# ISLR Chapter 3 lab

Adrien Osakwe

## Load Libraries

```
library(MASS)
library(ISLR2)
```

## Simple Linear Regression

```
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)

# Showing basic info from the generated model
lm.fit
```

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

```
#Detailed information on model
summary(lm.fit)
```

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

```r
# Confidence Intervals for coefficients
confint(lm.fit)
```

```
##                 2.5 %     97.5 %
## (Intercept) 33.448457 35.6592247
## lstat       -1.026148 -0.8739505
```

```r
#Predict a series of medv values given a set of lstat values with 95% CI
predict(lm.fit, data.frame(lstat = (c(3,10,20))), interval = "confidence")
```

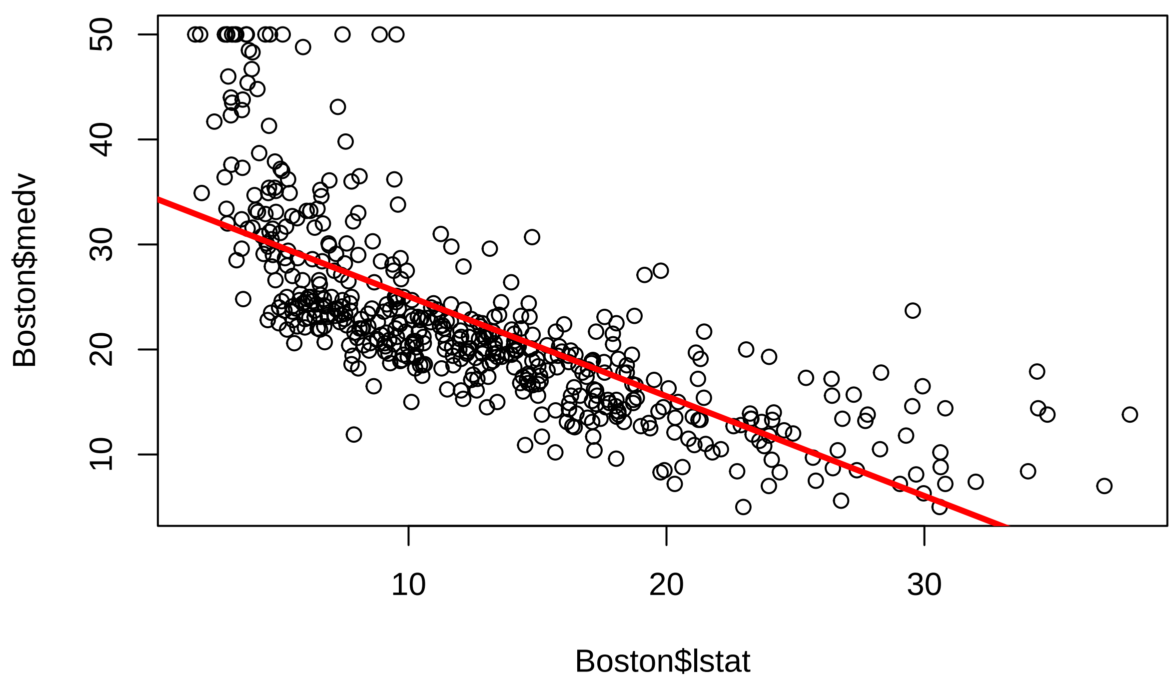
```
##        fit      lwr      upr
## 1 31.70369 30.79027 32.61712
## 2 25.05335 24.47413 25.63256
## 3 15.55285 14.77355 16.33216
```

```r
#Plotting Model
plot(Boston$lstat,Boston$medv)
abline(lm.fit)

#Trying some other stuff out
attach(Boston)
```

