MATH 4322 Homework 2 Solutions

Instructor: Dr. Cathy Poliak

9/24/2021

Problem 1

Suppose we have a data set with five predictors, X_1 =GPA, X_2 = IQ, X_3 = Gender (1 for Female and 0 for Male), X_4 = Interaction between GPA and IQ, and X_5 = Interaction between GPA and Gender. The response is starting salary after graduation (in thousands of dollars). Suppose we use least squares to fit the model, and get $\hat{\beta}_0 = 50$, $\hat{\beta}_1 = 20$, $\hat{\beta}_2 = 0.07$, $\hat{\beta}_3 = 35$,$\hat{\beta}_4 = 0.01$, $\hat{\beta}_5 = -10$.

(a) Which answer is correct, and why?

iii. For a fixed value of IQ and GPA, males earn more on average than females provided that the GPA is high enough because:

- The predicted regression equation for multiple variables is:
  - $\hat{Y} = 50 + 20X_1 + 0.07X_2 + 35X_3 + 0.01X_4 - 10X_5$
- The male regression equation is:
  - $\hat{Y}_m = 50 + 20X_1 + 0.07X_2 + 0.01X_4$
- The female regression equation is:
  - $\hat{Y}_f = 85 + 20X_1 + 0.07X_2 + 0.01X_4 - 10X_5$

(b) Predict the salary of a female with IQ of 110 and a GPA of 4.0.

- $\hat{Y}_f = 85 + 20X_1 + 0.07X_2 + 0.01X_4 - 10X_5$
- $\hat{Y}_f = 85 + 20(4.0) + 0.07(110) + 0.1(4 * 110) - 10(4 * 1)$

```
(Y_f = 85 + 20*(4.0) + 0.07*(110) + 0.01*(4 * 110) - 10*(4 * 1))
```

```
## [1] 137.1
```

(c) True or false: Since the coefficient for the GPA/IQ interaction term is very small, there is very little evidence of an interaction effect. Justify your answer.

- **False**, because it also depends on the SE of the *beta* estimator

Problem 2

We perform stepwise, forward stepwise, and backward stepwise selection on a single data set. For each approach, we obtain $p + 1$ models, containing $0, 1, 2, \ldots, p$ predictors. True or False:

(a) The predictors in the k-variable model identified by forward stepwise are a subset of the predictors in the $(k + 1)$-variable model identified by forward stepwise selection.

- **True**, because in forward selection each predictors is added at each step. $k+1$) variables are considered as predictors for the model.

(b) The predictors in the k-variable model identified by backward stepwise are a subset of the predictors in the $(k + 1)$-variable model identified by backward stepwise selection.

- **True**, by testing if any variable in test effective then those variable will be affected. The model will remain with k variables in next step.

(c) The predictors in the k-variable model identified by backward stepwise are a subset of the predictors in the $(k + 1)$-variable model identified by forward stepwise selection.

- **False**, the predictors of model with $k+1$ variables from backward selection method are form by deleting each term after checking the less effective variable and deleting them.

(d) The predictors in the k-variable model identified by forward stepwise are a subset of the predictors in the $(k + 1)$-variable model identified by backward stepwise selection.

- **False**, the k variables in the predictor model by backward selection method and cannot be subset of $k + 1$ variables in the model by foward selection because they're different

(e) The predictors in the k-variable model identified by stepwise are a subset of the predictors in the $(k + 1)$-variable model identified by stepwise selection.

- **True**, the predictors in the k variable model identified by best subset are subset of the predictors in the $k + 1$ variable model.

Problem 3

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

(a) Use the `lm()` function to perform a simple linear regression with *mpg* as the response and *horsepower* (*hp*) as the predictor. Use the `summary()` function to print the results. Comment on the output. For example:

```
library(ISLR)
attach(Auto)
mpg_horsepower = lm(mpg ~ horsepower)
summary(mpg_horsepower)
```

```
##
## 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
```

- i. Is there a relationship between the predictor and the response?
  - Yes, there is a relationship between the predictor and the response, the $p - value = 2e - 16 = 0 < a = 0.05$
- ii. How strong is the relationship between the predictor and the response?

```
cor(mpg, horsepower)
```

```
## [1] -0.7784268
```

- iii. Is the relationship between the predictor and the response positive or negative?
  - The relationship between the predictor and the response is negative, because it have a negative correlation.
- iv. What is the predicted mpg associated with a horsepower of 98? What are the associated 95% confidence and prediction intervals? Give an interpretation of these intervals.
  - $Y = 39.935 - 0.1578X$
  - The predicted *mpg* associated with a *horsepower* of 98 is:

```r
(Y = 39.935 - 0.1578 * 98)
```

```
## [1] 24.4706
```

- The *predict interval* is:

```r
horsedata = data.frame(horsepower = 98)
#for predict
predict(mpg_horsepower, horsedata, interval = "predict")
```

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

- This means the predicted *mpg* for a $horsepower = 98$ is between $[14.80, 34.12]$ with a 95% confidence
- The *confidence interval* is:

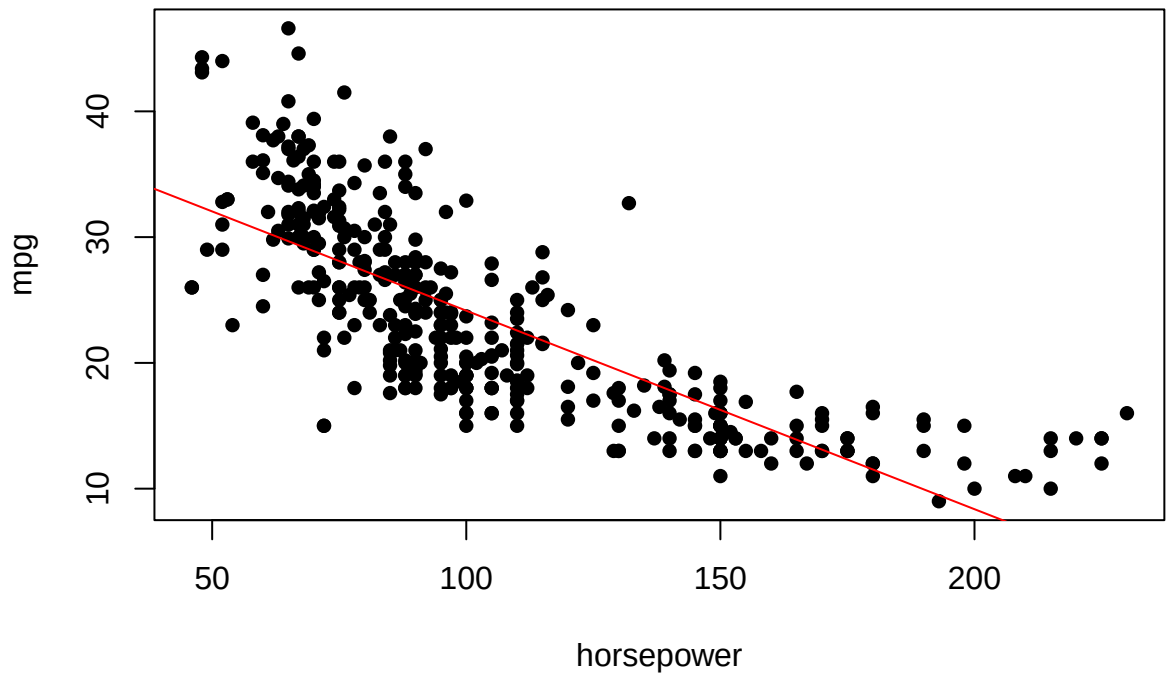
```r
#for confidence
predict(mpg_horsepower, horsedata, interval = "confidence")
```

```
##        fit      lwr      upr
## 1 24.46708 23.97308 24.96108
```

- This means we predict the **average** *mpg* with $horsepower = 98$ is between $[23.97, 24.96]$ with 95% confidence.

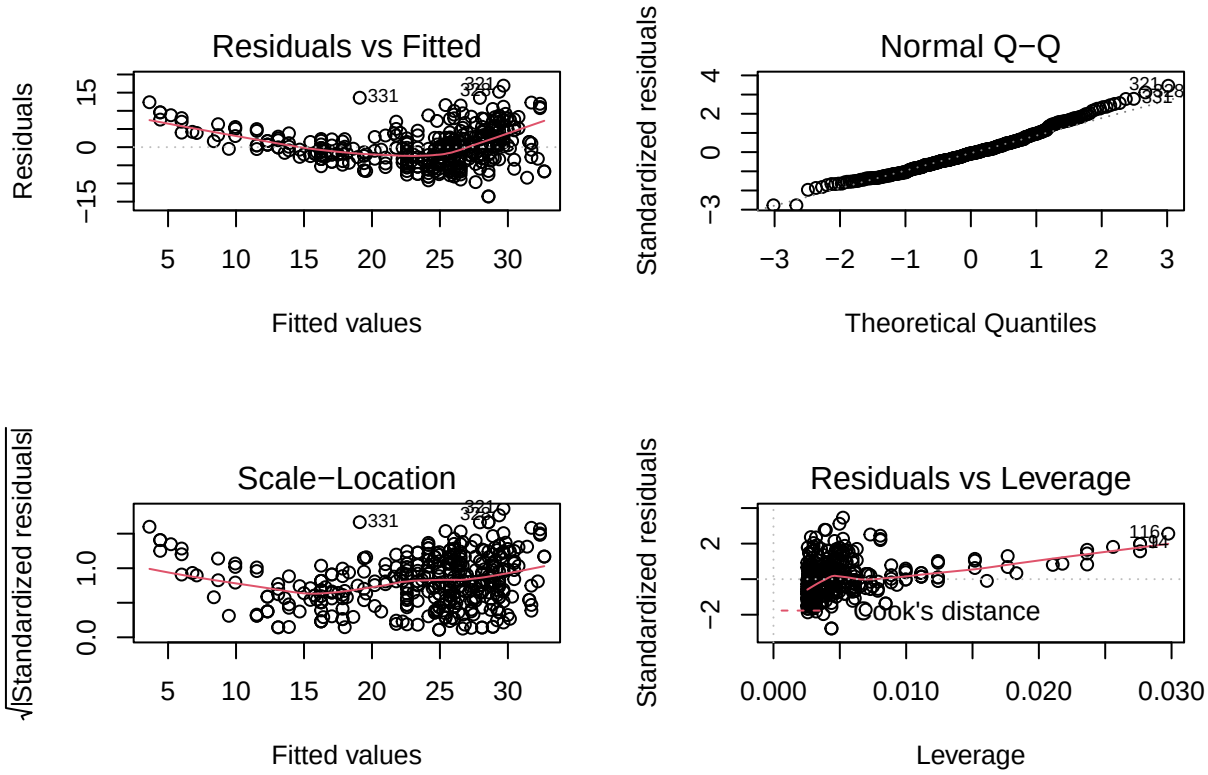
(b) Plot the response and the predictor. Use the `abline()` function to display the least squares regression line.

```r
plot(mpg~horsepower, pch=16)
abline(mpg_horsepower, col = "red")
```



