Outside Book Homework:

$f_1(x) = \beta_o + \beta_1 X$
$f_2(x) = \beta_o + \beta_1 X + \beta_2 X^2 + \beta_3 X^3$

set.seed(1)
x = seq(from = -2 to = 2, by = 0.1)
y = 100 + 2*x – X^2 + rnorm(41)

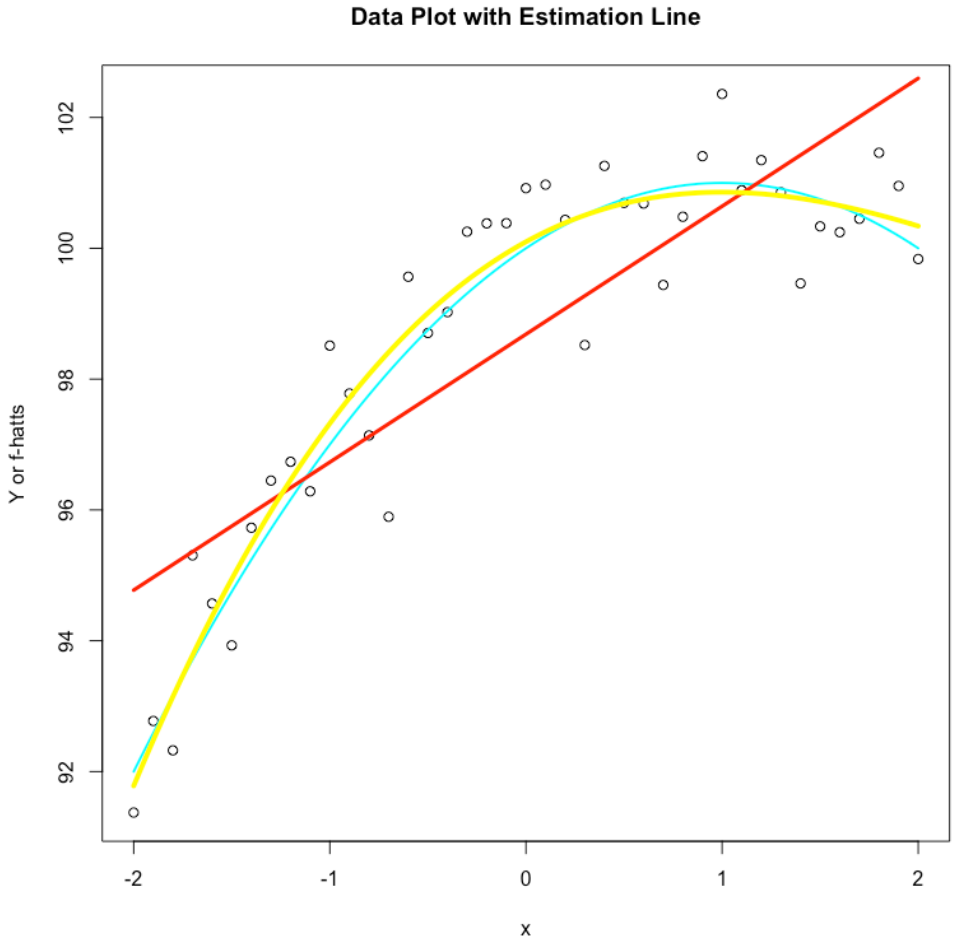
Problem 1:

```
> #Outside Book:
> rm(list = ls())
> set.seed(1)
> x = seq(from = -2, to = 2, by = .1)
> y = 100 + 2*x - x^2 + rnorm(41)
> # Problem 1:
> xsq = x*x
> xcu = xsq*x
> fx = 100 + 2*x - x^2
> f1xhatt = lm(y ~ x)
> f1xhatt

Call:
lm(formula = y ~ x)

Coefficients:
(Intercept)            x
     98.686        1.956

> f2xhatt = lm(y ~ x + xsq + xcu)
> f2xhatt

Call:
lm(formula = y ~ x + xsq + xcu)

Coefficients:
(Intercept)            x          xsq          xcu
   100.0993       1.6444      -1.0097       0.1238
```

$f(x) = 100 + 2x + x^2$
$\widehat{f_1}(x) = 98.686 + 1.9561X$
$\widehat{f_2}(x) = 100.0993 + 1.6444X - 1.0097X^2 + 0.1238X^3$

Problem 2:

```
# Problem 2:
plot (x, y, col = 1, xlab = "x", ylab = "Y or f-hatts", main = "Data Plot with Estimation
Line")
curve (100 + 2*x - x^2, add = TRUE, col = 5, lyt = 2, lwd = 2) #True Y.
curve(98.686 + 1.9561*x, add = TRUE, col = 10, lyt = 4, lwd = 3) #Linear
curve(100.0993 + 1.6444*x -1.0097*x^2 + 0.1238*x^3, add = TRUE, col = 15, lyt = 4, lwd = 4)
#Quadratic
```

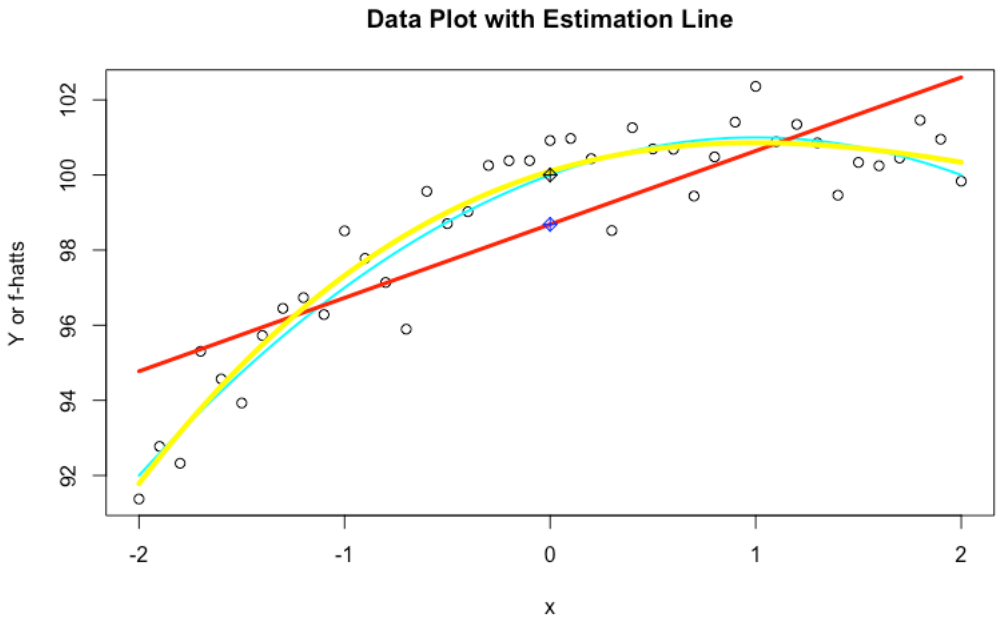


Problem 3:

```
# Probelm 3:
truef = function(x) {100 + 2*x - x^2}
truef(0)
predict (f1xhatt, data.frame( x = 0)) #Linear Function
points(0, 100, pch = 9, col = "black") #True Y
points(0, 98.68577, pch = 9, col = "blue") # Linear Function
```

$f(0) = 100$

$\hat{f}_1(0) = 98.68577$



Blue dot is from linear function, black dot is from true Y.