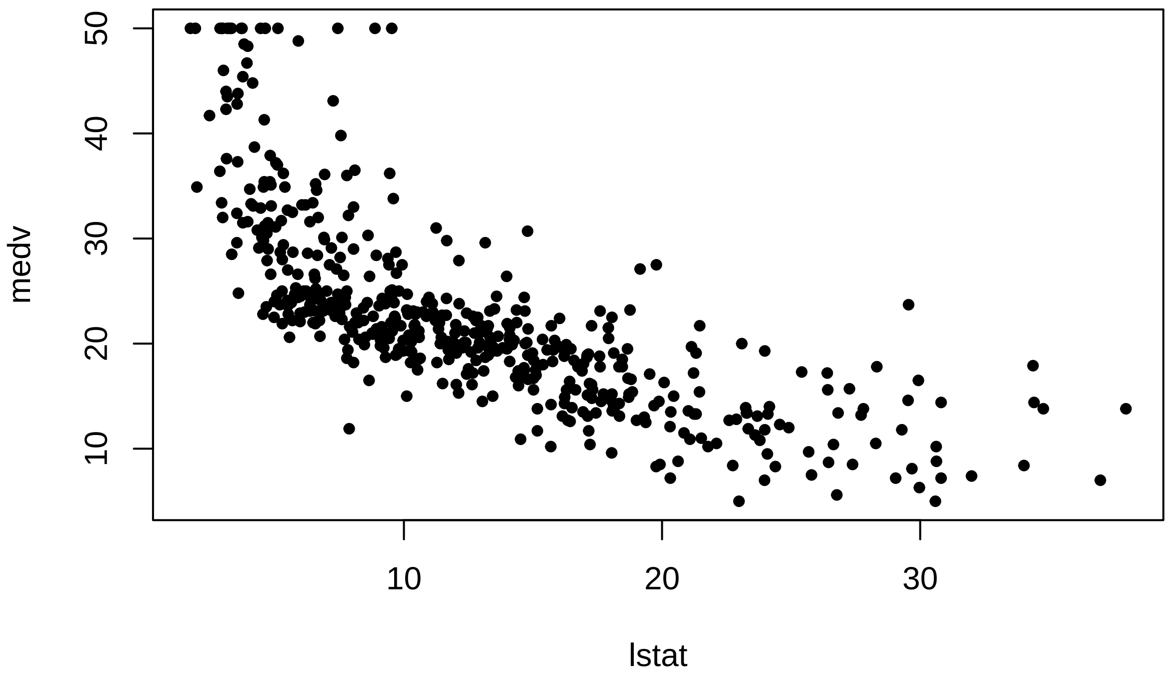
```
## The following objects are masked from Boston (pos = 4):
##
##     age, chas, crim, dis, indus, lstat, medv, nox, ptratio, rad, rm, tax, zn

## The following objects are masked from Boston (pos = 5):
##
##     age, chas, crim, dis, indus, lstat, medv, nox, ptratio, rad, rm, tax, zn

## The following objects are masked from Boston (pos = 7):
##
##     age, chas, crim, dis, indus, lstat, medv, nox, ptratio, rad, rm, tax, zn

## The following objects are masked from Boston (pos = 8):
##
##     age, chas, crim, dis, indus, lstat, medv, nox, ptratio, rad, rm, tax, zn
```

```
## The following objects are masked from Boston (pos = 12):
##
##     age, chas, crim, dis, indus, lstat, medv, nox, ptratio, rad, rm, tax, zn

## The following objects are masked from Boston (pos = 13):
##
##     age, chas, crim, dis, indus, lstat, medv, nox, ptratio, rad, rm, tax, zn
```

