Homework10_madhu

Madhu Jagdale

5/1/2021

For this document, you'll want to run the following R code chunk:

```
library(ggplot2) # graphics library
library(MASS)    # contains data sets, including Boston
library(ISLR)    # contains code and data from the textbook

## Warning: package 'ISLR' was built under R version 4.0.5

library(knitr)   # contains kable() function

options(scipen = 2)  # Suppresses scientific notation

require(ISLR)
data(Carseats)
```

Part I : Exploring predictors

(Before you answer the following, make sure that the missing values, if any, have been removed from the data.)

(a) Which of the predictors are quantitative (discrete random variables), and which are qualiative (continuous random variables)?

```
head(Carseats)

##   Sales CompPrice Income Advertising Population Price ShelveLoc Age
Education
## 1  9.50       138     73          11        276   120       Bad  42
17
## 2 11.22       111     48          16        260    83      Good  65
10
## 3 10.06       113     35          10        269    80    Medium  59
12
## 4  7.40       117    100           4        466    97    Medium  55
14
## 5  4.15       141     64           3        340   128       Bad  38
13
## 6 10.81       124    113          13        501    72       Bad  78
16
##   Urban  US
```

```
## 1   Yes Yes
## 2   Yes Yes
## 3   Yes Yes
## 4   Yes Yes
## 5   Yes  No
## 6    No Yes
```

```
str(Carseats)
```

```
## 'data.frame':    400 obs. of  11 variables:
##  $ Sales      : num  9.5 11.22 10.06 7.4 4.15 ...
##  $ CompPrice  : num  138 111 113 117 141 124 115 136 132 132 ...
##  $ Income     : num  73 48 35 100 64 113 105 81 110 113 ...
##  $ Advertising: num  11 16 10 4 3 13 0 15 0 0 ...
##  $ Population : num  276 260 269 466 340 501 45 425 108 131 ...
##  $ Price      : num  120 83 80 97 128 72 108 120 124 124 ...
##  $ ShelveLoc  : Factor w/ 3 levels "Bad","Good","Medium": 1 2 3 3 1 1 3 2
3 3 ...
##  $ Age        : num  42 65 59 55 38 78 71 67 76 76 ...
##  $ Education  : num  17 10 12 14 13 16 15 10 10 17 ...
##  $ Urban      : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 1 2 2 1 1 ...
##  $ US         : Factor w/ 2 levels "No","Yes": 2 2 2 2 1 2 1 2 1 2 ...
```

```
#Carseats = na.omit(Carseats)
```

[Quantitative predictors: Sales, Population, Age, CompPrice, Income, Advertising, Price, Education. Qualitative predictors: ShelveLoc, Urban, US]

(b) What is the range of each quantitative predictor? You can answer this using the range() function.

```
apply(Carseats[,1:11], 2, range)
```

```
##      Sales   CompPrice Income Advertising Population Price ShelveLoc Age
## [1,] " 0.00" " 77"     " 21"  " 0"        " 10"      " 24" "Bad"     "25"
## [2,] "16.27" "175"     "120"  "29"        "509"      "191" "Medium"  "80"
##      Education Urban US
## [1,] "10"      "No"  "No"
## [2,] "18"      "Yes" "Yes"
```

[Above table shows the min and max value of each quantitative predictor.]

(c) What is the mean and standard deviation of each quantitative predictor?

```
options(width = 95)
apply(Carseats[,-c(7,10,11)], 2, mean)
```