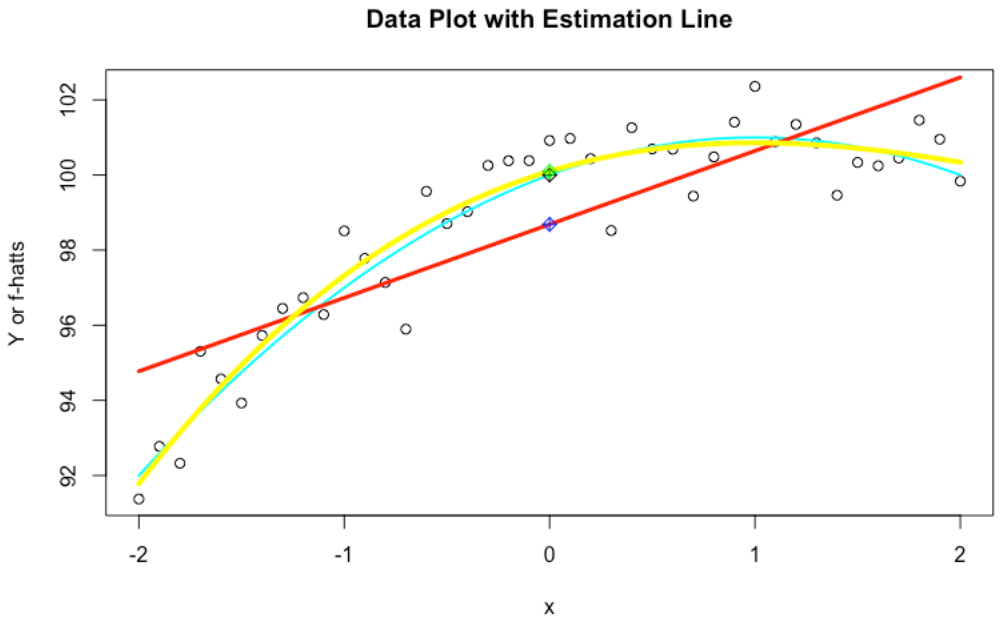
Yes, the difference between these two values is reducible with better estimate of parameters by choosing better model.

Problem 4:

$f(0) = 100$

$\hat{f}_2(0) = 100.0933$

```
# Probelm 4:
truef = function(x) {100 + 2*x - x^2}
truef(0)
predict (f2xhatt, data.frame( x = 0, xsq = 0, xcu = 0)) #Quadratic
points(0, 100.0993, pch = 9, col = "green") # Quadratic Function.
```



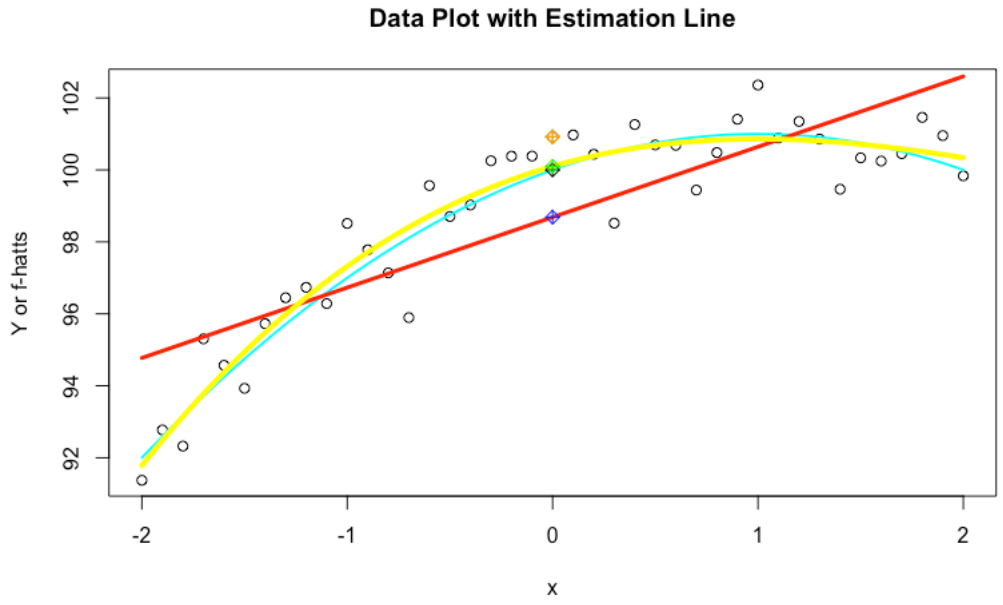
Blue dot is from linear function, black dot is from true Y, green dot is from quadratic function.

This error is reducible. Because still the prediction is different from the true value and always can be reduced.

Problem 5:

Y_{21} = 100.919
f(o) = 100

```
# Problem 5:
truef(x = 0)
y[21]
points(x[21], y[21], pch = 9, col = "orange") #
```



Blue dot is from linear function, black dot is from true Y, green dot is from quadratic function, Orange dot is from Y_{21}.

This error is irreducible. This is coming from the random error and thus irreducible.

Problem 6:

TEST MSE = Avg $[(y_o - \hat{f}(x_o))^2]$
(x, y) = (1, 94), (0, 100), (1, 100)
TEST MSE for $\widehat{f_1}(x)$ = $(94 - 96.73)^2 + (100 - 98.686)^2 + (100 - 100.642)^2 = 3.197$
TEST MSE for $\hat{2}(x)$ = $(94 - 97.32)^2 + (100 - 100.099)^2 + (100 - 100.86)^2 = 3.93s$

Problem 7:

```
> # Problem 7:
> set.seed(2)
> x7 = seq(from = -2, to = 2, by = .1)
> y7 = 100 + 2*x7 - x7^2 + rnorm(41)
> x7sq = x7*x7
> x7cu = x7sq*x7
>
> fx7 = 100 + 2*x7 - x7^2 #True Y
> f1x7hatt = lm(y7 ~ x7) #Linear
> f1x7hatt

Call:
lm(formula = y7 ~ x7)

Coefficients:
(Intercept)           x7
     98.686        1.836

> f2x7hatt = lm(y7 ~ x7 + x7sq + x7cu) #Quadratic.
> f2x7hatt

Call:
lm(formula = y7 ~ x7 + x7sq + x7cu)

Coefficients:
(Intercept)           x7         x7sq         x7cu
 100.373727     1.855582    -1.205624    -0.007936
```

Predicting New $\widehat{f_1}(0)$ and $\widehat{f_2}(0)$.

```
> truef7a = function(x7) {100 + 2*x7 - x7^2}
> truef7a(0)
[1] 100
> predict (f1x7hatt, data.frame( x7 = 0)) #Linear Function
       1
98.68585
>
> truef7b = function(x7) {100 + 2*x7 - x7^2}
> truef7b(0)
[1] 100
> predict (f2x7hatt, data.frame( x7 = 0, x7sq = 0, x7cu = 0)) #Quadratic
       1
100.3737
```

New $\widehat{f_1}(0)$ = 98.68585 and $\widehat{f_2}(0)$ = 100.373. This is still different results compared to when we set to set.seed(1). This is because we are selecting separate set of samples from same set of data. This is due to difference in the variance. **OLS is always unbiased.**

Problem 8:

From set.seed(1):

$\widehat{f_2}(0)$ = 100.093 and $f(0)$ = 100

Bias of $\widehat{f_2}(x)$ = 100.093 – 100 = 0.0933

From set.seed(2):

$\widehat{f_2}(x)$ = 100.3737 and $f(0)$ = 100

Bias of $\widehat{f_2}(x)$ = 100.3737 – 100 = 0.3737

Problem 9:
More flexible model tends to have less variance. With the larger sample size, the variance is almost zero. Since $\widehat{f_1}(x)$ (linear model) is less flexible than $\widehat{f_2}(x)$ (quadratic model), the quadratic model has smaller variances.

