Math 7553 – Spring 2018

HW #1 (Hand In)

Shuddha Chowdhury

Date: 06/02/2018

Chapter 2

Problem 2 (a) We collect a set of data on the top 500 firms in the US. For each firm we record profit, number of employees, industry and the CEO salary. We are interested in understanding which factors affect CEO salary.

Ans: This problem is regression and inference problem because the quantitative output of the CEO salary here is based on CEO firm's feature.
Here, n is 500 because there are 500 firms in the US
The value of p is 3 because of profit, number of employees and industry.

Problem 2 (b) we are considering launching a new product and wish to know whether it will be a success or a failure. We collect data on 20 similar products that were previously launched. For each product we have recorded whether it was a success or failure, price charged for the product, marketing budget, competition price, and ten other variables.

Ans: This problem is classification and prediction problem because it will predict product's success or failure.
Here, n is 20 – Because 20 similar products were previously launched.
The value of p is 13 because of price charged for the product, marketing budget, competition price, and ten other variables.

Problem 2 (c) We are interesting in predicting the % change in the US dollar in relation to the weekly changes in the world stock markets. Hence we collect weekly data for all of 2012. For each week we record the % change in the dollar, the % change in the US market, the % change in the British market, and the % change in the German market.

Ans: This problem is regression and prediction problem because quantitative output of % change.
Here, n is 52 – Because 52 weeks of 2012 weekly data is available.
The value of p is 3 because of % change in US market, % change in British market and % change in German market.

Problem 3. We now revisit the bias-variance decomposition.
Problem 3 (a) Provide a sketch of typical (squared) bias, variance, training error, test error, and Bayes (or irreducible) error curves, on a single plot, as we go from less flexible statistical learning methods Towards more flexible approaches. The x-axis should represent the amount of flexibility in the method, and the y-axis should represent the values for each curve. There should be five curves. Make sure to label each one.
Ans:

The graph is provided below in the below picture.

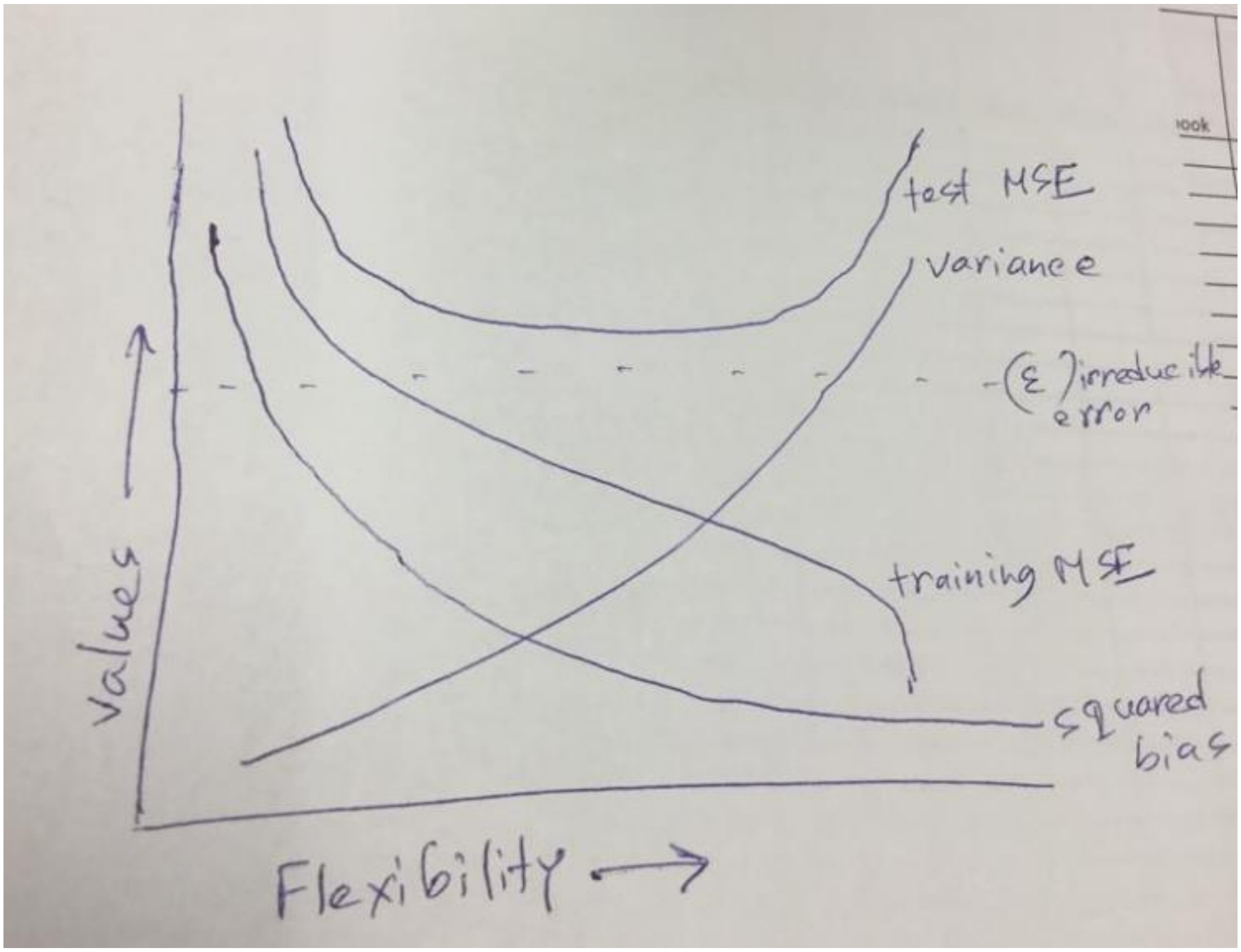


Fig 1: Sketch of typical (squared) bias, variance, training error, test error, and Bayes (or irreducible) error curves