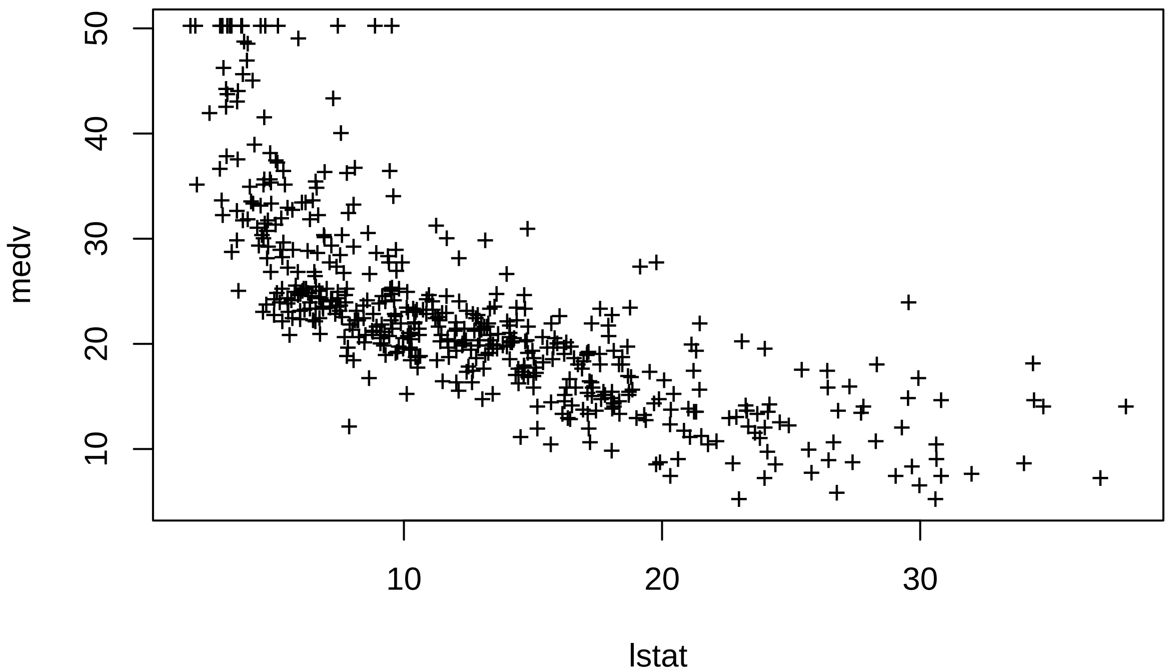
```
abline(lm.fit, lwd = 3, col = "red") # plots a thicker red line
```



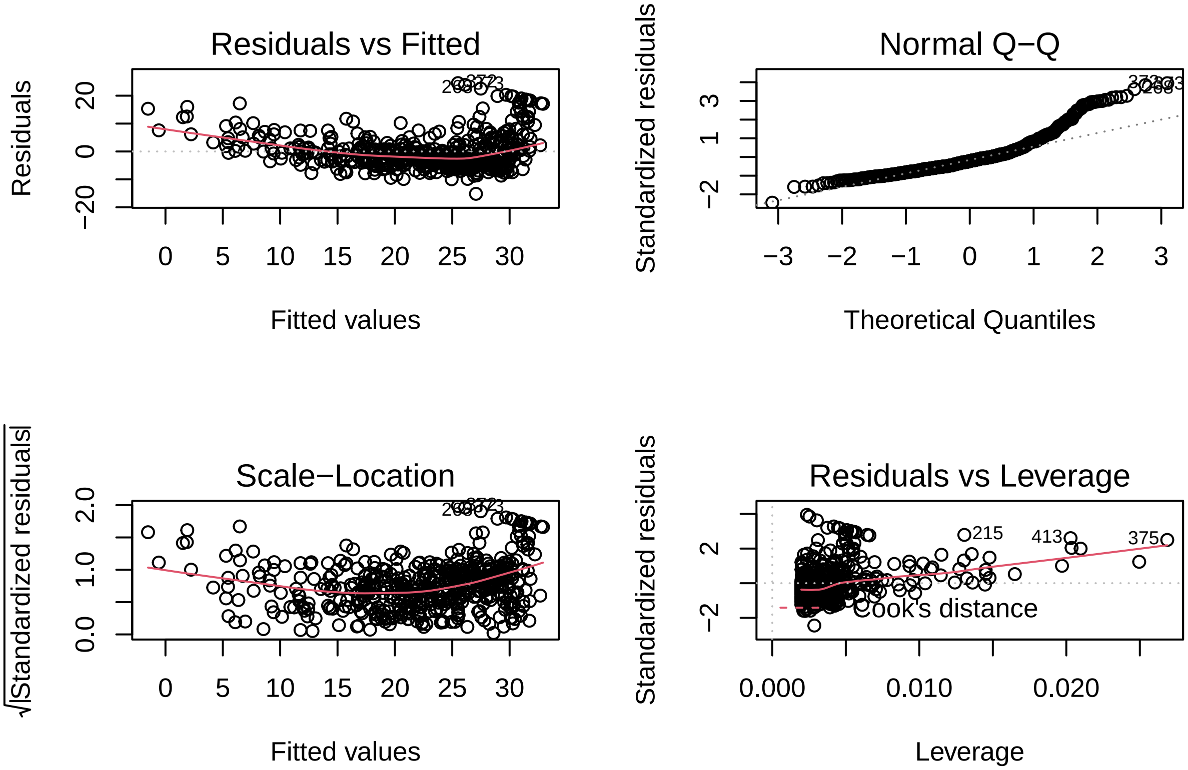

```
plot(lstat,medv,pch = 20) #changing symbol of points by symbol ID
```



