Homework 1

Tianyu Li, Jingyan Sun

January 30, 2019

#### Question 1 (P52-Q1)

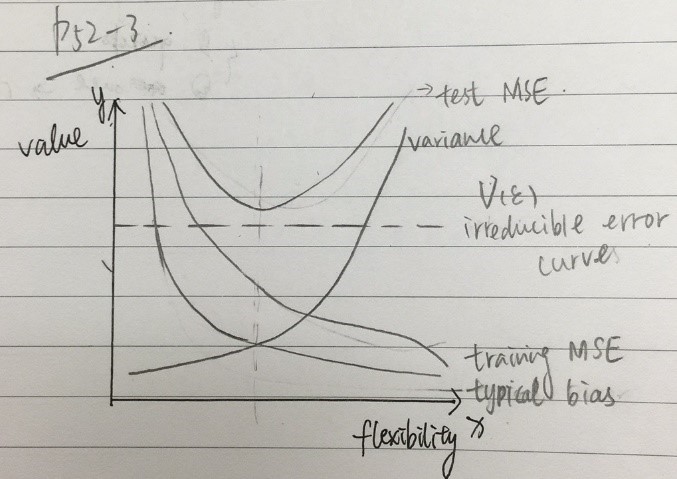
Solutions:

1. A Flexible Method could be better for this situation. Because the extremely large capacity of the sample, it is better to use a more complicated and flexible method to fit the data, which will be more accurate than use an inflexible method under this circumstance.
2. An inflexible Method could be better for this situation. Because the observation capacity is much smaller than the predictors, it is better to use a simpler and inflexible method to cater the basic need. The Flexible Method is better to adopt when condition the number of sample (or observation) is equal or larger than the number of predictors, otherwise it will be inefficient and overfitting the data.
3. A Flexible Method could be better for this situation. No-linear Relationship (or Regression) means the simple-inflexible method would fail to cater the need of the expression. For a more accurate expression, the Flexible Method with higher degree of freedom would fit the data better.
4. Flexible Methods could worsen the situation. With the extremely high variance, A very flexible methods would be fit here and cause the overfitting. An inflexible model here may avoid this situation and get the key features of the data.

#### Question 2 (P52-Q3)

Solutions:

(a)



(b)

For Test Error and Training Error (Corresponding to Test MSE and Training MSE in the figure 1): as the increase of the flexibility, the training MSE will decrease but the test MSE will show a trend of parabola (a>0) above irreducible error (the dash line). That is because as flexibility increases, the f curve fits the observed data (training data) closer; in opposite, a large flexibility will cause Test data an overfitting, so the possibility of error increases after the lowest point. In real case, lower in test MSE is more important than training MSE.

For Typical bias and Variance: these two curves show the opposite trend. The bias will decrease because a flexible method will fit the sample closer. The variance will increase because a flexible method will overfit. In real case, a Bias-Variance Trade-off should be applied to find the appropriate degree of flexibility.

For Irreducible Error: The value of it should remain a constant as it is “irreducible” no matter which model is chosen. And it should be lower than the Test MSE as it is the lowest possible error.

#### Question 3 (P53-Q6)

The difference between Parametric Method and Non-Parametric Method: Parametric Method reduces the problem of estimating f down to one of estimating a set of parameters. It needs a two-step model to make an assumption about the functional form of f and to uses training data to fit the model. Non-Parametric Method does not make explicit assumptions about the functional form of f and needs a large sample to accurate the estimate f.

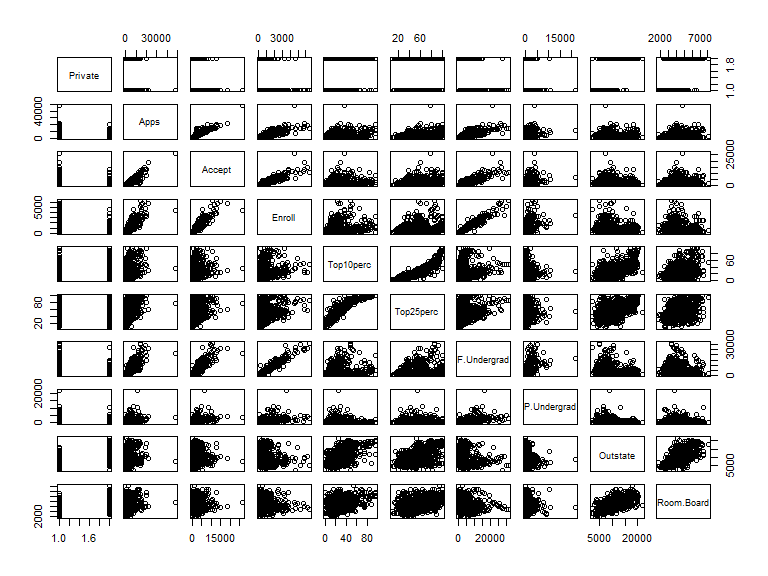
The advantage of using Parametric Method for regression or classification: simplify the problem of estimating f; reduce the requirements for the large sample number comparing to Non-Parametric Methods. the potential disadvantage of the Parametric Method is that the model we choose will usually not match the true unknown form of f, the estimate f might very different from the true f which will cause the resulting model fail to fit the data well.