Problem 10:
Since $\widehat{f_1}(x)$ (linear model) is less flexible than $\widehat{f_2}(x)$ (quadratic model), the quadratic model has smaller bias. The model should be very flexible for bias to be zero.

Problem 11:
I would choose f_2 as this model is more flexible to minimize MSE when large sample is available. The large sample reduce variance to almost zero.

Book Problems = Chapter 2: 1 (a-c), 2, 5, 7, 8 & 9.

1:
(a) Answer: There are two ways to reduce bias and variance. Bias can be reduced using a flexible model to find a better fitting model and thus better predict y. Also, we can reduce variance by either taking larger sample size or using a simpler model. Extremely larger sample size has small variance very close to zero. So, variance is not really a problem for this dataset. Bias could be a problem, if more conservative model was used. But since we are using more flexible learning method, the performance of model should be better.

(b) Answer: Flexible method would perform better. Small number of observations increase variance which makes model worse. The flexible model overfit the data and capture noise in the data. Also, large number of predictors, though helpful to capture more information by including more variables in the model, might degrade model because the model is not strong enough to make accurate prediction due to lack of sufficient data to model extremely large number of observations.

(c) Answer: Linear models are less flexible and highly non-linear model are more flexible as highly non-linear model can fit all data points in the given dataset. The linear model has lower degree of freedom compared to non-linear model. The higher degree of freedom means more flexible model gives a better fit.

2:
(a) Answer:

This is a regression problem, as CEO salary is continuous or quantitative variable. We are more interested in inference, understanding which factors affects CEO salary.
N = 500 (firms in the US).
P = 3, viz. profit, number of employees, industry. s

(b)
Answer:
This is a classification problem, as dependent variable is success or failure. We are interested in prediction of success and failure of new product based on previous data.
N = 20 (similar products).
P = 13, Viz. Product Price, Marketing Budget, Competition Price, and 10 other variables.

(c)
Answer:
This is a regression problem, and we are interested in predicting the percentage change in the USD/Euro exchange rate in relation to weekly change in world stock markets.
N = 52. I am assuming there are 52 weeks in the year 2012.
P = 3, Viz. % change in US market, % change in British market, and % Change in German market

5:
Answer:

Very flexible models (eg. quadratic models) provides better fit to the data compared to less flexible models such as linear regression and also create less bias. But very flexible models require large number of data, need to predict large number of parameters, and also tends to overfit the data and capture all the noise/errors in the data which increase variance.

Less flexible model such as linear regression are more interpretable and thus provide a better inference compared to more flexible models like boosting and splines. If we are merely interested in prediction, more flexible method is more appropriate and preferred as we don't care about interpretability.

7:

(a) Compute the Euclidean distance between each observation and the test point x1 = x2 = x3 = 0.

Say, X = (a, b, c) and Y = (d, e, f). Distance XY = $\sqrt{((d-a)^2 + (e-b)^2 + (f-c)^2)}$.

Obs.	X1	X2	X3	**Distance (0 0, 0)**	Y
1	0	3	0	= $\sqrt{(3-0)^2} = 3$	Red
2	2	0	0	= $\sqrt{(2-0)^2} = 2$	Red

3	0	1	3	$= \sqrt{((1-0)^2 + (3-0)^2)} = 3.16228$	Red
4	0	1	2	$\sqrt{5} = 2.2361$	Green
5	-1	0	1	$\sqrt{2} = 1.414$	Green
6	1	1	1	$\sqrt{3} = 1.7321$	Red

(b) Answer:

Our prediction is Green. For K = 1, We are making decision based on one observation only. So, we have to select the closest one from the test point which is given by the Euclidean distance between test and train points. The Distance = 1.41 is the closest point to (0, 0, 0) which is observation #5. Thus, observation #5 is the closest neighbor.

(c) Answer:

For K = 3, we have three points to choose. So, nearest three points based on distance should be taken into consideration. So, potential solutions are observations #5(Green), #6 (Red) or #2 (Red). Since, we have more reds (Probability = 2/3) than green (Probability = 1/3), the probability of test point being red is high. Thus, we predict the point (0, 0, 0) is red.

(d) Answer:

We would expect the best value for K to be larger. Because for highly non-linear decision boundary, large K is more flexible.

8:

8 (a):

```
> # a) Use read.csv() to read data into R.
> college  = read.csv(file = "college.csv",  header = TRUE, sep = " ")
> head(college)
  Private Apps Accept Enroll Top10perc Top25perc F.Undergrad P.Undergrad Outstate
1     Yes 1660   1232    721        23        52        2885         537     7440
2     Yes 2186   1924    512        16        29        2683        1227    12280
3     Yes 1428   1097    336        22        50        1036          99    11250
4     Yes  417    349    137        60        89         510          63    12960
5     Yes  193    146     55        16        44         249         869     7560
6     Yes  587    479    158        38        62         678          41    13500
  Room.Board Books Personal PhD Terminal S.F.Ratio perc.alumni Expend Grad.Rate
1       3300   450     2200  70       78      18.1          12   7041        60
2       6450   750     1500  29       30      12.2          16  10527        56
3       3750   400     1165  53       66      12.9          30   8735        54
4       5450   450      875  92       97       7.7          37  19016        59
5       4120   800     1500  76       72      11.9           2  10922        15
6       3335   500      675  67       73       9.4          11   9727        55
```

8 (b)

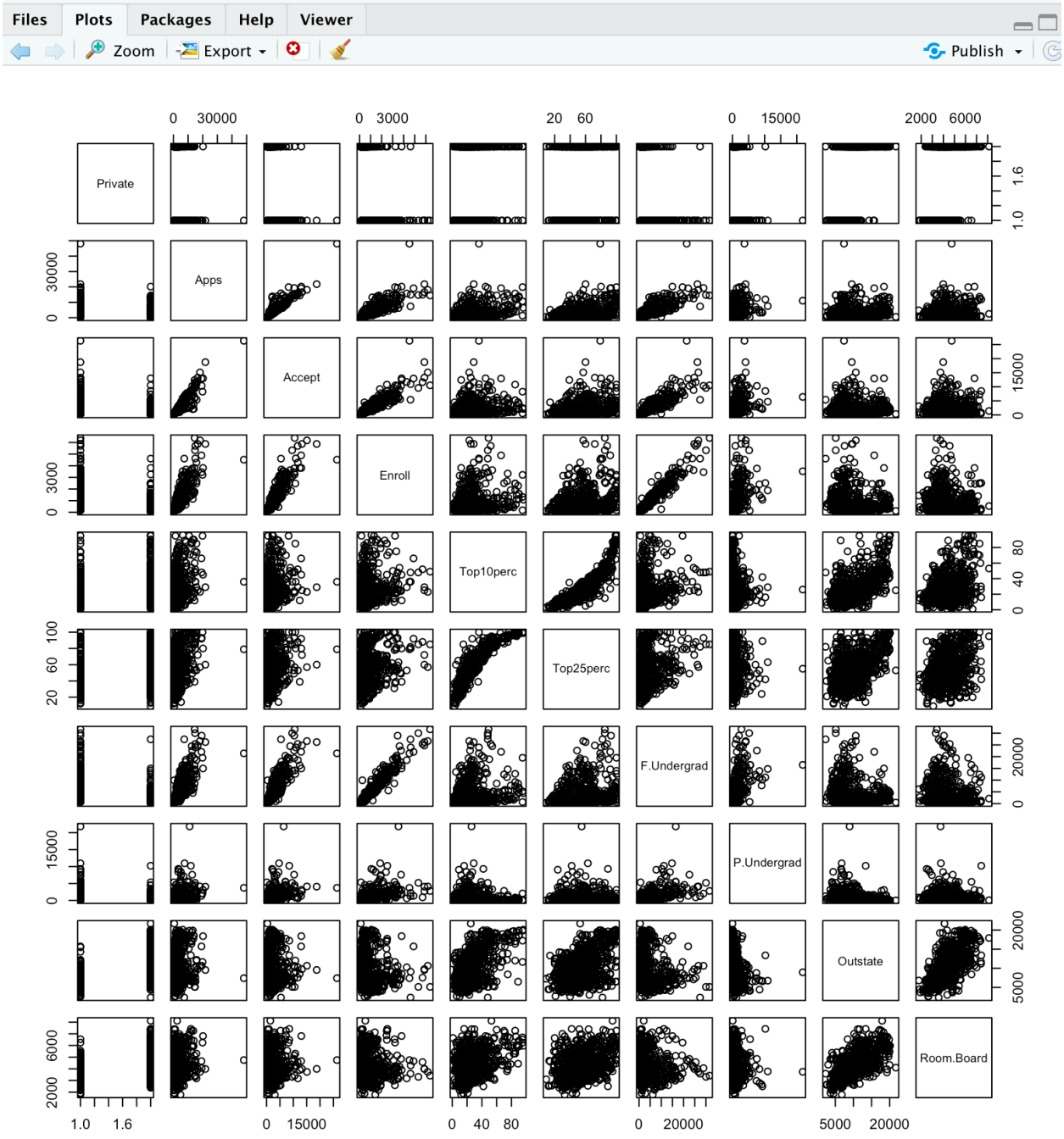
```
# b)
fix(college)
rownames(college) = college [, 1]
college1 = college [, -1]
fix(college1)
```