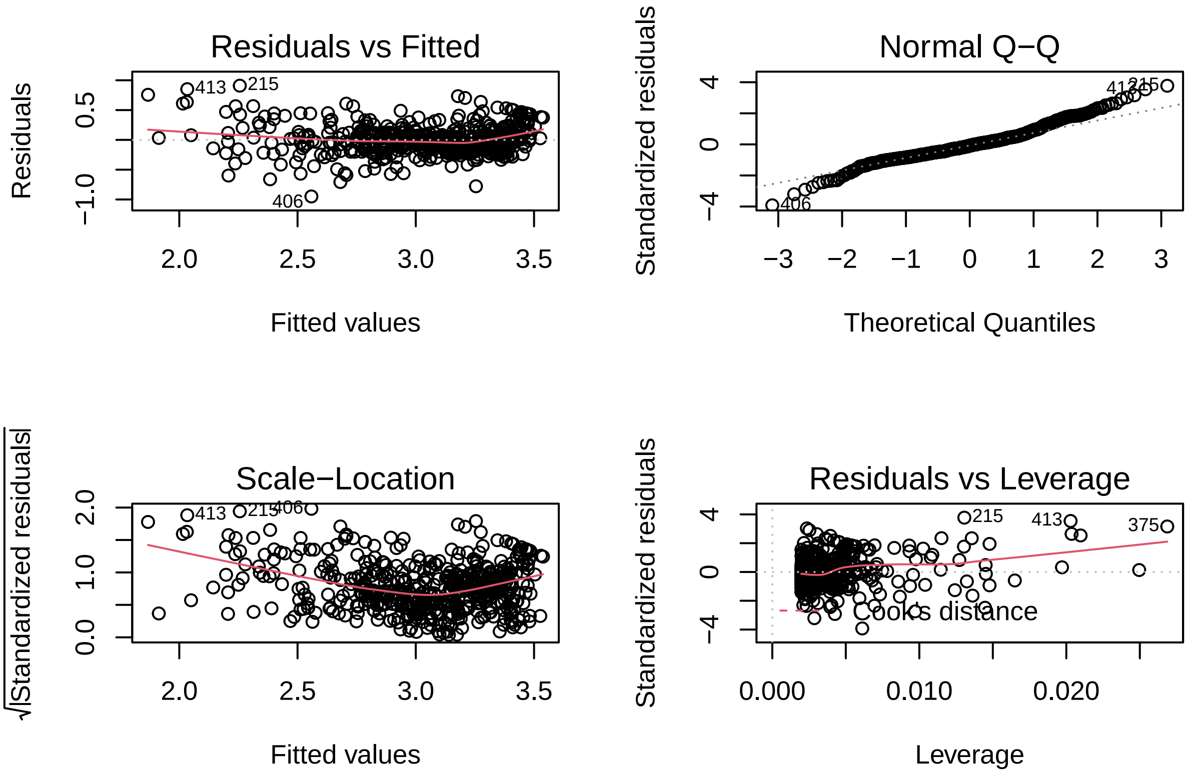

```
plot(lstat, medv, pch = "+")
```




```
#Diagnostic Plots - There are 4 plots if we plot lm.fit directly
par(mfrow = c(2,2))
plot(lm.fit)
```