Problem 3 (b) Explain why each of the five curves has the shape displayed in part

Ans: From the graph we see that all five lines are greater than zero. We see that the squared bias decreases monotonically because increases in flexibility yield a closer field. We also see that the variance increases monotonically because increases in flexibility yield over fit. A monotonic function is a function between ordered sets that preserves or reverses the given order. From the graph, training error decreases monotonically because increases in flexibility yield a closer fit. Test error is a concave up curve because increase in flexibility yields a fit before it overfits. Here we see that Bayes (irreducible error) – defines the lower limit. The test error is bounded below by the irreducible error due to variance in the error in the output values (0 <= value). The Bayes error rate is defined for classification problems and is determined by the ratio of data points which lie at the wrong side of the decision boundary. (0 <= value < 1)

Problem – 8

This exercise relates to the College data set, which can be found in the file College.csv. It contains a number of variables for 777 different universities and colleges in the US. The variables are

· Private : Public/private indicator
· Apps : Number of applications received
· Accept : Number of applicants accepted
· Enroll : Number of new students enrolled
· Top10perc : New students from top 10% of high school class
· Top25perc : New students from top 25% of high school class
· F.Undergrad : Number of full-time undergraduates
· P.Undergrad : Number of part-time undergraduates
· Outstate : Out-of-state tuition
· Room.Board : Room and board costs
· Books : Estimated book costs
· Personal : Estimated personal spending
· PhD : Percent of faculty with Ph.D.'s
· Terminal : Percent of faculty with terminal degree
· S.F.Ratio : Student/faculty ratio
· perc.alumni : Percent of alumni who donate
· Expend : Instructional expenditure per student
· Grad.Rate : Graduation rate

Before reading the data into R, it can be viewed in Excel or a text editor.

Problem 8 (a) Use the read.csv() function to read the data into R. Call the loaded data "college". Make sure that you have the directory set to the correct location for the data.

Ans:

```
library(ISLR)
data(College)
college <- read.csv("C:\\Users\\shc422\\Desktop\\Dataset\\College.csv")
```

Reference: How to Read CSV in R - http://rprogramming.net/read-csv-in-r/

Problem 8 (b) Look at the data using the fix() function. You should notice that the first column is just the name of each university. We don't really want R to treat this as data. However, it may be handy to have these names for later.

Ans:

```
#R-Markdown has been used as a typesetting tool for this assignment.

library(ISLR)
data(College)
college <- read.csv("C:\\Users\\shc422\\Desktop\\Dataset\\College.csv")
head(college[, 1:5])

##                              X Private Apps Accept Enroll
## 1 Abilene Christian University     Yes 1660   1232    721
## 2           Adelphi University     Yes 2186   1924    512
## 3               Adrian College     Yes 1428   1097    336
## 4          Agnes Scott College     Yes  417    349    137
```

```
## 5    Alaska Pacific University     Yes  193    146     55
## 6            Albertson College     Yes  587    479    158
```

```
rownames <- college[, 1]
college <- college[, -1]
head(college[, 1:5])
```

```
##   Private Apps Accept Enroll Top10perc
## 1     Yes 1660   1232    721        23
## 2     Yes 2186   1924    512        16
## 3     Yes 1428   1097    336        22
## 4     Yes  417    349    137        60
## 5     Yes  193    146     55        16
## 6     Yes  587    479    158        38
```

```
#Reference: R head function - https://www.rdocumentation.org/packages/utils/v
ersions/3.4.3/topics/head
```

Problem 8 (c)

(i) Use the summary () function to produce a numerical summary of the variables in the data set.

```
summary(college)
```