#### Question 4

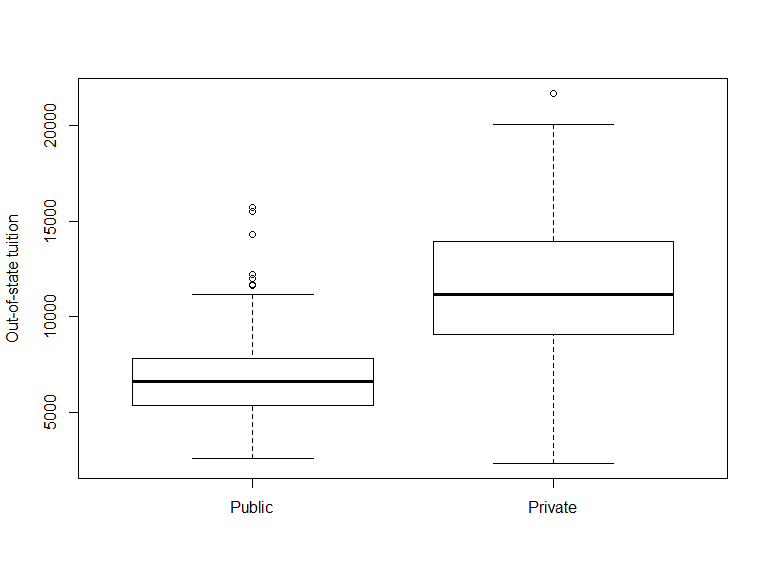
# Author: Tianyu Li  
# Created on Jan 21st 2019  
# Edited on Jan 28th 2019  
#  
# R script for Homework 1 Question 4(Section 2.4, page 54-55, question 8)  
# The College.csv file should be in working direction   
setwd('Z:/R\_working\_directory/DS502HW1');  
  
#(a) Read the file  
college = read.csv(file = 'College.csv', header = TRUE);  
  
#(b) Give row names  
rownames(college) = college[,1];  
fix(college);  
  
college = college[,-1];  
fix(college);  
  
#(c) i.Summary function  
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

#(c) ii.Pairs function  
pairs(college[,1:10]);



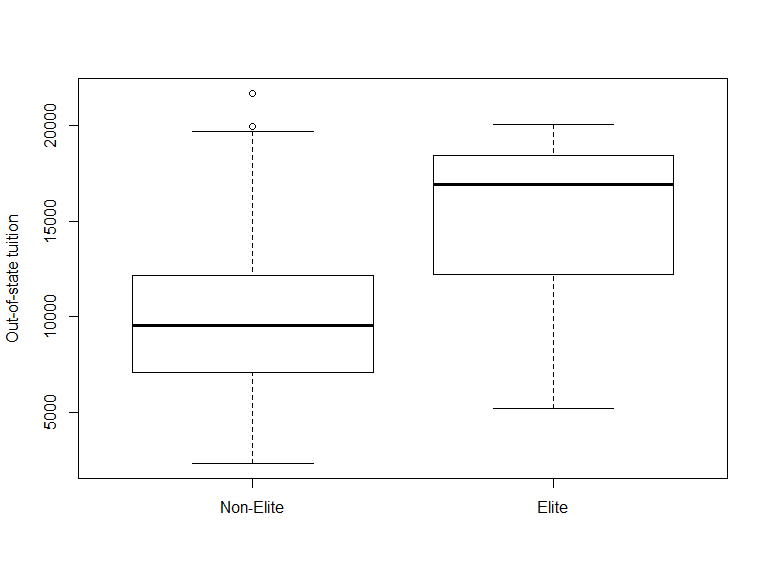
#(c) iii.Plot function  
boxplot(college$Outstate ~ college$Private, names = c("Public", "Private"),  
 ylab = "Out-of-state tuition");