8. C. I:

```
> # 8.c.i: Summary:
> 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       Room.Board
 Min.   :  9.0   Min.   :  139   Min.   :    1.0   Min.   : 2340   Min.   :1780
 1st Qu.: 41.0   1st Qu.:  992   1st Qu.:   95.0   1st Qu.: 7320   1st Qu.:3597
 Median : 54.0   Median : 1707   Median :  353.0   Median : 9990   Median :4200
 Mean   : 55.8   Mean   : 3700   Mean   :  855.3   Mean   :10441   Mean   :4358
 3rd Qu.: 69.0   3rd Qu.: 4005   3rd Qu.:  967.0   3rd Qu.:12925   3rd Qu.:5050
 Max.   :100.0   Max.   :31643   Max.   :21836.0   Max.   :21700   Max.   :8124
     Books           Personal         PhD            Terminal       S.F.Ratio
 Min.   :  96.0   Min.   : 250   Min.   :  8.00   Min.   : 24.0   Min.   : 2.50
 1st Qu.: 470.0   1st Qu.: 850   1st Qu.: 62.00   1st Qu.: 71.0   1st Qu.:11.50
 Median : 500.0   Median :1200   Median : 75.00   Median : 82.0   Median :13.60
 Mean   : 549.4   Mean   :1341   Mean   : 72.66   Mean   : 79.7   Mean   :14.09
 3rd Qu.: 600.0   3rd Qu.:1700   3rd Qu.: 85.00   3rd Qu.: 92.0   3rd Qu.:16.50
 Max.   :2340.0   Max.   :6800   Max.   :103.00   Max.   :100.0   Max.   :39.80
  perc.alumni        Expend        Grad.Rate
 Min.   : 0.00   Min.   : 3186   Min.   : 10.00
 1st Qu.:13.00   1st Qu.: 6751   1st Qu.: 53.00
 Median :21.00   Median : 8377   Median : 65.00
 Mean   :22.74   Mean   : 9660   Mean   : 65.46
 3rd Qu.:31.00   3rd Qu.:10830   3rd Qu.: 78.00
 Max.   :64.00   Max.   :56233   Max.   :118.00
```

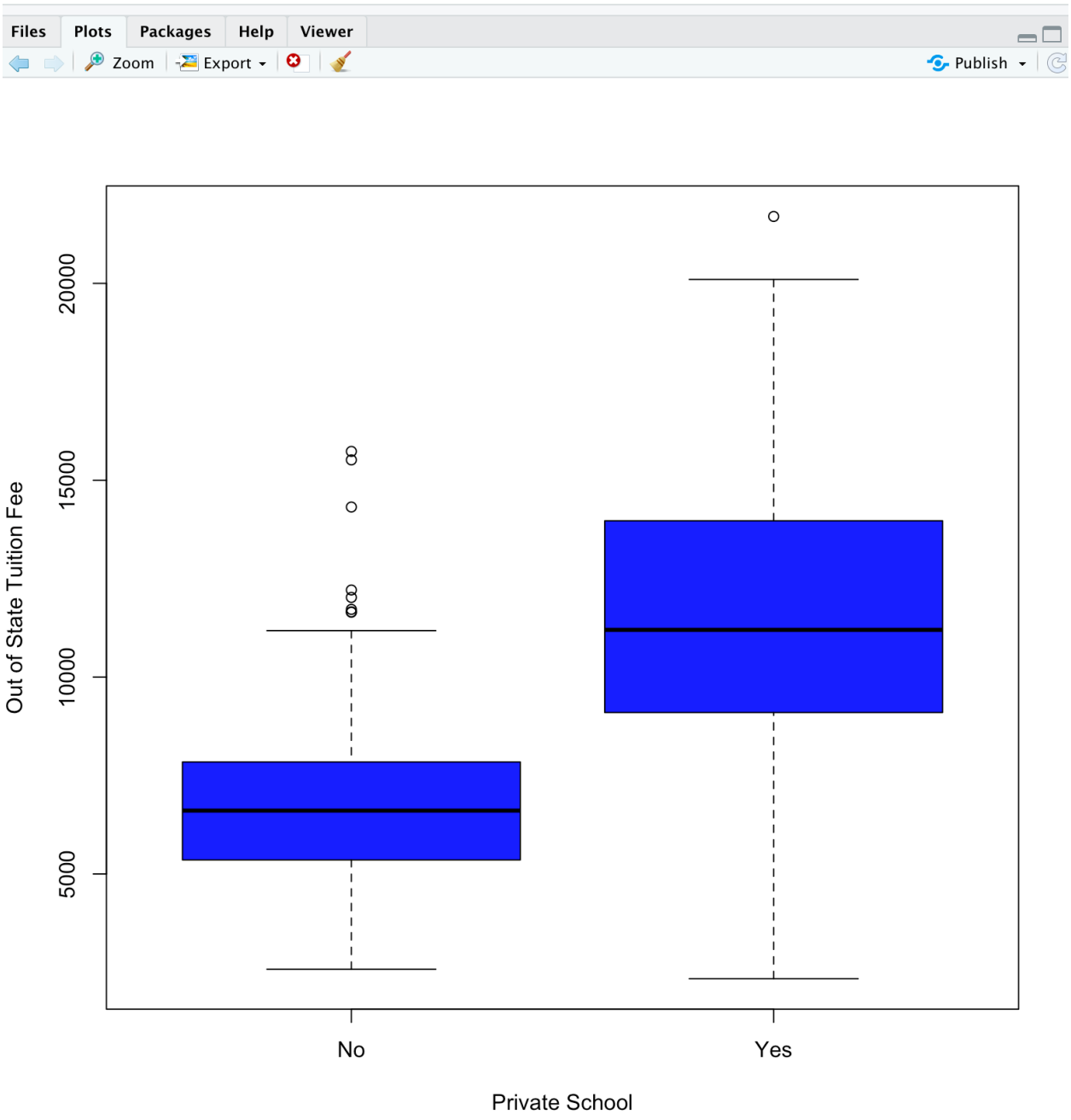
8.C. II:

```
> pairs(college[,1:10])
> # 8.c.ii: Pair Plot
> pairs(college[,1:10])
>
```



