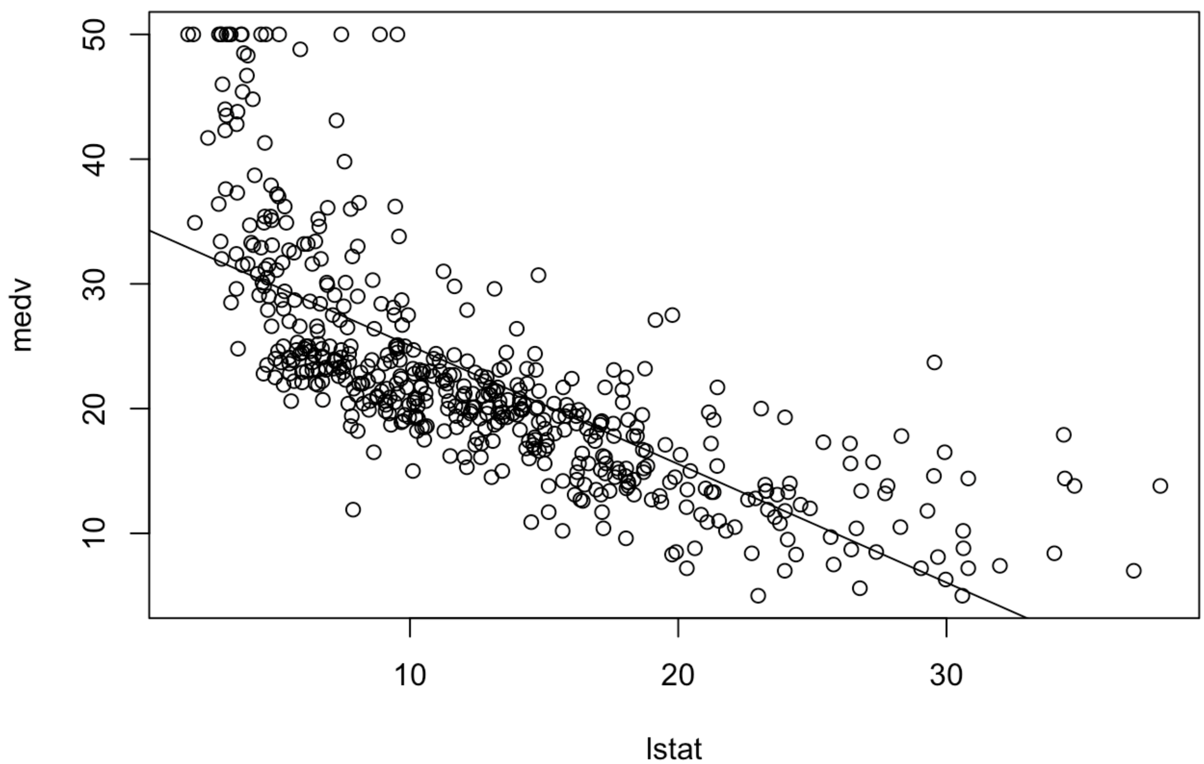
**Jeffrey Kerley**
**Jakcqc**
**2/16/2022**