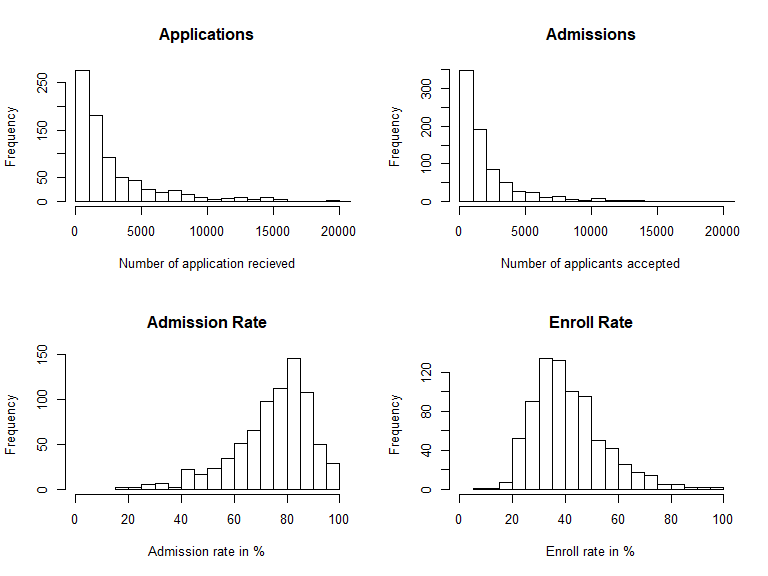
#(c) iv.Factor function  
Elite = rep("No", nrow(college));  
Elite[college$Top10perc > 50] = "Yes";  
Elite = as.factor(Elite);  
college = data.frame(college, Elite);  
  
summary(college$Elite);

## No Yes   
## 699 78

boxplot(college$Outstate ~ college$Elite, names = c("Non-Elite", "Elite"),  
 ylab = "Out-of-state tuition");



#(c) v.Hist function  
par(mfrow=c(2,2));  
  
hist(college$Apps, breaks = 40, xlim = range(0, 20000), main = "Applications",  
 xlab = "Number of application recieved");  
hist(college$Accept, breaks = 20, xlim = range(0, 20000), main = "Admissions",  
 xlab = "Number of applicants accepted");  
hist(100 \* college$Accept / college$Apps, breaks = 20, xlim = range(0, 100),  
 main = "Admission Rate", xlab = "Admission rate in %");  
hist(100 \* college$Enroll / college$Accept, breaks = 20, xlim = range(0, 100),  
 main = "Enroll Rate", xlab = "Enroll rate in %");



#(c) vi.Continue exploring  
summary(lm(Grad.Rate ~ . , data = college));

##   
## Call:  
## lm(formula = Grad.Rate ~ ., data = college)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -53.991 -7.100 -0.300 7.174 54.034   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 33.8925541 4.8522723 6.985 6.24e-12 \*\*\*  
## PrivateYes 3.4262050 1.7151733 1.998 0.046119 \*   
## Apps 0.0012936 0.0004428 2.921 0.003588 \*\*   
## Accept -0.0006909 0.0008638 -0.800 0.424030   
## Enroll 0.0021440 0.0023111 0.928 0.353840   
## Top10perc 0.0465274 0.0851268 0.547 0.584838   
## Top25perc 0.1374511 0.0564462 2.435 0.015118 \*   
## F.Undergrad -0.0004648 0.0004026 -1.155 0.248635   
## P.Undergrad -0.0014809 0.0003907 -3.790 0.000162 \*\*\*  
## Outstate 0.0010197 0.0002339 4.360 1.48e-05 \*\*\*  
## Room.Board 0.0019067 0.0005926 3.217 0.001348 \*\*   
## Books -0.0022140 0.0029189 -0.758 0.448388   
## Personal -0.0016620 0.0007703 -2.158 0.031270 \*   
## PhD 0.0882924 0.0571134 1.546 0.122543   
## Terminal -0.0751566 0.0624063 -1.204 0.228845   
## S.F.Ratio 0.0746163 0.1595478 0.468 0.640153   
## perc.alumni 0.2796432 0.0492353 5.680 1.92e-08 \*\*\*  
## Expend -0.0004596 0.0001552 -2.961 0.003158 \*\*   
## EliteYes 0.4618984 2.5235781 0.183 0.854821   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 12.75 on 758 degrees of freedom  
## Multiple R-squared: 0.4616, Adjusted R-squared: 0.4488   
## F-statistic: 36.1 on 18 and 758 DF, p-value: < 2.2e-16

# We chose graduate rate as the dependent variable and tried to fit in a linear model of other columns.  
# We found that the Number of part-time student is negatively correlated to the graduate rate significantly and   
# the outstate tuition and percent of alumni who donate are postively correlated to the graduate rate significantly.  
# Besides, the number of applicants, room and board costs and instructional expenditure per student are  
# also correlated to the graduate rate.