8.C. III:

```
> plot(college$Private, college$Outstate, col = "blue", xlab = "Private School", ylab =
 "Out of State Tuition Fee")
>
```

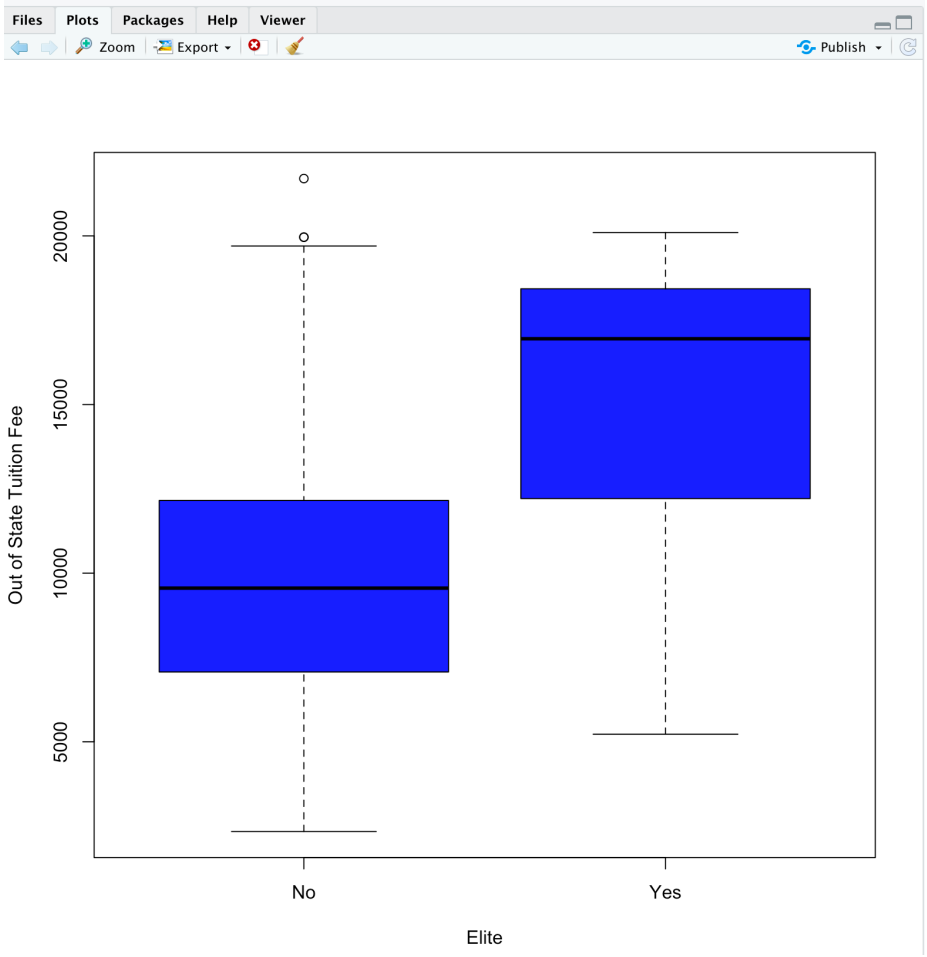


8.C. IV:

```
> # 8.c.iv.Create a new qualitative variable:
> Elite = rep("No", nrow(college))
> Elite[college$Top10perc >50] = "Yes"
> Elite = as.factor(Elite)
> college = data.frame(college, Elite)
> summary(Elite)
 No Yes
699  78
```

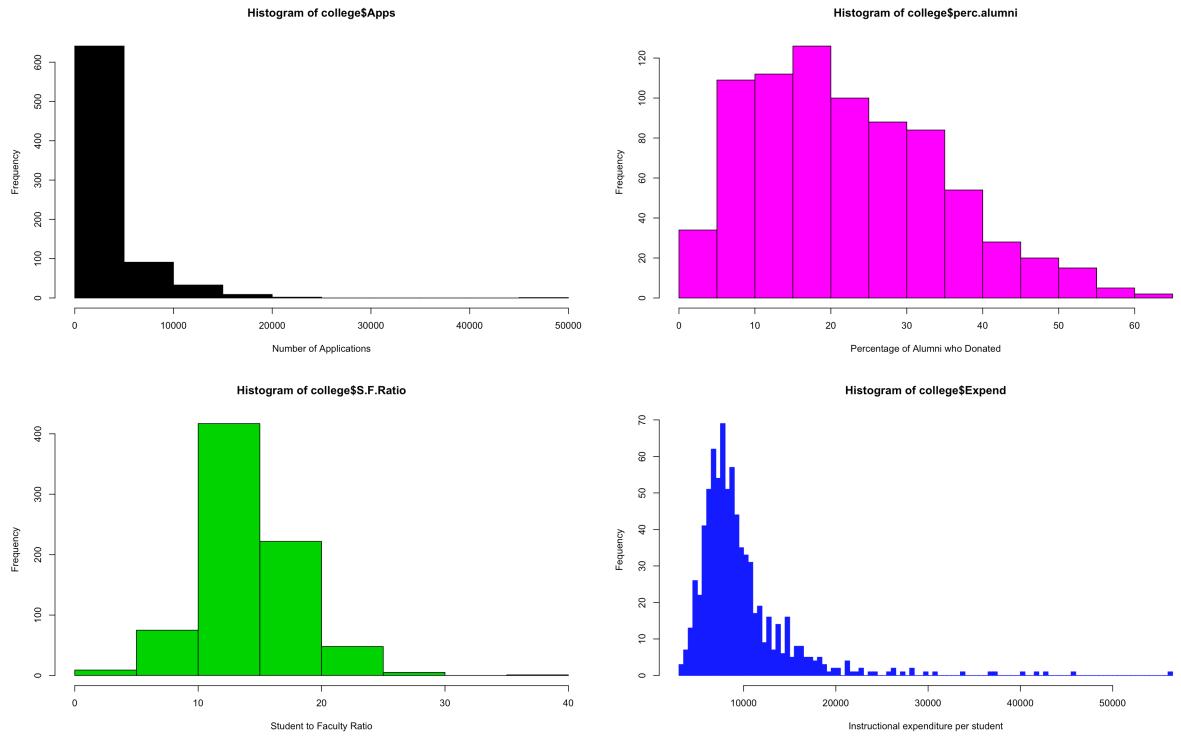
There are 78 elite universities and 699 non-elite universities.

```
> plot(college$Elite, college$Outstate, col = "blue", xlab = "Elite", ylab = "Out of Sta
te Tuition Fee")
>
```



8.C.V:

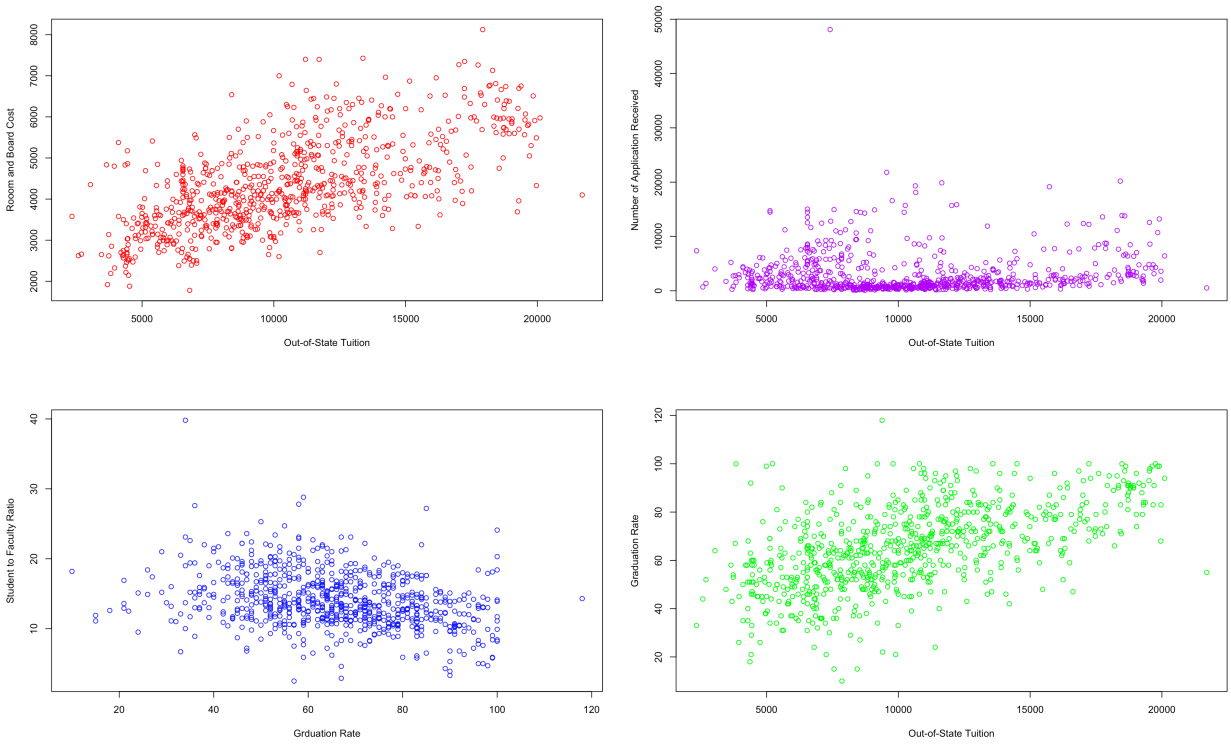
```
> #8.C.V: Create Histogram: Hist()
> par(mfrow = c(2,2))
> hist(college$Apps, xlab = "Number of Applications", ylab = "Frequency", col = 9)
> hist(college$perc.alumni, col = 6, xlab = "Percentage of Alumni who Donated", ylab =
 "Frequency",)
> hist(college$S.F.Ratio, col=3, breaks=10, xlab = "Student to Faculty Ratio", ylab = "F
requency",)
> hist(college$Expend, breaks=100, col = 12, xlab = "Instructional expenditure per stude
nt", ylab = "Fequency", border = 12, axes = TRUE, freq = T)
>
```



These histograms represent four different variables plotted from the “college” data. The last histogram in the picture plots probability density function.

8.C. VI:

```
> # 8. C. VI. Continuing exploring data:
> par(mfrow = c(2,2))
> plot(college$Outstate, college$Room.Board, xlab = "Out-of-State Tuition", ylab = "Rooo
m and Board Cost", col = "red")
> plot(college$Outstate, college$Apps, xlab = "Out-of-State Tuition", ylab = "Number of
 Application Received", col = "purple")
> plot(college$Grad.Rate, college$S.F.Ratio, xlab = "Grduation Rate", ylab = "Student to
 Faculty Ratio", col = "blue")
> plot(college$Outstate, college$Grad.Rate, xlab = "Out-of-State Tuition", ylab = "Gradu
ation Rate", col = "green")
>
```



The round and board cost are positively correlated with the out of state tuition. The number of applications received is high where out of tuition is low as we can see most points are clustered between 5,000 to 10,000 on the top right corner chart. The student faculty ration seems like decreasing with the increase in graduation rate, but the slope is not very steep. The graduation rate is positively correlated with the out of state tuition fee.

s

9 (a):

```
# 9. a)
data("Auto")
help(Auto)
Auto = na.omit(Auto)
origin1 = as.factor(Auto$origin)
auto = data.frame(Auto, origin1)
summary(auto)
```

```
> str(auto)
'data.frame':   392 obs. of  10 variables:
 $ mpg         : num  18 15 18 16 17 15 14 14 14 15 ...
 $ cylinders   : num  8 8 8 8 8 8 8 8 8 8 ...
 $ displacement: num  307 350 318 304 302 429 454 440 455 390 ...
 $ horsepower  : num  130 165 150 150 140 198 220 215 225 190 ...
 $ weight      : num  3504 3693 3436 3433 3449 ...
 $ acceleration: num  12 11.5 11 12 10.5 10 9 8.5 10 8.5 ...
 $ year        : num  70 70 70 70 70 70 70 70 70 70 ...
 $ origin      : num  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 ...
 $ origin1     : Factor w/ 3 levels "1","2","3": 1 1 1 1 1 1 1 1 1 1 ...
> summary(auto)
      mpg          cylinders      displacement     horsepower        weight
 Min.   : 9.00   Min.   :3.000   Min.   : 68.0   Min.   : 46.0   Min.   :1613
 1st Qu.:17.00   1st Qu.:4.000   1st Qu.:105.0   1st Qu.: 75.0   1st Qu.:2225
 Median :22.75   Median :4.000   Median :151.0   Median : 93.5   Median :2804
 Mean   :23.45   Mean   :5.472   Mean   :194.4   Mean   :104.5   Mean   :2978
 3rd Qu.:29.00   3rd Qu.:8.000   3rd Qu.:275.8   3rd Qu.:126.0   3rd Qu.:3615
 Max.   :46.60   Max.   :8.000   Max.   :455.0   Max.   :230.0   Max.   :5140

  acceleration        year           origin                      name     origin1
 Min.   : 8.00   Min.   :70.00   Min.   :1.000   amc matador       :  5   1:245
 1st Qu.:13.78   1st Qu.:73.00   1st Qu.:1.000   ford pinto        :  5   2: 68
 Median :15.50   Median :76.00   Median :1.000   toyota corolla    :  5   3: 79
 Mean   :15.54   Mean   :75.98   Mean   :1.577   amc gremlin       :  4
 3rd Qu.:17.02   3rd Qu.:79.00   3rd Qu.:2.000   amc hornet        :  4
 Max.   :24.80   Max.   :82.00   Max.   :3.000   chevrolet chevette:  4
                                                 (Other)           :365
```

Quantitative: mpg, cylinders, displacement, horsepower, weight, acceleration, year
Qualitative: Origin, name.

9.(B):

```
> sapply(Auto[, 1:7], range)
      mpg cylinders displacement horsepower weight acceleration year
[1,]  9.0         3           68         46   1613          8.0   70
[2,] 46.6         8          455        230   5140         24.8   82
```

This value above is range of each variable in the dataset. For example, mpg has range of 9 to 46.6 cylinder has range of 3 to 8 or range of 5. Other variables can be interpreted in same way.

9.C:

```
> sapply(Auto[, 1:7], mean)
         mpg    cylinders displacement   horsepower       weight acceleration
   23.445918     5.471939   194.411990   104.469388  2977.584184    15.541327
        year
   75.979592
> sapply(Auto[, 1:7], sd)
         mpg    cylinders displacement   horsepower       weight acceleration
    7.805007     1.705783   104.644004    38.491160   849.402560     2.758864
        year
    3.683737
```

The mean and standard deviation of variable mpg is 23.445918 and 7.805007 respectively. Other variables can be interpreted similarly.