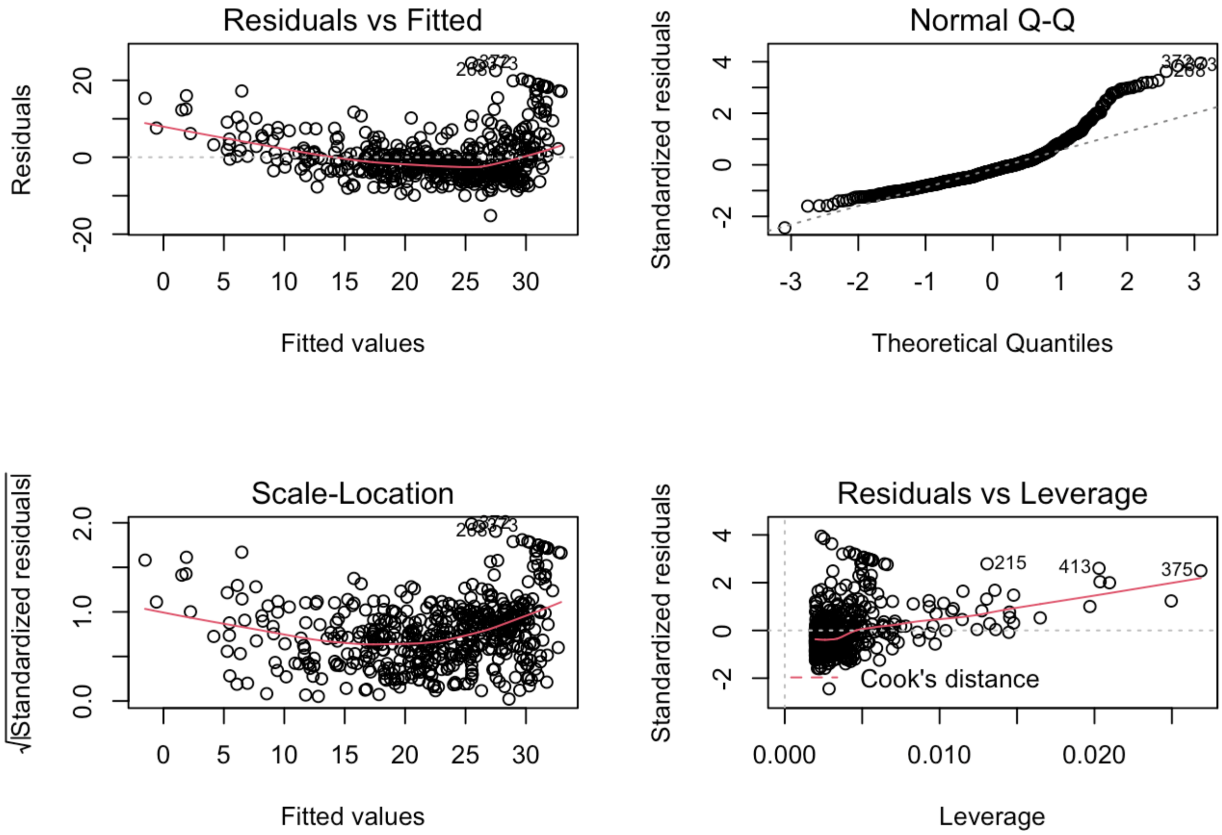
**Problem 1:**

```
> head(Boston)
     crim zn indus chas   nox    rm  age    dis rad tax ptratio lstat medv
1 0.00632 18  2.31    0 0.538 6.575 65.2 4.0900   1 296    15.3  4.98 24.0
2 0.02731  0  7.07    0 0.469 6.421 78.9 4.9671   2 242    17.8  9.14 21.6
3 0.02729  0  7.07    0 0.469 7.185 61.1 4.9671   2 242    17.8  4.03 34.7
4 0.03237  0  2.18    0 0.458 6.998 45.8 6.0622   3 222    18.7  2.94 33.4
5 0.06905  0  2.18    0 0.458 7.147 54.2 6.0622   3 222    18.7  5.33 36.2
6 0.02985  0  2.18    0 0.458 6.430 58.7 6.0622   3 222    18.7  5.21 28.7
> lm.fit <- lm(medv ~ lstat , data = Boston)
> attach (Boston)
The following objects are masked from Boston (pos = 3):

    age, chas, crim, dis, indus, lstat, medv, nox, ptratio, rad, rm, tax, zn

The following objects are masked from Boston (pos = 4):

    age, chas, crim, dis, indus, lstat, medv, nox, ptratio, rad, rm, tax, zn

The following objects are masked from Boston (pos = 5):

    age, chas, crim, dis, indus, lstat, medv, nox, ptratio, rad, rm, tax, zn

> lm.fit <- lm(medv ~ lstat)
> summary (lm.fit)

Call:
lm(formula = medv ~ lstat)

Residuals:
    Min      1Q  Median      3Q     Max
-15.168  -3.990  -1.318   2.034  24.500

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 34.55384    0.56263   61.41   <2e-16 ***
lstat       -0.95005    0.03873  -24.53   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
Residual standard error: 6.216 on 504 degrees of freedom
Multiple R-squared:  0.5441,  Adjusted R-squared:  0.5432
F-statistic: 601.6 on 1 and 504 DF,  p-value: < 2.2e-16

> names (lm.fit)
 [1] "coefficients"  "residuals"     "effects"       "rank"          "fitted.values"
 [6] "assign"        "qr"            "df.residual"   "xlevels"       "call"
[11] "terms"         "model"
> confint (lm.fit)
               2.5 %     97.5 %
(Intercept) 33.448457 35.6592247
lstat       -1.026148 -0.8739505
> predict (lm.fit , data.frame(lstat = (c(5, 10, 15))), interval = "confidence")
       fit      lwr      upr
1 29.80359 29.00741 30.59978
2 25.05335 24.47413 25.63256
3 20.30310 19.73159 20.87461
> predict (lm.fit , data.frame(lstat = (c(5, 10, 15))), interval = "prediction")
       fit       lwr      upr
1 29.80359 17.565675 42.04151
2 25.05335 12.827626 37.27907
3 20.30310  8.077742 32.52846
> plot (lstat , medv)
> abline (lm.fit)

> abline (lm.fit , lwd = 3)
> abline (lm.fit , lwd = 3, col = " red ")
> plot (lstat , medv , col = " red ")
> plot (lstat , medv , pch = 20)
> plot (lstat , medv , pch = "+")
> plot (1:20, 1:20, pch = 1:20)
> par (mfrow = c(2, 2))
```

