Homework 3

Drew Dahlquist

2/16/2022

1.

```
library(MASS)
library(ISLR)
```

```
Boston = read.csv("Boston.csv",stringsAsFactors=TRUE)
fix(Boston)
names(Boston)
```

```
##  [1] "crim"    "zn"      "indus"   "chas"    "nox"     "rm"      "age"
##  [8] "dis"     "rad"     "tax"     "ptratio" "black"   "lstat"   "medv"
```

```
lm.fit=lm(medv~lstat, data=Boston)
attach(Boston)
lm.fit=lm(medv~lstat)
```

```
lm.fit
```

```
##
## Call:
## lm(formula = medv ~ lstat)
##
## Coefficients:
## (Intercept)        lstat
##       34.55        -0.95
```

```
names(lm.fit)
```

```
##  [1] "coefficients"  "residuals"     "effects"       "rank"
##  [5] "fitted.values" "assign"        "qr"            "df.residual"
##  [9] "xlevels"       "call"          "terms"         "model"
```

```
coef(lm.fit)
```

```
## (Intercept)       lstat
##  34.5538409  -0.9500494
```

```
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")
```

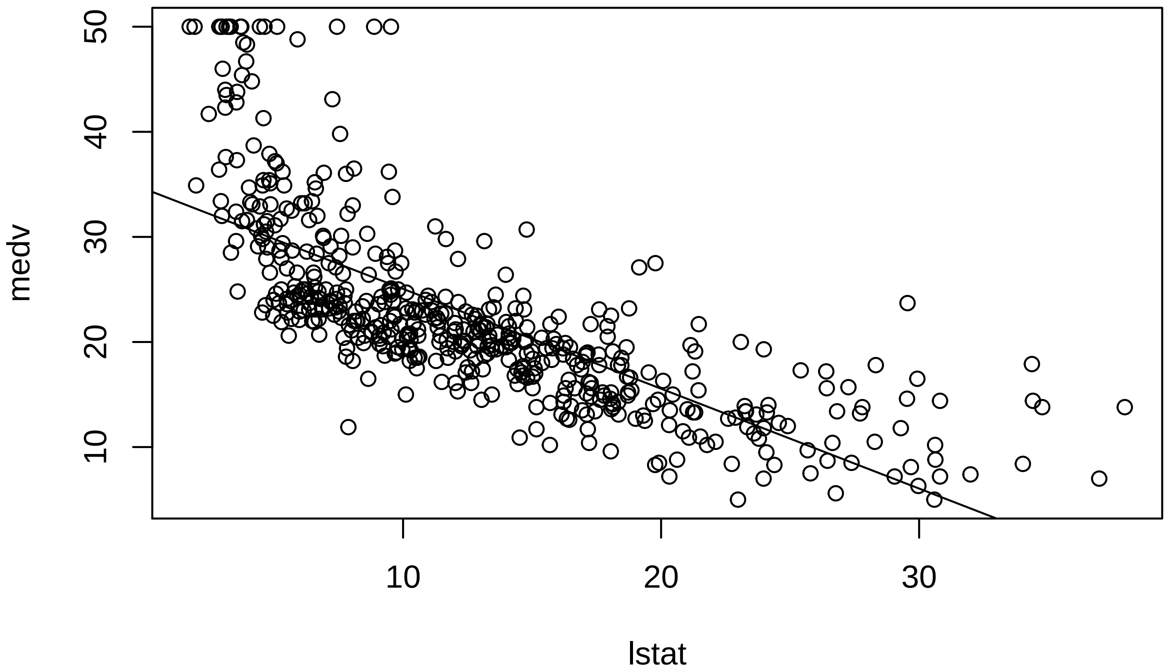
```
##        fit      lwr      upr
## 1 29.80359 29.00741 30.59978
```

```
## 2 25.05335 24.47413 25.63256
## 3 20.30310 19.73159 20.87461
```

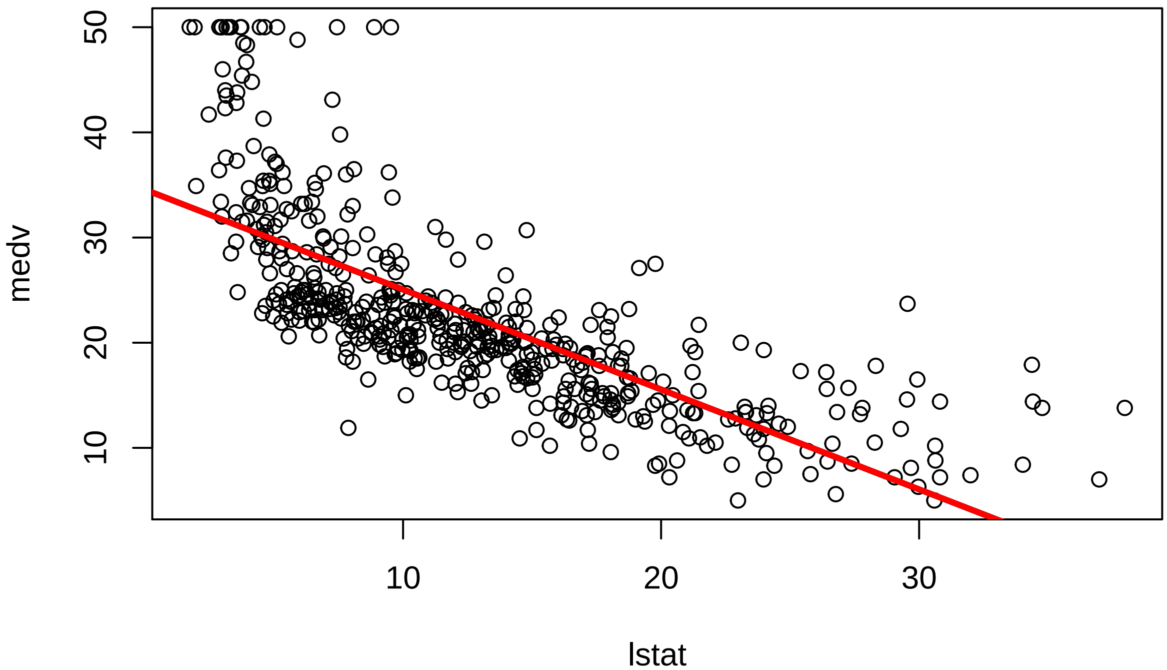
```
predict(lm.fit,data.frame(lstat=c(5,10,15)),
        interval="prediction")
```

```
##        fit       lwr      upr
## 1 29.80359 17.565675 42.04151
## 2 25.05335 12.827626 37.27907
## 3 20.30310  8.077742 32.52846
```

