10,11)],sd)
#calculate z-score for crim, tax,ptratio
rbind((range[c(1,3,5)]-mean)/sd,(range[c(2,4,6)]-mean)/sd)
```

```
##            crim       tax   ptratio
## [1,] -0.4193669 -1.312691 -2.704703
## [2,]  9.9241096  1.796416  1.637208
```

As shown, tax has a relatively clustered range with only under -1.31σ to 1.8σ even though its range of absolute values are the highest $711 - 187 = 524$.

The pupil - teacher ratio has the lowest absolute value for its range of value with only 7.4. When change to z-score, the range of this ratio is also seems to have a limited upside while having a particularly low downside with a value of -2.7σ.

Finally, the criminal rate is the most fluctuated parameters in all three with a range of -0.42σ to 9.92σ. The range suggest that there is no particularly lower than average crime rate but the exist extreme upsides.

3. How many of the suburbs in this data set bound the Charles river?

```
#insert r code here
sum(Boston["chas"] == 1)
```

```
## [1] 35
```

4. What is the median pupil-teacher ratio among the towns in this data set?

```
#insert r code here
median(sapply(Boston["ptratio"], as.numeric))
```

```
## [1] 19.05
```

5. In this data set, how many of the suburbs average more than seven rooms per dwelling?

```
#insert r code here
sum(Boston["rm"]> 7)
```

```
## [1] 64
```

More than eight rooms per dwelling?

```
#insert r code here
sum(Boston["rm"]> 8)
```

```
## [1] 13
```

Question 3

This question should be answered using the Weekly data set, which is part of the ISLR package. This data contains 1,089 weekly returns for 21 years, from the beginning of 1990 to the end of 2010.

1. What does the data represent?

```
#insert r code here
library("ISLR")
?Weekly
print("The data is weekly percentage returns for the S&P 500 stock index
between 1990 and 2010. ")
```

```
## [1] "The data is weekly percentage returns for the S&P 500 stock index
between 1990 and 2010. "
```

2. Use the full data set to perform a logistic regression with Direction as the response and the five lag variables plus Volume as predictors. Use the summary function to print the results. Do any of the predictors appear to be statistically significant? If so, which ones?

```
#insert r code here
glm.fit=glm(Direction~Lag1+Lag2+Lag3+Lag4+Lag5+Volume,family=binomial,data=We
ekly)
summary(glm.fit)
```

```
##
## Call:
## glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +
##     Volume, family = binomial, data = Weekly)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6949  -1.2565   0.9913   1.0849   1.4579
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.26686    0.08593   3.106   0.0019 **
## Lag1        -0.04127    0.02641  -1.563   0.1181
## Lag2         0.05844    0.02686   2.175   0.0296 *
## Lag3        -0.01606    0.02666  -0.602   0.5469
## Lag4        -0.02779    0.02646  -1.050   0.2937
## Lag5        -0.01447    0.02638  -0.549   0.5833
## Volume      -0.02274    0.03690  -0.616   0.5377
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
```

```
##     Null deviance: 1496.2  on 1088  degrees of freedom
## Residual deviance: 1486.4  on 1082  degrees of freedom
## AIC: 1500.4
##
## Number of Fisher Scoring iterations: 4
```

"Lag 2" is the only statistically significant predictor at 5% confidence level

3. Fit a logistic regression model using a training data period from 1990 to 2008, using the predictors from the previous problem that you determined were statistically significant. Test your model on the held out data (that is, the data from 2009 and 2010) and express its accuracy.

```
#insert r code here
attach(Weekly)
train=(Year<=2008)
trainq3=Weekly[train,]
testq3=Weekly[!train,]
no=nrow(testq3)
glm.fit2=glm(Direction~Lag2,family=binomial,data=trainq3)
glm.fit2t=predict(glm.fit2,testq3)
glm.pred=rep("Down",no)
glm.pred[glm.fit2t>.5]="Up"
logmean=mean(glm.pred==testq3$Direction)
logmean

## [1] 0.4423077

detach(Weekly)
```

4. Repeat Part 3 using LDA.