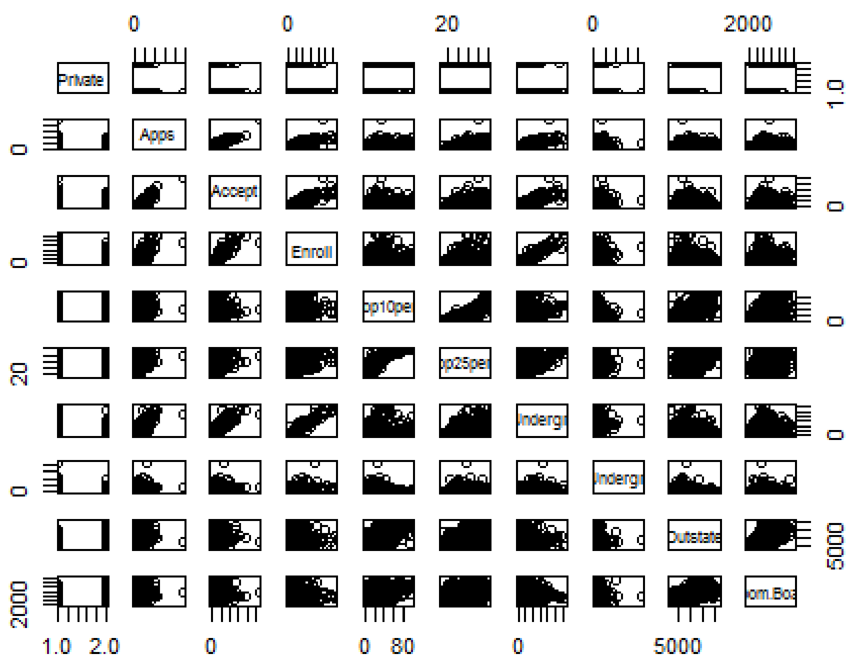
```
##  Private        Apps           Accept          Enroll       Top10perc
##  No :212   Min.   :   81   Min.   :   72   Min.   :  35   Min.   : 1.00
##  Yes:565   1st Qu.:  776   1st Qu.:  604   1st Qu.: 242   1st Qu.:15.00
##            Median : 1558   Median : 1110   Median : 434   Median :23.00
##            Mean   : 3002   Mean   : 2019   Mean   : 780   Mean   :27.56
##            3rd Qu.: 3624   3rd Qu.: 2424   3rd Qu.: 902   3rd Qu.:35.00
##            Max.   :48094   Max.   :26330   Max.   :6392   Max.   :96.00
##    Top25perc      F.Undergrad     P.Undergrad        Outstate
##  Min.   :  9.0   Min.   :  139   Min.   :    1.0   Min.   : 2340
##  1st Qu.: 41.0   1st Qu.:  992   1st Qu.:   95.0   1st Qu.: 7320
##  Median : 54.0   Median : 1707   Median :  353.0   Median : 9990
##  Mean   : 55.8   Mean   : 3700   Mean   :  855.3   Mean   :10441
##  3rd Qu.: 69.0   3rd Qu.: 4005   3rd Qu.:  967.0   3rd Qu.:12925
##  Max.   :100.0   Max.   :31643   Max.   :21836.0   Max.   :21700
##    Room.Board       Books           Personal         PhD
##  Min.   :1780   Min.   :  96.0   Min.   : 250   Min.   :  8.00
##  1st Qu.:3597   1st Qu.: 470.0   1st Qu.: 850   1st Qu.: 62.00
##  Median :4200   Median : 500.0   Median :1200   Median : 75.00
##  Mean   :4358   Mean   : 549.4   Mean   :1341   Mean   : 72.66
##  3rd Qu.:5050   3rd Qu.: 600.0   3rd Qu.:1700   3rd Qu.: 85.00
##  Max.   :8124   Max.   :2340.0   Max.   :6800   Max.   :103.00
##     Terminal        S.F.Ratio      perc.alumni        Expend
##  Min.   : 24.0   Min.   : 2.50   Min.   : 0.00   Min.   : 3186
##  1st Qu.: 71.0   1st Qu.:11.50   1st Qu.:13.00   1st Qu.: 6751
##  Median : 82.0   Median :13.60   Median :21.00   Median : 8377
```

```
##   Mean    : 79.7   Mean   :14.09   Mean   :22.74   Mean   : 9660
##   3rd Qu.: 92.0   3rd Qu.:16.50   3rd Qu.:31.00   3rd Qu.:10830
##   Max.   :100.0   Max.   :39.80   Max.   :64.00   Max.   :56233
##     Grad.Rate
##   Min.   : 10.00
##   1st Qu.: 53.00
##   Median : 65.00
##   Mean   : 65.46
##   3rd Qu.: 78.00
##   Max.   :118.00
```

```
#Reference: R summary function - https://www.rdocumentation.org/packages/base/version
s/3.4.3/topics/summary
```

ii) Use the pairs() function to produce a scatterplot matrix of the first ten columns or variables of the data.

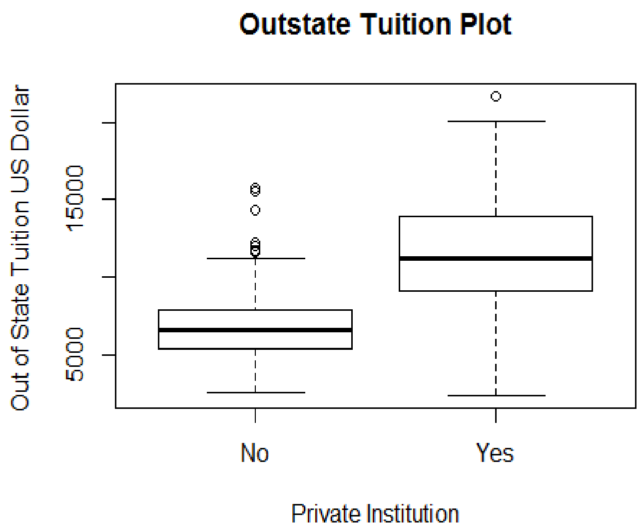
```
pairs(college[, 1:10])
```



```
# Reference: R pair function - https://www.rdocumentation.org/packages/pairwise/versi
ons/0.4.3-2/topics/pair
```

iii) Use the plot() function to produce side-by-side boxplots of "Outstate" versus "Private".

```
plot(college$Private, college$Outstate, xlab = "Private Institution", ylab ="
Out of State Tuition US Dollar ", main = "Outstate Tuition Plot")
```