(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(mpg_horsepower)
```



* These plot show some outliers observation numbers: 321, 331, 328
* High leverage: 116, 94
* It appears that the linearity fit is a little curvy

Problem 4

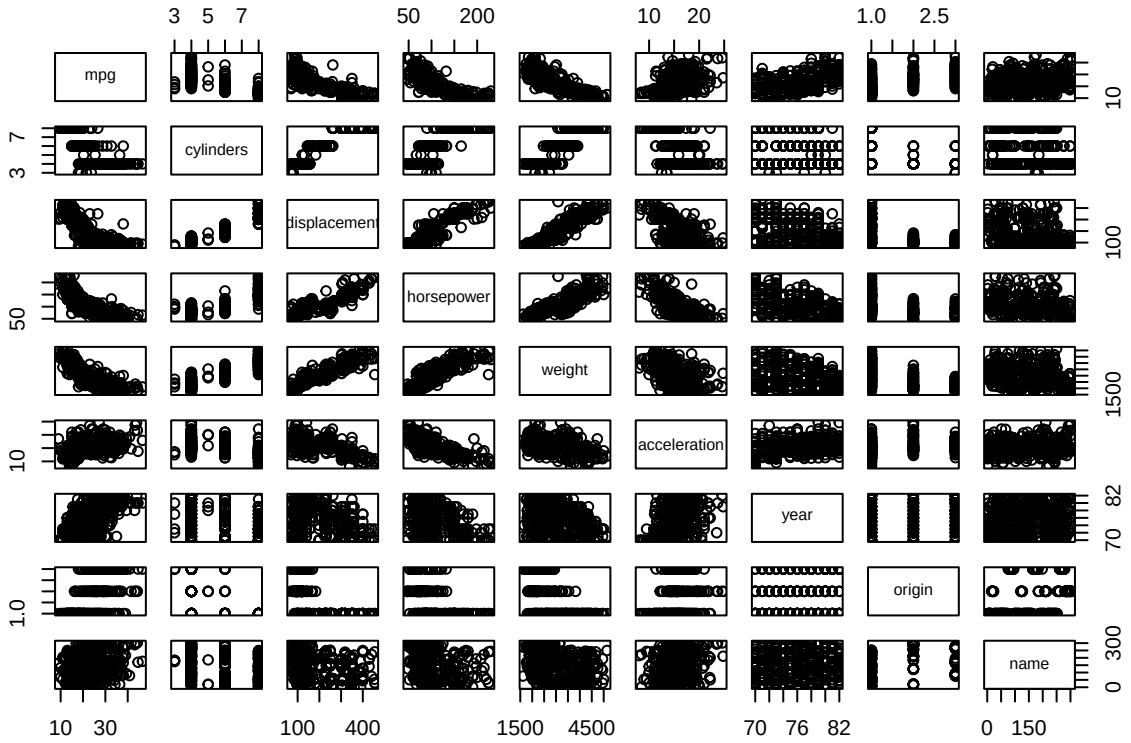
This question involves the use of multiple linear regression on the *Auto* data set.

(a) Produce a scatterplot matrix which includes all of the variables in the data set.

```
head(Auto)
```

```
##   mpg cylinders displacement horsepower weight acceleration year origin
## 1  18         8          307        130   3504         12.0   70      1
## 2  15         8          350        165   3693         11.5   70      1
## 3  18         8          318        150   3436         11.0   70      1
## 4  16         8          304        150   3433         12.0   70      1
## 5  17         8          302        140   3449         10.5   70      1
## 6  15         8          429        198   4341         10.0   70      1
##                       name
## 1 chevrolet chevelle malibu
## 2         buick skylark 320
## 3        plymouth satellite
## 4             amc rebel sst
## 5               ford torino
## 6          ford galaxie 500
```

```
plot(Auto)
```



(b) Compute the matrix of correlations between the variables using the function `cor()`. You will need to exclude the name variable, `cor()` which is qualitative.