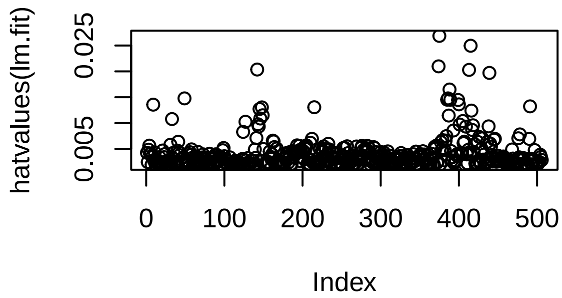



```
# The residuals vs fitted plot shows that there is a non-linear trend
# that was not captured by the model
#Scale Location plot shows certain values have an SR above sqrt(3)
#indicating potential outliers
#Residuals vs Leverage shows that there are observations with both a
#high leverage AND SR, meaning they should probably be removed
# The Q-Q plot has a considerable fat tail on the RHS indicating
# non-normality --> transforming the data could fix this - let's try it !


#Log transformed medv linear regression
par(mfrow = c(2,2))
plot(lm(log(medv) ~ lstat))
```




```
#This seems to improve a lot but not completely, still have apparent outliers
#to deal with
#Will have to look into this more in the future to see how to accomodate the fat
#tail of the Q-Q plot

#Note: can also plot these diagnostics individually with i.e. hatvalues for
#leverage, residuals() for residuals, and rstudent() for studentized residuals.
#Trying it out for leverage stats below:

#Leverage Statistics
#hatvalues function calculates the leverage statistics for the predictors
plot(hatvalues(lm.fit))
#find index with the largest leverage
which.max(hatvalues(lm.fit))
```

```
## 375
## 375
```




