#3.6 Lab: Linear Regression

#3.6.1

In [1]:
```
library(MASS)
library(ISLR)
```

In [4]:
```
attach(Boston)
#To access the Boston data from the MASS package
```

In [5]:
```
head(Boston)
```

crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	black	lstat	medv
0.00632	18	2.31	0	0.538	6.575	65.2	4.0900	1	296	15.3	396.90	4.98	24.0
0.02731	0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	396.90	9.14	21.6
0.02729	0	7.07	0	0.469	7.185	61.1	4.9671	2	242	17.8	392.83	4.03	34.7
0.03237	0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	2.94	33.4
0.06905	0	2.18	0	0.458	7.147	54.2	6.0622	3	222	18.7	396.90	5.33	36.2
0.02985	0	2.18	0	0.458	6.430	58.7	6.0622	3	222	18.7	394.12	5.21	28.7

In [7]:
```
names(Boston)
```

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

In [12]:
```
#simple linear fit
lm.fit = lm(medv~lstat)
```

In [13]:
```
lm.fit = lm(medv~lstat, data = Boston)
```

In [11]:
```
lm.fit
```

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

Coefficients:
(Intercept)        lstat
      34.55        -0.95
```

In [14]:
```
names(lm.fit)
```

'coefficients' 'residuals' 'effects' 'rank' 'fitted.values' 'assign' 'qr' 'df.residual' 'xlevels' 'call' 'terms' 'model'

In [15]:
```
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
```

In [16]:
```
coef(lm.fit)
```

(Intercept)	34.5538408793831
lstat	-0.950049353757991

In [17]:
```
confint(lm.fit)
```

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

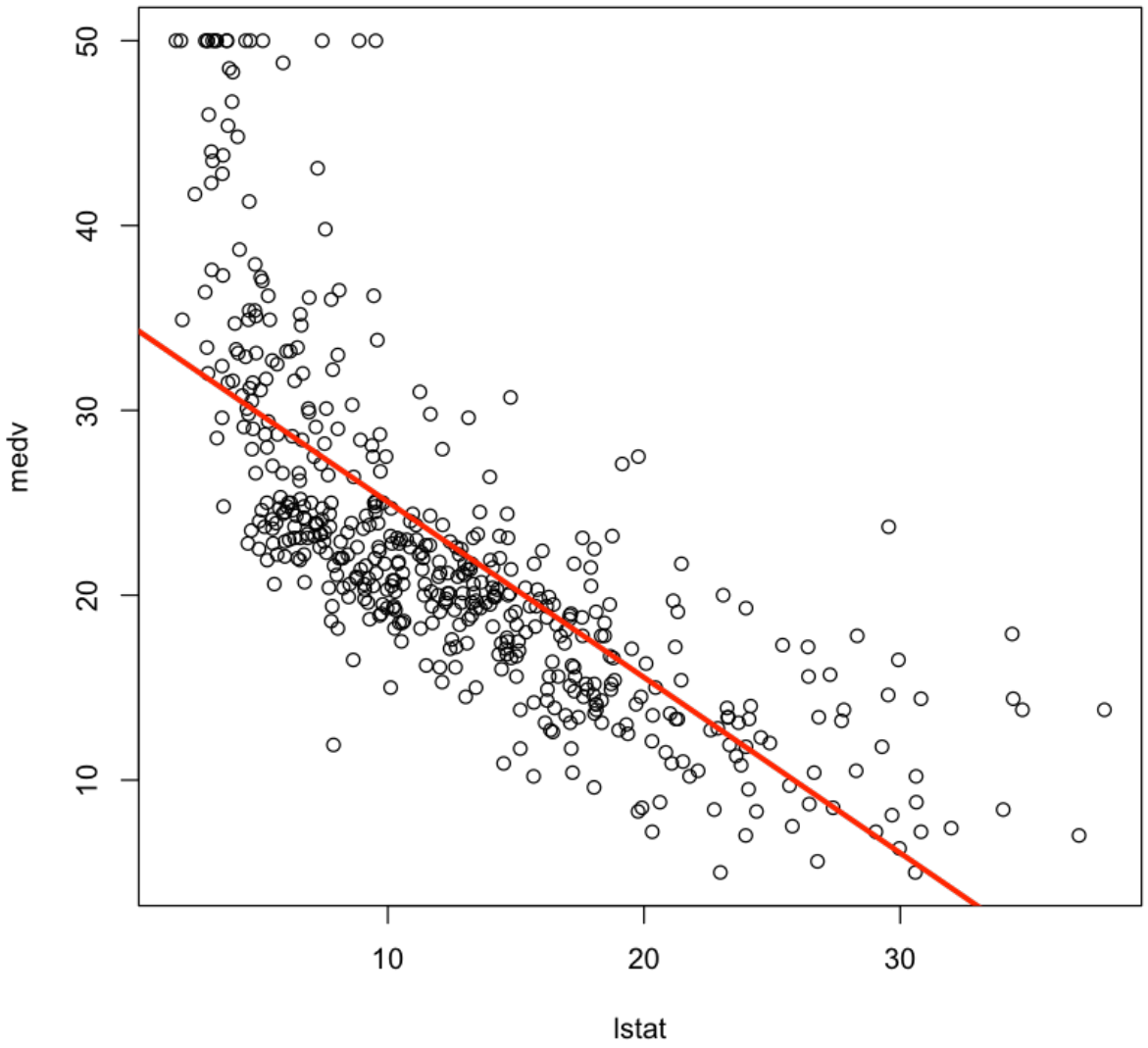
In [19]:
```
predict(lm.fit,data.frame(lstat=c(5,10,15)), interval = "confidenc