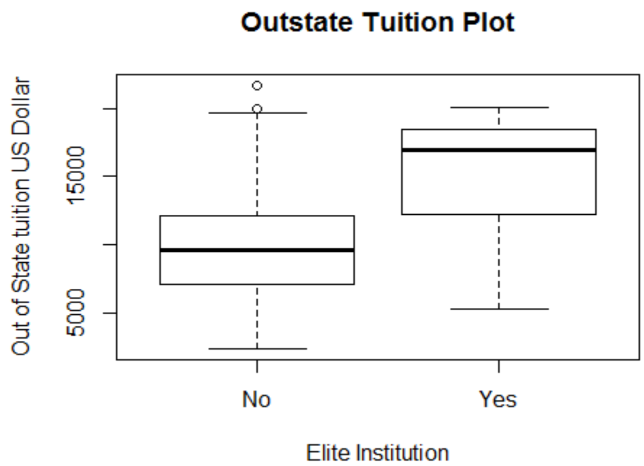



```
# Reference: R plot function - https://www.rdocumentation.org/packages/graphics/versions/3.4.3/topics/plot
```

iv) Create a new qualitative variable, called "Elite", by binning the "Top10perc" variable. Use the summary() function to see how many elite universities there are. Now use the plot() function to produce side-by-side boxplots of "Outstate" versus "Elite".

```
Elite <- rep("No", nrow(college))
Elite[college$Top10perc > 50] <- "Yes"
Elite <- as.factor(Elite)
```

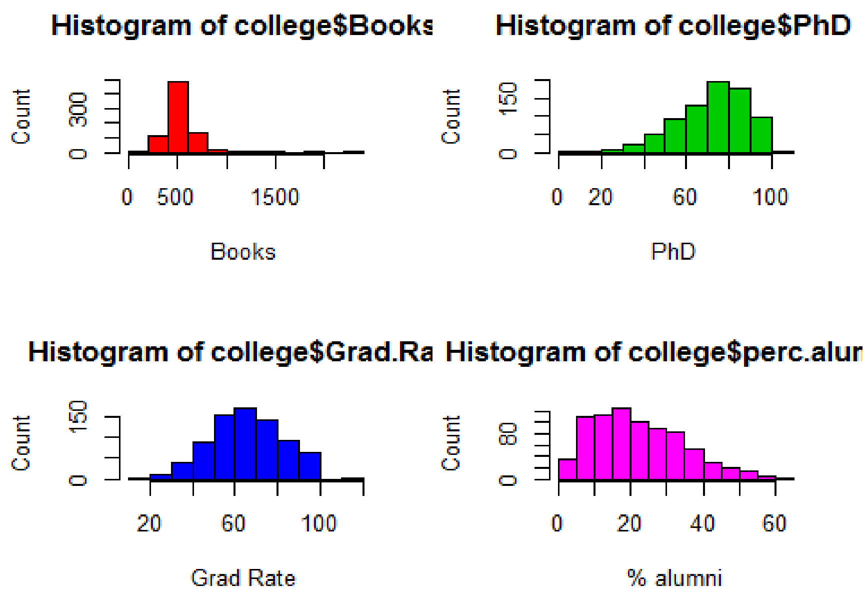
```
college$Elite <- Elite
summary(college$Elite)

##  No Yes
## 699  78

plot(college$Elite, college$Outstate, xlab = "Elite Institution", ylab ="Out of State tuition US Dollar", main = "Outstate Tuition Plot")
```



v) Use the hist() function to produce some histograms with numbers of bins for a few of the quantitative variables.

```
par(mfrow = c(2,2))
hist(college$Books, col = 2, xlab = "Books", ylab = "Count")
hist(college$PhD, col = 3, xlab = "PhD", ylab = "Count")
hist(college$Grad.Rate, col = 4, xlab = "Grad Rate", ylab = "Count")
hist(college$perc.alumni, col = 6, xlab = "% alumni", ylab = "Count")
```

```
# Reference: R hist function - https://www.rdocumentation.org/packages/graphics/versi
ons/3.4.3/topics/hist
```



vi) Continue exploring the data, and provide a brief summary of what you discover.

```
summary(college$PhD)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    8.00   62.00   75.00   72.66   85.00  103.00
```

```
weird.phd <- college[college$PhD == 103, ]
nrow(weird.phd)
```

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

```
rownames[as.numeric(rownames(weird.phd))]
```

```
## [1] Texas A&M University at Galveston
## 777 Levels: Abilene Christian University ... York College of Pennsylvania
```

```
#Reference: R row.names function - https://www.rdocumentation.org/packages/base/versi
ons/3.4.3/topics/row.names
```

Question – 9. This exercise involves the "Auto" data set studied in the lab. Make sure the missing values have been removed from the data.

a)

```
auto <- read.csv("C:\\Users\\shc422\\Desktop\\Dataset\\Auto(1).csv", na.strings = "?")
auto <- na.omit(auto)
str(auto)
```

```
## 'data.frame':    392 obs. of  9 variables:
##  $ mpg         : num  18 15 18 16 17 15 14 14 14 15 ...
##  $ cylinders   : int  8 8 8 8 8 8 8 8 8 8 ...
##  $ displacement: num  307 350 318 304 302 429 454 440 455 390 ...
##  $ horsepower  : int  130 165 150 150 140 198 220 215 225 190 ...
##  $ weight      : int  3504 3693 3436 3433 3449 4341 4354 4312 4425 3850 ...
##  $ acceleration: num  12 11.5 11 12 10.5 10 9 8.5 10 8.5 ...
##  $ year        : int  70 70 70 70 70 70 70 70 70 70 ...
##  $ origin      : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ name        : Factor w/ 304 levels "amc ambassador brougham",..: 49 36 231 14 161

 141 54 223 241 2 ...
##  - attr(*, "na.action")=Class 'omit'  Named int [1:5] 33 127 331 337 355
##   .. ..- attr(*, "names")= chr [1:5] "33" "127" "331" "337" ...
```

Here all the variables rather than "horsepower" and "name" are quantitative.