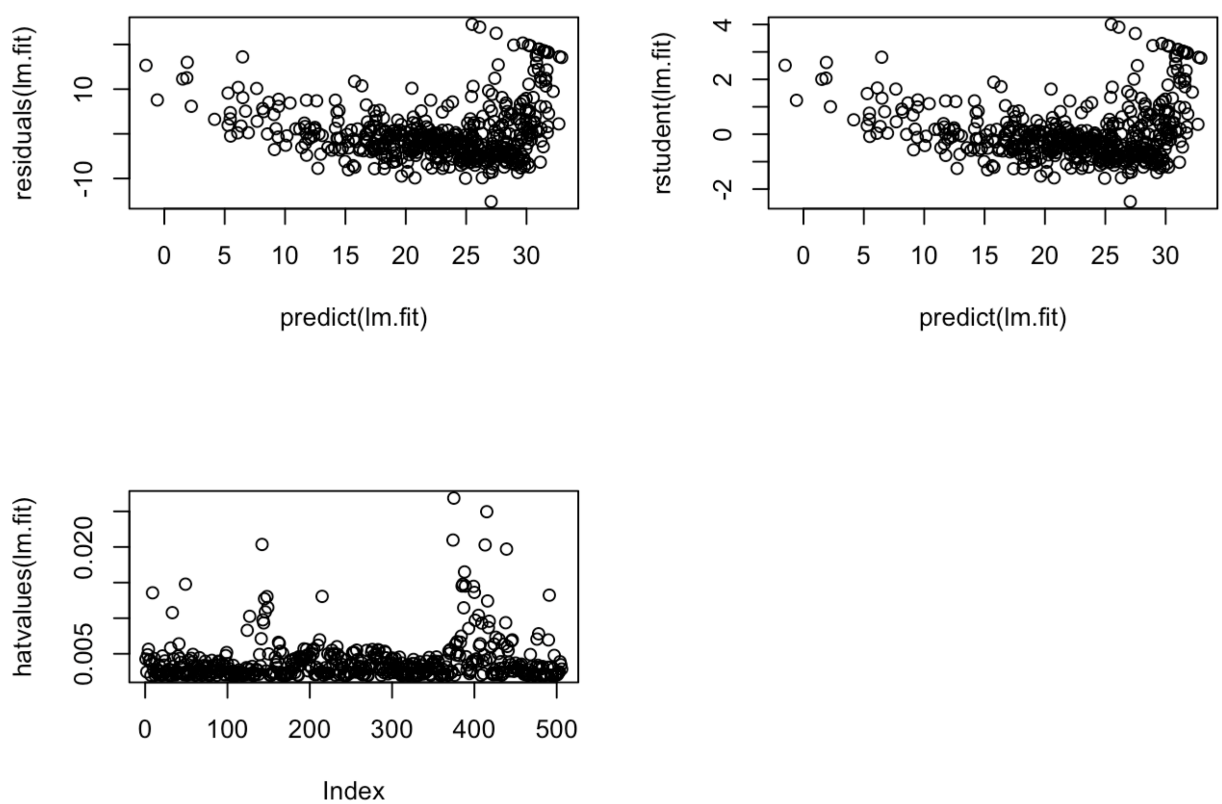



> plot (lm.fit)




```
> plot ( predict (lm.fit), residuals (lm.fit))
> plot ( predict (lm.fit), rstudent (lm.fit))
> plot ( hatvalues (lm.fit))
> which.max ( hatvalues (lm.fit))
375
```

```
> lm.fit <- lm(medv ~ lstat + age , data = Boston)
> summary (lm.fit)

Call:
lm(formula = medv ~ lstat + age, data = Boston)

Residuals:
```