```
?Auto
#9 variables
(correlation = cor(Auto[,-9]))
```

```
##                     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) Use the `lm()` function to perform a multiple linear regression with *mpg* as the response and all other variables except name as the predictors. Use the `summary()` function to print the results. Comment on the output. For instance:

```
auto.new = Auto[, -9] #8 variables
auto.new$origin = as.factor(auto.new$origin) #set to factor
auto.new$cylinders = as.factor(auto.new$cylinders)
auto.lm = lm(mpg~., data = auto.new)  #linear model
summary(auto.lm)
```

```
##
## Call:
## lm(formula = mpg ~ ., data = auto.new)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.6797 -1.9373 -0.0678  1.6711 12.7756
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -2.208e+01  4.541e+00  -4.862 1.70e-06 ***
## cylinders4    6.722e+00  1.654e+00   4.064 5.85e-05 ***
## cylinders5    7.078e+00  2.516e+00   2.813  0.00516 **
## cylinders6    3.351e+00  1.824e+00   1.837  0.06701 .
## cylinders8    5.099e+00  2.109e+00   2.418  0.01607 *
## displacement  1.870e-02  7.222e-03   2.590  0.00997 **
## horsepower   -3.490e-02  1.323e-02  -2.639  0.00866 **
## weight       -5.780e-03  6.315e-04  -9.154  < 2e-16 ***
## acceleration  2.598e-02  9.304e-02   0.279  0.78021
## year          7.370e-01  4.892e-02  15.064  < 2e-16 ***
## origin2       1.764e+00  5.513e-01   3.200  0.00149 **
## origin3       2.617e+00  5.272e-01   4.964 1.04e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.098 on 380 degrees of freedom
## Multiple R-squared:  0.8469, Adjusted R-squared:  0.8425
## F-statistic: 191.1 on 11 and 380 DF,  p-value: < 2.2e-16
```

i. Is there a relationship between the predictors and the response?

- $p - value = 2.2e - 16 < a = 0.05$, therefore there is a relationship between the predictors and the response.

ii. Which predictors appear to have a statistically significant relationship to the response?

- The predictors that appear to have a statistically significant relationship to the response is *cylinders4, cylinders5, cylinders6, cylinders8, displacement, horsepower, weight, year, origin2 and origin3$*, because their $p - value$ is significant.

iii. What does the coefficient for the year variable suggest?

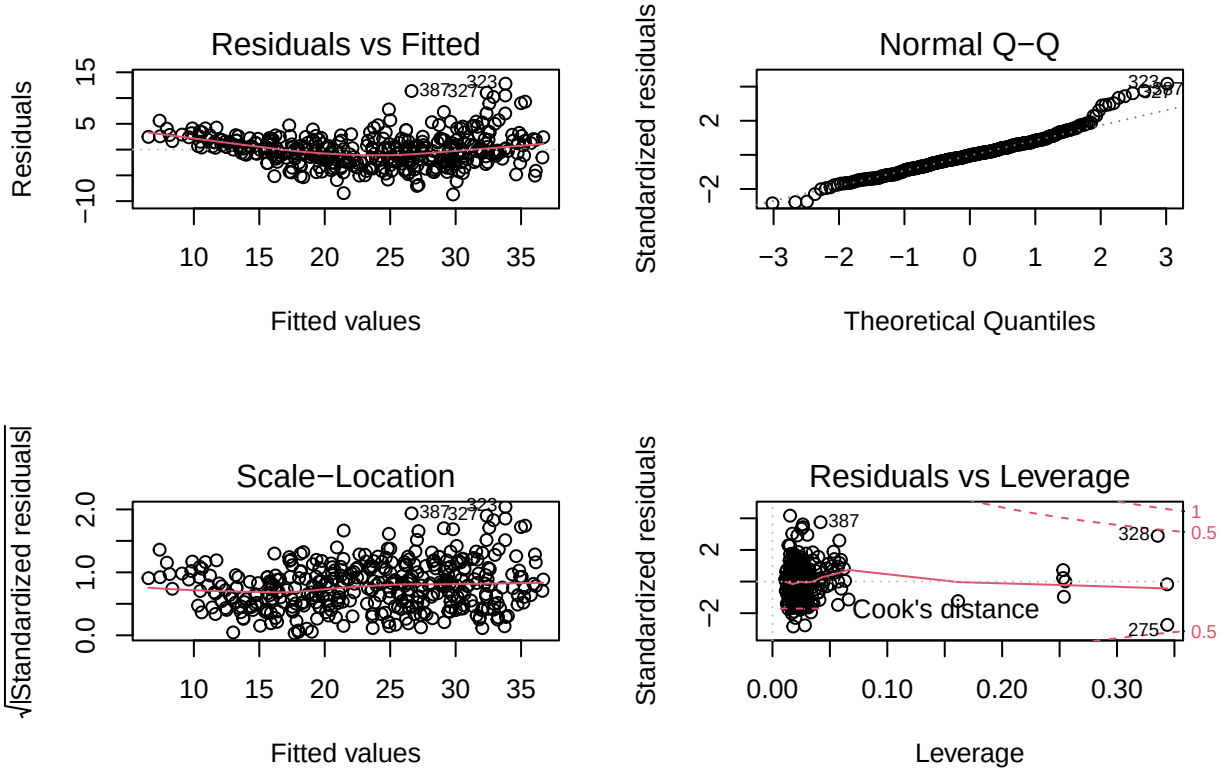
- For testing each one predictor separately , $H_0 : B_j = 0$ it appears that only *acceleration* does not have a statistically significant to *mpg*