b. What is the range of each quantitative predictor?

```
summary(auto[, -c(4, 9)])
```

```
##       mpg          cylinders      displacement       weight
##  Min.   : 9.00   Min.   :3.000   Min.   : 68.0   Min.   :1613
##  1st Qu.:17.00   1st Qu.:4.000   1st Qu.:105.0   1st Qu.:2225
##  Median :22.75   Median :4.000   Median :151.0   Median :2804
##  Mean   :23.45   Mean   :5.472   Mean   :194.4   Mean   :2978
##  3rd Qu.:29.00   3rd Qu.:8.000   3rd Qu.:275.8   3rd Qu.:3615
```

```
##  Max.   :46.60   Max.   :8.000   Max.   :455.0   Max.   :5140
##   acceleration        year           origin
##  Min.   : 8.00   Min.   :70.00   Min.   :1.000
##  1st Qu.:13.78   1st Qu.:73.00   1st Qu.:1.000
##  Median :15.50   Median :76.00   Median :1.000
##  Mean   :15.54   Mean   :75.98   Mean   :1.577
##  3rd Qu.:17.02   3rd Qu.:79.00   3rd Qu.:2.000
##  Max.   :24.80   Max.   :82.00   Max.   :3.000
```

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

```
sapply(auto[, -c(4, 9)], mean)

##          mpg    cylinders displacement       weight acceleration
##    23.445918     5.471939   194.411990  2977.584184    15.541327
##         year       origin
##    75.979592     1.576531
```

```
sapply(auto[, -c(4, 9)], sd)

##          mpg    cylinders displacement       weight acceleration
##    7.8050075    1.7057832  104.6440039  849.4025600    2.7588641
##         year       origin
##    3.6837365    0.8055182
```

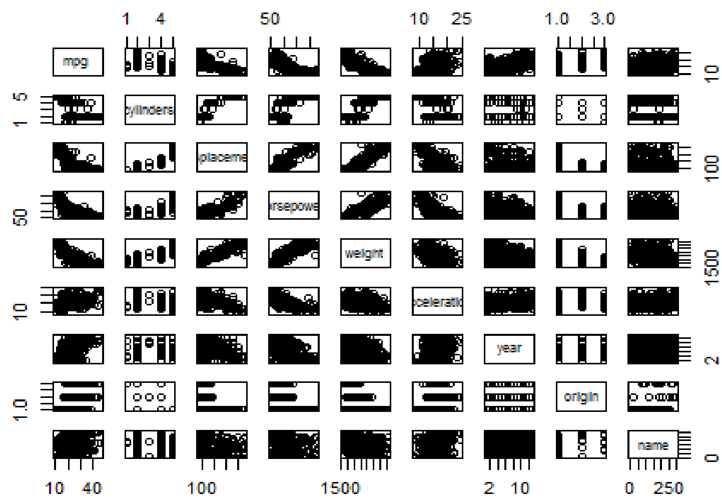
d)Now remove the 10th through 85th observations. What is the range, mean, and standard deviation of each predicto r in the subset of the data that remains ?

```
subset <- auto[-c(10:85), -c(4,9)]
sapply(subset, range)

##       mpg cylinders displacement weight acceleration year origin
## [1,] 11.0         3           68   1649          8.5   70      1
## [2,] 46.6         8          455   4997         24.8   82      3

sapply(subset, mean)

##          mpg    cylinders displacement       weight acceleration
##    24.404430     5.373418   187.240506  2935.971519    15.726899
##         year       origin
##    77.145570     1.601266

sapply(subset, sd)

##          mpg    cylinders displacement       weight acceleration
##     7.867283     1.654179    99.678367   811.300208     2.693721
##         year       origin
##     3.106217     0.819910
```

```
#Reference: R sapply function - https://www.datacamp.com/community/tutorials/r-tutori
al-apply-family
```

e) 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.

```r
auto$cylinders <- as.factor(auto$cylinders)
auto$year <- as.factor(auto$year)
auto$origin <- as.factor(auto$origin)
pairs(auto)
```



From the graph we see that we will to get more mileage per gallon on a 4 cylinders vehicle than the others. Weight, displacement and horsepower have an inverse effect with mpg. There is an overall increase in mpg over the years. It almost doubled in one decade. We also notice that Japanese cars have higher mpg than US or European cars.

f) Suppose that we wish to predict gas mileage ("mpg") on the basis of other variables. Do your plots suggest that any of the other variables might be useful in predicting "mpg"?

```r
auto$horsepower <- as.numeric(auto$horsepower)
cor(auto$weight, auto$horsepower)

## [1] 0.8645377

cor(auto$weight, auto$displacement)

## [1] 0.9329944

cor(auto$displacement, auto$horsepower)

## [1] 0.897257
```

```
#Reference: Correlation function - https://www.rdocumentation.org/packages/stats/versions/3.4.3/topics/cor
```

Chapter -3

Problem – 8

This question involves the use of simple linear regression on the Auto data set.

Problem 8 (a) Use the lm() function to perform a simple linear regression with mpg as the response and horsepower as the predictor. Use the summary() function to print the results. Comment on the output.

```
auto <- read.csv("C:\\Users\\shc422\\Desktop\\Dataset\\Auto(1).csv", header=T
, na.strings="?")
 auto <- na.omit(auto)
 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
```

```
attach(auto)
lm.fit = lm(mpg ~ horsepower)
summary(lm.fit)

##
## Call:
## lm(formula = mpg ~ horsepower)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.5710  -3.2592  -0.3435   2.7630  16.9240
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 39.935861   0.717499   55.66   <2e-16 ***
## horsepower  -0.157845   0.006446  -24.49   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.906 on 390 degrees of freedom
## Multiple R-squared:  0.6059, Adjusted R-squared:  0.6049
## F-statistic: 599.7 on 1 and 390 DF,  p-value: < 2.2e-16
```

```
#R Tutorial Series: Simple Linear Regression- https://www.r-bloggers.com/r-tutorial-s
eries-simple-linear-regression/
```

Problem 8 (a)
i. Is there a relationship between the predictor and the response?