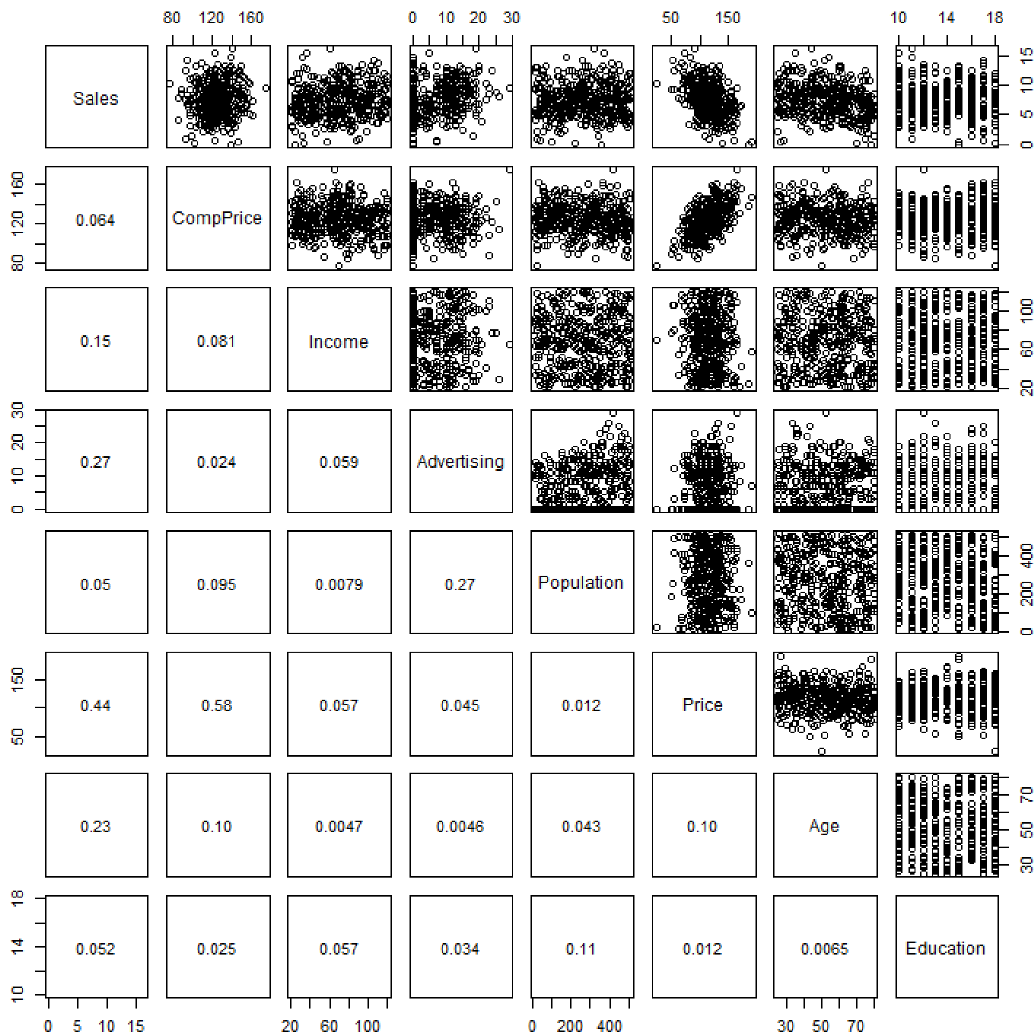
```
##       Sales   CompPrice      Income Advertising  Population       Price
Age
##    7.496325  124.975000   68.657500    6.635000  264.840000  115.795000
53.322500
##   Education
##   13.900000
```

```
apply(Carseats[,-c(7,10,11)], 2, sd)

##       Sales   CompPrice      Income Advertising  Population       Price
Age
##    2.824115   15.334512   27.986037    6.650364  147.376436   23.676664
16.200297
##   Education
##    2.620528
```

[Above both table shows the mean and standard deviation of each quantitative predictor.]