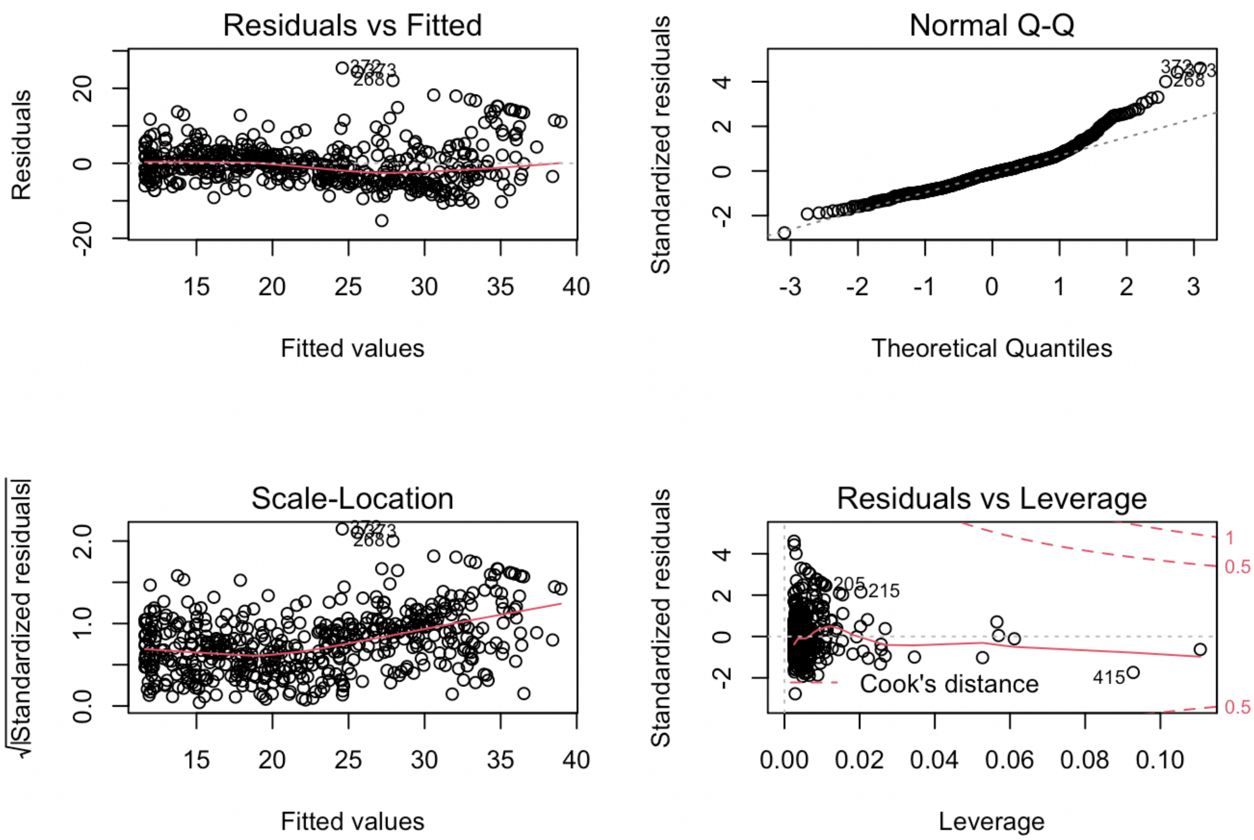



```
    Min      1Q  Median      3Q     Max 
-15.981  -3.978  -1.283   1.968  23.158 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 33.22276    0.73085  45.458  < 2e-16 ***
lstat       -1.03207    0.04819 -21.416  < 2e-16 ***
age          0.03454    0.01223   2.826  0.00491 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.173 on 503 degrees of freedom
```

```
Multiple R-squared:  0.5513,  Adjusted R-squared:  0.5495
F-statistic:   309 on 2 and 503 DF,  p-value: < 2.2e-16

> lm.fit <- lm(medv ~ ., data = Boston)
> summary (lm.fit)

Call:
lm(formula = medv ~ ., data = Boston)

Residuals:
     Min       1Q   Median       3Q      Max
-15.1304  -2.7673  -0.5814   1.9414  26.2526

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  41.617270   4.936039   8.431 3.79e-16 ***
crim         -0.121389   0.033000  -3.678 0.000261 ***
zn            0.046963   0.013879   3.384 0.000772 ***
indus         0.013468   0.062145   0.217 0.828520
chas          2.839993   0.870007   3.264 0.001173 **
nox         -18.758022   3.851355  -4.870 1.50e-06 ***
rm            3.658119   0.420246   8.705  < 2e-16 ***
age           0.003611   0.013329   0.271 0.786595
dis          -1.490754   0.201623  -7.394 6.17e-13 ***
rad           0.289405   0.066908   4.325 1.84e-05 ***
tax          -0.012682   0.003801  -3.337 0.000912 ***
ptratio      -0.937533   0.132206  -7.091 4.63e-12 ***
lstat        -0.552019   0.050659 -10.897  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.798 on 493 degrees of freedom
Multiple R-squared:  0.7343,  Adjusted R-squared:  0.7278
F-statistic: 113.5 on 12 and 493 DF,  p-value: < 2.2e-16

> library(car)
Loading required package: carData
> vif(lm.fit)
    crim       zn    indus     chas      nox       rm      age      dis      rad      tax
1.767486 2.298459 3.987181 1.071168 4.369093 1.912532 3.088232 3.954037 7.445301
9.002158
 ptratio    lstat
1.797060 2.870777
> lm.fit1 <- lm(medv ~ . - age , data = Boston)
```

```
> summary (lm.fit1)

Call:
lm(formula = medv ~ . - age, data = Boston)

Residuals:
     Min       1Q   Median       3Q      Max 
-15.1851  -2.7330  -0.6116   1.8555  26.3838 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  41.525128   4.919684   8.441 3.52e-16 ***
crim         -0.121426   0.032969  -3.683 0.000256 ***
zn            0.046512   0.013766   3.379 0.000785 ***
indus         0.013451   0.062086   0.217 0.828577    
chas          2.852773   0.867912   3.287 0.001085 ** 
nox         -18.485070   3.713714  -4.978 8.91e-07 ***
rm            3.681070   0.411230   8.951  < 2e-16 ***
dis          -1.506777   0.192570  -7.825 3.12e-14 ***
rad           0.287940   0.066627   4.322 1.87e-05 ***
tax          -0.012653   0.003796  -3.333 0.000923 ***
ptratio      -0.934649   0.131653  -7.099 4.39e-12 ***
lstat        -0.547409   0.047669 -11.483  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.794 on 494 degrees of freedom
Multiple R-squared:  0.7343,  Adjusted R-squared:  0.7284 
F-statistic: 124.1 on 11 and 494 DF,  p-value: < 2.2e-16

> lm.fit1 <- update (lm.fit , ~ . - age)
> summary (lm(medv ~ lstat * age , data = Boston))

Call:
lm(formula = medv ~ lstat * age, data = Boston)

Residuals:
    Min      1Q  Median      3Q     Max 
-15.806  -4.045  -1.333   2.085  27.552 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) 36.0885359  1.4698355  24.553  < 2e-16 ***
lstat       -1.3921168  0.1674555  -8.313 8.78e-16 ***
```

```
age         -0.0007209  0.0198792  -0.036   0.9711
lstat:age    0.0041560  0.0018518   2.244   0.0252 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.149 on 502 degrees of freedom
Multiple R-squared:  0.5557,  Adjusted R-squared:  0.5531
F-statistic: 209.3 on 3 and 502 DF,  p-value: < 2.2e-16

> lm.fit2 <- lm(medv ~ lstat + I(lstat^2))
> summary (lm.fit2)

Call:
lm(formula = medv ~ lstat + I(lstat^2))

Residuals:
     Min       1Q   Median       3Q      Max
-15.2834  -3.8313  -0.5295   2.3095  25.4148

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 42.862007   0.872084   49.15   <2e-16 ***
lstat       -2.332821   0.123803  -18.84   <2e-16 ***
I(lstat^2)   0.043547   0.003745   11.63   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.524 on 503 degrees of freedom
Multiple R-squared:  0.6407,  Adjusted R-squared:  0.6393
F-statistic: 448.5 on 2 and 503 DF,  p-value: < 2.2e-16

> lm.fit <- lm(medv ~ lstat)
> anova (lm.fit , lm.fit2)
Analysis of Variance Table

Model 1: medv ~ lstat
Model 2: medv ~ lstat + I(lstat^2)
  Res.Df   RSS Df Sum of Sq     F    Pr(>F)
1    504 19472
2    503 15347  1    4125.1 135.2 < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> par (mfrow = c(2, 2))
> plot (lm.fit2)
```




```
> lm.fit5 <- lm(medv ~ poly (lstat , 5))
> summary (lm.fit5)

Call:
lm(formula = medv ~ poly(lstat, 5))

Residuals:
     Min       1Q   Median       3Q      Max
-13.5433  -3.1039  -0.7052   2.0844  27.1153

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)       22.5328     0.2318  97.197  < 2e-16 ***
poly(lstat, 5)1 -152.4595     5.2148 -29.236  < 2e-16 ***
poly(lstat, 5)2   64.2272     5.2148  12.316  < 2e-16 ***
poly(lstat, 5)3  -27.0511     5.2148  -5.187 3.10e-07 ***
poly(lstat, 5)4   25.4517     5.2148   4.881 1.42e-06 ***
poly(lstat, 5)5  -19.2524     5.2148  -3.692 0.000247 ***
---
```

```
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.215 on 500 degrees of freedom
Multiple R-squared:  0.6817,  Adjusted R-squared:  0.6785
F-statistic: 214.2 on 5 and 500 DF,  p-value: < 2.2e-16

> summary (lm(medv ~ log(rm), data = Boston))

Call:
lm(formula = medv ~ log(rm), data = Boston)

Residuals:
    Min      1Q  Median      3Q     Max
-19.487  -2.875  -0.104   2.837  39.816

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -76.488      5.028  -15.21   <2e-16 ***
log(rm)       54.055      2.739   19.73   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.915 on 504 degrees of freedom
Multiple R-squared:  0.4358,  Adjusted R-squared:  0.4347
F-statistic: 389.3 on 1 and 504 DF,  p-value: < 2.2e-16

> head (Carseats)
  Sales CompPrice Income Advertising Population Price ShelveLoc Age Education Urban  US
1  9.50       138     73          11        276   120       Bad  42        17   Yes Yes
2 11.22       111     48          16        260    83      Good  65        10   Yes Yes
3 10.06       113     35          10        269    80    Medium  59        12   Yes Yes
4  7.40       117    100           4        466    97    Medium  55        14   Yes Yes
5  4.15       141     64           3        340   128       Bad  38        13   Yes  No
6 10.81       124    113          13        501    72       Bad  78        16    No Yes
> lm.fit <- lm(Sales ~ . + Income:Advertising + Price:Age , data = Carseats)
> summary (lm.fit)

Call:
lm(formula = Sales ~ . + Income:Advertising + Price:Age, data = Carseats)

Residuals:
    Min      1Q  Median      3Q     Max
-2.9208 -0.7503  0.0177  0.6754  3.3413
```