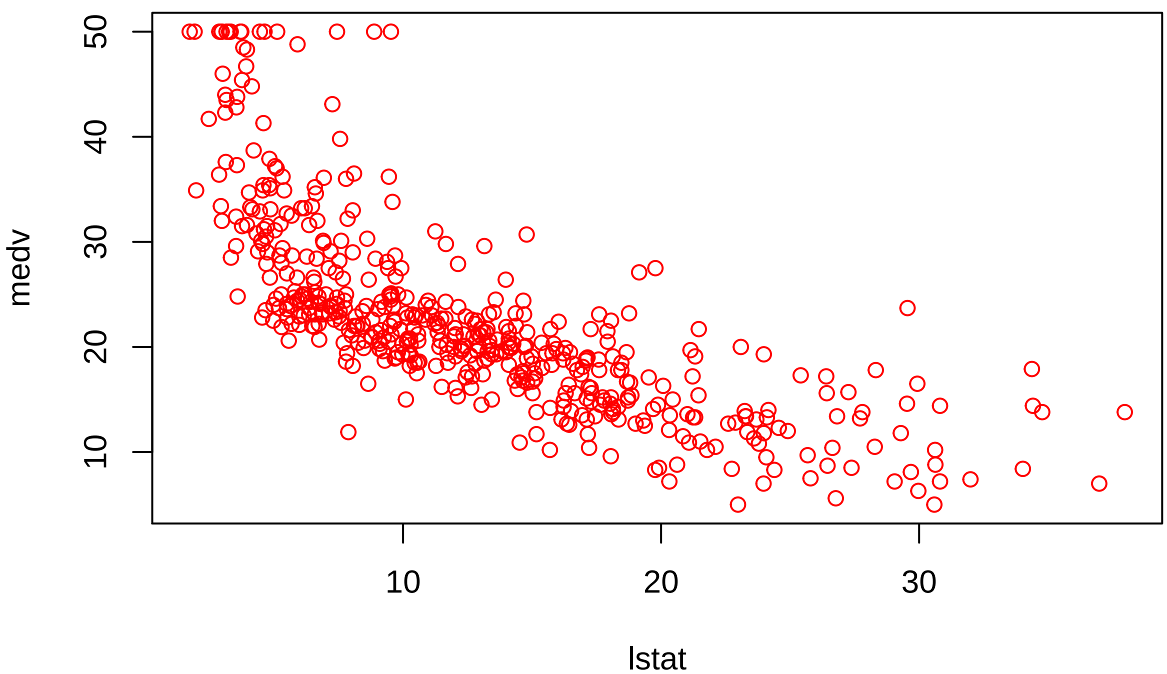
```
plot(lstat,medv)
abline(lm.fit)
```



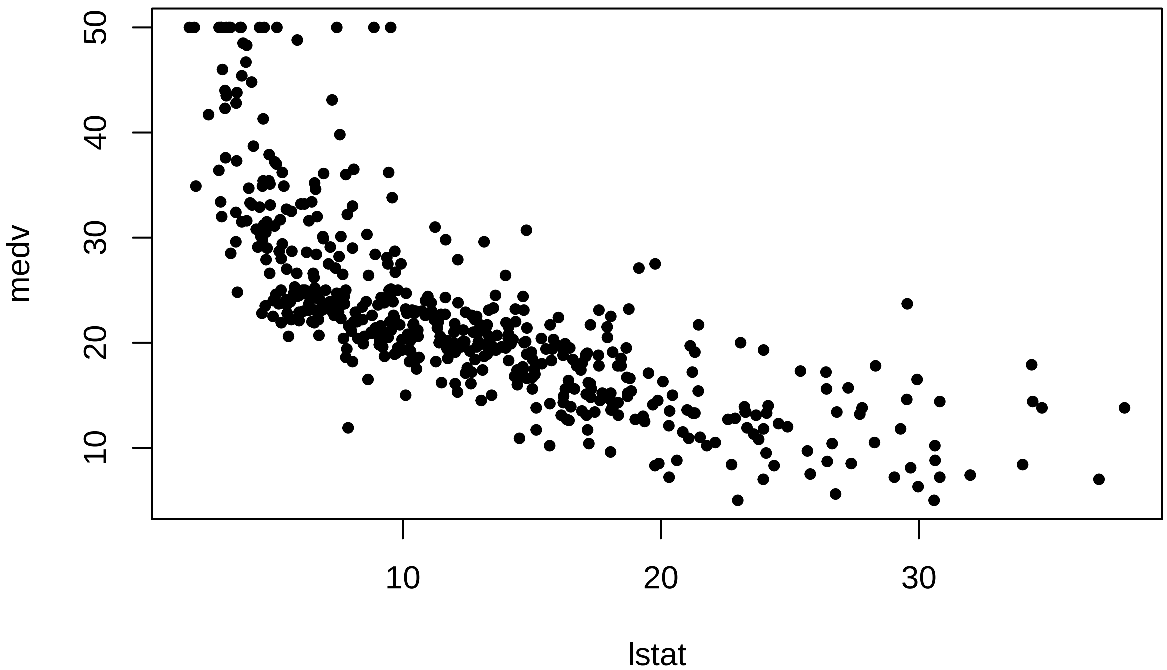

```
plot(lstat,medv)
abline(lm.fit)
abline(lm.fit,lwd=3)
abline(lm.fit,lwd=3,col="red")
```