(d) Use the `plot()` function to produce diagnostic plots of the linear regression fit based on the predictors that appear to have a statistically signifianct relationship to the response. Comment on any problems you see with the fit. Do the residual plots suggest any unusually large outliers? Does the leverage plot identify any observations with unusually high leverage?

```
auto.new2 = auto.new[, -6] #take out the 6th variable in auto.new
auto.lm2 = lm(mpg~., data = auto.new2) #linear model2
summary(auto.lm2)
```

```
##
## Call:
## lm(formula = mpg ~ ., data = auto.new2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.7037 -1.9501 -0.0552  1.7105 12.7932
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -2.162e+01  4.231e+00  -5.111 5.09e-07 ***
## cylinders4    6.784e+00  1.637e+00   4.144 4.20e-05 ***
## cylinders5    7.147e+00  2.501e+00   2.857 0.004510 **
## cylinders6    3.403e+00  1.813e+00   1.877 0.061262 .
## cylinders8    5.137e+00  2.102e+00   2.444 0.014983 *
## displacement  1.848e-02  7.169e-03   2.578 0.010312 *
## horsepower   -3.706e-02  1.071e-02  -3.459 0.000604 ***
## weight       -5.696e-03  5.535e-04 -10.291  < 2e-16 ***
## year          7.358e-01  4.868e-02  15.114  < 2e-16 ***
## origin2       1.763e+00  5.506e-01   3.203 0.001476 **
## origin3       2.621e+00  5.264e-01   4.979 9.71e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.094 on 381 degrees of freedom
## Multiple R-squared:  0.8469, Adjusted R-squared:  0.8429
## F-statistic: 210.7 on 10 and 381 DF,  p-value: < 2.2e-16
```

```
par(mfrow = c(2,2))
plot(auto.lm2)
```



* These plots show some outliers observation numbers: 387, 323, 327
* High leverage: 387, 328, 275
* It appears that the linearity fit is good.

(e) Use the * and/or : symbols to fit linear regression models with interaction effects. Do any interactions appear to be statistically significant?

```
#interactiion model between displacment: horsepower and horsepower: weight
auto.int = lm(mpg ~ cylinders + displacement * horsepower + horsepower*weight + year + origin, data = au
summary(auto.int)
```

```
##
## Call:
## lm(formula = mpg ~ cylinders + displacement * horsepower + horsepower *
##     weight + year + origin, data = auto.new2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.7565 -1.4899 -0.0843  1.4168 12.0178
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                -7.583e+00  4.316e+00  -1.757 0.079734 .
## cylinders4                  5.856e+00  1.516e+00   3.863 0.000132 ***
## cylinders5                  7.464e+00  2.297e+00   3.250 0.001259 **
## cylinders6                  5.197e+00  1.728e+00   3.008 0.002803 **
## cylinders8                  6.455e+00  2.042e+00   3.161 0.001700 **
## displacement               -2.243e-02  1.660e-02  -1.351 0.177530
```

```
## horsepower              -1.842e-01  2.162e-02  -8.521 3.79e-16 ***
## weight                  -7.717e-03  1.513e-03  -5.099 5.41e-07 ***
## year                     7.523e-01  4.523e-02  16.635  < 2e-16 ***
## origin2                  1.056e+00  5.251e-01   2.011 0.045084 *
## origin3                  1.695e+00  4.971e-01   3.411 0.000718 ***
## displacement:horsepower  1.968e-04  9.529e-05   2.066 0.039544 *
## horsepower:weight        2.768e-05  1.047e-05   2.644 0.008533 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.84 on 379 degrees of freedom
## Multiple R-squared:  0.8716, Adjusted R-squared:  0.8676
## F-statistic: 214.4 on 12 and 379 DF,  p-value: < 2.2e-16
```

- It appears that there might be interaction effects with horsepower and displacement also horsepower and weight. However, when we add these interaction terms, the displacement is no longer significant.