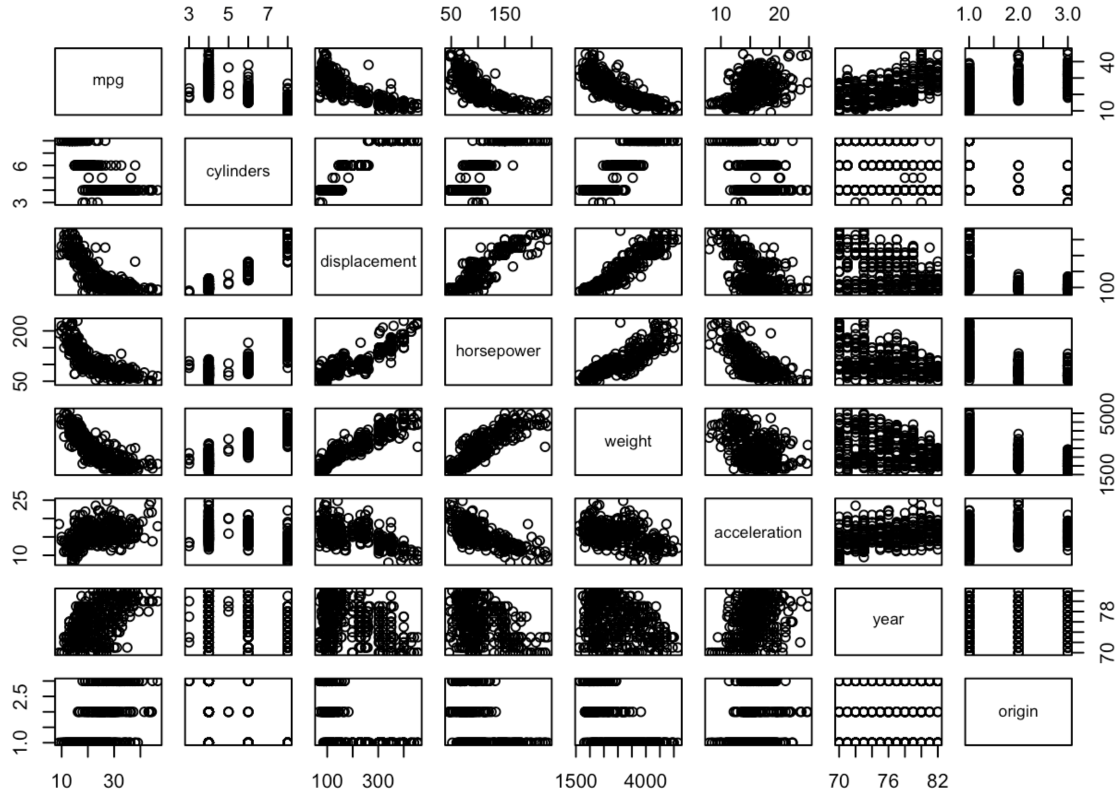
```
Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)         6.5755654  1.0087470   6.519 2.22e-10 ***
CompPrice           0.0929371  0.0041183  22.567  < 2e-16 ***
Income              0.0108940  0.0026044   4.183 3.57e-05 ***
Advertising         0.0702462  0.0226091   3.107 0.002030 **
Population          0.0001592  0.0003679   0.433 0.665330
Price              -0.1008064  0.0074399 -13.549  < 2e-16 ***
ShelveLocGood       4.8486762  0.1528378  31.724  < 2e-16 ***
ShelveLocMedium     1.9532620  0.1257682  15.531  < 2e-16 ***
Age                -0.0579466  0.0159506  -3.633 0.000318 ***
Education          -0.0208525  0.0196131  -1.063 0.288361
UrbanYes            0.1401597  0.1124019   1.247 0.213171
USYes              -0.1575571  0.1489234  -1.058 0.290729
Income:Advertising  0.0007510  0.0002784   2.698 0.007290 **
Price:Age           0.0001068  0.0001333   0.801 0.423812
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.011 on 386 degrees of freedom
Multiple R-squared:  0.8761,  Adjusted R-squared:  0.8719
F-statistic:   210 on 13 and 386 DF,  p-value: < 2.2e-16

> attach (Carseats)
> contrasts (ShelveLoc)
       Good Medium
Bad       0      0
Good      1      0
Medium    0      1
```