#### Question 5

# Author: Tianyu Li  
# Created on Jan 28th 2019  
# Edited on Jan 29th 2019  
#  
# R script for Homework 1 Question 5(Section 2.4, page 56, question 9)  
# The Auto.csv file should be in working direction   
setwd('Z:/R\_working\_directory/DS502HW1');  
  
# Read the file  
auto = read.csv(file = 'Auto.csv', header = TRUE);  
  
# Remove missing values  
auto[auto == '?'] <- NA;  
auto = na.omit(auto);  
auto$horsepower = as.numeric(as.character(auto$horsepower));  
  
#(a) The last 2 predictors are qualitative, the others are quantative.  
# The "origin" should stands for the continent so it is qualitative.  
  
#(b) Range of each quantitative predictor  
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

#(c) Mean and standard deviation of each quantitative predictor  
sapply(auto[, 1:7], mean);

## mpg cylinders displacement horsepower weight   
## 23.445918 5.471939 194.411990 104.469388 2977.584184   
## acceleration year   
## 15.541327 75.979592

sapply(auto[, 1:7], sd);

## mpg cylinders displacement horsepower weight   
## 7.805007 1.705783 104.644004 38.491160 849.402560   
## acceleration year   
## 2.758864 3.683737

#(d) Remove the 10th through 85th observations.  
subAuto = auto[-(10:85),];  
sapply(subAuto[, 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

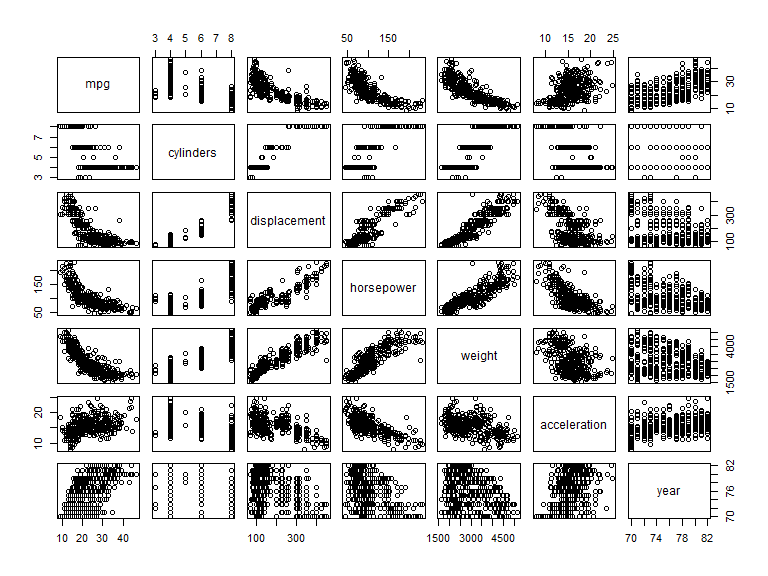
sapply(subAuto[, 1:7], mean);

## mpg cylinders displacement horsepower weight   
## 24.404430 5.373418 187.240506 100.721519 2935.971519   
## acceleration year   
## 15.726899 77.145570

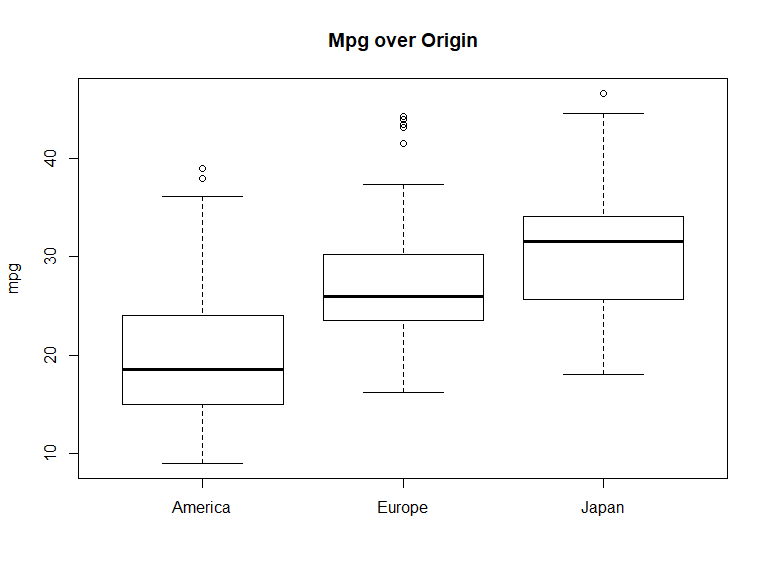
sapply(subAuto[, 1:7], sd);