(d) Using the full data set, investigate the predictors *graphically*, using scatterplots or other tools of your choice. Create some plots highlighting the relationships among the predictors. Comment on your findings.

```
#pairs(Carseats[,1:11])

panel.cor <- function(x, y, digits = 2, prefix = "", cex.cor, ...)
{
    usr <- par("usr"); on.exit(par(usr))
    par(usr = c(0, 1, 0, 1))
    r <- abs(cor(x, y))
    txt <- format(c(r, 0.123456789), digits = digits)[1]
    txt <- paste0(prefix, txt)
    if(missing(cex.cor)) cex.cor <- 0.4/strwidth(txt)
    text(0.5, 0.5, txt, cex = pmax(1, cex.cor * r))
}

#pairs(Carseats[,c("Sales","CompPrice", "Income","Advertising", "Population",
"Price", "ShelveLoc", "Age", "Education", "Urban", "US")],lower.panel =
panel.cor)
numericdata <- Carseats[,-c(7,10,11)]
#pairs(m)

pairs(numericdata, lower.panel = panel.cor)
```




```
options(width = 80)
round(cor(Carseats[,-c(7,10,11)]), 2)

##              Sales CompPrice Income Advertising Population Price   Age
Education
## Sales         1.00      0.06   0.15        0.27       0.05 -0.44 -0.23
-0.05
## CompPrice     0.06      1.00  -0.08       -0.02      -0.09  0.58 -0.10
0.03
## Income        0.15     -0.08   1.00        0.06      -0.01 -0.06  0.00
-0.06
## Advertising  0.27     -0.02   0.06        1.00       0.27  0.04  0.00
-0.03
## Population    0.05     -0.09  -0.01        0.27       1.00 -0.01 -0.04
-0.11
## Price        -0.44      0.58  -0.06        0.04      -0.01  1.00 -0.10
0.01
```

```
## Age          -0.23     -0.10   0.00         0.00      -0.04 -0.10   1.00
0.01
## Education   -0.05      0.03  -0.06        -0.03      -0.11  0.01   0.01
1.00

#c=subset(Carseats, select=-ShelveLoc)
#b=subset(c, select=-US)
#a=subset(b, select=-Urban)
#cor(a)
```

[Above scatter plots shows the relationship among the predictors. As we can see price and sales has the most correlation between all the variables.]

(e) Suppose that we wish to predict sales of car seats (in each location) (that is, the random variable `Sales`) on the basis of the other variables. Do your plots suggest that any of the other variables might be useful in predicting `Sales`? Justify your answer

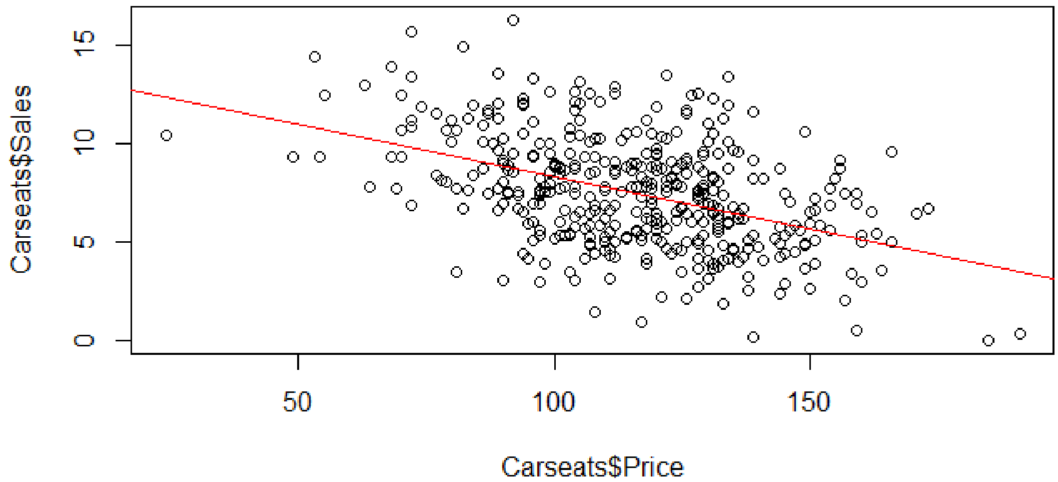
[The other variables might be useful in predicting `Sales` is price variable. We can predict that if there is a increase in the price sales decreases.]

Part II: Splitting the sample into training and sample

For this problem, continue using the `Carseats` data set.

a) Construct a scatterplot of `Sales` vs `Price`.

```
attach(Carseats)

fit.slm <- lm(Sales~Price , data=Carseats)
#qplot(data = Carseats, x = Sales, y = Price)



plot(Carseats$Price,Carseats$Sales)
abline(fit.slm, col="red")
```



[As we can see from the scatter plot if we increases the price of carseats the sale of carseats decreses.]

b) Use the `sample()` command to construct `train`, a vector of observation indexes to be used for the purpose of training your model. This will partition the data set into the *training* set and the *testing* set.
   - Describe what the `sample()` function as used above actually does.