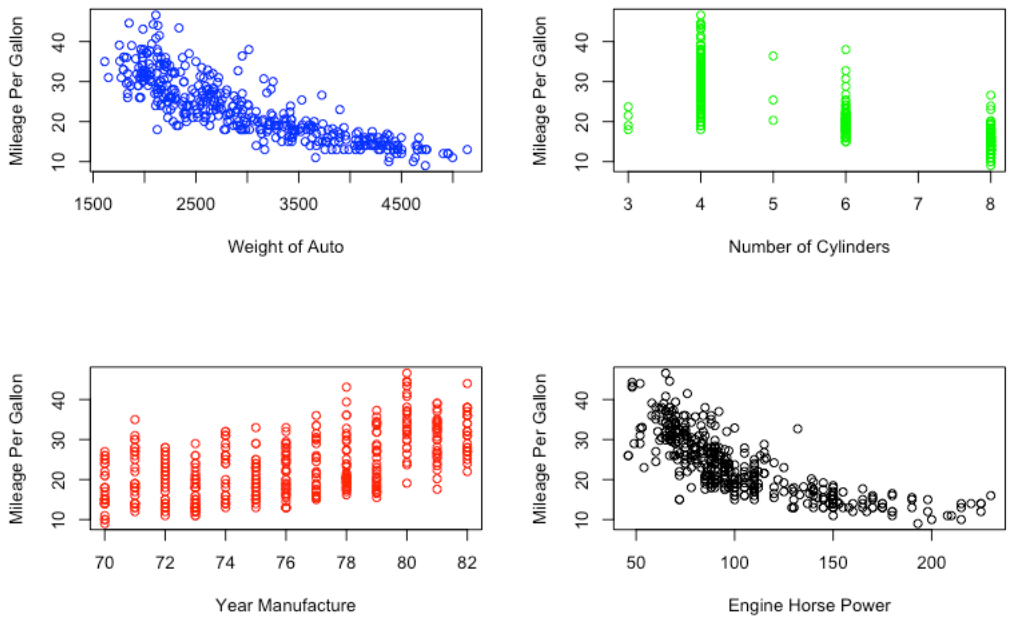
9.D:

```
> # 9. d)
> newAuto = Auto[-(10:85), ]
> # Range, Mean and Standard Deviation.
> sapply(newAuto[, 1:7], range)
      mpg cylinders displacement horsepower weight acceleration year
[1,] 11.0         3           68         46   1649          8.5   70
[2,] 46.6         8          455        230   4997         24.8   82
> sapply(newAuto[, 1:7], mean)
         mpg    cylinders displacement   horsepower       weight acceleration
   24.404430     5.373418   187.240506   100.721519  2935.971519    15.726899
        year
   77.145570
> sapply(newAuto[, 1:7], sd)
         mpg    cylinders displacement   horsepower       weight acceleration
    7.867283     1.654179    99.678367    35.708853   811.300208     2.693721
        year
    3.106217
```

Range, mean and standard deviation of variable mpg from newly created dataset "newAuto" are 11.0 to 46.6, 24.404430 and 7.867283 respectively. Other variables can be interpreted in same way.

9.E.:

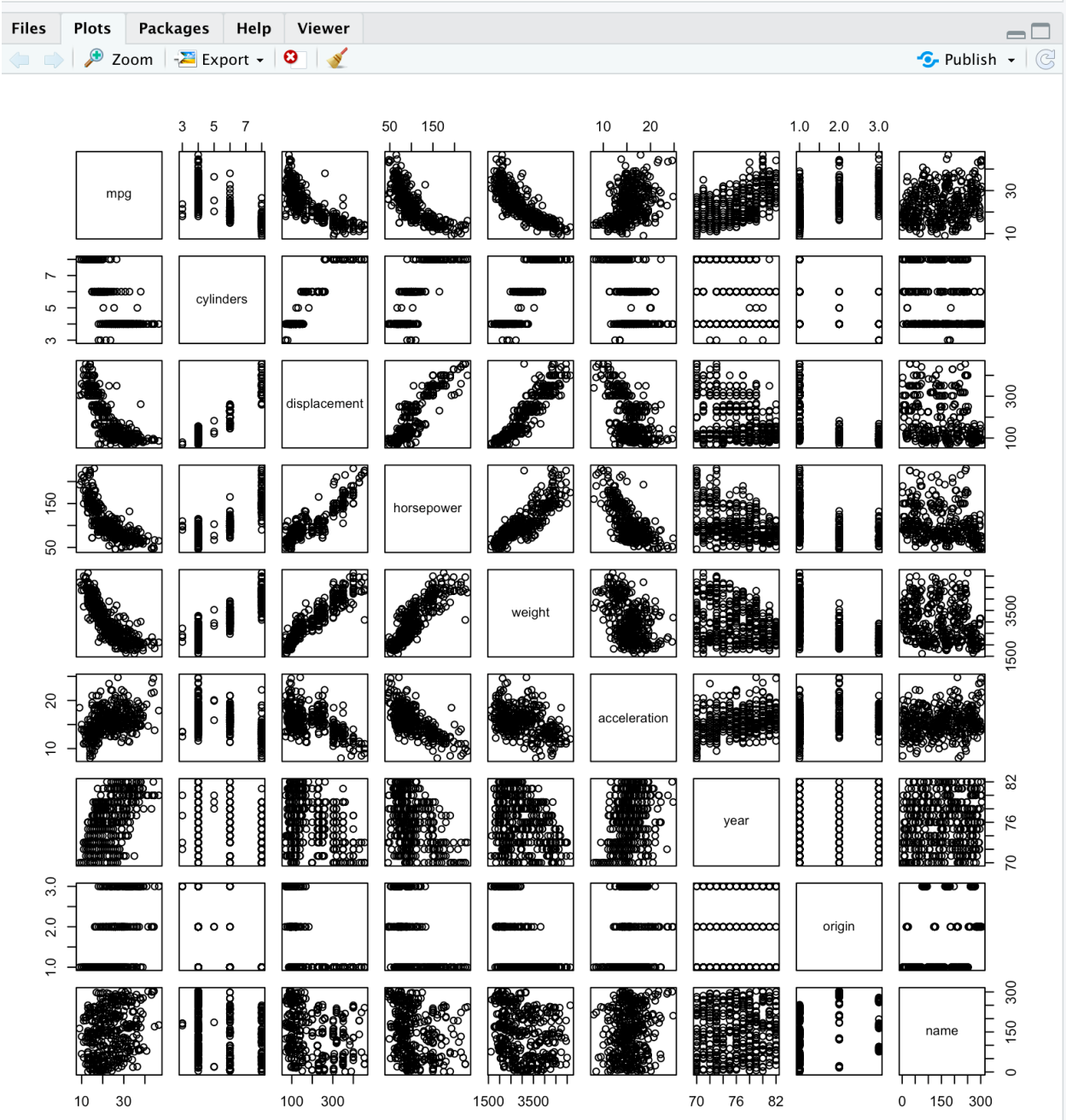
```
> # 9. e)
> par(mfrow = c(2,2))
> plot(y = Auto$mpg, x = Auto$weight, ylab = "Mileage Per Gallon", xlab = "Weight of Aut
o", col = "blue")
> plot(y = Auto$mpg, x = Auto$cylinders, ylab = "Mileage Per Gallon", xlab = "Number of
 Cylinders", col = "green")
> plot(y = Auto$mpg, x = Auto$year, ylab = "Mileage Per Gallon", xlab = "Year Manufactur
e", col = "red")
> plot(y = Auto$mpg, x = Auto$horsepower, ylab = "Mileage Per Gallon", xlab = "Engine Ho
rse Power", col = "black")
```



The milage (mile per gallon) decrease with the increase in auto weight, number of cylinders, and engine horsepower but increases with the year of manufacturing.