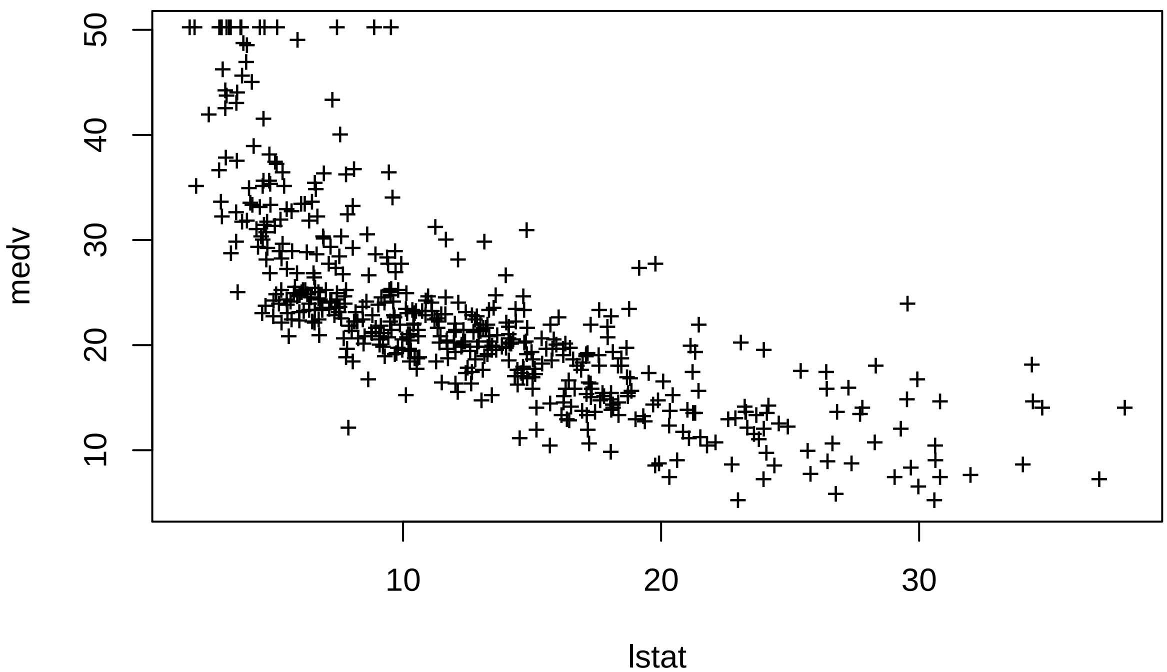

```
plot(lstat,medv,col="red")
```



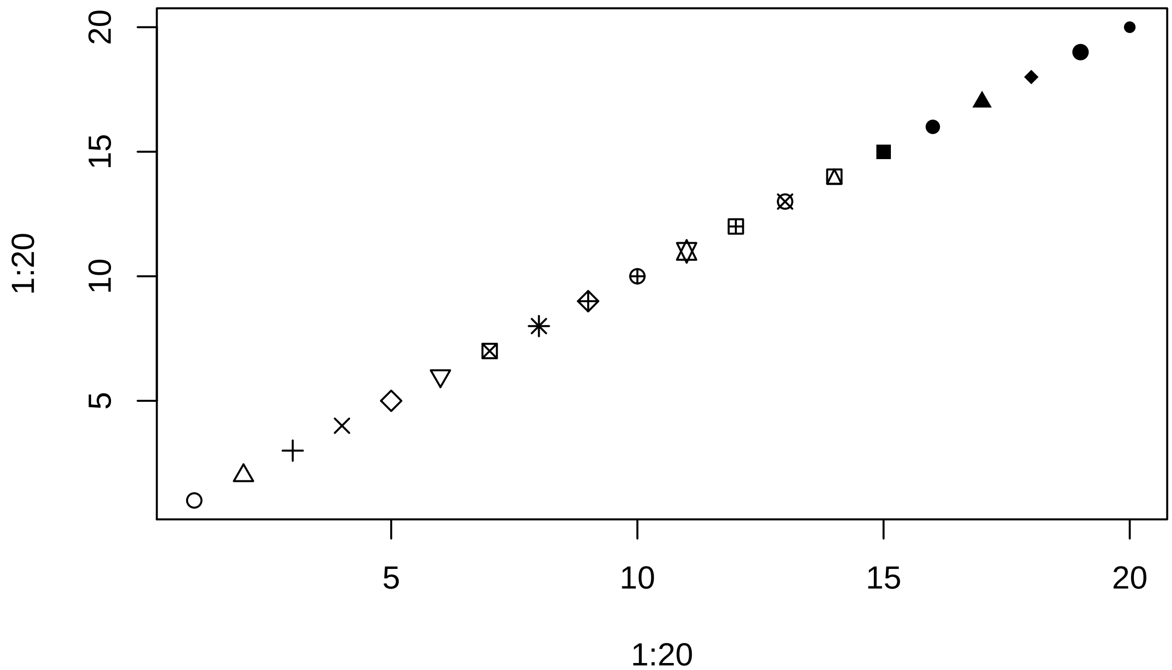

```
plot(lstat,medv,pch=20)
```



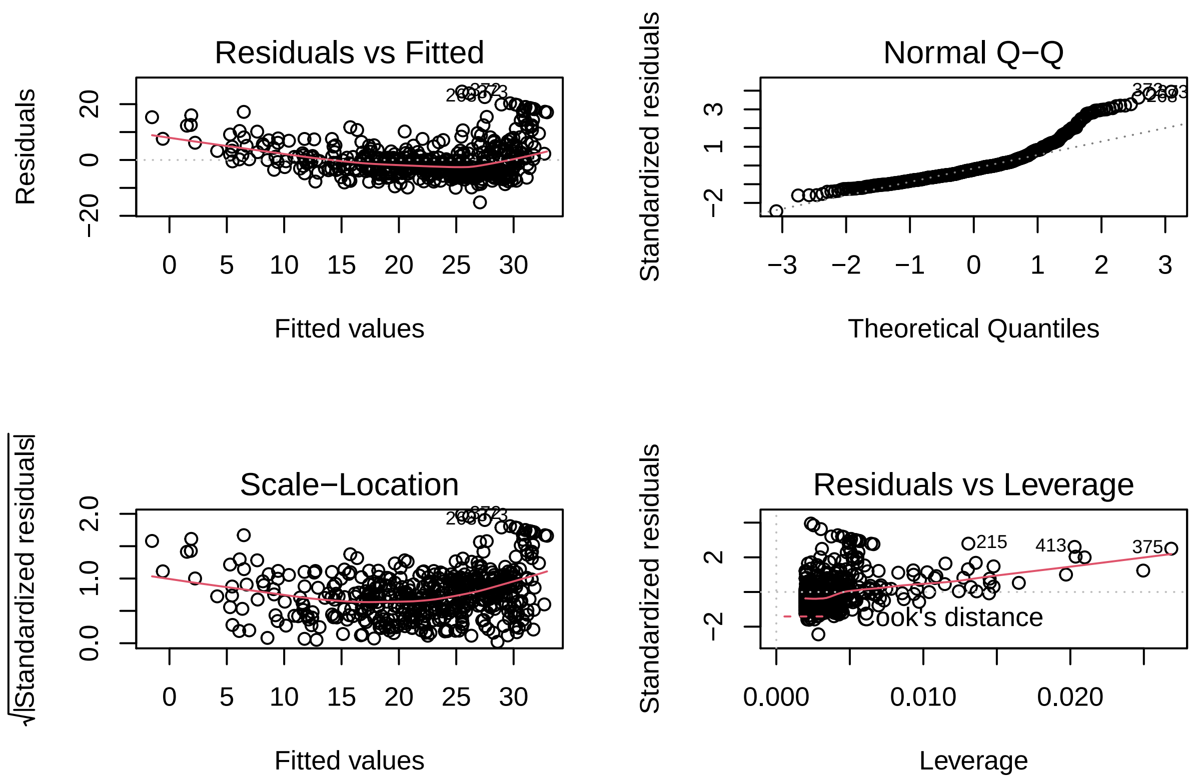