#predict()used to produce confidence intervals and prediction inte
#for the prediction of medv for a given value of lstat.
```

fit	lwr	upr
29.80359	29.00741	30.59978
25.05335	24.47413	25.63256
20.30310	19.73159	20.87461

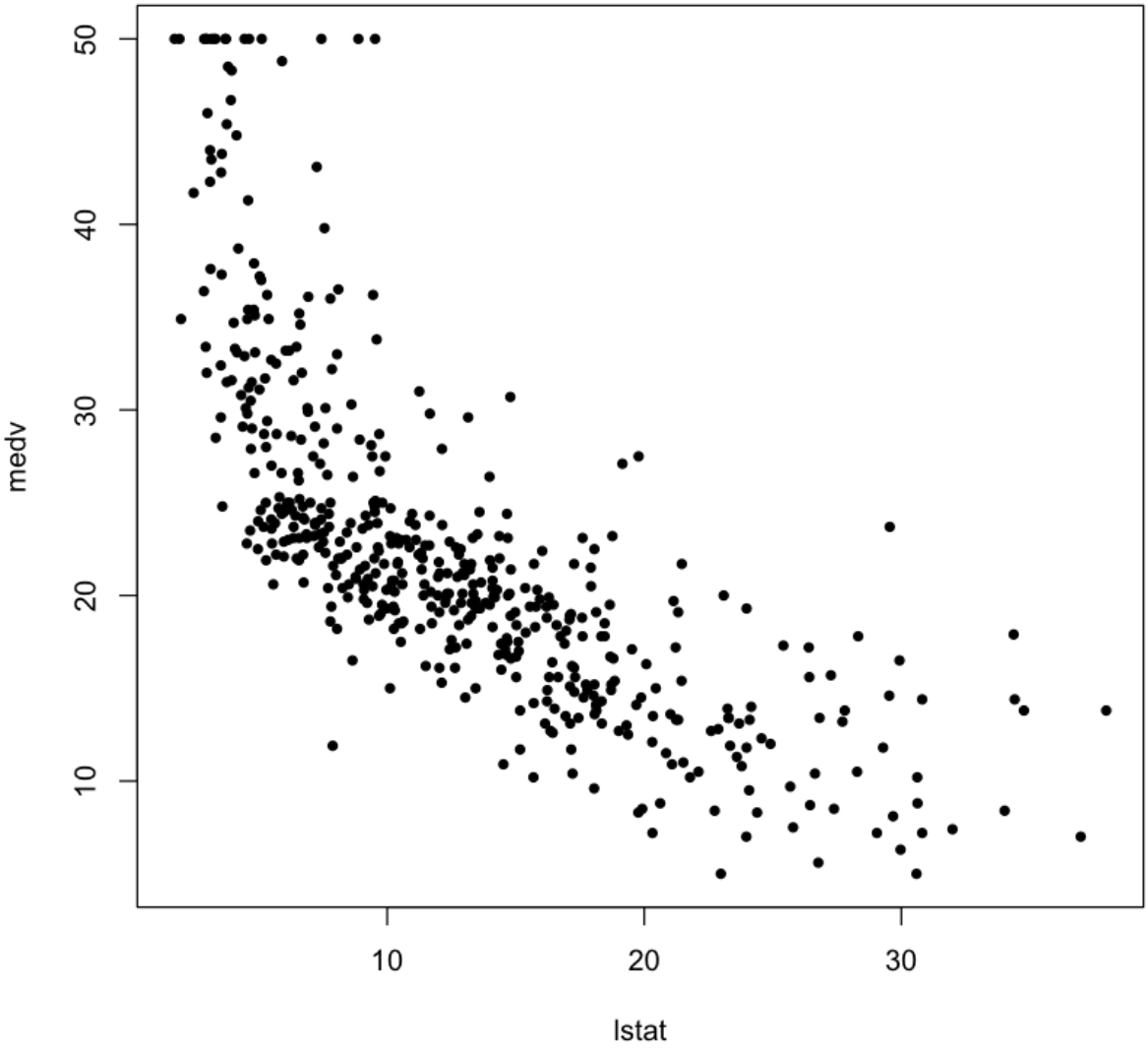
In [20]:
```
predict(lm.fit,data.frame(lstat=c(5,10,15)), interval = "predictic

```

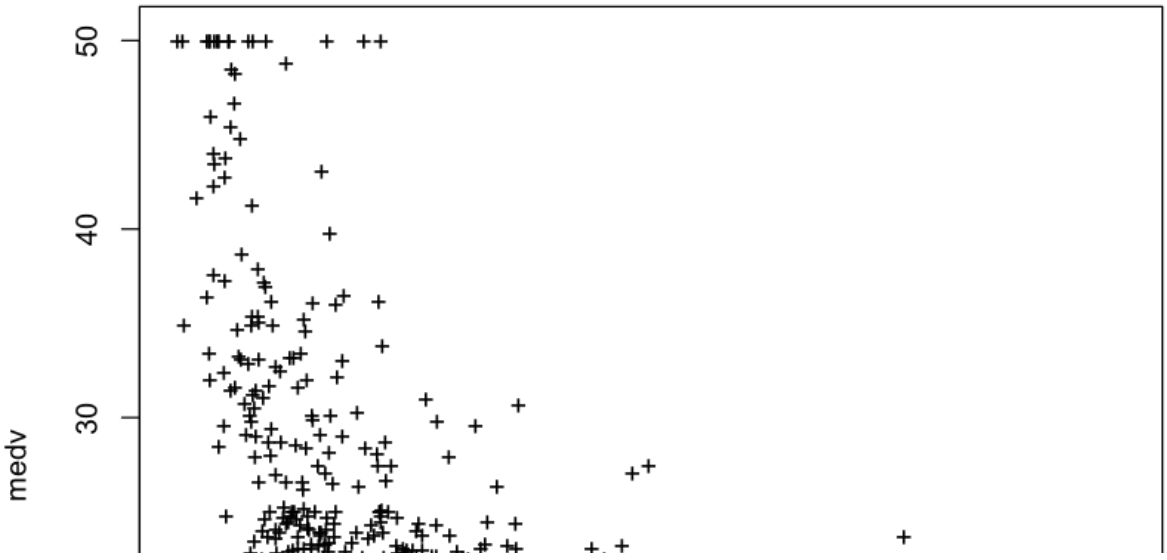
fit	lwr	upr
29.80359	17.565675	42.04151
25.05335	12.827626	37.27907
20.30310	8.077742	32.52846

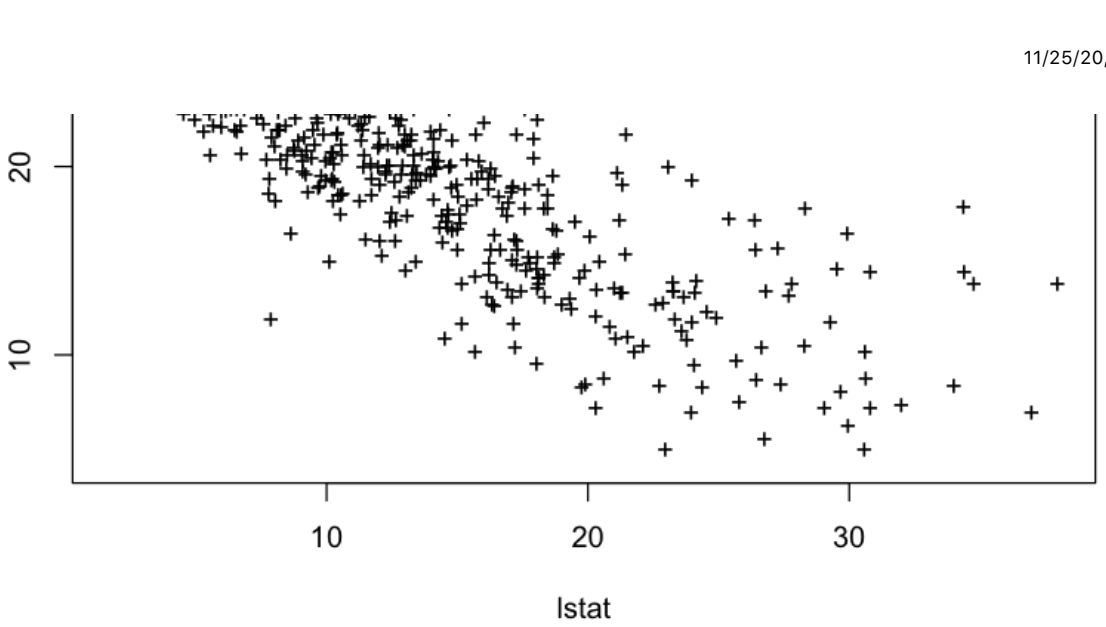
In [25]:
```