**Problem 2:**

**<u>a)</u>**

```
> Auto <- read.csv("Auto.csv", na.strings ="?") # With the option, R recognizes ? as NA.
> Auto <- na.omit(Auto) # Remove data rows including NA.
> Auto$origin <- as.factor(Auto$origin) # Coerce the type of origin into factor
> pairs(Auto[,1:8])
```

**<u>b)</u> Acceleration and Year predictor values, both appear to have an association with the response var MPG**




**c)**

1) **If we look at the correlation between all entries, the confirmation of association between mpg : year, and mpg : acceleration can be seen with positive values that are highlighted below. Furthermore, when we see a plot with a positive trend, we can track the response and predictor values associated, and see in fact, that the correlation is a positive one. Weight : horsepower, weight : displacement, follow this trend as well.**
2) **The potential collinearity problems in this data are as follows: there are several variables in the data that are too closely correlated, as underlined below. It might be difficult to find a good model/fit for the data as a result of these closely correlated variables, especially to find statistically significant variables.**

```
> count <- 1
> countTwo <- 2
> while(count > 0){
+   while(countTwo > 0){
+
+     print(cor(Auto[count], Auto[countTwo]))
+     countTwo = countTwo + 1
+     if(countTwo > 7)
```

```
+     {
+       countTwo = 0
+     }
+   }
+   count = count + 1
+   countTwo = count + 1
+   if(count > 6)
+   {
+     count = 0
+   }
+ }
```

     cylinders
mpg -0.7776175
    displacement
mpg   -0.8051269
    horsepower
mpg -0.7784268
        weight
mpg -0.8322442
    **acceleration**
**mpg    0.4233285**
        **year**
**mpg 0.580541**
          displacement
cylinders    0.9508233
          horsepower
cylinders  0.8429834
             weight
cylinders 0.8975273
          acceleration
cylinders   -0.5046834
                year
cylinders -0.3456474
             horsepower
displacement   0.897257
                weight
displacement 0.9329944
             acceleration
displacement   -0.5438005
                   year
displacement -0.3698552
              weight
horsepower 0.8645377
           acceleration

horsepower   -0.6891955
year
horsepower -0.4163615
acceleration
weight   -0.4168392
year
weight -0.3091199
year
acceleration 0.2903161

**d)**

1. **Yes there is a relationship between the predictors and response.**
2. **Displacement, year, origin (namely origin 2 and 3).**
3. **The coefficient for year, shows that there this variable has a strong influence/bias on the data.**

```
> lm.fitMPG <- lm(Auto$mpg ~ . - name, data = Auto)
> plot(lm.fitMPG)
> summary(lm.fitMPG)

Call:
lm(formula = Auto$mpg ~ . - name, data = Auto)

Residuals:
    Min      1Q  Median      3Q     Max
-9.0095 -2.0785 -0.0982  1.9856 13.3608

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)  -1.795e+01  4.677e+00  -3.839 0.000145 ***
cylinders    -4.897e-01  3.212e-01  -1.524 0.128215
displacement  2.398e-02  7.653e-03   3.133 0.001863 **
horsepower   -1.818e-02  1.371e-02  -1.326 0.185488
weight       -6.710e-03  6.551e-04 -10.243  < 2e-16 ***
acceleration  7.910e-02  9.822e-02   0.805 0.421101
year          7.770e-01  5.178e-02  15.005  < 2e-16 ***
origin2       2.630e+00  5.664e-01   4.643 4.72e-06 ***
origin3       2.853e+00  5.527e-01   5.162 3.93e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.307 on 383 degrees of freedom
Multiple R-squared:  0.8242,  Adjusted R-squared:  0.8205
F-statistic: 224.5 on 8 and 383 DF,  p-value: < 2.2e-16
```