```
plot(lstat,medv,pch="+")
```



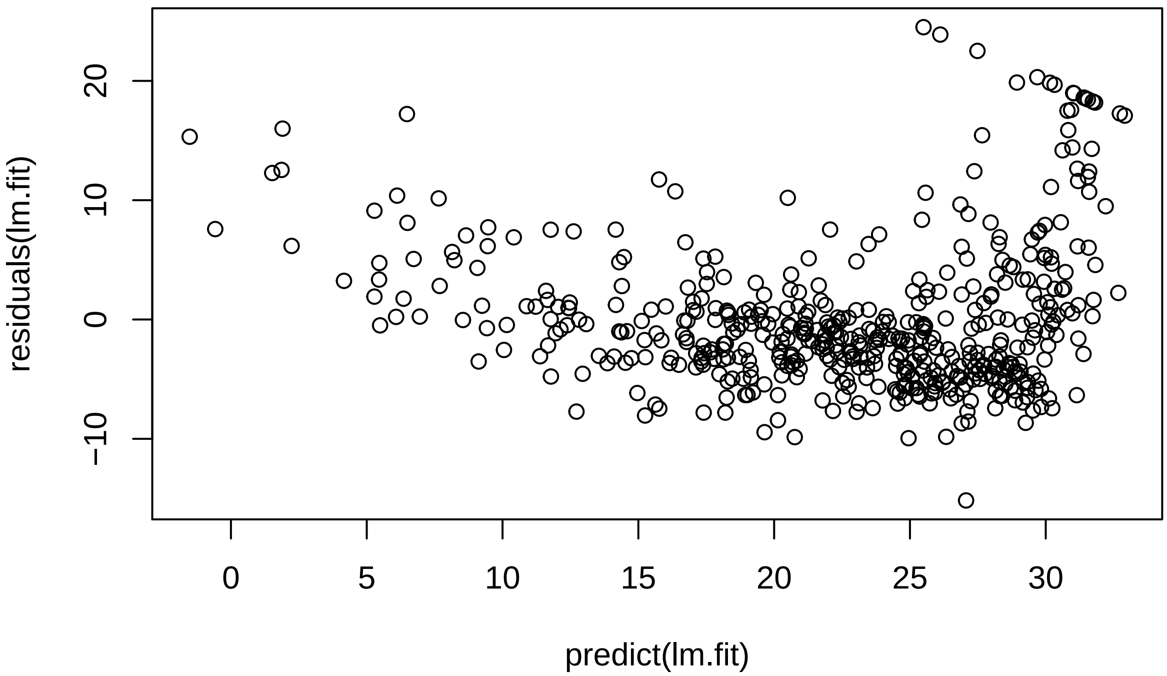

```
plot(1:20,1:20,pch=1:20)
```



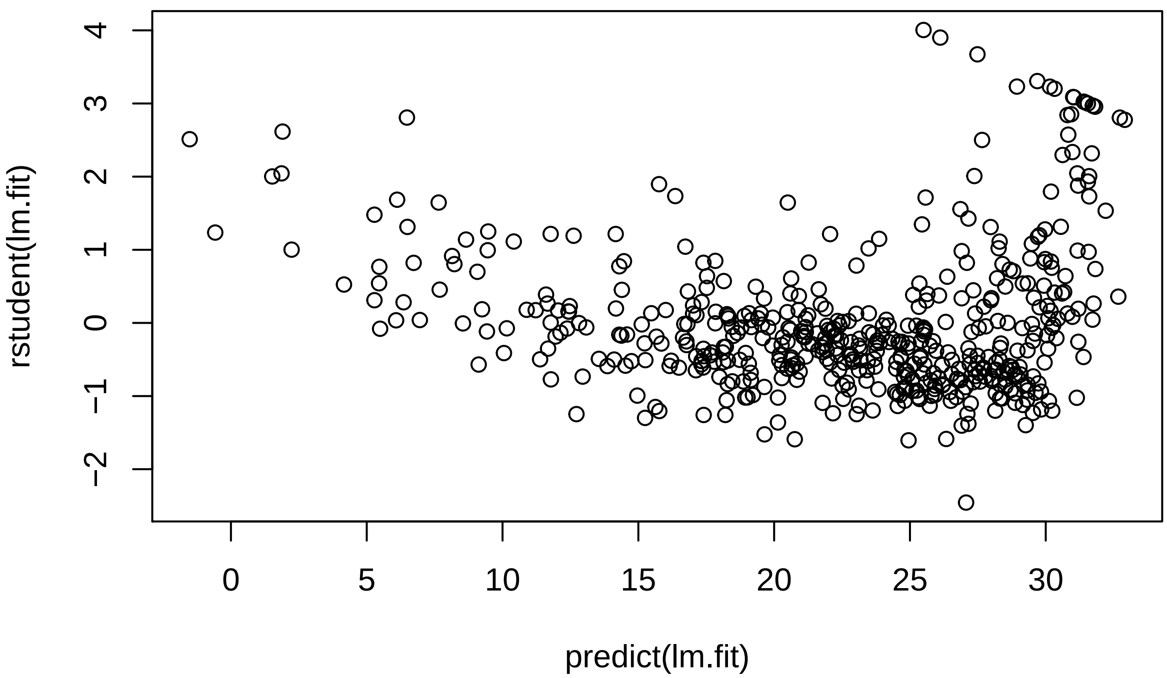

```r
par(mfrow=c(2,2))
plot(lm.fit)
```