Ans : There is a relationship between horsepower and mpg. When we determine that testing the null hypothesis of al l the regression coefficient equal to zero. We see that the F- statistic is far larger than 1 and the p value of the F- stati stic is close to zero we can reject the null hypothesis and conclude there is a statistically significant relationship betw een horsepower and mpg.

ii. How strong is the relationship between the predictor and the response?
Ans : If we want to calculate the residual error relative to the response we use the mean of the response and the RSE. The mpg's mean is 23.445. The RSE of the lm.fit is 4.90. It indicates a percentage error of 20.92. The R^2 of the lm.fit 0.6059 mean 60.59% of the variance in mpg is explained by horsepower.

iii) Is the relationship between the predictor and the response positive or negative?

Ans : The relationship between mpg and horsepower is negative. The more horsepower an automobile has the linear regression indicates the less mpg fuel efficiency the automobile have.

iv)What is the predicted mpg associated with a horsepower of 98? What are the associated 95% confidence and prediction intervals?

Ans:

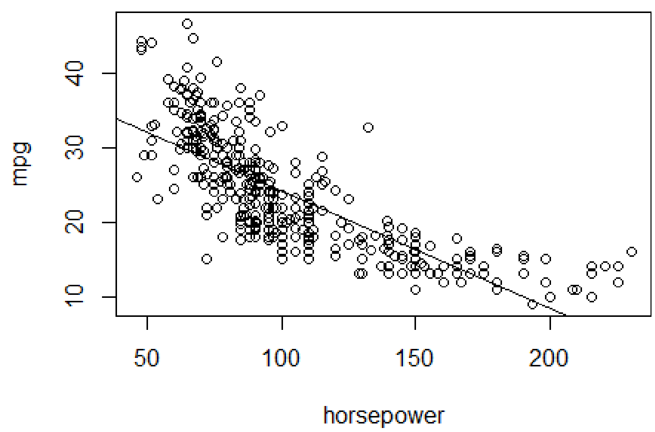
```
predict(lm.fit, data.frame(horsepower=c(98)), interval="confidence")

##        fit      lwr      upr
## 1 24.46708 23.97308 24.96108

predict(lm.fit, data.frame(horsepower=c(98)), interval="prediction")

##        fit     lwr      upr
## 1 24.46708 14.8094 34.12476
```

Problem 8 (b)

Plot the response and the predictor. Use the abline() function to display the least squares regression line.

Ans:

```
plot(horsepower, mpg)
abline(lm.fit)
```

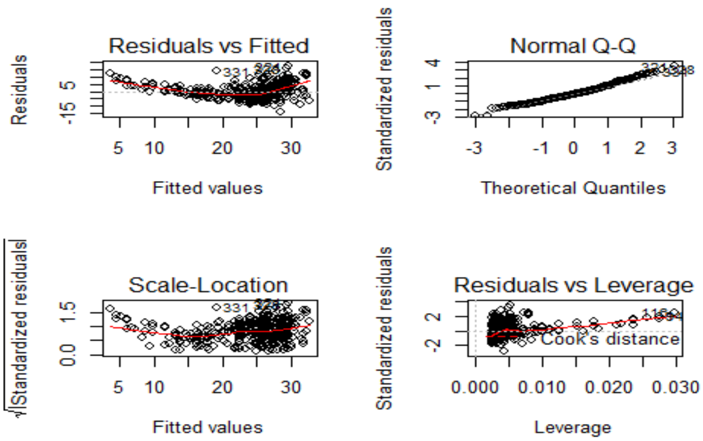


Reference: abline R function- http://www.sthda.com/english/wiki/abline-r-function-an-easy-way-to-add-straight-lines-to-a-plot-using-r-software

Problem 8 (c)

Use the plot() function to produce diagnostic plots of the least squares regression fit. Comment on any problems you see with the fit.

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



There is some evidence of non-linearity based on the residuals plots.

Problem 13

In this exercise you will create some simulated data and will fit simple linear regression models to it. Make sure to use set.seed(1) prior to starting part (a) to ensure consistent results.

Problem 13 (a)

Using the rnorm() function, create a vector, x, containing 100 observations drawn from a N(0, 1) distribution. This represents a feature, X.

```
set.seed(1)
x = rnorm(100)
```

Problem 13 (b)

Using the rnorm() function, create a vector, eps, containing 100 observations drawn from a N(0, 0.25) distribution i.e. a normal distribution with mean zero and variance 0.25.

```
eps = rnorm(100, 0, sqrt(0.25))
```

Problem 13 (c)

Using x and eps, generate a vector y according to the model $Y = -1 + 0.5X + E$. (1)

What is the length of the vector y? What are the values of β_0 and β_1 in this linear model?