### Multiple Linear Regression

```
#Now adding age to the model to create a multiple linear regression
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 = 5):
##
##     age, chas, crim, dis, indus, lstat, medv, nox, ptratio, rad, rm, tax, zn

## The following objects are masked from Boston (pos = 6):
##
##     age, chas, crim, dis, indus, lstat, medv, nox, ptratio, rad, rm, tax, zn

## The following objects are masked from Boston (pos = 8):
##
##     age, chas, crim, dis, indus, lstat, medv, nox, ptratio, rad, rm, tax, zn

## The following objects are masked from Boston (pos = 9):
##
##     age, chas, crim, dis, indus, lstat, medv, nox, ptratio, rad, rm, tax, zn
```

```
## The following objects are masked from Boston (pos = 13):
##
##     age, chas, crim, dis, indus, lstat, medv, nox, ptratio, rad, rm, tax, zn

## The following objects are masked from Boston (pos = 14):
##
##     age, chas, crim, dis, indus, lstat, medv, nox, ptratio, rad, rm, tax, zn
```

```
lm.fit <- lm(medv ~ lstat + age)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = medv ~ lstat + age)
##
## 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
```

```
#It seems that R-squared barely changes upon adding age as a predictor. Age's
#coefficient has a significant p value but much less than lstat
# The F-statistic actually drops although it is still much larger than one


## Adding all predictors
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
```

```
#R-squared and RSE decrease, F-stat is still well above 1


#Looking at Variance Inflation Factor to identify collinearity
library(car)
vif(lm.fit)
```

```
##     crim       zn    indus     chas      nox       rm      age      dis      rad      tax  ptratio
## 1.767486 2.298459 3.987181 1.071168 4.369093 1.912532 3.088232 3.954037 7.445301 9.002158 1.797060 2
```

```
# tax and rad are pretty high relative to the others
#nox, dis and indus are also high
#Looking into what these variables are may give insight into why we are seeing
#collinearity and we can then decide which predictors to keep


#All but some predictors in model
#Will remove indus and age because of the high p values as well as tax given
#the high VIF
lm.fit <- update(lm.fit, ~.-age)
lm.fit <- update(lm.fit, ~.-indus)
lm.fit <- update(lm.fit, ~.-tax)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = medv ~ crim + zn + chas + nox + rm + dis + rad +
##     ptratio + lstat, data = Boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.8550  -2.9880  -0.5477   1.8770  26.4105
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)  39.98405    4.94523   8.085 4.78e-15 ***
## crim         -0.11854    0.03330  -3.559 0.000407 ***
## zn            0.03658    0.01357   2.695 0.007279 **
## chas          3.13944    0.86975   3.610 0.000338 ***
## nox         -21.37566    3.50093  -6.106 2.07e-09 ***
## rm            3.85056    0.41105   9.368  < 2e-16 ***
## dis          -1.45079    0.18905  -7.674 8.92e-14 ***
## rad           0.10457    0.04071   2.569 0.010495 *
## ptratio      -1.00175    0.13049  -7.677 8.74e-14 ***
## lstat        -0.55346    0.04797 -11.537  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.847 on 496 degrees of freedom
## Multiple R-squared:  0.7273, Adjusted R-squared:  0.7223
## F-statistic: 146.9 on 9 and 496 DF,  p-value: < 2.2e-16
```

```
#Barely changes R-squared, RSE and F
vif(lm.fit)
```

```
##     crim       zn     chas      nox       rm      dis      rad  ptratio    lstat
## 1.764257 2.154051 1.049185 3.538215 1.793239 3.407113 2.701027 1.715836 2.523246
```

```
#Also seem much lower VIF scores --> rad and tax must have been collinear
#One is for property tax and the other for accessibility to radial highways
#which facilitate access to different regions by car
#Could look into this more to understand why they would be collinear
```

### Interaction Terms

```
#Building model with lstat, age and their interaction as a product
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
## -15.806  -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
```

```
#The interaction between lstat and age seems to be much more important than age
#Let's try with others
#lstat is important, so is its interaction with age, lets add the interaction of
#crime and lstat

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

```
##
## Call:
## lm(formula = medv ~ lstat * age + lstat * crim, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -16.620  -3.957  -1.256   1.878  27.367
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 35.331467   1.466012  24.100  < 2e-16 ***
## lstat       -1.359118   0.165753  -8.200 2.05e-15 ***
## age          0.023603   0.021129   1.117 0.264496
## crim        -0.452590   0.107968  -4.192 3.27e-05 ***
## lstat:age    0.003151   0.001919   1.642 0.101152
## lstat:crim   0.017656   0.005078   3.477 0.000551 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.043 on 500 degrees of freedom
## Multiple R-squared:  0.5725, Adjusted R-squared:  0.5682
## F-statistic: 133.9 on 5 and 500 DF,  p-value: < 2.2e-16
```