**e)**
**If we look at the contrasts for origin, we see a 0,1 encoding, which explains why there is some level of predictiveness in origin2 and origin3, but not for origin1.**

```
> Auto$origin
  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1 1 1 3 2 2 2 2 2 1 1 1 1 1 3 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
2 2
 [52] 2 3 3 2 1 3 1 2 1 1 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 2 2 2 2 1 3 3 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 2
[103] 1 1 1 1 1 3 1 3 3 1 1 2 1 1 2 2 2 2 1 2 3 1 1 1 1 3 1 3 1 1 1 1 1 1 1 1 1 2 2 2 3 3 1 2 2 3 3 2
1 1 1
[154] 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1 1 1 3 2 3 1 2 1 2 2 2 2 3 2 2 1 1 2 1 1 1 1 1 1 1 1 1 1 1 2 3 1 1 1 1
2 3 3
[205] 1 2 1 2 3 2 1 1 1 1 3 1 2 1 3 1 1 1 1 1 1 1 1 1 1 1 1 2 1 3 1 1 1 3 2 3 2 3 2 1 3 3 3 1 1 1 1 1
1 1 1
[256] 1 1 1 1 1 1 1 1 1 1 3 3 1 3 1 1 3 2 2 2 2 2 3 1 1 1 1 1 1 1 1 1 1 1 1 1 2 3 1 1 2 1 2 1 1 1 3 2
1 1 1
[307] 1 2 3 1 3 1 1 1 1 2 3 3 3 3 3 1 3 2 2 2 2 3 3 2 3 3 2 3 1 1 1 1 1 3 1 3 3 3 3 3 1 1 1 2 3 3 3 3
2 2 3
[358] 3 1 1 1 1 1 1 1 1 1 1 1 1 2 3 3 1 1 3 3 3 3 3 3 1 1 1 1 3 1 1 1 2 1 1 1
Levels: 1 2 3
> contrasts(t(Auto$origin))
  2 3
1 0 0
2 1 0
3 0 1
```

**f)**
**A different result is found between mpg and cylinder in the single regression. A P value that is < 0.0001 is found in the single regression, where as in the multi it was P > 0.0001. This inconsistiny can arise from having a lack of information in fitting the model proper. Just like the student credit card default problem from class, here the cylinders had a “false” positive, but a true understanding was understood when adding more information and fitting a multi regression.**

```
> lm.fitMPGC  <- lm(Auto$mp ~ Auto$cylinders, data = Auto)
> plot(lm.fitMPGC)
> summary(lm.fitMPGC)

Call:
lm(formula = Auto$mp ~ Auto$cylinders, data = Auto)

Residuals:
    Min       1Q   Median       3Q      Max
```

```
-14.2413  -3.1832  -0.6332   2.5491  17.9168

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)     42.9155     0.8349   51.40   <2e-16 ***
Auto$cylinders  -3.5581     0.1457  -24.43   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.914 on 390 degrees of freedom
Multiple R-squared:  0.6047,	Adjusted R-squared:  0.6037
F-statistic: 596.6 on 1 and 390 DF,  p-value: < 2.2e-16
```

**g)**

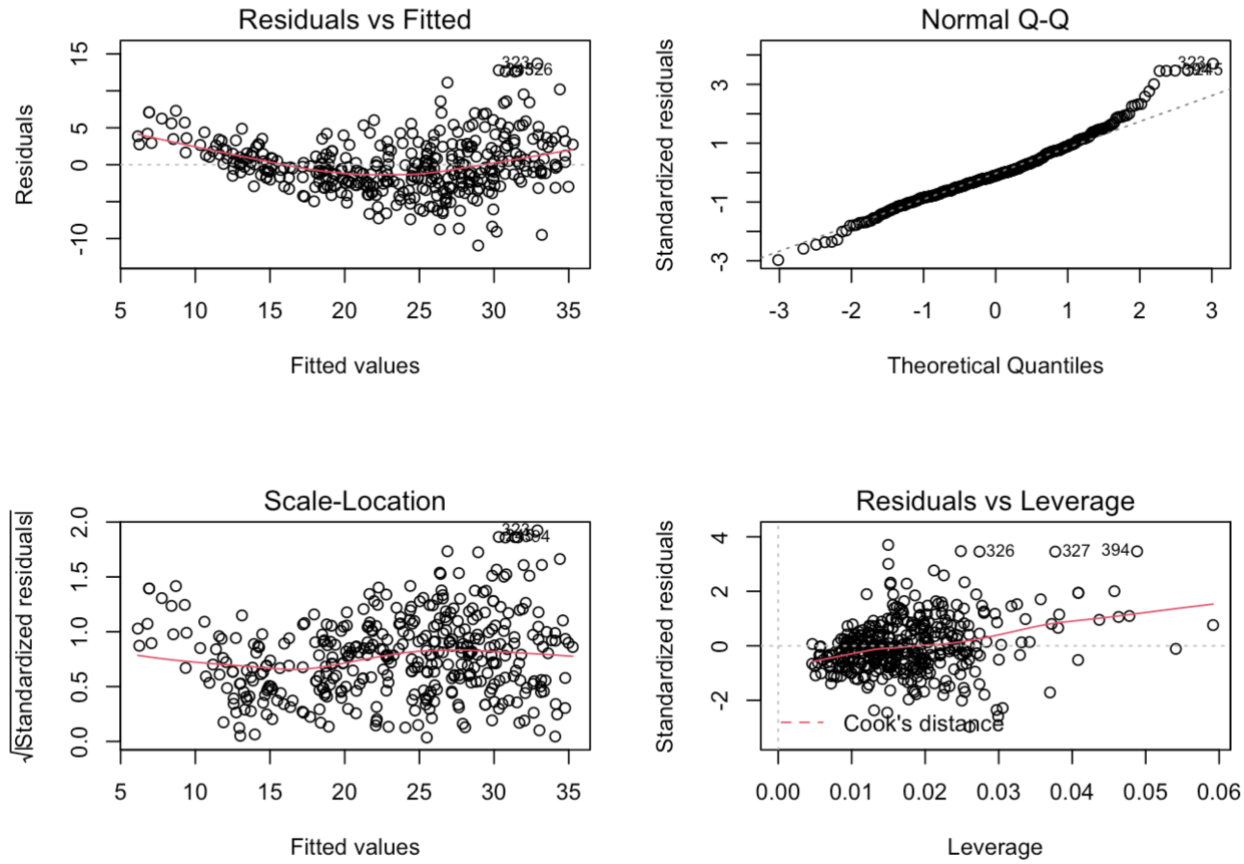
1. **Below a value of 10.737771 is seen for the GVIF of cylinders, which as stated can be taken as a problematic var in respect to collinearity problems.**
2. **cylinders, horsepower, and acceleration sd errors and t-stats changed in that the errors all stayed the same, but the t-stats got smaller.**
3. **The colinearity problem is solved, as we can see the >10 values are no longer.**

```
> vif(lm.fitMPG)
                  GVIF Df GVIF^(1/(2*Df))
cylinders    10.737771  1        3.276854
displacement 22.937950  1        4.789358
horsepower    9.957265  1        3.155513
weight       11.074349  1        3.327814
acceleration  2.625906  1        1.620465
year          1.301373  1        1.140777
origin        2.096060  2        1.203236

lm.fitMPG <- lm(Auto$mpg ~ . - name - displacement - weight, data = Auto)
> plot(lm.fitMPG)
> summary(lm.fitMPG)

Call:
lm(formula = Auto$mpg ~ . - name - displacement - weight, data = Auto)

Residuals:
     Min       1Q   Median       3Q      Max
-10.9382  -2.2983  -0.3841   2.1021  13.6975

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  -7.05536    5.09623  -1.384    0.167
```

```
cylinders    -1.17736    0.22952  -5.130 4.61e-07 ***
horsepower   -0.08836    0.01130  -7.819 5.15e-14 ***
acceleration -0.40980    0.09676  -4.235 2.86e-05 ***
year          0.67654    0.05724  11.820  < 2e-16 ***
origin2       2.35961    0.59865   3.942 9.61e-05 ***
origin3       3.62212    0.57316   6.320 7.26e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.727 on 385 degrees of freedom
Multiple R-squared:  0.7755,  Adjusted R-squared:  0.772
F-statistic: 221.6 on 6 and 385 DF,  p-value: < 2.2e-16

> vif(lm.fitMPG)
                 GVIF Df GVIF^(1/(2*Df))
cylinders    4.314828  1        2.077216
horsepower   5.324960  1        2.307588
acceleration 2.006098  1        1.416368
year         1.251327  1        1.118627
origin       1.675771  2        1.137768
```