plot(lstat ,medv)
abline(lm.fit)
abline(lm.fit,lwd = 3, col="red")
#abline() draws a line with intercept a and slope b,
#The lwd=3 command causes the width of the regression line
#to be increased by a factor of 3
```



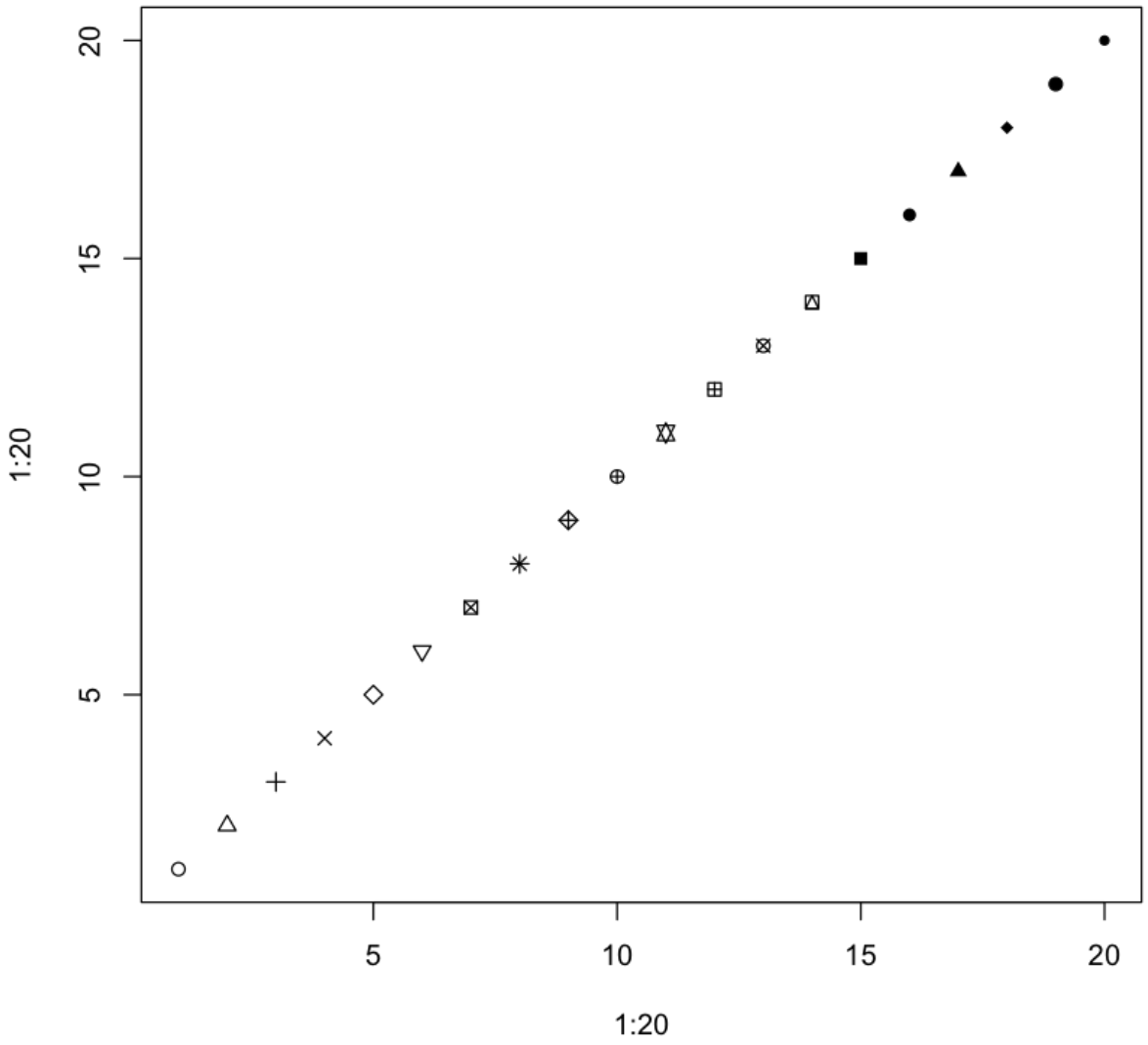
In [27]:
```