```
#Does not improve the model much... could try this out again later
```

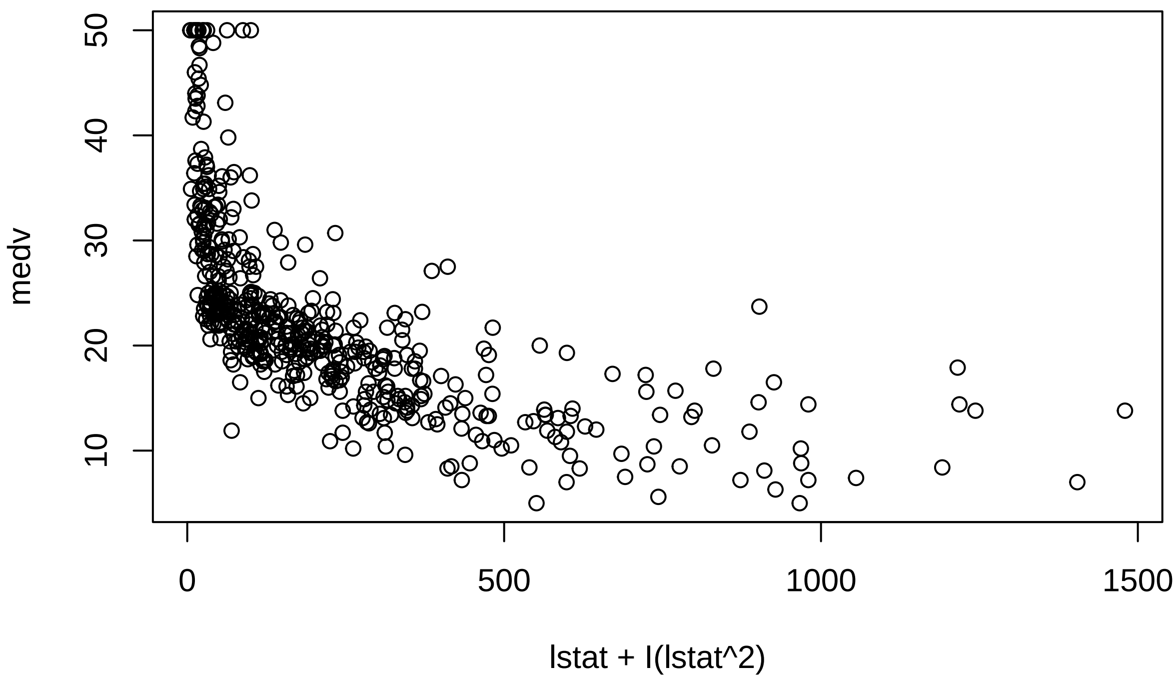
### Non-linear Transformations of the Predictors