(f) Try a few different transformations of the variables, such as *log(X)*, $\sqrt{X}$, X^2. Comment on your findings.

```
auto.lm3 = lm(mpg ~ cylinders + displacement + sqrt(horsepower) + weight + origin, data = auto.new2)
summary(auto.lm3)
```

```
##
## Call:
## lm(formula = mpg ~ cylinders + displacement + sqrt(horsepower) +
##     weight + origin, data = auto.new2)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.994 -2.235 -0.542  1.758 15.765
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)      44.0684287  3.1692690  13.905  < 2e-16 ***
## cylinders4        7.8227761  2.0337518   3.846  0.00014 ***
## cylinders5        9.9647779  3.0964663   3.218  0.00140 **
## cylinders6        4.0868709  2.2409849   1.824  0.06898 .
## cylinders8        6.2616424  2.6039750   2.405  0.01666 *
## displacement      0.0063803  0.0085501   0.746  0.45599
## sqrt(horsepower) -1.7726717  0.2759663  -6.424 3.96e-10 ***
## weight           -0.0037309  0.0006861  -5.438 9.65e-08 ***
## origin2           0.0051860  0.6652473   0.008  0.99378
## origin3           2.6162364  0.6490513   4.031 6.71e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.845 on 382 degrees of freedom
## Multiple R-squared:  0.7629, Adjusted R-squared:  0.7573
## F-statistic: 136.6 on 9 and 382 DF,  p-value: < 2.2e-16
```

- The adjusted R-squared actually got lower when i $sqrt(horsepower)$

Problem 5

This problem focuses on the **collinearity** problem.

(a) Perform the following commands in R:

```
set.seed (1)
x1=runif (100)
x2 =0.5* x1+rnorm (100) /10
y=2+2* x1 +0.3* x2+rnorm (100)
```

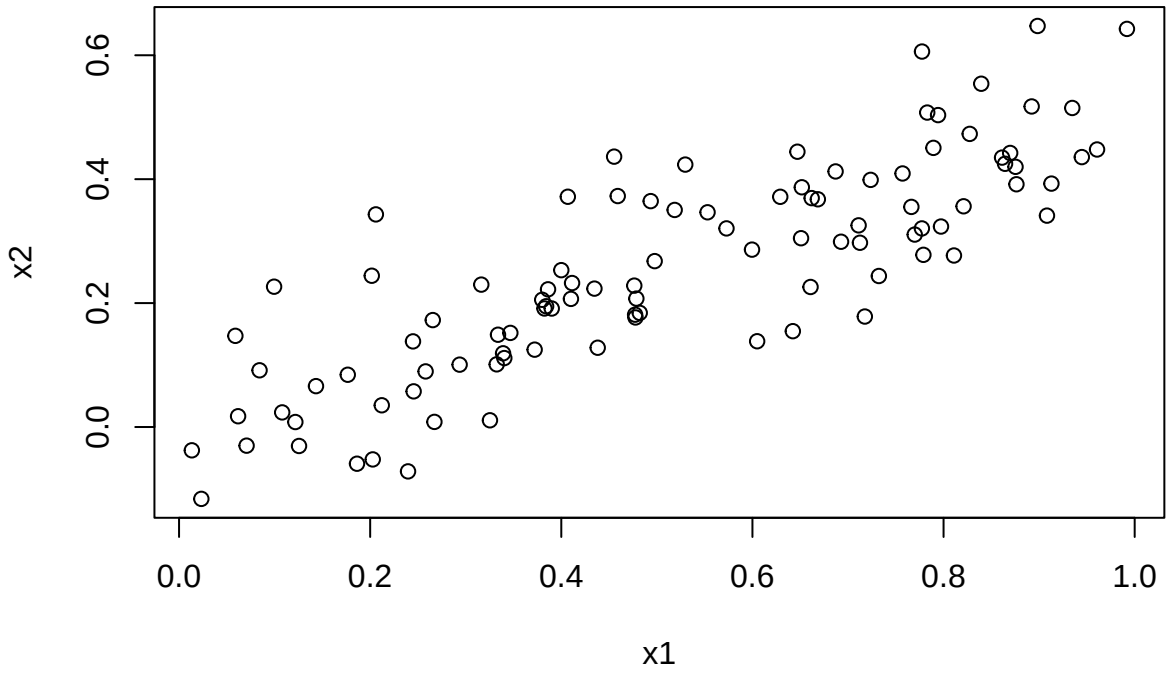
The last line corresponds to creating a linear model in which y is a function of x_1 and x_2. Write out the form of the linear model. What are the regression coefficients?

(b) What is the correlation between x_1 and x_2? Create a scatterplot displaying the relationship between the variables.

```
cor(x1, x2)
```

```
## [1] 0.8351212
```

```
plot(x1, x2)
```



(c) Using this data, fit a least squares regression to predict y using x_1 and x_2. Describe the results obtained. What are $\hat{\beta}_0$, $\hat{\beta}_1$, and $\hat{\beta}_2$? How do these relate to the true β_0, β_1, and β_2? Can you reject the null hypothesis $H_0 : \beta_1 = 0$? How about the null hypothesis $H_0 : \beta_2 = 0$?

```
model.fit = lm(y~x1 + x2)
summary(model.fit)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8311 -0.7273 -0.0537  0.6338  2.3359
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.1305     0.2319   9.188 7.61e-15 ***
## x1            1.4396     0.7212   1.996   0.0487 *
## x2            1.0097     1.1337   0.891   0.3754
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.056 on 97 degrees of freedom
## Multiple R-squared:  0.2088, Adjusted R-squared:  0.1925
## F-statistic:  12.8 on 2 and 97 DF,  p-value: 1.164e-05
```

- $\beta_0 = 2.1305$, $\beta_1 = 1.4396$, and $\beta_2 = 1.009$. We can RH_0 for β_1 because the $p - value < a$. We FRH_0 β_2 because the $p - value > a$.