## mpg cylinders displacement horsepower weight   
## 7.867283 1.654179 99.678367 35.708853 811.300208   
## acceleration year   
## 2.693721 3.106217

#(e) Create some plots  
pairs(auto[, 1:7]);



# We found that the displacement and weight look postively correlated to the horsepower.  
  
#(f) Predictors that might be useful in predicting mpg  
# From the figures of last question, we found that mpg is postively correlated to year and  
# negatively correlated to displacement, horsepower and weight.   
boxplot(auto$mpg ~ auto$origin, names = c("America", "Europe", "Japan"), ylab = "mpg",   
 main = "Mpg over Origin");



# We also found that for the origin column, 1 should stands for America, 2 stands for Europe  
# and 3 stands for Asia. In general, asian cars is highest on mpg while american cars is lowest.

#### Question 6 (P120-Q1)

Solutions:

In this case, we can assume the advertising budgets of TV, Radio and Newspaper do not affect the sales. Multiple Linear Regression Model is

The null hypothesis with , ().

Since the corresponding p-value for TV and Radio are highly significant but not significant for Newspaper, we reject the and , and accept . Which means in this case, the TV and Radio advertising could affect the sales, but the Newspaper advertising do not affect sales significantly.

#### Question 7 (P121-Q5)

Solutions:

1. When , substitute into :

-----eqn.1

1. When , let eqn.1 equivalent to function in terms of :

#### Question 8 (P121-Q6)

Solutions:

Reminds the Eqn.3.4 on textbook is

The function of y is

Assume the point , substitute the

So, we conclude that the point always on the least square line.

#### Question 9

# Author: Tianyu Li  
# Created on Jan 29th 2019  
#  
# R script for Homework 1 Question 9(Section 3.7, page 121-122, question 8)  
# The Auto.csv file should be in working direction   
setwd('Z:/R\_working\_directory/DS502HW1');  
  
# Read the file  
auto = read.csv(file = 'Auto.csv', header = TRUE);  
  
# Remove missing values  
auto[auto == '?'] <- NA;  
auto = na.omit(auto);  
auto$horsepower = as.numeric(as.character(auto$horsepower));  
  
#(a) Perform a simple linear regression  
temp = lm(mpg ~ horsepower, data = auto);  
summary(temp);

##   
## Call:  
## lm(formula = mpg ~ horsepower, data = auto)  
##   
## 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

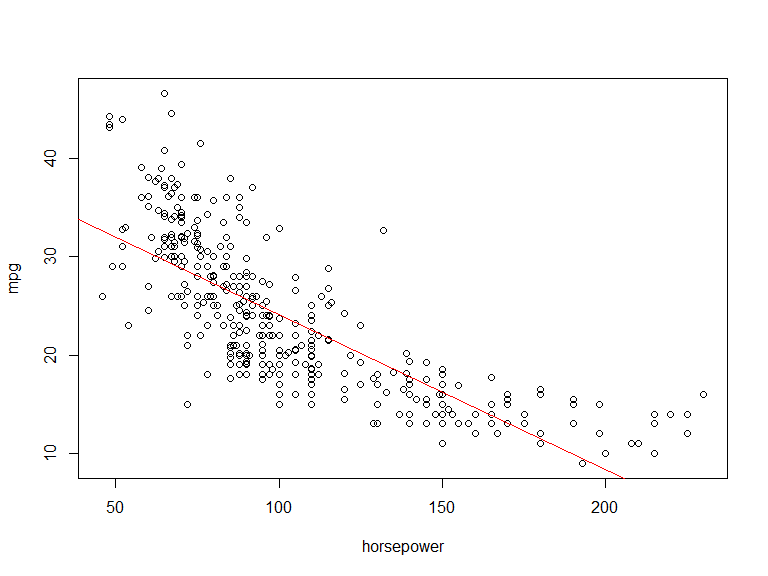
# i Yes  
# ii The relationship is strong as the value of R^2 is 0.6059  
# iii Negative, the coefficient is -0.157845.  
# iv Predict mpg associated with a horsepower of 98  
predict(temp, data.frame(horsepower = 98), interval = "confidence");