```
#insert r code here
attach(Weekly)
lda.fit=lda(Direction~Lag2,data=trainq3)
lda.fitt=predict(lda.fit,testq3)
ldamean=mean(lda.fitt$class==testq3$Direction)
ldamean

## [1] 0.625

detach(Weekly)
```

5. Repeat Part 3 using QDA.

```
#insert r code here
attach(Weekly)
qda.fit=qda(Direction~Lag2,data=trainq3)
qda.fitt=predict(qda.fit,testq3)
```

```r
qdamean=mean(qda.fitt$class==testq3$Direction)
qdamean

## [1] 0.5865385

detach(Weekly)
```

6. Repeat Part 3 using KNN with K = 1, 2, 3.

```r
#insert r code here
library("class")
attach(Weekly)
knntrain=Lag2[train]
knntest=Lag2[!train]
train.Direction=Direction[train]
knnmean=rep(0,3)
for(i in c(1:3)){
  knn.pred=knn(data.frame(knntrain),data.frame(knntest),train.Direction,k=i)
  knnmean[i]=mean(knn.pred==testq3$Direction)
}
knnmean

## [1] 0.5096154 0.5288462 0.5576923

detach(Weekly)
```

7. Which of these methods in Parts 3, 4, 5, and 6 appears to provide the best results on this data?

```r
#insert r code here
which.max(c(logmean,ldamean,qdamean,knnmean))

## [1] 2
```

As seen above, LDA has the highest accuracy of all 4 methods in predicting “Up” and “Down” based on “Lag 2”

Question 4

Write a function that works in R to gives you the parameters from a linear regression on a data set between two sets of values (in other words you only have to do the 2-D case and your output will be the coefficients beta_0 and beta_1). Include in the output the standard error of your variables. You cannot use the lm command in this function or any of the other built in regression models. For example your output could be a 2x2 matrix with the parameters in the first column and the standard errors in the second column. For up to 5 bonus points, format your output so that it displays and operates similar in function to the output of the lm command.(i.e. in a data frame that includes all potentially useful outputs)

```r
#insert r code here
diylm=function (y,x){
  xave=mean(x)
  yave=mean(y)
  tss=0
  no=length(x)
  df=no-1-1
  num=0
  den=0
  err=rep(0,no)
  tval=c(0,0)
  pval=c(0,0)
  star=c(0,0)
  rss=0
  for(i in c(1:no)){
    num=num+(x[i]-xave)*(y[i]-yave)
    den=den+(x[i]-xave)^2
    tss=tss+(y[i]-yave)^2
  }
  beta1=num/den
  beta0=yave-xave*beta1
  for(i in c(1:no)){
    err[i]=y[i]-beta0-beta1*x[i]
    rss=rss+err[i]^2
  }
  varerr=sd(err)^2
  se0=varerr*(1/no+xave^2/den)
  se1=varerr/den
  rse=sqrt(rss/df)
  rsq=1-rss/tss
  adjrsq=1-(1-rsq)*(no-1)/df
```