plot(lstat,medv,pch=20)
plot(lstat,medv,pch="+")
#pch used to create different plotting symbols.
```


medv
lstat
10
20
30
40
50

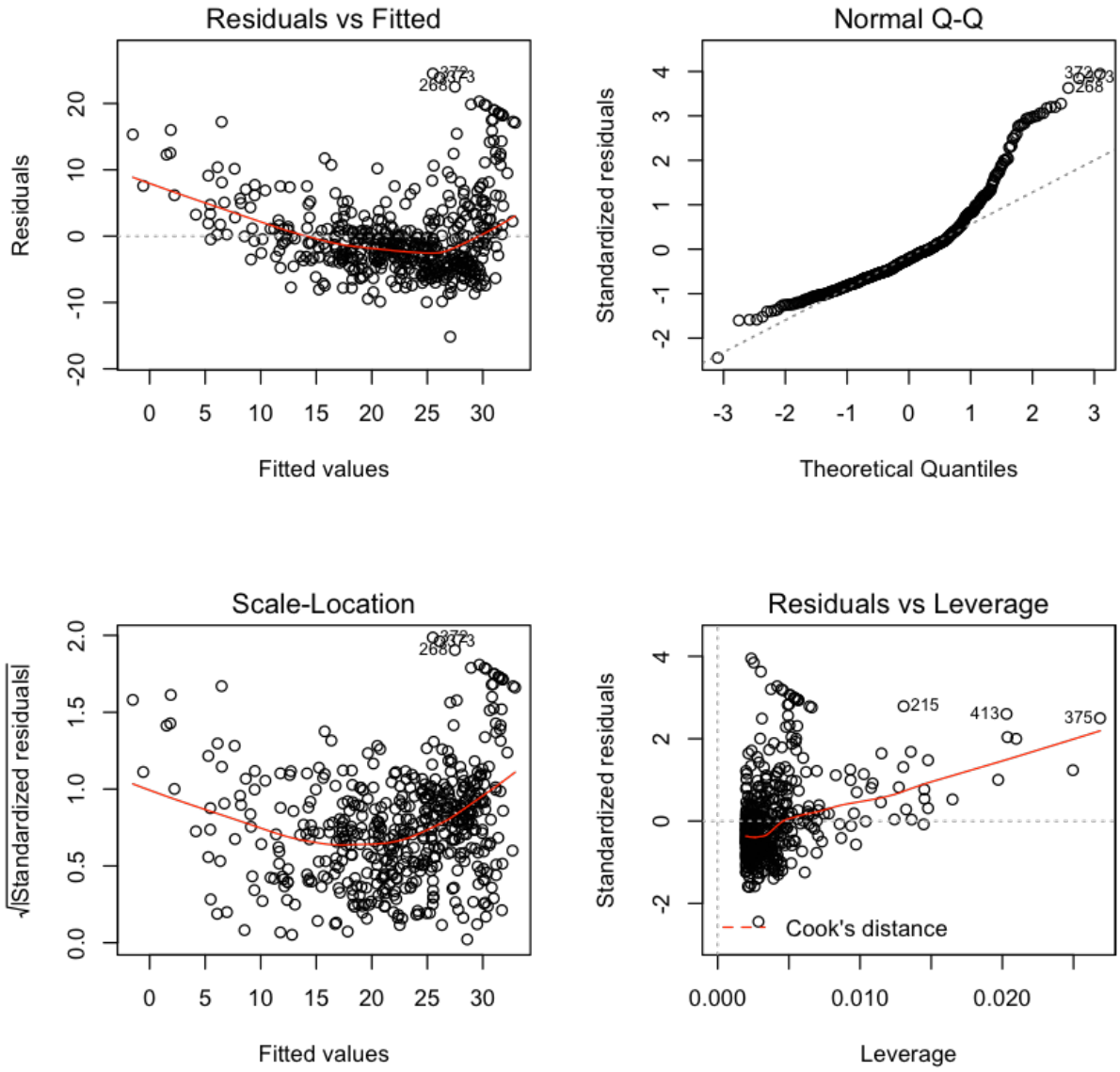

medv
30
40
50


20
10
10
20
30
lstat

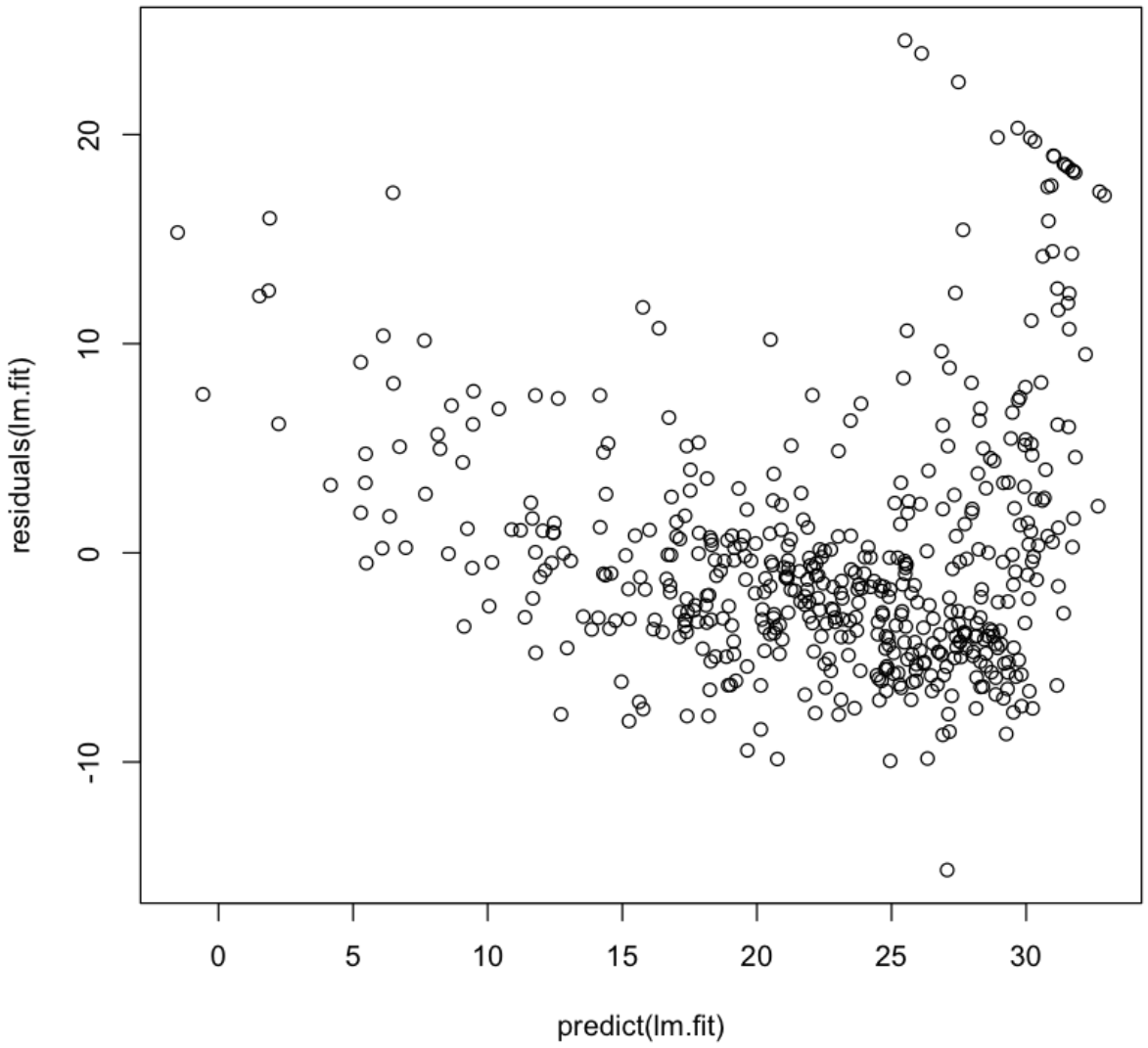
In [28]:
```