## fit lwr upr  
## 1 24.46708 23.97308 24.96108

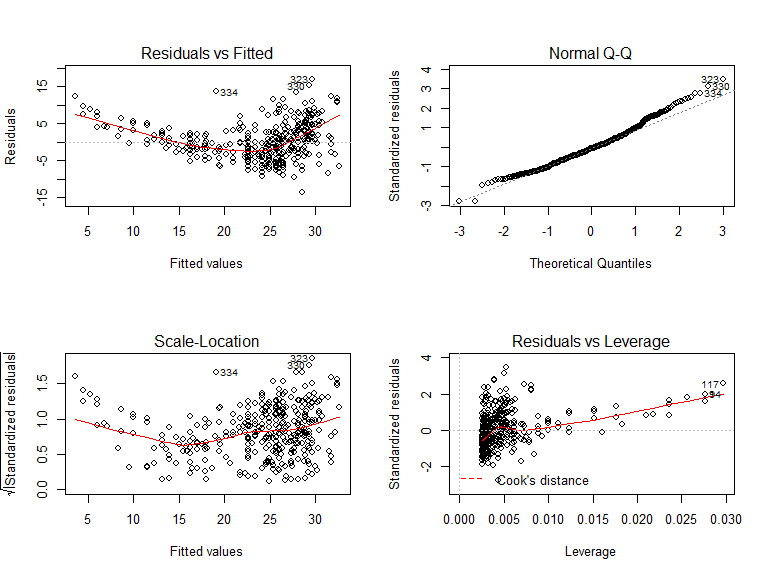
predict(temp, data.frame(horsepower = 98), interval = "prediction");

## fit lwr upr  
## 1 24.46708 14.8094 34.12476

#(b) Plot the response and the predictor  
par(mfrow=c(1,1));  
plot(mpg ~ horsepower, data = auto);  
abline(temp, col = 'red');



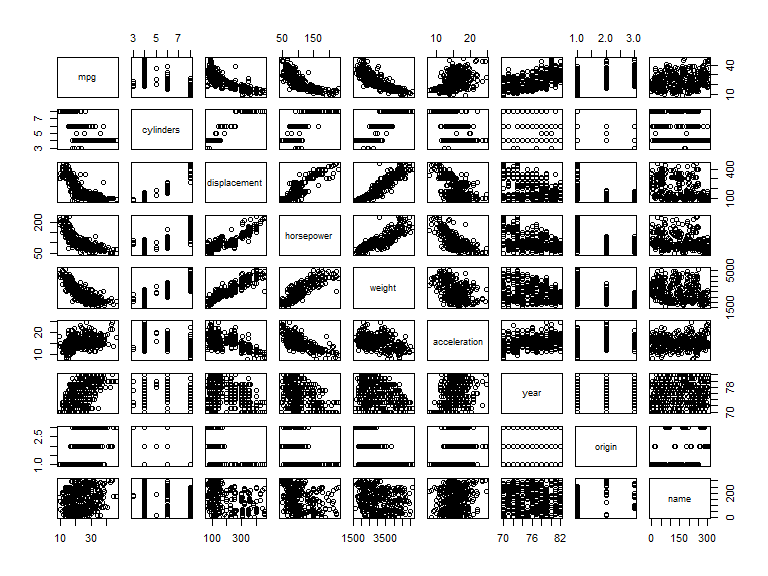
#(c) Produce diagnostic plots of the least squares regression fit  
par(mfrow = c(2, 2))  
plot(temp);



# The plots (the first and third one) show that the relationship is non-linear.

#### Question 10

# Author: Tianyu Li  
# Created on Jan 29th 2019  
#  
# R script for Homework 1 Question 10(Section 3.7, page 122, question 9)  
# The Auto.csv file should be in working direction   
setwd('Z:/R\_working\_directory/DS502HW1');  
  
# Read the file  
auto = read.csv(file = 'Auto.csv', header = TRUE);  
  
# Remove missing values  
auto[auto == '?'] <- NA;  
auto = na.omit(auto);  
auto$horsepower = as.numeric(as.character(auto$horsepower));  
  
#(a) Produce a scatterplot matrix  
pairs(auto);