#Splitting data into training and testing sets using 75% of sample size.

```
sampleSize <- floor(0.75 * nrow(numericdata))

set.seed(123)
trainingSet <- sample(seq_len(nrow(numericdata)), size = sampleSize)
trainingSet1 <- numericdata[trainingSet, ]
testingSet1 <- numericdata[-trainingSet, ]
head(trainingSet1)

##     Sales CompPrice Income Advertising Population Price Age Education
## 179 10.66       104     71          14         89    81  25        14
## 14  10.96       115     28          11         29    86  53        18
## 195  7.23       112     98          18        481   128  45        11
## 306  8.03       115     29          26        394   132  33        13
## 118  8.80       145     53           0        507   119  41        12
## 299 10.98       148     63           0        312   130  63        15

head(testingSet1)

##    Sales CompPrice Income Advertising Population Price Age Education
## 1   9.50       138     73          11        276   120  42        17
## 3  10.06       113     35          10        269    80  59        12
## 6  10.81       124    113          13        501    72  78        16
```

```
## 8  11.85        136      81            15          425    120  67          10
## 15 11.17        107     117            11          148    118  52          18
## 17  7.58        118      32             0          284    110  63          13
```

[When used in sample(n, size) syntax, the sample function produces a random sample of size size from 1:n. Sampling is done without replacement]

c) Valiadate the partition you obtained for the data. Do you see any issues?

  - *Hint*: this problem is *not* asking you to balance the training data set. It is instead asking you to determine *whether* balancing might be required. To determine that, you should use a hypothesis test! Which test? In `R`, there is *one function* you need to call to get the output for the comparison of two samples. Explain the answer; justify your conclusion.

```
t.test(trainingSet1,testingSet1)

## 
##  Welch Two Sample t-test
## 
## data:  trainingSet1 and testingSet1
## t = 0.32889, df = 1402.1, p-value = 0.7423
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -6.447885  9.045493
## sample estimates:
## mean of x mean of y 
##  82.27737  80.97856
```

[As we can see the mean of training set is 82.27737 and the mean of testing set is 80.97856. There is no more difference in the mean of both the sets. I think there will be no need to balance the dataset.]

Part III: fitting and evaluting a multiple linear regression model

This question should be answered using the `Carseats` data set. Use your **training** data set for fitting the model.

(a) Fit a multiple regression model to predict `Sales` using `Price`, `Urban`, and `US`.

```
lm.fit = lm(Sales ~ Price+Urban+US, data= Carseats)

summary(lm.fit)

## 
## Call:
## lm(formula = Sales ~ Price + Urban + US, data = Carseats)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -6.9206 -1.6220 -0.0564  1.5786  7.0581 
## 
```

```
## Coefficients:
##             Estimate Std. Error t value   Pr(>|t|)
## (Intercept) 13.043469   0.651012  20.036    < 2e-16 ***
## Price       -0.054459   0.005242 -10.389    < 2e-16 ***
## UrbanYes    -0.021916   0.271650  -0.081      0.936
## USYes        1.200573   0.259042   4.635 0.00000486 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.472 on 396 degrees of freedom
## Multiple R-squared:  0.2393, Adjusted R-squared:  0.2335
## F-statistic: 41.52 on 3 and 396 DF,  p-value: < 2.2e-16
```

```
kable(summary(lm.fit)$coef, digits = c(3, 3, 3, 4), format = "markdown")
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	13.043	0.651	20.036	0.0000
Price	-0.054	0.005	-10.389	0.0000
UrbanYes	-0.022	0.272	-0.081	0.9357
USYes	1.201	0.259	4.635	0.0000