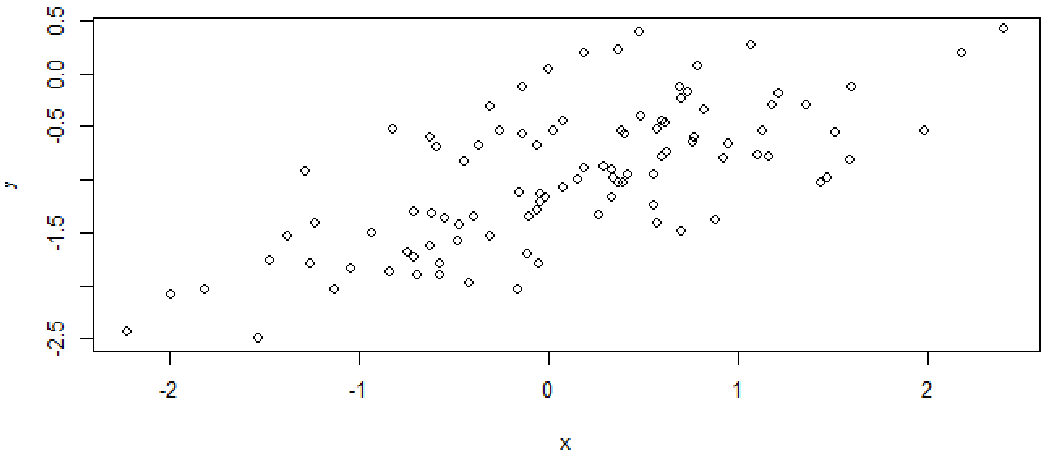
```
y = -1 + 0.5*x + eps
```

y is of length 100. β_0 is -1, β_1 is 0.5.

Problem 13 (d)

Create a scatterplot displaying the relationship between x and y. Comment on what you observe.

```
plot(x,y)
```



There is a linear relationship between x and y with a positive slope, with a variance as to be expected.

Problem 13 (e)

Fit a least squares linear model to predict y using x. Comment on the model obtained. How do ˆ β0 and ˆ β1 compare to β0 and β1?

```
lm.fit = lm(y~x)
summary(lm.fit)

##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.93842 -0.30688 -0.06975  0.26970  1.17309
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.01885    0.04849 -21.010  < 2e-16 ***
## x            0.49947    0.05386   9.273 4.58e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4814 on 98 degrees of freedom
## Multiple R-squared:  0.4674, Adjusted R-squared:  0.4619
## F-statistic: 85.99 on 1 and 98 DF,  p-value: 4.583e-15
```

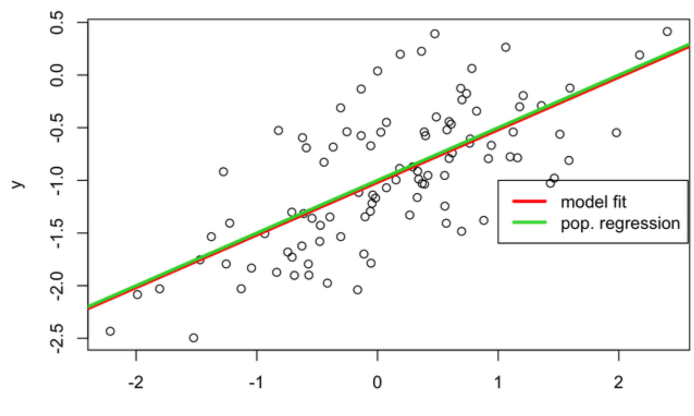
The linear regression fits a model close to the true value of the coefficient as it is constructed. This model has a large F- statistic with a near zero p-value. So we can reject the null hypothesis.

Problem 13 (f)

Display the least squares line on the scatterplot obtained in (d). Draw the population regression line on the plot, in a different color. Use the legend () command to create an appropriate legend.

Ans :

```
plot(x, y)
abline(lm.fit, lwd=3, col=2)
abline(-1, 0.5, lwd=3, col=3)
legend(-1, legend = c("model fit", "pop. regression"), col=2:3, lwd=3)
```



Problem 13 (g)

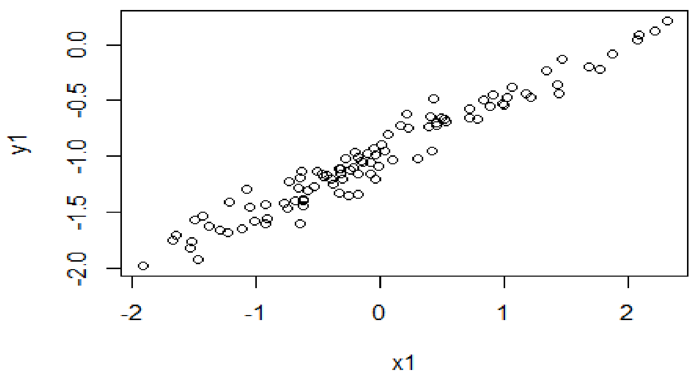
```
lm.fit_sq = lm(y~x+I(x^2))
summary(lm.fit_sq)

##
## Call:
## lm(formula = y ~ x + I(x^2))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.98252 -0.31270 -0.06441  0.29014  1.13500
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.97164    0.05883 -16.517  < 2e-16 ***
## x            0.50858    0.05399   9.420  2.4e-15 ***
## I(x^2)      -0.05946    0.04238  -1.403    0.164
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.479 on 97 degrees of freedom
## Multiple R-squared:  0.4779, Adjusted R-squared:  0.4672
## F-statistic:  44.4 on 2 and 97 DF,  p-value: 2.038e-14
```

There is evidence that model fit has increased over the training data given the slight increase in R2 and RSE. Although, the p-value of the t-statistic suggests that there isn't a relationship between y and x2.