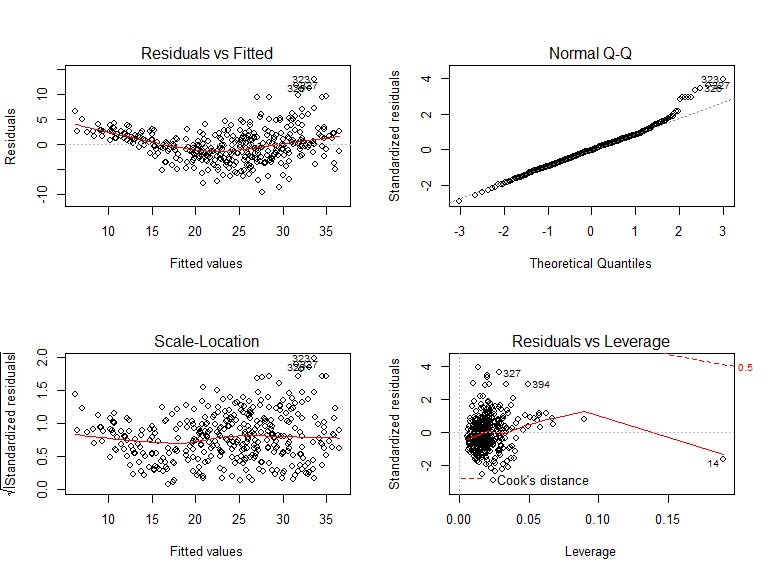
#(b) Compute the matrix of correlations  
cor(auto[, 1:8]);

## mpg cylinders displacement horsepower weight  
## mpg 1.0000000 -0.7776175 -0.8051269 -0.7784268 -0.8322442  
## cylinders -0.7776175 1.0000000 0.9508233 0.8429834 0.8975273  
## displacement -0.8051269 0.9508233 1.0000000 0.8972570 0.9329944  
## horsepower -0.7784268 0.8429834 0.8972570 1.0000000 0.8645377  
## weight -0.8322442 0.8975273 0.9329944 0.8645377 1.0000000  
## acceleration 0.4233285 -0.5046834 -0.5438005 -0.6891955 -0.4168392  
## year 0.5805410 -0.3456474 -0.3698552 -0.4163615 -0.3091199  
## origin 0.5652088 -0.5689316 -0.6145351 -0.4551715 -0.5850054  
## acceleration year origin  
## mpg 0.4233285 0.5805410 0.5652088  
## cylinders -0.5046834 -0.3456474 -0.5689316  
## displacement -0.5438005 -0.3698552 -0.6145351  
## horsepower -0.6891955 -0.4163615 -0.4551715  
## weight -0.4168392 -0.3091199 -0.5850054  
## acceleration 1.0000000 0.2903161 0.2127458  
## year 0.2903161 1.0000000 0.1815277  
## origin 0.2127458 0.1815277 1.0000000

#(c) Perform a multiple linear regression with mpg  
temp = lm(mpg ~ . - name, data = auto);  
summary(temp);

##   
## Call:  
## lm(formula = mpg ~ . - name, data = auto)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -9.5903 -2.1565 -0.1169 1.8690 13.0604   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -17.218435 4.644294 -3.707 0.00024 \*\*\*  
## cylinders -0.493376 0.323282 -1.526 0.12780   
## displacement 0.019896 0.007515 2.647 0.00844 \*\*   
## horsepower -0.016951 0.013787 -1.230 0.21963   
## weight -0.006474 0.000652 -9.929 < 2e-16 \*\*\*  
## acceleration 0.080576 0.098845 0.815 0.41548   
## year 0.750773 0.050973 14.729 < 2e-16 \*\*\*  
## origin 1.426141 0.278136 5.127 4.67e-07 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 3.328 on 384 degrees of freedom  
## Multiple R-squared: 0.8215, Adjusted R-squared: 0.8182   
## F-statistic: 252.4 on 7 and 384 DF, p-value: < 2.2e-16

#i Yes, the adjusted R-squared value is 0.8185, which is high.  
#ii From the summary table, we may conclude that 'displacement', 'weight',  
# 'year', 'origin' have a statistically significant relationship to the response.  
#iii It suggests that in average, 0.750773 mpg will increase as a year increase.  
# Which means the fuel efficiency is improving.  
  
  
#(d) Produce diagnostic plots of the linear regression fit.  
par(mfrow=c(2,2));  
plot(temp);



# No unusually large outliers are observed.   
# The plot shows that the 14th data have a relatively high leverage.  
temp2 = lm(mpg ~ . - name, data = auto[-(14),]);  
summary(temp2);

##   
## Call:  
## lm(formula = mpg ~ . - name, data = auto[-(14), ])  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -9.551 -2.147 -0.048 1.889 13.056   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -1.771e+01 4.644e+00 -3.813 0.00016 \*\*\*  
## cylinders -5.469e-01 3.242e-01 -1.687 0.09247 .   
## displacement 2.306e-02 7.745e-03 2.977 0.00309 \*\*   
## horsepower -1.105e-02 1.422e-02 -0.777 0.43769   
## weight -6.916e-03 7.046e-04 -9.815 < 2e-16 \*\*\*  
## acceleration 1.163e-01 1.010e-01 1.151 0.25043   
## year 7.551e-01 5.093e-02 14.825 < 2e-16 \*\*\*  
## origin 1.427e+00 2.775e-01 5.142 4.35e-07 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 3.32 on 383 degrees of freedom  
## Multiple R-squared: 0.822, Adjusted R-squared: 0.8188   
## F-statistic: 252.7 on 7 and 383 DF, p-value: < 2.2e-16