<u>**h)**</u>

1. **There appears to be a non-linear relationship between the residuals and fitted values, as a slight curve, u shape, can be seen in the plot.**
2. **Yes, when the fitted values > 30, we can see outliers present.**
3. **No evidence of unusually large leverage points.**




**i)**

**Below a clear view of the statistical significance of the horsepower:origin (2 and 3) interactive term can be seen. In both models without name, and one also without displacement and weight, both showed P < 0.0001, proving the significance.**

```
> lm.fitHOInt <- lm(Auto$mpg ~ . - name + (horsepower:origin), data = Auto)
> summary(lm.fitHOInt)

Call:
lm(formula = Auto$mpg ~ . - name + (horsepower:origin), data = Auto)

Residuals:
    Min      1Q  Median      3Q     Max
-9.3534 -1.8467 -0.1525  1.5595 11.9755

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)     -1.616e+01  4.318e+00  -3.741 0.000211 ***
cylinders       -5.380e-01  2.962e-01  -1.816 0.070080 .
displacement    -3.473e-03  7.784e-03  -0.446 0.655735
horsepower       6.273e-03  1.297e-02   0.484 0.628994
```

```
weight            -4.482e-03  6.603e-04  -6.788 4.38e-11 ***
acceleration       -1.600e-01  9.496e-02  -1.685 0.092866 .
year               7.566e-01  4.783e-02  15.819  < 2e-16 ***
origin2            1.428e+01  1.776e+00   8.040 1.14e-14 ***
origin3            1.337e+01  1.801e+00   7.425 7.45e-13 ***
horsepower:origin2 -1.498e-01  2.162e-02  -6.931 1.79e-11 ***
horsepower:origin3 -1.347e-01  2.183e-02  -6.170 1.75e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.048 on 381 degrees of freedom
Multiple R-squared:  0.8514,  Adjusted R-squared:  0.8475
F-statistic: 218.3 on 10 and 381 DF,  p-value: < 2.2e-16

> lm.fitHOInt <- lm(Auto$mpg ~ . - name - displacement - weight + (horsepower:origin), data =
Auto)
> summary(lm.fitHOInt)

Call:
lm(formula = Auto$mpg ~ . - name - displacement - weight + (horsepower:origin),
   data = Auto)

Residuals:
    Min       1Q   Median       3Q      Max
-10.5389  -1.9760  -0.5574   1.6785  12.5593

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)        -7.00018    4.50609  -1.553    0.121
cylinders          -1.77639    0.21079  -8.427 7.25e-16 ***
horsepower         -0.05498    0.01048  -5.245 2.60e-07 ***
acceleration       -0.53472    0.08635  -6.192 1.53e-09 ***
year               0.69777    0.05066  13.773  < 2e-16 ***
origin2            16.95794    1.82200   9.307  < 2e-16 ***
origin3            16.59896    1.84691   8.987  < 2e-16 ***
horsepower:origin2 -0.17823    0.02144  -8.313 1.65e-15 ***
horsepower:origin3 -0.16112    0.02209  -7.293 1.75e-12 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.294 on 383 degrees of freedom
Multiple R-squared:  0.8255,  Adjusted R-squared:  0.8219
F-statistic: 226.5 on 8 and 383 DF,  p-value: < 2.2e-16
```