1 plot(1:20,1:20,pch=1:20)
```



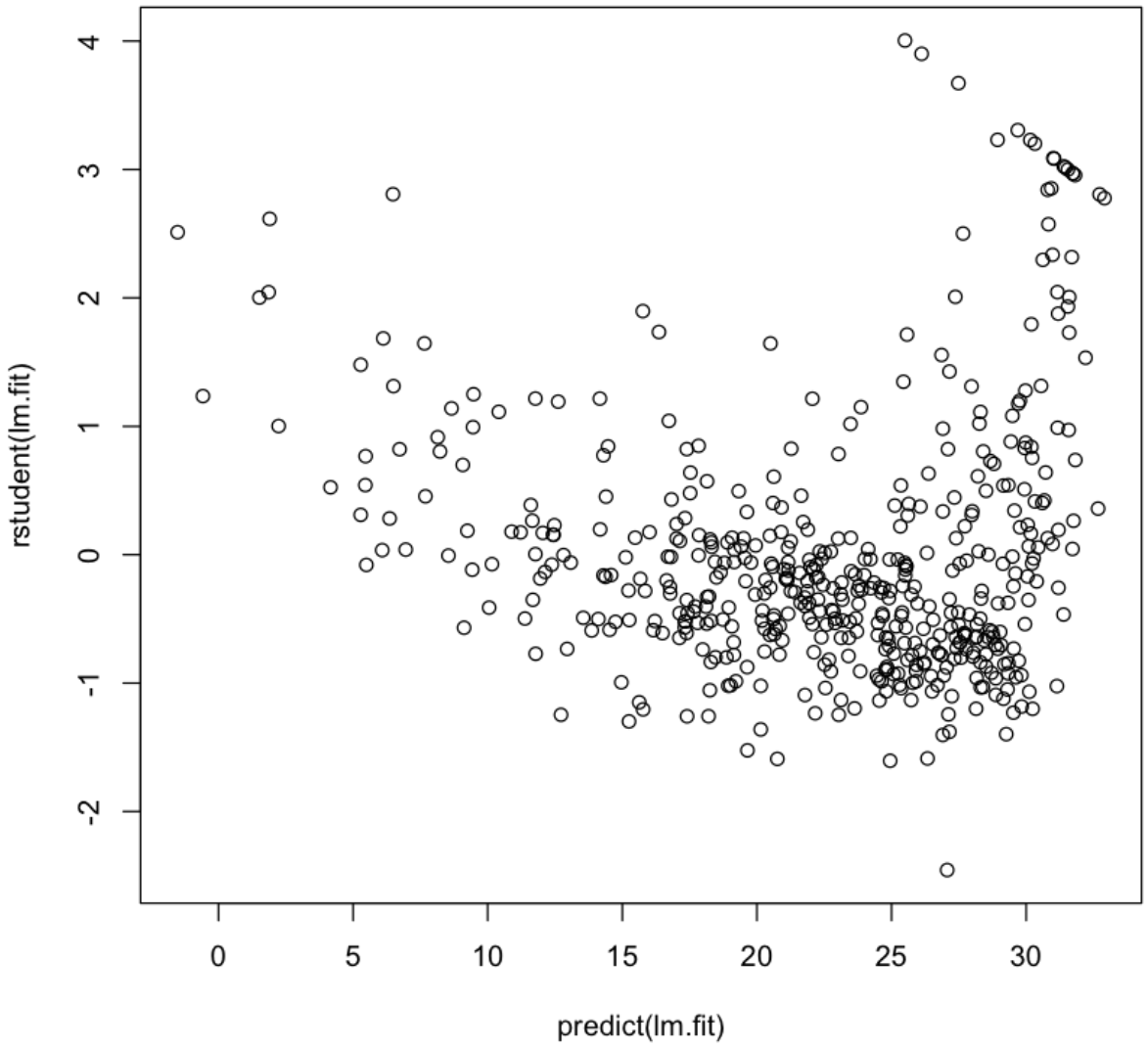
In [29]:
```
par(mfrow=c(2,2))
plot(lm.fit)
```



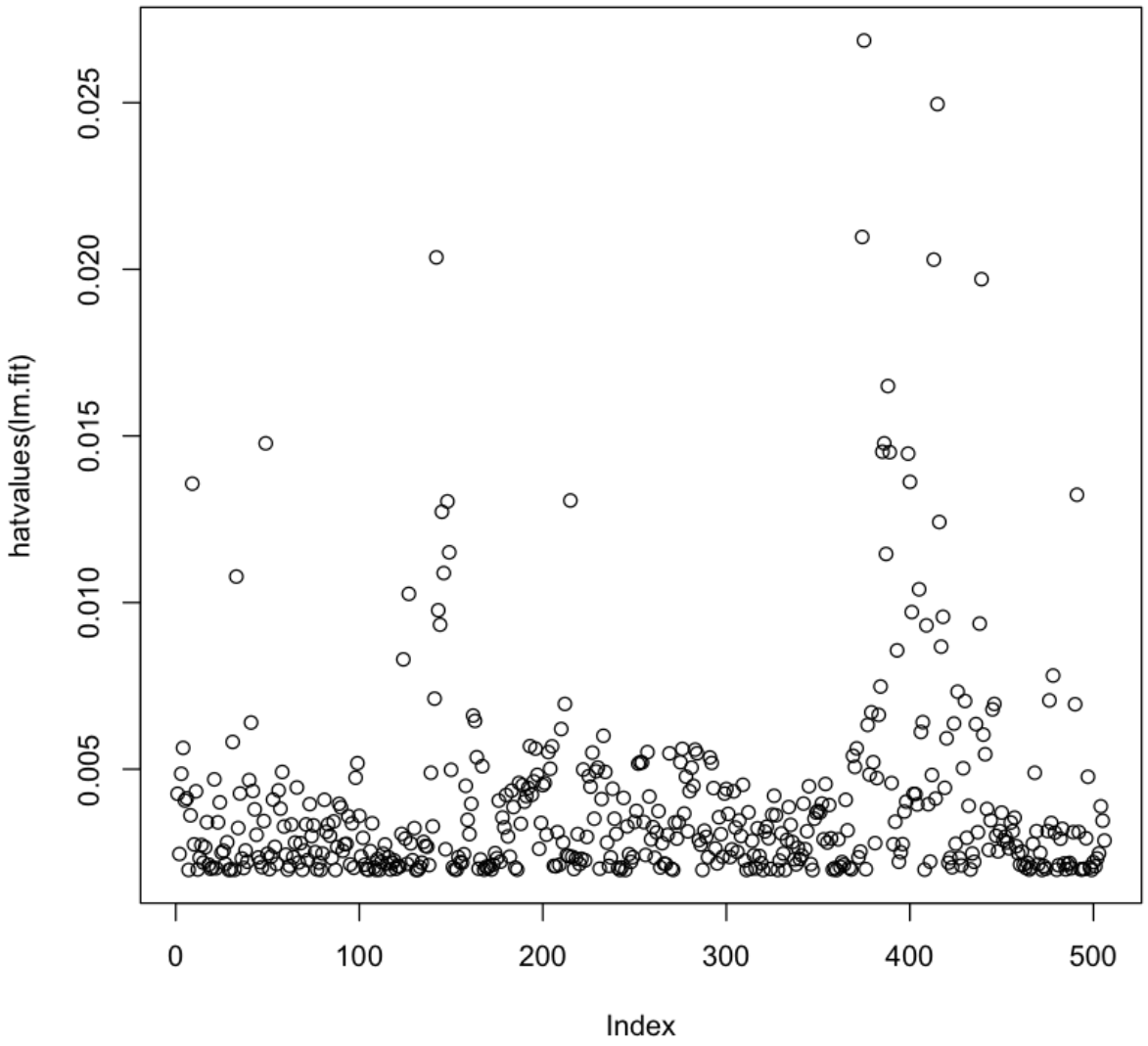
In [30]:
```
1 plot(predict(lm.fit), residuals(lm.fit))
```



In [31]:
```
plot(predict(lm.fit), rstudent(lm.fit))