```
lm.fit <- lm(medv ~ lstat + I(lstat^2), data = Boston)
summary(lm.fit)
```

```
##
## 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
```

```r
plot(lstat + I(lstat^2),medv)
```

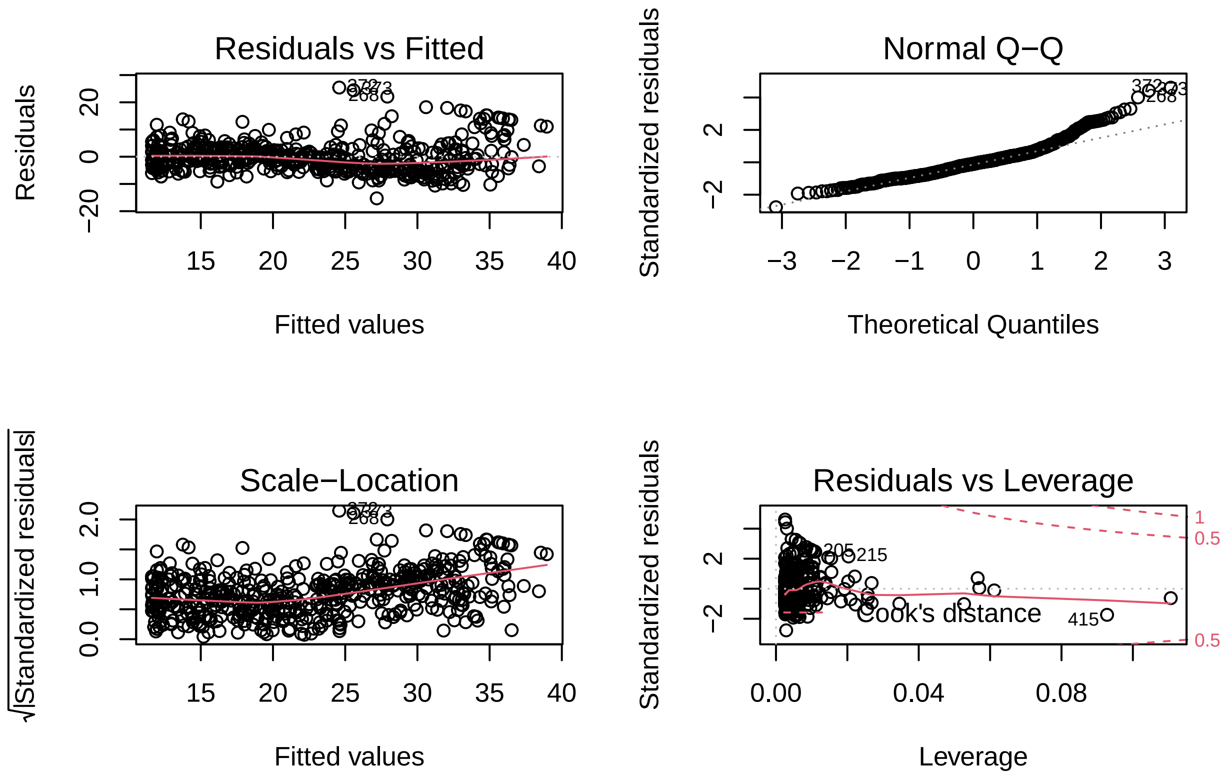



```r
#ANOVA comparison
lm.fit1 <- lm(medv ~ lstat, data = Boston)
anova(lm.fit1,lm.fit)
```

```
## 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
```

```
#Anova tells us that this new model is much better than just medv ~ lstat
par( mfrow = c(2, 2))
plot(lm.fit )
```




```
#Rv.F is pretty good, Q-Q- is as before but we may have no choice but to remove
#those outliers to accomodate this


#Other polynomial functions
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
```

```
#Anova again reveals that this model is much better
anova(lm.fit,lm.fit5)
```

```
## Analysis of Variance Table
##
## Model 1: medv ~ lstat + I(lstat^2)
## Model 2: medv ~ poly(lstat, 5)
##   Res.Df   RSS Df Sum of Sq      F    Pr(>F)
## 1    503 15347
## 2    500 13597  3    1750.2 21.453 4.372e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#Another? Turns out x^5 is as good as it gets for lstat
lm.fit6 <- lm(medv ~ poly(lstat,6))
anova(lm.fit5,lm.fit6)
```

```
## Analysis of Variance Table
##
## Model 1: medv ~ poly(lstat, 5)
## Model 2: medv ~ poly(lstat, 6)
##   Res.Df   RSS Df Sum of Sq      F Pr(>F)
## 1    500 13597
## 2    499 13555  1    42.364 1.5596 0.2123
```

```
#Already tried out log-transformation earlier so will just move on
```

## Qualitative Predictors

```
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
```

```r
#Initial model
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
#check code used for the dummy variables
attach(Carseats)
```

```
## The following objects are masked from Carseats (pos = 5):
##
##     Advertising, Age, CompPrice, Education, Income, Population, Price, Sales, ShelveLoc, Urban, US

## The following objects are masked from Carseats (pos = 8):
##
##     Advertising, Age, CompPrice, Education, Income, Population, Price, Sales, ShelveLoc, Urban, US

## The following objects are masked from Carseats (pos = 11):
##
##     Advertising, Age, CompPrice, Education, Income, Population, Price, Sales, ShelveLoc, Urban, US
```

```r
contrasts(ShelveLoc) #creates two new predictors: shelvelocgood and
```

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

```r
#shelvelocmedium --> 3 valid binary combinations (0,0) (1,0) (0,1)
```

### Lab is Complete!!