9.F:

```
> # 9. f)
> pairs(Auto)
>
```



The plot shows that the mpg decreases with the increase in cylinders, displacement, horsepower, weight but mpg increases with increase in acceleration, better in new year models. These are the variables that can be used to predict the mpg in the auto dataset. Other two variables origin and name does not have a visibly definite pattern.

```
rm(list = ls())
# Mac:
setwd("//Users//bmishra//...")
# Window:
setwd("C:\\Users\\bmishra\\...")
library(readr)
library(ISLR)
library(MASS)

# Chapter 2:

# Q. 8:
# 8.a) Use read.csv() to read data into R.
college  = read.csv(file = "college.csv",  header = TRUE, sep = " ")
```

```
head(college)

# 8.b)
fix(college)
rownames(college) = college [, 1]
college1 = college [, -1]
fix(college1)

# 8. c)
# 8.c.i: Summary:
summary(college)

# 8.c.ii: Pair Plot
pairs(college[,1:10])

# 8.c.iii. Plot
college = na.omit(college)
plot(college$Private, college$Outstate, col = "blue", xlab = "Private School", ylab = "Out of State Tuition Fee")

# 8.c.iv.Create a new qualitative variable:
Elite = rep("No", nrow(college))
Elite[college$Top10perc >50] = "Yes"
Elite = as.factor(Elite)
college = data.frame(college, Elite)
summary(Elite)
plot(college$Elite, college$Outstate, col = "blue", xlab = "Elite", ylab = "Out of State Tuition Fee")

#8.C.V: Create Histogram: Hist()
par(mfrow = c(2,2))
hist(college$Apps, xlab = "Number of Applications", ylab = "Frequency", col = 9)
hist(college$perc.alumni, col = 6, xlab = "Percentage of Alumni who Donated", ylab = "Frequency",)
hist(college$S.F.Ratio, col=3, breaks=10, xlab = "Student to Faculty Ratio", ylab = "Frequency",)
hist(college$Expend, breaks=100, col = 12, xlab = "Instructional expenditure per student", ylab = "Fequency", border = 12, axes = TRUE, freq = T)
```

```
# 8. C. VI. Continuing exploring data:
par(mfrow = c(2,2))
plot(college$Outstate, college$Room.Board, xlab = "Out-of-State Tuition", ylab = "Rooom and Board Cost", col = "red")
plot(college$Outstate, college$Apps, xlab = "Out-of-State Tuition", ylab = "Number of Application Received", col = "purple")
plot(college$Grad.Rate, college$S.F.Ratio, xlab = "Grduation Rate", ylab = "Student to Faculty Ratio", col = "blue")
plot(college$Outstate, college$Grad.Rate, xlab = "Out-of-State Tuition", ylab = "Graduation Rate", col = "green")

# 9:
# 9. a)
data("Auto")
help(Auto)
Auto = na.omit(Auto)
origin1 = as.factor(Auto$origin)
auto = data.frame(Auto, origin1)
summary(auto)
str(auto)

# 9. b)
sapply(Auto[, 1:7], range)
# 9. c)
sapply(Auto[, 1:7], mean)
sapply(Auto[, 1:7], sd)

# 9. d)
newAuto = Auto[-(10:85), ]
# Range, Mean and Standard Deviation.
sapply(newAuto[, 1:7], range)
sapply(newAuto[, 1:7], mean)
sapply(newAuto[, 1:7], sd)

# 9. e)
par(mfrow = c(2,2))
plot(y = Auto$mpg, x = Auto$weight, ylab = "Mileage Per Gallon", xlab = "Weight of Auto", col = "blue")
```

```
plot(y = Auto$mpg, x = Auto$cylinders, ylab = "Mileage Per Gallon", xlab = "Number of
Cylinders", col = "green")
plot(y = Auto$mpg, x = Auto$year, ylab = "Mileage Per Gallon", xlab = "Year
Manufacture", col = "red")
plot(y = Auto$mpg, x = Auto$horsepower, ylab = "Mileage Per Gallon", xlab = "Engine
Horse Power", col = "black")

# 9. f)
pairs(Auto)

##############################################################################
###################

#Outside Book:
rm(list = ls())
set.seed(1)
x = seq(from = -2, to = 2, by = .1)
y = 100 + 2*x - x^2 + rnorm(41)

# Problem 1:
xsq = x*x
xcu = xsq*x
fx = 100 + 2*x - x^2 #True Y
f1xhatt = lm(y ~ x) #Linear
f1xhatt
f2xhatt = lm(y ~ x + xsq + xcu) #Quadratic.
f2xhatt
plot(x,y)
# Predicting f1:
f1.linear = round(cbind(y,fx= 100+2*x - x^2, f1xhatt = predict(f1xhatt),
error=resid(f1xhatt)),2)
f1.linear [, 2]
# Predicting f2
f2.quad = round(cbind(y,fx= 100+2*x - x^2, f2xhatt = predict(f2xhatt),
error=resid(f2xhatt)),2)
f2.quad [, 2]

# Problem 2:
plot (x, y, col = 1, xlab = "x", ylab = "Y or f-hatts", main = "Data Plot with Estimation Line")
```

```
curve (100 + 2*x - x^2, add = TRUE, col = 5, lyt = 2, lwd = 2) #True Y.
curve(98.686 + 1.9561*x, add = TRUE, col = 10, lyt = 4, lwd = 3) #Linear
curve(100.0993 + 1.6444*x -1.0097*x^2 + 0.1238*x^3, add = TRUE, col = 15, lyt = 4, lwd
= 4) #Quadratic

# Probelm 3:
truef = function(x) {100 + 2*x - x^2}
truef(0)
predict (f1xhatt, data.frame( x = 0)) #Linear Function
points(0, 100, pch = 9, col = "black") #True Y
points(0, 98.68577, pch = 9, col = "blue") # Linear Function

# Probelm 4:
truef = function(x) {100 + 2*x - x^2}
truef(0)
predict (f2xhatt, data.frame( x = 0, xsq = 0, xcu = 0)) #Quadratic
points(0, 100.0993, pch = 9, col = "green") # Quadratic Function.

# Problem 5:
truef(x = 0)
y[21]
points(x[21], y[21], pch = 9, col = "orange") #

# Problem 6: #Pending Revise Code. Did Manually.
test.data1 = data.frame(x = c(-1, 0, 1), y = c(94, 100, 100))
f1xhatt_pred = predict(f1xhatt, test.data1)
f1xhatt.mse = (sum(test.data1$y-f1hatt_pred)^2)/(length(test.data1$y)) #MSE fmla.
f1hatt.mse
test.data2 = data.frame(x = c(-1, 0, 1), y = c(94, 100, 100))
f2xhatt_pred = predict(f2xhatt, test.data)
f2xhatt.mse = (sum(test.data$y-f2hatt_pred)^2)/(length(test.data$y)) #MSE fmla.
f2hatt.mse

# Problem 7:
set.seed(2)
x7 = seq(from = -2, to = 2, by = .1)
y7 = 100 + 2*x7 - x7^2 + rnorm(41)
x7sq = x7*x7
x7cu = x7sq*x7
```

```
fx7 = 100 + 2*x7 - x7^2 #True Y
f1x7hatt = lm(y7 ~ x7) #Linear
f1x7hatt
f2x7hatt = lm(y7 ~ x7 + x7sq + x7cu) #Quadratic.
f2x7hatt

truef7a = function(x7) {100 + 2*x7 - x7^2}
truef7a(0)
predict (f1x7hatt, data.frame( x7 = 0)) #Linear Function

truef7b = function(x7) {100 + 2*x7 - x7^2}
truef7b(0)
predict (f2x7hatt, data.frame( x7 = 0, x7sq = 0, x7cu = 0)) #Quadratic
```