#residuals and studentized residuals plots can be ploted
#from linear model using residuals() and rstudent()
```



In [32]:
```
plot(hatvalues(lm.fit))
```



In [33]:
```
which.max(hatvalues (lm.fit))
#identifies the index of the largest element of a vector. it tells
#which observation has the largest leverage statistic.
```

375: 375

#3.6.3 Multiple Regression

In [37]:
```
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
```

In [38]:
```
lm.fit <- lm(medv ~ ., data = Boston)
summary(lm.fit)

# . can be add all variables as predictors
```

```
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
```

In [39]:
```
?summary.lm
```

In [62]:
```
summary(lm.fit)$r.sq
#R^2
```

0.740642664109409

In [63]:
```
summary(lm.fit)$sigma
#RSE
```

4.74529818169963

In [121]:
```
#library(ISLR)

install.packages("ISLR")
```

```
Updating HTML index of packages in '.Library'
Making 'packages.html' ... done
```

In [123]:
```
#install.packages("car")
```

In [104]:
```

#library(car)
#vif(lm.fit)
```

In [68]:
```
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
```

In [69]:
```
lm.fit1=update(lm.fit, ~.-age)
```

In [77]:
```
#Interactions
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
```

In [79]:
```
#Non-linear Transformations of the Predictors

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
```

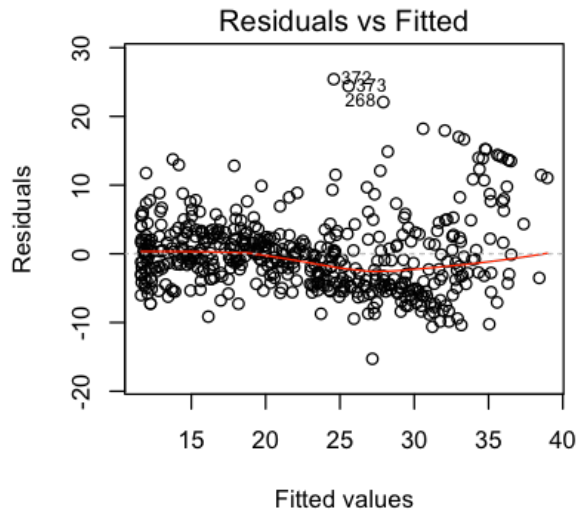
In [ ]:
```
#The near-zero p-value associated with the quadratic term
#suggests that it leads to an improved model.
```

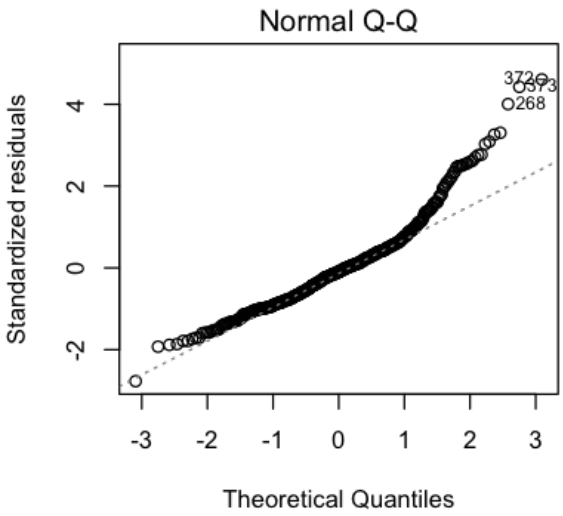
In [80]:
```
#We use the anova() function to further quantify the extent
#to which the quadratic fit is superior to the linear fit
lm.fit=lm(medv~lstat)
anova(lm.fit ,lm.fit2)
```

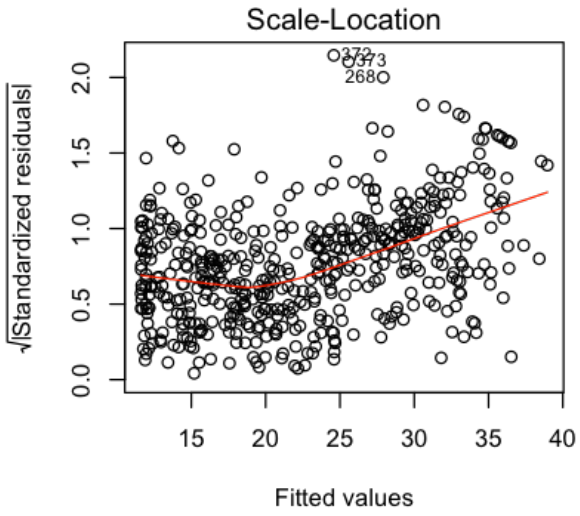
Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
504	19472.38	NA	NA	NA	NA
503	15347.24	1	4125.138	135.1998	7.630116e-28

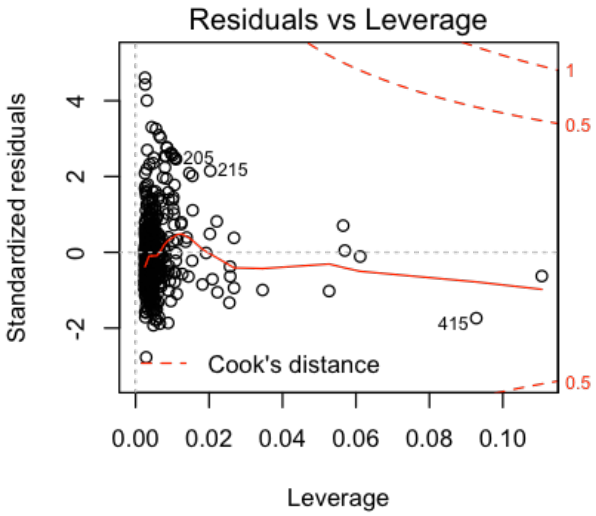
In [ ]:
```
#Model 1 containing only one predictor-lstat
#Model 2 is larger quadratic model that has two 2 predictors:lstat
#The anova() performs a hypothesis test comparing the two models.
#both the models fit the data equally well,
#in model 2, F is 135, and p-value is near to zero, so model 2
#is superior than Model 1
```

In [81]:
```
par(mfrow=c(2,2)) > plot(lm.fit2)