```
my.output <- summary(lm.fit)
```

(b) Provide an interpretation of each coefficient in the model. Be careful—some of the variables in the model are qualitative!

[The coefficient of Price is -0.054, which is close to 0 numerically and is not statistically significant. Holding all else in the model constant, Price does not appear to have much association with Sales. In otherwords, when price increases by $1000, the number of carseats sold decrease by 54,459.] [Also urban cars does not appear to have mch association with sales because the coefficiant of urban is -0.022. A store's sale is not affected by whether or not it is in a Urban area.] [The coefficient of US is 1.201. This indicates that, all else in the model held constant, cars manufactured in the USA carry a sales tag that is on average $1.201 thousand dollars higher than cars manufactered outside the USA. The coefficient is statistically significant at the 0.05 level.A store in the US sales 1200 more carseats (in average) than a store that is abroad.]

(c) Write out the model in equation form, being careful to handle the qualitative variables properly.

[The model may be written as

Sales=13.0434689+(−0.0544588)×Price+(−0.0219162)×Urban+(1.2005727)×US +ε

with Urban=1 if the store is in an urban location and 0 if not, and US=1 if the store is in the US and 0 if not.]

(d) For which of the predictors can you reject the null hypothesis $H_0: \beta_j = 0$?

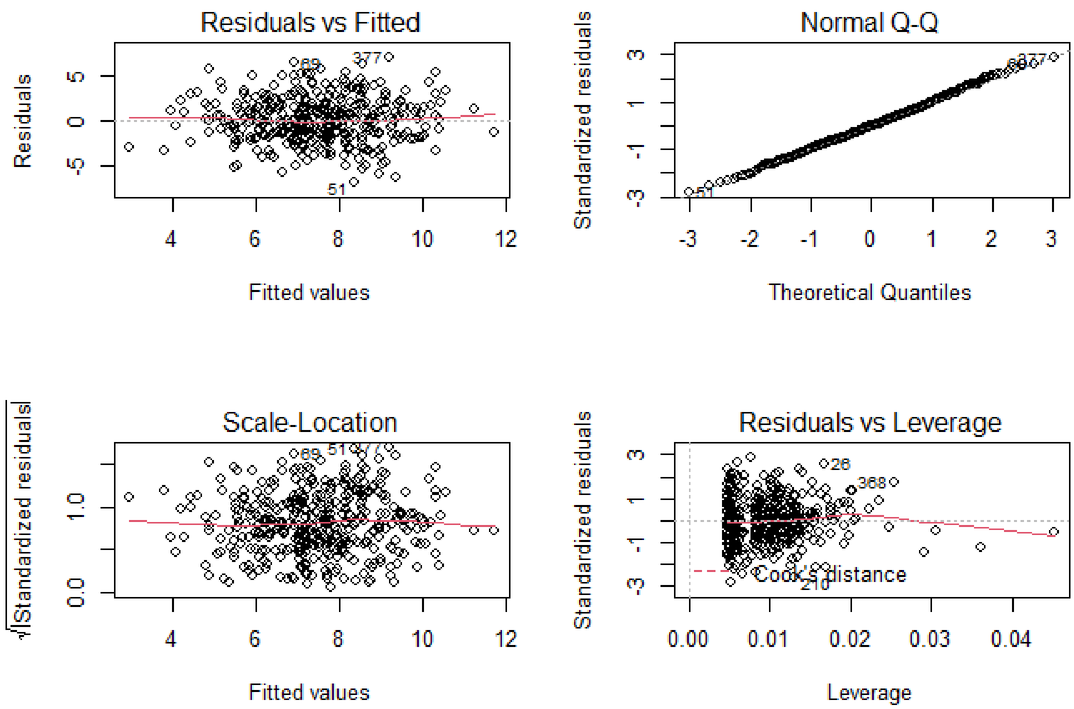
[We can reject the null hypothesis for the "Price" and "US" variables because P value are too much less for these two variables.]

(e) On the basis of your response to the previous question, fit a smaller model that only uses the predictors for which there is evidence of association with the outcome.

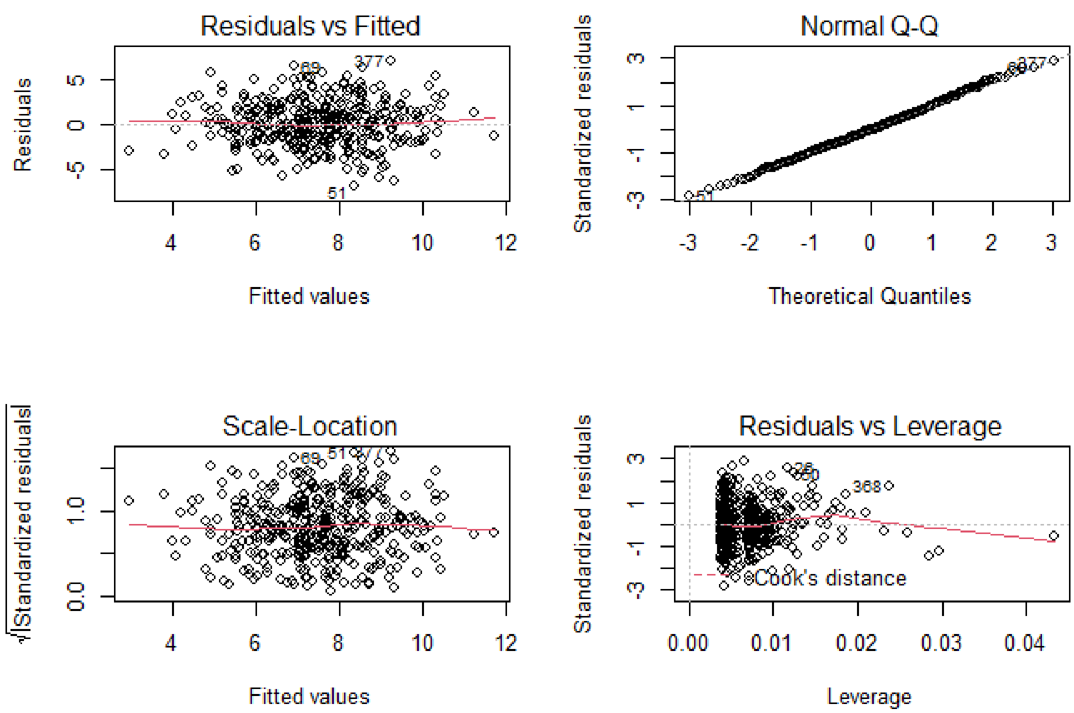
```
lm.fit2 = lm(Sales ~ Price+US, data= Carseats)
summary(lm.fit2)