# By removing the 14th data, the R-squared value increase for 0.0006  
  
#(e) Fit linear regression models with interaction effects.  
temp3 = lm(mpg ~ . - name + horsepower:weight + horsepower:displacement, data = auto);  
summary(temp3);

##   
## Call:  
## lm(formula = mpg ~ . - name + horsepower:weight + horsepower:displacement,   
## data = auto)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -8.5879 -1.5160 -0.0954 1.3493 11.9604   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.364e+00 4.476e+00 0.305 0.760723   
## cylinders 4.152e-01 3.143e-01 1.321 0.187275   
## displacement -4.432e-02 1.652e-02 -2.684 0.007596 \*\*   
## horsepower -2.256e-01 2.339e-02 -9.646 < 2e-16 \*\*\*  
## weight -6.623e-03 1.556e-03 -4.256 2.63e-05 \*\*\*  
## acceleration -1.770e-01 9.123e-02 -1.941 0.053037 .   
## year 7.515e-01 4.468e-02 16.818 < 2e-16 \*\*\*  
## origin 7.046e-01 2.511e-01 2.806 0.005276 \*\*   
## horsepower:weight 2.541e-05 1.036e-05 2.453 0.014625 \*   
## displacement:horsepower 3.194e-04 9.601e-05 3.327 0.000964 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2.893 on 382 degrees of freedom  
## Multiple R-squared: 0.8657, Adjusted R-squared: 0.8626   
## F-statistic: 273.7 on 9 and 382 DF, p-value: < 2.2e-16

# The interaction between displacement and horsepower appear to be statistically significant.  
  
  
#(f) Try a few different transformations of the variables  
cor(data.frame(auto$weight, log(auto$weight), sqrt(auto$weight), (auto$weight)^2), auto$mpg);

## [,1]  
## auto.weight -0.8322442  
## log.auto.weight. -0.8441938  
## sqrt.auto.weight. -0.8400951  
## X.auto.weight..2 -0.8066816

cor(data.frame(auto$horsepower, log(auto$horsepower), sqrt(auto$horsepower), (auto$horsepower)^2), auto$mpg);

## [,1]  
## auto.horsepower -0.7784268  
## log.auto.horsepower. -0.8175174  
## sqrt.auto.horsepower. -0.8023114  
## X.auto.horsepower..2 -0.7122970

cor(data.frame(auto$cylinders, log(auto$cylinders), sqrt(auto$cylinders), (auto$cylinders)^2), auto$mpg);

## [,1]  
## auto.cylinders -0.7776175  
## log.auto.cylinders. -0.7768177  
## sqrt.auto.cylinders. -0.7783516  
## X.auto.cylinders..2 -0.7703552

cor(data.frame(auto$displacement, log(auto$displacement), sqrt(auto$displacement), (auto$displacement)^2), auto$mpg);

## [,1]  
## auto.displacement -0.8051269  
## log.auto.displacement. -0.8284533  
## sqrt.auto.displacement. -0.8213314  
## X.auto.displacement..2 -0.7523545

cor(data.frame(auto$acceleration, log(auto$acceleration), sqrt(auto$acceleration), (auto$acceleration)^2), auto$mpg);

## [,1]  
## auto.acceleration 0.4233285  
## log.auto.acceleration. 0.4359007  
## sqrt.auto.acceleration. 0.4306775  
## X.auto.acceleration..2 0.4037617

cor(data.frame(auto$origin, log(auto$origin), sqrt(auto$origin), (auto$origin)^2), auto$mpg);

## [,1]  
## auto.origin 0.5652088  
## log.auto.origin. 0.5742758  
## sqrt.auto.origin. 0.5708022  
## X.auto.origin..2 0.5483534

cor(data.frame(auto$year, log(auto$year), sqrt(auto$year), (auto$year)^2), auto$mpg);

## [,1]  
## auto.year 0.5805410  
## log.auto.year. 0.5765192  
## sqrt.auto.year. 0.5785682  
## X.auto.year..2 0.5842529

# Weight, horsepower, displacement, acceleration and origin fit the log transformation best,  
# Cylinders fit the square root transformation best and year fit square transfromation best.