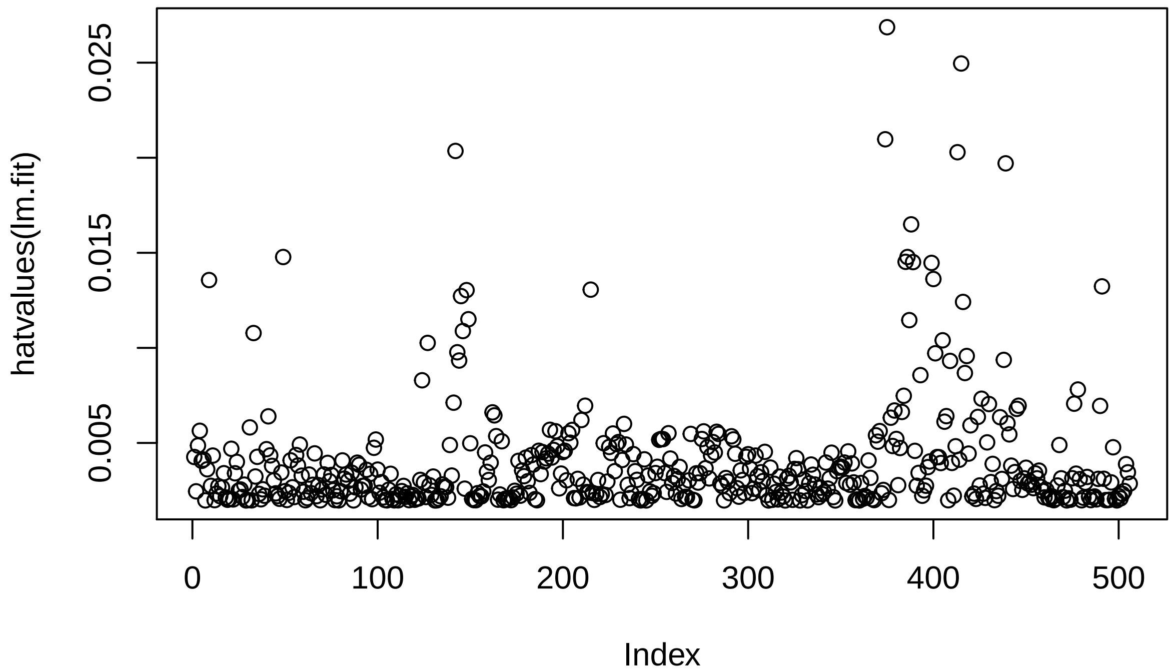

```r
plot(predict(lm.fit), residuals(lm.fit))
```




```r
plot(predict(lm.fit), rstudent(lm.fit))
```




```r
plot(hatvalues(lm.fit))
```




```
which.max(hatvalues(lm.fit))
```

```
## 375
## 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
## -15.595  -2.730  -0.518   1.777  26.199
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.646e+01  5.103e+00   7.144 3.28e-12 ***
## crim        -1.080e-01  3.286e-02  -3.287 0.001087 **
## zn           4.642e-02  1.373e-02   3.382 0.000778 ***
## indus        2.056e-02  6.150e-02   0.334 0.738288
## chas         2.687e+00  8.616e-01   3.118 0.001925 **
## nox         -1.777e+01  3.820e+00  -4.651 4.25e-06 ***
## rm           3.810e+00  4.179e-01   9.116  < 2e-16 ***
## age          6.922e-04  1.321e-02   0.052 0.958229
## dis         -1.476e+00  1.995e-01  -7.398 6.01e-13 ***
## rad          3.060e-01  6.635e-02   4.613 5.07e-06 ***
## tax         -1.233e-02  3.760e-03  -3.280 0.001112 **
## ptratio     -9.527e-01  1.308e-01  -7.283 1.31e-12 ***
## black        9.312e-03  2.686e-03   3.467 0.000573 ***
## lstat       -5.248e-01  5.072e-02 -10.347  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.745 on 492 degrees of freedom
## Multiple R-squared:  0.7406, Adjusted R-squared:  0.7338
## F-statistic: 108.1 on 13 and 492 DF,  p-value: < 2.2e-16
```

```r
library(car)
```

```
## Loading required package: carData
```

```
## Warning: package 'carData' was built under R version 4.0.5
```

```r
vif(lm.fit)
```

```
##     crim       zn    indus     chas      nox       rm      age      dis
## 1.792192 2.298758 3.991596 1.073995 4.393720 1.933744 3.100826 3.955945
##      rad      tax  ptratio    black    lstat
## 7.484496 9.008554 1.799084 1.348521 2.941491
```

```r
lm.fit1=lm(medv~.-age,data=Boston)
summary(lm.fit1)
```

```
##
## Call:
## lm(formula = medv ~ . - age, data = Boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.6054  -2.7313  -0.5188   1.7601  26.2243
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  36.436927   5.080119   7.172 2.72e-12 ***
## crim         -0.108006   0.032832  -3.290 0.001075 **
## zn            0.046334   0.013613   3.404 0.000719 ***
```

```