##
## Call:
## lm(formula = Sales ~ Price + US, data = Carseats)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.9269 -1.6286 -0.0574  1.5766  7.0515
##
## Coefficients:
##             Estimate Std. Error t value   Pr(>|t|)
## (Intercept) 13.03079    0.63098  20.652    < 2e-16 ***
## Price       -0.05448    0.00523 -10.416    < 2e-16 ***
## USYes        1.19964    0.25846   4.641 0.00000471 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.469 on 397 degrees of freedom
## Multiple R-squared:  0.2393, Adjusted R-squared:  0.2354
## F-statistic: 62.43 on 2 and 397 DF,  p-value: < 2.2e-16

my.output1 <- summary(lm.fit2)
```

(f) How well do the models in (a) and (e) fit the data?

```
par(mfrow = c(2, 2))
plot(lm.fit)
```




```
par(mfrow = c(2, 2))
plot(lm.fit2)
```



[Based on their respective R-square values(in summary tables), these two models are mediocre (only 24% change in response explained).] [Based on the RSE and R^2 of the linear regressions, they both fit the data similarly, with linear regression from (e) fitting the data slightly better.]

(g) Using the model from (e), obtain 95% confidence intervals for the coefficient(s).

```
confint(lm.fit2)

##                   2.5 %      97.5 %
## (Intercept) 11.79032020 14.27126531
## Price       -0.06475984 -0.04419543
## USYes        0.69151957  1.70776632
```

(h) Compute the *training MSE* and the *testing MSE*.

```
mean(my.output$residuals^2)

## [1] 6.052087

mean(my.output1$residuals^2)

## [1] 6.052186
```

[MSE for training and testing dataset are almost the same.]

Part IV: adding interaction terms

Use the * symbols to fit linear regression models with interaction effects. Do any interactions appear to be statistically significant?