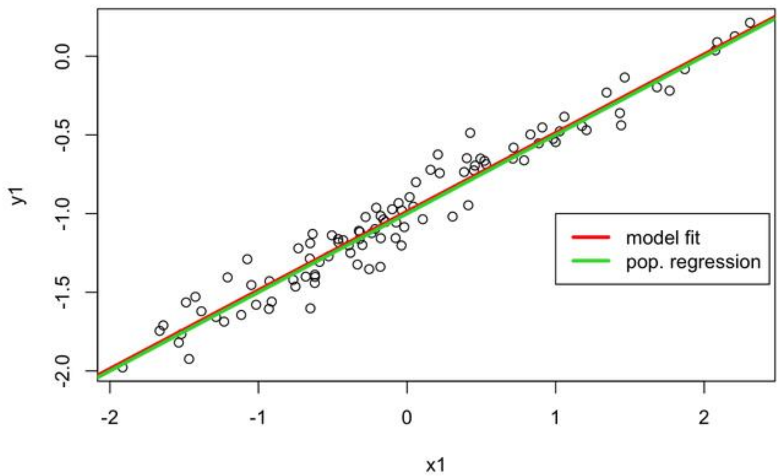
Problem 13 (h)
Repeat (a)–(f) after modifying the data generation process in such a way that there is less noise in the data. The model (3.39) should remain the same. You can do this by decreasing the variance of the normal distribution used to generate the error term in (b). Describe your results.

```
set.seed(1)
eps1 = rnorm(100, 0, 0.125)
x1 = rnorm(100)> y1 = -1 + 0.5*x1 + eps1
y1 = -1 + 0.5*x1 + eps1
plot(x1, y1)
```




```
lm.fit1 = lm(y1~x1)
 summary(lm.fit1)

##
## Call:
## lm(formula = y1 ~ x1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.29052 -0.07545  0.00067  0.07288  0.28664
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.98639    0.01129  -87.34   <2e-16 ***
## x1           0.49988    0.01184   42.22   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1128 on 98 degrees of freedom
```

```
## Multiple R-squared:  0.9479, Adjusted R-squared:  0.9474
## F-statistic:  1782 on 1 and 98 DF,  p-value: < 2.2e-16

 plot(x, y)
 abline(lm.fit, lwd=3, col=2)
 abline(-1, 0.5, lwd=3, col=3)
 legend(-1, legend = c("model fit", "pop. regression"), col=2:3, lwd=3)
```



Here we can see that the error observed in R^2 and RSE decrease considerably.

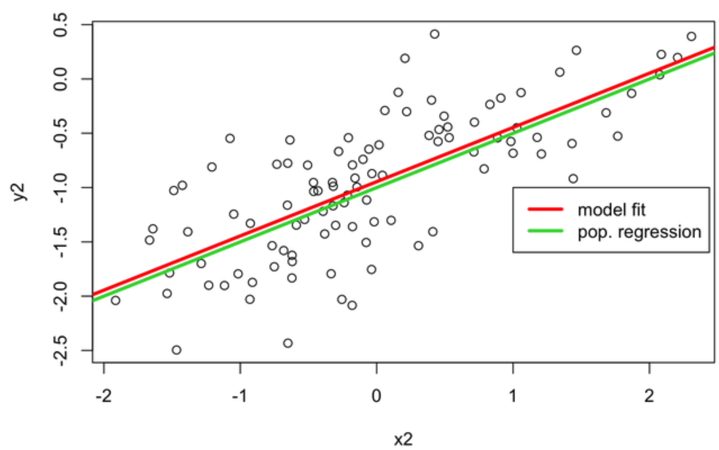
Problem 13 (i)

Repeat (a)–(f) after modifying the data generation process in such a way that there is more noise in the data. The model (3.39) should remain the same. You can do this by increasing the variance of the normal distribution used to generate the error term _ in (b). Describe your results.

```
set.seed(1)
eps2 = rnorm(100, 0, 0.5)
x2 = rnorm(100)
y2 = -1 + 0.5*x2 + eps2
plot(x2, y2)
```

```
lm.fit2 = lm(y2~x2)
 summary(lm.fit2)

##
## Call:
## lm(formula = y2 ~ x2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.16208 -0.30181  0.00268  0.29152  1.14658
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept) -0.94557    0.04517  -20.93   <2e-16 ***
## x2           0.49953    0.04736   10.55   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4514 on 98 degrees of freedom
## Multiple R-squared:  0.5317, Adjusted R-squared:  0.5269
## F-statistic: 111.2 on 1 and 98 DF, p-value: < 2.2e-16
```



RSE increase considerably here.

Problem 13 (j)

What are the confidence intervals for β0 and β1 based on the original data set, the noisier data set, and the less noisy data set? Comment on your results.

```
confint(lm.fit)

##                  2.5 %     97.5 %
## (Intercept) -1.1150804 -0.9226122
## x            0.3925794  0.6063602

confint(lm.fit1)

##                 2.5 %     97.5 %
## (Intercept) -1.008805 -0.9639819
## x1           0.476387  0.5233799

confint(lm.fit2)

##                  2.5 %     97.5 %
## (Intercept) -1.0352203 -0.8559276
## x2           0.4055479  0.5935197
```

Here All intervals here seem to be centered on approximately 0.5, with the second fit's interval being narrower than the first fit's interval and the last fit's interval is wider than the first fit's interval.

```
#Reference: [1] set.seed function - http://rfunction.com/archives/62
            [2] rnorm function - https://www.rdocumentation.org/packages/compositions
/versions/1.40-1/topics/rnorm
```