## indus         0.020562   0.061433   0.335 0.737989
## chas          2.689026   0.859598   3.128 0.001863 **
## nox         -17.713540   3.679308  -4.814 1.97e-06 ***
## rm            3.814394   0.408480   9.338  < 2e-16 ***
## dis          -1.478612   0.190611  -7.757 5.03e-14 ***
## rad           0.305786   0.066089   4.627 4.75e-06 ***
## tax          -0.012329   0.003755  -3.283 0.001099 **
## ptratio      -0.952211   0.130294  -7.308 1.10e-12 ***
## black         0.009321   0.002678   3.481 0.000544 ***
## lstat        -0.523852   0.047625 -10.999  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.74 on 493 degrees of freedom
## Multiple R-squared:  0.7406, Adjusted R-squared:  0.7343
## F-statistic: 117.3 on 12 and 493 DF,  p-value: < 2.2e-16
```

```
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),data=Boston)
summary(lm.fit2)
```

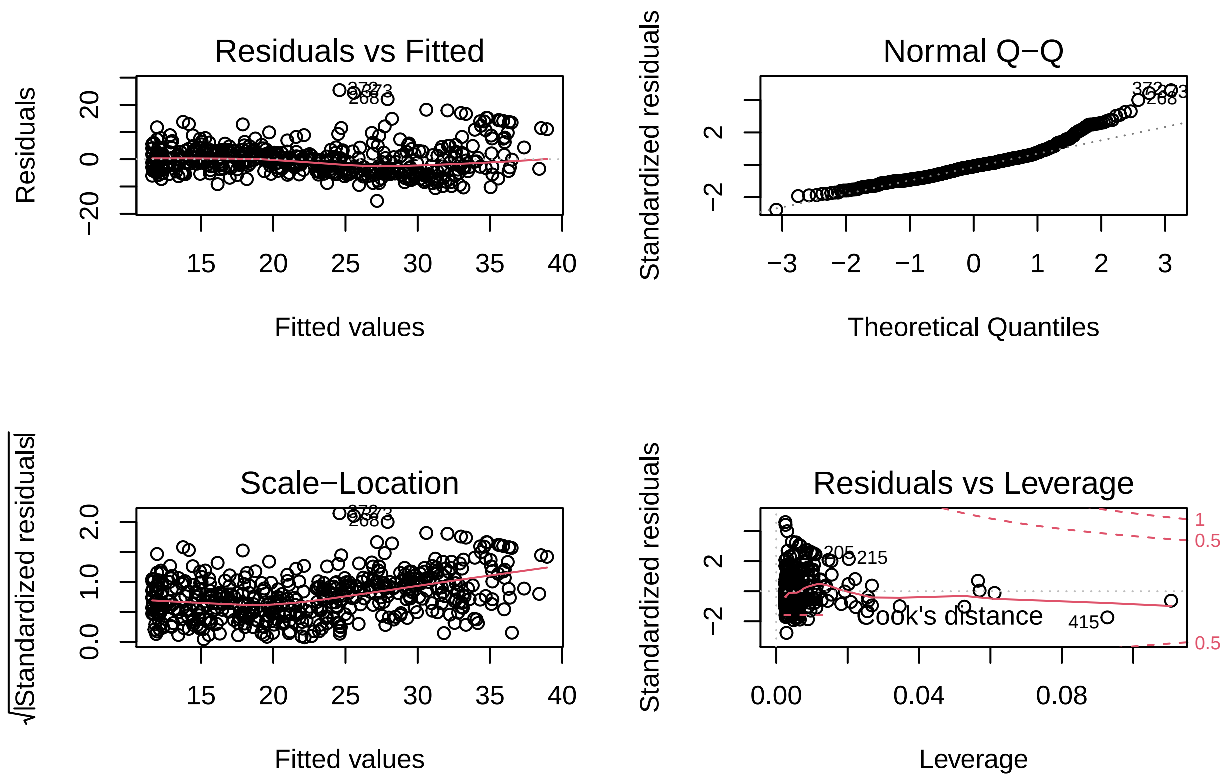
```
##
## Call:
## lm(formula = medv ~ lstat + I(lstat^2), data = Boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.2834  -3.8313  -0.5295   2.3095  25.4148
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 42.862007   0.872084   49.15   <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,data=Boston)
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),data=Boston)
summary(lm.fit5)
```

```
##
## Call:
## lm(formula = medv ~ poly(lstat, 5), data = Boston)
##
```

```
## 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
```

```
Carseats = read.csv("Carseats.csv",stringsAsFactors=TRUE)
fix(Carseats)
names(Carseats)
```

```
##  [1] "Sales"       "CompPrice"   "Income"      "Advertising" "Population"
##  [6] "Price"       "ShelveLoc"   "Age"         "Education"   "Urban"
## [11] "US"
```

```
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
```

```r
attach(Carseats)
contrasts(ShelveLoc)
```

```
##        Good Medium
## Bad       0      0
## Good      1      0
## Medium    0      1
```

```r
LoadLibraries=function(){
  library(ISLR)
  library(MASS)
  print("The libraries have been loaded.")
}
```

```r
LoadLibraries
```

```
## function(){
##   library(ISLR)
##   library(MASS)
##   print("The libraries have been loaded.")
## }
```

```r
LoadLibraries()
```

```
## [1] "The libraries have been loaded."
```