```
data("Carseats", package = "ISLR")
lm.fit3 <-  lm(Sales ~ Price * US, data = Carseats)
summary(lm.fit3)

##
## Call:
## lm(formula = Sales ~ Price * US, data = Carseats)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.9299 -1.6375 -0.0492  1.5765  7.0430
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 12.974798   0.953079  13.614  < 2e-16 ***
## Price       -0.053986   0.008163  -6.613 1.22e-10 ***
## USYes        1.295775   1.252146   1.035    0.301
## Price:USYes -0.000835   0.010641  -0.078    0.937
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 2.472 on 396 degrees of freedom
## Multiple R-squared:  0.2393, Adjusted R-squared:  0.2335
## F-statistic: 41.52 on 3 and 396 DF,  p-value: < 2.2e-16
```

[The average of the price if the carseat is from the US is decreased with -0.000835 this amount.]

Hints & shortcuts & some more code

Exploring a new data set

One of the following may be helpful as you explore the data set:

```
View(Carseats)
help(Carseats)
str(Carseats)
```

Loops and selections from data frames

You may want to consider some of the following functions or commands as you write code to solve the exam.

```
tmp_data_set <- mtcars
tmp_col <- tmp_data_set[,1]
tmp_rows <- tmp_data_set[c(1,2,3,5),]
sapply(tmp_data_set[,1:7], max) # applies a function (in this case, `max`) to
all of the indicated columns of the data frame

##     mpg     cyl    disp      hp    drat      wt    qsec
##  33.900   8.000 472.000 335.000   4.930   5.424  22.900
```

Getting a 'nice' printout of the coefficients table

Run the following R chunk.

```
kable(coef(summary(lm.fit)), digits = c(4, 5, 2, 4))
```

Overlaying a linear regression line on a data scatterplot with ggplot

Here is a ggplot command that overlays a linear regression line on a scatterplot of `PREDICTORNAME` vs. `RESPONSENAME`. Of course, you should edit the `xlab` and `ylab` arguments to produce more meaningful axis labels.
Run the following R chunk.

```
qplot(data = NAMEDATASET, x = PREDICTORNAME, y = RESPONSENAME,
      xlab = "type name of predictor variable here", ylab = "type name of
response variable here") + stat_smooth(method = "lm")
```

You can use this code to get a plot to answer the following type of a question: does the linear model appear to fit the data well?

Computing the MSE

Once you run `lm` and save `your.output<- summary(lm)`, the mean squared error is given by `mean(your.output$residuals^2)`. You could write a function to calculate this, e.g.:

```
mse <- function(my.output)
    mean(my.output$residuals^2)
```

You can also use the `MSE` function for the predicted and true values, which you previously saved as `y_predicted` and `y_true`:

```
MSE(y_predicted, y_true)
```

Of course, you have to remember that there is a *training MSE* and a *testing MSE*, computed on the two different subsets of the sample data.

Fitting linear regression with interaction effects

To illustrate how one fits a model with interaction effects, let's run some simple code on a different data set:

```
data("Auto", package = "ISLR")
lm.fit <-  lm(mpg ~ cylinders * displacement + displacement * weight, data =
Auto)
summary(lm.fit)

##
## Call:
## lm(formula = mpg ~ cylinders * displacement + displacement *
##     weight, data = Auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.2934  -2.5184  -0.3476   1.8399  17.7723
##
## Coefficients:
##                            Estimate   Std. Error t value Pr(>|t|)
## (Intercept)             52.623409829  2.237444964  23.519  < 2e-16 ***
## cylinders                0.760640513  0.766949203   0.992    0.322
## displacement            -0.073512773  0.016694640  -4.403 1.38e-05 ***
## weight                  -0.009888167  0.001329428  -7.438 6.69e-13 ***
## cylinders:displacement -0.002986051  0.003425720  -0.872    0.384
## displacement:weight      0.000021277  0.000005002   4.254 2.64e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.103 on 386 degrees of freedom
```

```
## Multiple R-squared:  0.7272, Adjusted R-squared:  0.7237
## F-statistic: 205.8 on 5 and 386 DF,  p-value: < 2.2e-16
```

The model fitted is

mpg = 52.62 + 0.76 cylinders + -0.07 displacement + -0.01 weight + -0.003 cylinders * displacement + 0 displacement * weight.

Pro-tip: In the above line, I used an in-line `r` chunk!

As we learned in the lecture, when using interaction terms, we follow the *hierarchical model* rule.

From the output, we see that interaction between displacement and weight is statistically significant, while the interaction between cylinders and displacement is not.