(d) Now fit a least squares regression to predict y using only x_1. Comment on your results. Can you reject the null hypothesis $H_0 : \beta_1 = 0$?

```
model.x1 = lm(y~x1)
summary(model.x1)
```

```
##
## Call:
## lm(formula = y ~ x1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.89495 -0.66874 -0.07785  0.59221  2.45560
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.1124     0.2307   9.155 8.27e-15 ***
## x1            1.9759     0.3963   4.986 2.66e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.055 on 98 degrees of freedom
## Multiple R-squared:  0.2024, Adjusted R-squared:  0.1942
## F-statistic: 24.86 on 1 and 98 DF,  p-value: 2.661e-06
```

- We RH_0 because the $p - value$ is significantly small, almost close to 0.

(e) Now fit a least squares regression to predict y using only x_2. Comment on your results. Can you reject the null hypothesis $H_0 : \beta_1 = 0$?

```
model.x2 = lm(y~x2)
summary(model.x2)
```

```
##
## Call:
## lm(formula = y ~ x2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.62687 -0.75156 -0.03598  0.72383  2.44890
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.3899     0.1949   12.26  < 2e-16 ***
## x2            2.8996     0.6330    4.58 1.37e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.072 on 98 degrees of freedom
## Multiple R-squared:  0.1763, Adjusted R-squared:  0.1679
## F-statistic: 20.98 on 1 and 98 DF,  p-value: 1.366e-05
```

- We RH_0 because the $p - value$ is significantly small, almost close to 0.

(f) Do the results obtained in (c)–(e) contradict each other? Explain your answer.

- **Yes**, because we $FRH_0 : \beta_2$ in (c) and we find that we could RH_0 in (e) when we get the model using only x_2.

(g) Now suppose we obtain one additional observation, which was unfortunately mismeasured.

```
x1=c(x1 , 0.1)
x2=c(x2 , 0.8)
y=c(y,6)
```

Re-fit the linear models from (c) to (e) using this new data. What effect does this new observation have on the each of the models? In each model, is this observation an outlier? A high-leverage point? Both? Explain your answers.

```
new.model = lm(y ~ x1 + x2)
summary(new.model)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.73348 -0.69318 -0.05263  0.66385  2.30619
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.2267     0.2314   9.624 7.91e-16 ***
## x1            0.5394     0.5922   0.911  0.36458
## x2            2.5146     0.8977   2.801  0.00614 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.075 on 98 degrees of freedom
## Multiple R-squared:  0.2188, Adjusted R-squared:  0.2029
## F-statistic: 13.72 on 2 and 98 DF,  p-value: 5.564e-06
```

- For this model with $x1$ and $x2$, we can RH_0 because the $p - value < a$.

Problem 6

In this exercise, we will generate simulated data, and will then use this data to perform best subset selection.

(a) Use the `rnorm()` function to generate a predictor X of length $n = 100$, as well as a noise vector ϵ of length $n = 100$.

```
set.seed(1) #get the same result
predictorx = rnorm(100)
noise_vector = rnorm(100)
```

(b) Generate a response vector Y of length n = 100 according to the model

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \epsilon,$$

where β_0, β_1, β_2, and β_3 are constants of your choice.

```
beta0 = 2
beta1 = 3
beta2 = 4
beta3 = 5
Y = 2 + 3*predictorx + 4*predictorx^2 + 5*predictorx^3 + noise_vector
```