```
  fstat=(tss-rss)/(rss/df)
  fpval=pf(fstat,df1=1,df2=df,lower.tail = F)
  Estimate=c(beta0,beta1)
  Std.Error=c(sqrt(se0),sqrt(se1))
  for(i in c(1,2)){
    tval[i]=Estimate[i]/Std.Error[i]
    pval[i]=2*pt(abs(tval[i]),df=df,lower=F)
    if(pval[i]<=0.001){
      star[i]="***"
    } else if (0.001<pval[i]&&pval[i]<=0.01){
      star[i]="**"
    } else if (0.01<pval[i]&&pval[i]<=0.05){
      star[i]="*"
    } else if (0.05<pval[i]&&pval[i]<=0.1){
      star[i]="."
    } else {
      star[i]=" "
    }
  }
  Min=min(err)
  FirstQ=quantile(err,0,25)
  Median=median(err)
  ThirdQ=quantile(err,0.75)
  Max=max(err)
  res=data.frame(Min,FirstQ,Median,ThirdQ,Max)
  total=data.frame(Estimate,Std.Error,tval,pval,star)
  rownames(total,c("(Intercept)","Dependent Variables"))
  print(res)
  print(total)
  cat(sprintf("Residual standard error: %.3f",rse))
  cat(sprintf(" on %.2f degrees of freedom\n",df))
  cat(sprintf("Multiple R-squared:  %.4f",rsq))
  cat(sprintf(", Adjusted R-squared:  %.4f \n",adjrsq))
  cat(sprintf("F-statistic: %.2f",fstat))
  cat(sprintf(" on 1 and %f DF",df))
  cat(sprintf(" p-value: %.4f",fpval))
}
```

Compare the output of your function to that of the lm command in R.

```
#insert r code here
attach(cars)
lm1=lm(mpg~weight,data = cars)
summary(lm1)

##
## Call:
## lm(formula = mpg ~ weight, data = cars)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -11.9736  -2.7556  -0.3358   2.1379  16.5194
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 46.216524   0.798673   57.87   <2e-16 ***
## weight      -0.007647   0.000258  -29.64   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.333 on 390 degrees of freedom
## Multiple R-squared:  0.6926, Adjusted R-squared:  0.6918
## F-statistic: 878.8 on 1 and 390 DF,  p-value: < 2.2e-16

diylm(mpg,weight)

##          Min    FirstQ     Median   ThirdQ      Max
## 0% -11.97357 -11.97357 -0.3358445 2.137901 16.51937
##       Estimate    Std.Error      tval          pval star
## 1 46.216524549 0.7976504892  57.94082 1.037546e-193  ***
## 2 -0.007647343 0.0002576332 -29.68306 4.254202e-102  ***
## Residual standard error: 4.333 on 390.00 degrees of freedom
## Multiple R-squared:  0.6926, Adjusted R-squared:  0.6918
## F-statistic: 878.83 on 1 and 390.000000 DF p-value: 0.0000

detach(cars)
```

Question 5

Using the Advertising data set (Sales, TV, Radio, Newspaper), do the following:

1. Randomly split the data into two different pieces of roughly equal size.

```
library(boot)
ads=read.table("http://www-bcf.usc.edu/~gareth/ISL/Advertising.csv",
               header = TRUE,
               sep = ",")
name <- names(ads) %in% c("sales","TV","radio","newspaper")
ads <- ads[name]
index <- sample(1:nrow(ads),nrow(ads)/2)
train <- ads[index,]
test <- ads[-index,]
```

2. Pick one set to run a linear regression to predict sales based on all TV and Radio, and then test your accuracy using the other set

```
attach(ads)
lm1=lm(sales~TV+radio,data = train)
mean((sales-predict(lm1,train))^2)

## [1] 43.71107

mean((sales-predict(lm1,test))^2)

## [1] 55.82538

detach(ads)
```

3. Repeat the previous problem using all three predictors (including newspaper). What do you determine from this result?

```
attach(ads)
lm2=lm(sales~TV+radio+newspaper,data = train)
mean((sales-predict(lm2,train))^2)

## [1] 43.84939

mean((sales-predict(lm2,test))^2)

## [1] 55.67032

detach(ads)
```

From (2) and (3), it can be seen that the training MSE of (2) is little bit higher the training MSE of (3) while the testing MSE of (2) is only 0.14 lower than that of (3). Thus, the MSE does not improve with significance as we add newspaper into the regression.

4. Determine the LOOCV error for the linear regression using all three predictors.

```
attach(ads)
glm.reg=glm(sales~TV+radio+newspaper,data=ads)
cv.err=cv.glm(ads,glm.reg)
cv.err$delta

## [1] 2.946900 2.946486

detach(ads)
```