# with lstat2 included in the model, there is
#little noticible pattern in the residuals.
```









In [92]:
```
#cubic fit
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
```

In [94]:
```
lm.fit5=lm(medv~poly(lstat,7))
summary(lm.fit5)
```

```
Call:
lm(formula = medv ~ poly(lstat, 7))

Residuals:
     Min       1Q   Median       3Q      Max 
-14.3746  -3.1382  -0.7072   2.0646  26.9642 

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)    
(Intercept)       22.5328     0.2319  97.168  < 2e-16 ***
poly(lstat, 7)1 -152.4595     5.2164 -29.227  < 2e-16 ***
poly(lstat, 7)2   64.2272     5.2164  12.313  < 2e-16 ***
poly(lstat, 7)3  -27.0511     5.2164  -5.186 3.13e-07 ***
poly(lstat, 7)4   25.4517     5.2164   4.879 1.44e-06 ***
poly(lstat, 7)5  -19.2524     5.2164  -3.691 0.000248 ***
poly(lstat, 7)6    6.5088     5.2164   1.248 0.212708    
poly(lstat, 7)7    1.9416     5.2164   0.372 0.709888    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.216 on 498 degrees of freedom
Multiple R-squared:  0.6828,    Adjusted R-squared:  0.6783 
F-statistic: 153.1 on 7 and 498 DF,  p-value: < 2.2e-16
```

In [87]:
```
?lm.fit
```

In [ ]:
```
#including additional polynomial terms, up to fifth order,
#leads to an improvement in the model fit!
#no polynomial terms beyond fifth order have
#significant p-values in a regression fit.
```

#3.6.6 Qualitative predictors

In [101]:
```
library(ISLR)

```

In [102]:
```
attach(Carseats)
```

In [106]:
```
head(Carseats)
```

Sales	CompPrice	Income	Advertising	Population	Price	ShelveLoc	Age	Education	Urban
9.50	138	73	11	276	120	Bad	42	17	Yes
11.22	111	48	16	260	83	Good	65	10	Yes
10.06	113	35	10	269	80	Medium	59	12	Yes
7.40	117	100	4	466	97	Medium	55	14	Yes
4.15	141	64	3	340	128	Bad	38	13	Yes
10.81	124	113	13	501	72	Bad	78	16	No

In [107]:
```
names(Carseats)
```

'Sales' 'CompPrice' 'Income' 'Advertising' 'Population' 'Price' 'ShelveLoc' 'Age' 'Education' 'Urban' 'US'

In [ ]:
```
# Shelveloc is qualitative data, bad-median-good
#R generates dummy variables automatically
```

In [108]:
```
lm.fit=lm(Sales~.+Income:Advertising+Price:Age,data=Carseats)
summary(lm.fit)

```

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

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
```

In [109]:
```
attach(Carseats)
contrasts (ShelveLoc )
```

```
The following objects are masked from Carseats (pos = 3):

    Advertising, Age, CompPrice, Education, Income, Population, Price
,
    Sales, ShelveLoc, US, Urban
```

	Good	Medium
Bad	0	0
Good	1	0
Medium	0	1

In [110]:
```
# Shelveloc is qualitative data, bad-median-good
#R generates dummy variables automatically
?contrasts
```

3.6.7 Writing functions

In [118]:
```
LoadLibraries = function() {
    library(ISLR)
    library(MASS)
    print("The libraries have been loaded.")
}
```

In [120]:
```
LoadLibraries()
```

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