2.

(2a)

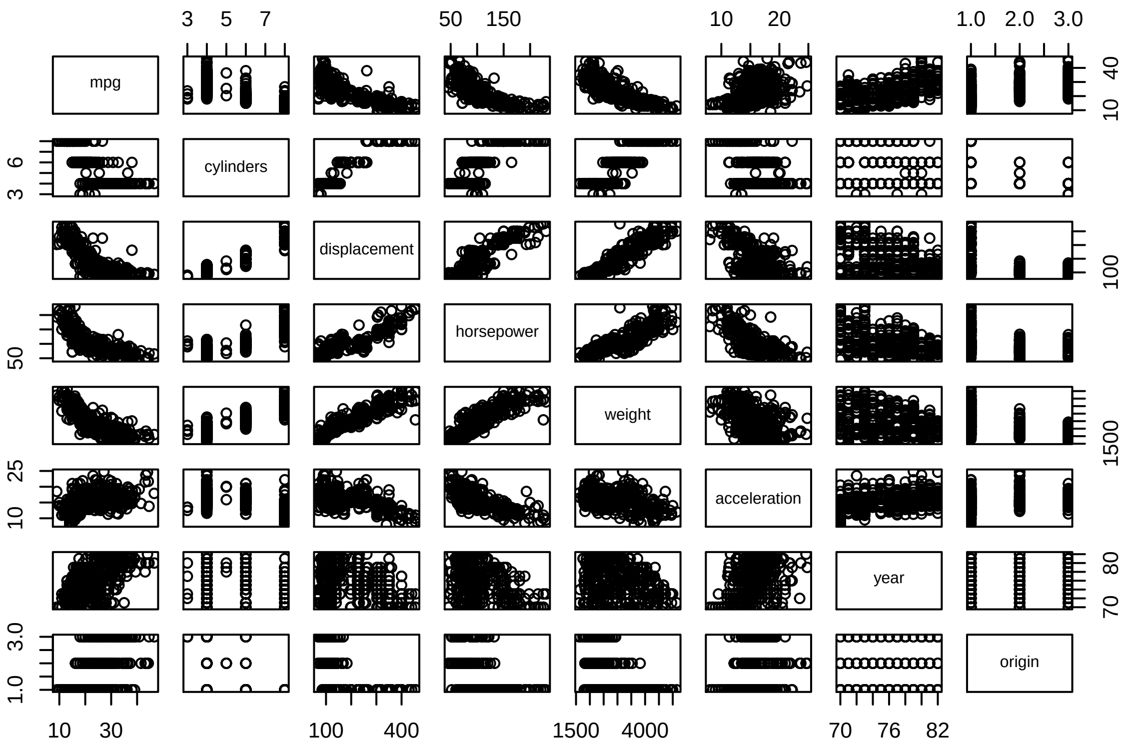
```r
Auto = read.csv("Auto.csv", na.strings="?")
Auto = na.omit(Auto)
```

```
Auto$origin = as.factor(Auto$origin)
```

(2b)

The predictor variables that look to be associated with mpg are displacement, horsepower, and weight. Also, cylinders and year look to have a slight assocaition as well, but not as strong as the others mentioned.

```
pairs(Auto[,1:8])
```



(2c)

Yes, the outcomes looks to be consistent with what I found from the scatterplot matrix since the predictors that I expected to be correlated with mpg do infact have a high correlation.

There are potential collinearity problems in this data since multiple predictors that look to affect mpg are also highly correlated with eachother (e.g., cylinders, displacement, and horsepower).

```
cor(Auto[,1:7])
```

```
##                     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
##              acceleration       year
## mpg             0.4233285  0.5805410
## cylinders      -0.5046834 -0.3456474
## displacement   -0.5438005 -0.3698552
## horsepower     -0.6891955 -0.4163615
## weight         -0.4168392 -0.3091199
```

```
## acceleration    1.0000000  0.2903161
## year            0.2903161  1.0000000
```

(2d)

Yes, there is a relationship between predictors and the response, seen by an R^2 statistic of 0.8242.