$Y = 2 + 3 * predictorx^{+}4 * predictorx^2 + 5x_3 * predictorx^3 + \epsilon$

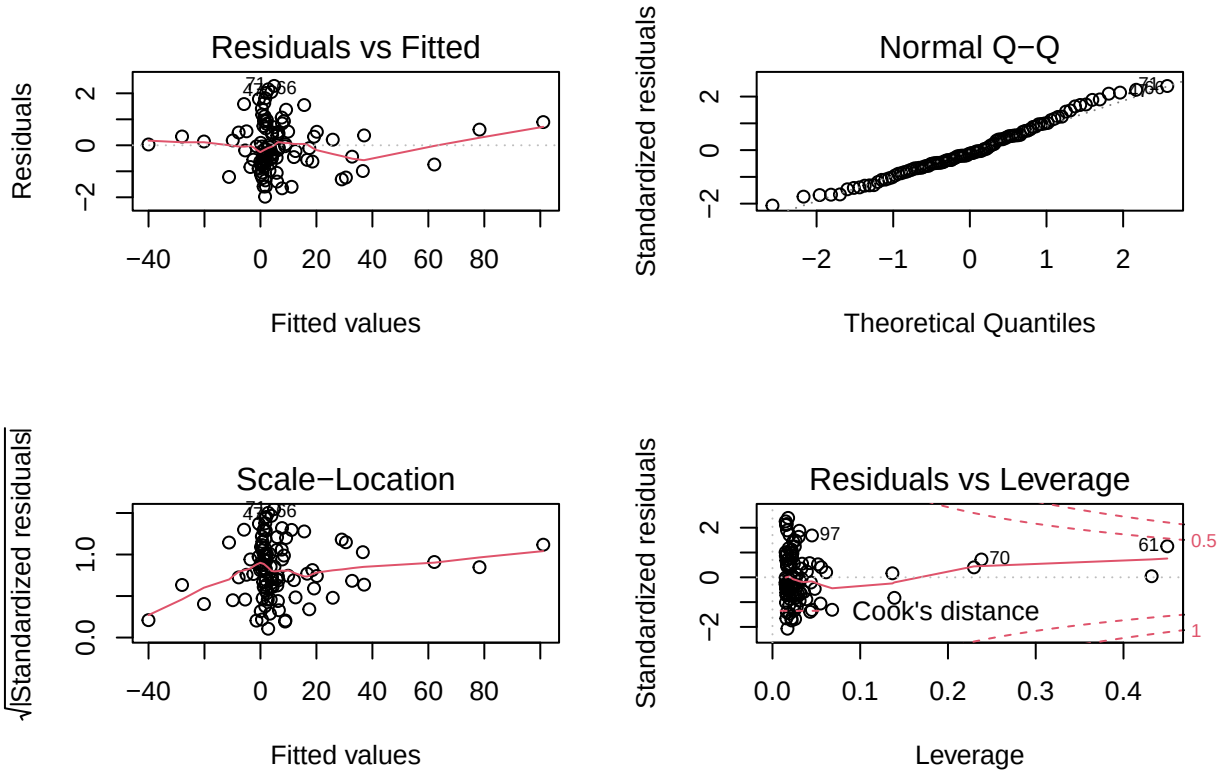
(c) Use the `regsubsets()` function to perform best subset selection in order to choose the best model containing the predictors $X, X^2, \ldots, X^{10}$. What is the best model obtained according to C_p, BIC, and adjusted R^2? Show some plots to provide evidence for your answer, and report the coefficients of the best model obtained. Note you will need to use the `data.frame()` function to create a single data set containing both X and Y.

```
library(leaps)
dataset = data.frame(cbind(Y, predictorx))
model.reg = regsubsets(Y~poly(predictorx, 10), data = dataset)
model.sum = summary(model.reg)
model.stat = cbind(model.sum$adjr2,model.sum$cp, model.sum$bic)
colnames(model.stat) = c("Adjr2", "Cp", "BIC")
print(model.stat)
```

```
##          Adjr2          Cp       BIC
## [1,] 0.6764762 9718.345972 -104.6532
## [2,] 0.8721720 3744.191499 -193.9324
## [3,] 0.9968305    2.185943 -560.0758
## [4,] 0.9968761    1.866261 -557.9643
## [5,] 0.9969003    2.193128 -555.1972
## [6,] 0.9969003    3.235128 -551.6599
## [7,] 0.9968706    5.119994 -547.1838
## [8,] 0.9968395    7.027330 -542.6827
```

- The best model is B_3 because it have the lowest Cp and BIC

```
par(mfrow = c(2,2))
plot(lm(Y~poly(predictorx, 3)))
```



(d) Repeat (c), using forward stepwise selection and also using backwards stepwise selection. How does your answer compare to the results in (c)?

```
step(lm(Y~poly(predictorx, 10)), direction = "backward")
```

```
## Start:  AIC=4.64
## Y ~ poly(predictorx, 10)
##
##                         Df Sum of Sq     RSS    AIC
## <none>                                  84.1   4.64
## - poly(predictorx, 10) 10     28861 28944.7 568.80
```

```
##
## Call:
## lm(formula = Y ~ poly(predictorx, 10))
##
## Coefficients:
##            (Intercept)   poly(predictorx, 10)1   poly(predictorx, 10)2
##                 6.5842                140.2677                 59.4663
##  poly(predictorx, 10)3   poly(predictorx, 10)4   poly(predictorx, 10)5
##                75.1300                 1.2571                  1.4802
##  poly(predictorx, 10)6   poly(predictorx, 10)7   poly(predictorx, 10)8
##                 0.1190                -0.3298                 -0.1079
##  poly(predictorx, 10)9  poly(predictorx, 10)10
##                -0.2958                -0.9512
```

```
step(lm(Y~poly(predictorx, 10)), direction = "forward")
```

```
## Start:  AIC=4.64
## Y ~ poly(predictorx, 10)
```

```
##
## Call:
## lm(formula = Y ~ poly(predictorx, 10))
##
## Coefficients:
##            (Intercept)   poly(predictorx, 10)1   poly(predictorx, 10)2
##                 6.5842                140.2677                 59.4663
##  poly(predictorx, 10)3   poly(predictorx, 10)4   poly(predictorx, 10)5
##                75.1300                 1.2571                  1.4802
##  poly(predictorx, 10)6   poly(predictorx, 10)7   poly(predictorx, 10)8
##                 0.1190                -0.3298                 -0.1079
##  poly(predictorx, 10)9  poly(predictorx, 10)10
##                -0.2958                -0.9512
```