Displacement, Weight, Year, and Origin have a statistically significant relationship to the response.

The coefficient for the year variable suggests that we expect to have a higher mpg for newer cars.

```
lm.fit.d=lm(mpg~.-name, data=Auto)
summary(lm.fit.d)
```

```
##
## Call:
## lm(formula = 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
```

(2e)

```
contrasts(Auto$origin)
```

```
##   2 3
## 1 0 0
## 2 1 0
## 3 0 1
```

(2f)

The result is not consistent with the significance of cylinders obtained in part d, this is likely due to collinearity within other predictors used to fit the model.

```
lm.fit.f=lm(mpg~cylinders, data=Auto)
summary(lm.fit.f)
```

```
##
## Call:
## lm(formula = mpg ~ 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 ***
## 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
```

(2g)

The parameter estimates have changed for most of the predictors. For cylinders, horsepower, and acceleration the standard errors have all increased whereas the t values have all decreased.

According to the VIF for Model-2, the collinearity problem appears to be fixed.

```
vif(lm.fit.d)
```

```
##                   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.fit.d.2=lm(mpg~cylinders+horsepower+acceleration+year+origin, data=Auto)
summary(lm.fit.d)
```

```
##
## Call:
## lm(formula = 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
```

```r
summary(lm.fit.d.2)
```

```
##
## Call:
## lm(formula = mpg ~ cylinders + horsepower + acceleration + year +
##     origin, 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
```

```r
vif(lm.fit.d.2)
```

```
##                  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
```

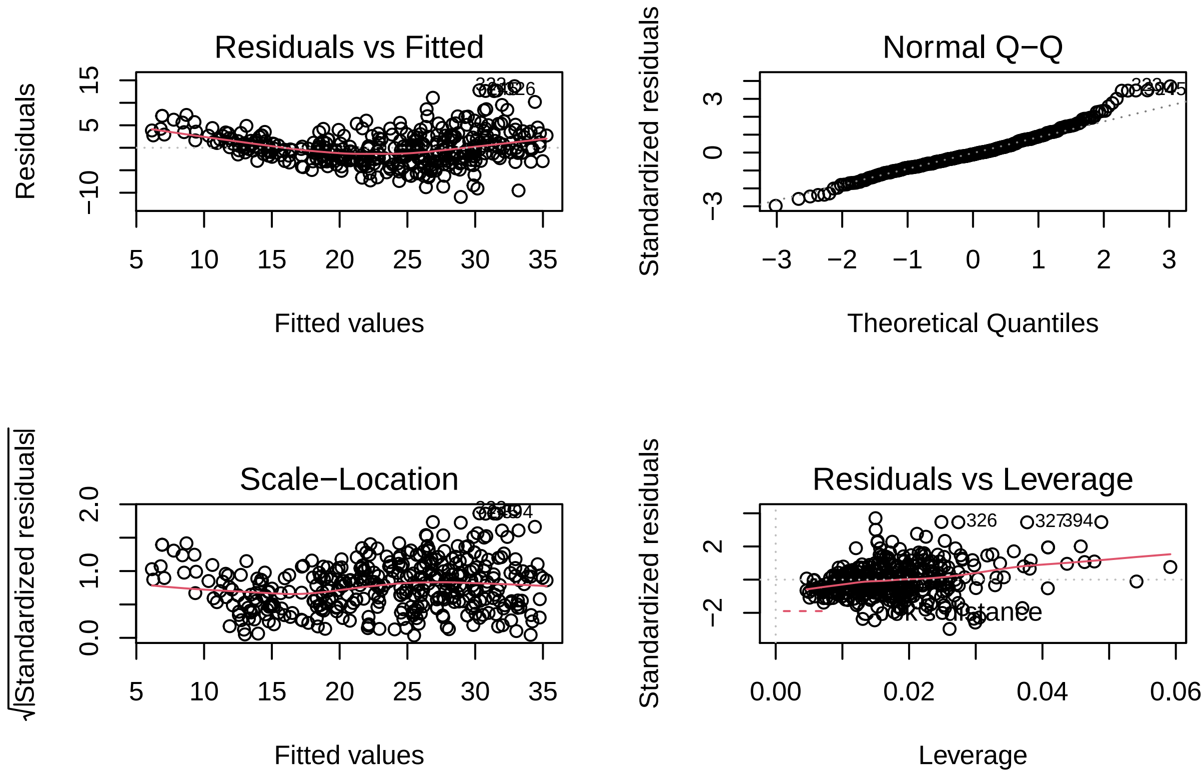
(2h)

The residual plots suggest there may be a very small amount of nonlinearity between the response and predictors.

There are a few outliers.

The leverage plot identifies a couple high leverage points.

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



(2i)

Yes, the interactions appears to be statistically significant.

```
lm.fit.i=lm(mpg~horsepower*origin, data=Auto)
summary(lm.fit.i)
```

```
##
## Call:
## lm(formula = mpg ~ horsepower * origin, data = Auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.7415  -2.9547  -0.6389   2.3978  14.2495
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       34.476496   0.890665  38.709  < 2e-16 ***
## horsepower        -0.121320   0.007095 -17.099  < 2e-16 ***
## origin2           10.997230   2.396209   4.589 6.02e-06 ***
## origin3           14.339718   2.464293   5.819 1.24e-08 ***
## horsepower:origin2 -0.100515   0.027723  -3.626 0.000327 ***
## horsepower:origin3 -0.108723   0.028980  -3.752 0.000203 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.422 on 386 degrees of freedom
## Multiple R-squared:  0.6831, Adjusted R-squared:  0.679
## F-statistic: 166.4 on 5 and 386 DF,  p-value: < 2.2e-16
```