ST443 Lab02

07 October 2020

lab 2.1 Install and activate packages

```
# install.packages("MASS")
# install.packages("ISLR")

library(MASS)
library(ISLR)
```

lab 2.2 Simple Linear Regression

```
# fix(Boston)
names(Boston)
```

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

```
# ?Boston
# lm.fit=lm(medv~lstat)
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
```

```
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"
##  [5] "fitted.values" "assign"        "qr"            "df.residual"
##  [9] "xlevels"       "call"          "terms"         "model"
```

```
lm.fit$coefficients
```

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

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

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

lab 2.3 Inference

Confidence intervals for the coefficient estimates

```
confint(lm.fit)
```

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

Preditive values, confidence intervals and prediction intervals for the prediction of medv for a given value of lstat $CI(\hat{y}_i) = CI(\hat{\beta}_0 + x_i \cdot \hat{\beta}_1)$, $PI(\hat{y}_i) = CI(\hat{\beta}_0 + x_i \cdot \hat{\beta}_1 + e_i)$,

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

```
##        1        2        3
## 29.80359 25.05335 20.30310
```

```
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")
```

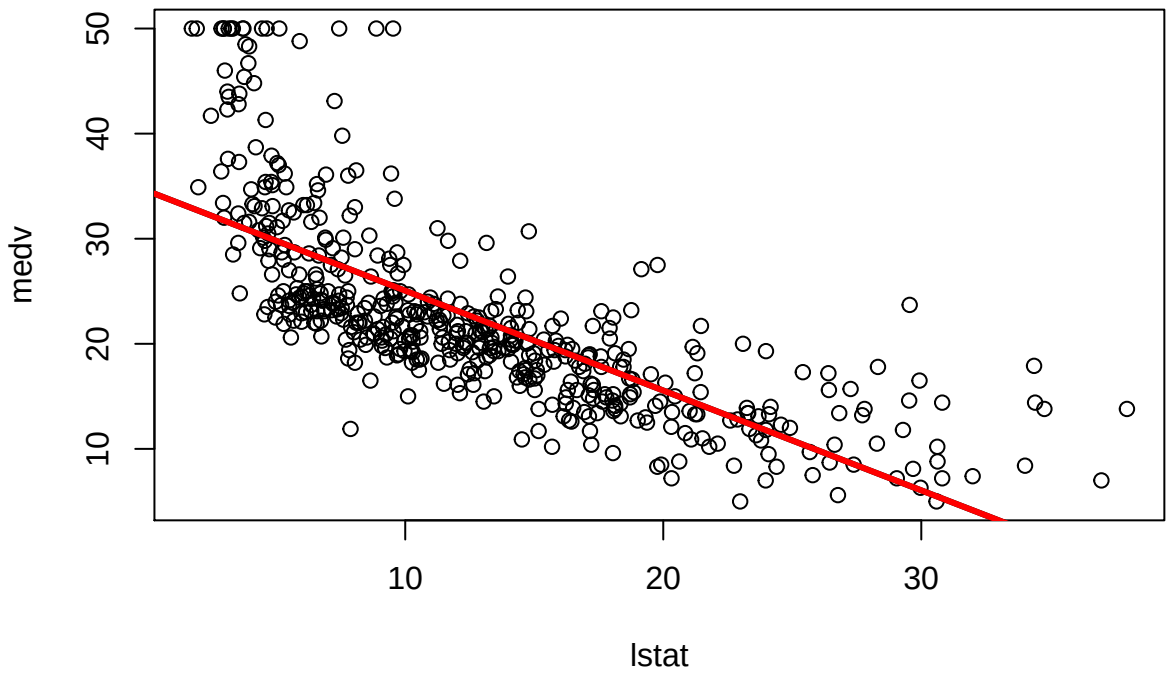
```
##        fit       lwr      upr
## 1 29.80359 17.565675 42.04151
## 2 25.05335 12.827626 37.27907
## 3 20.30310  8.077742 32.52846
```

lab 2.4 Diagnostic

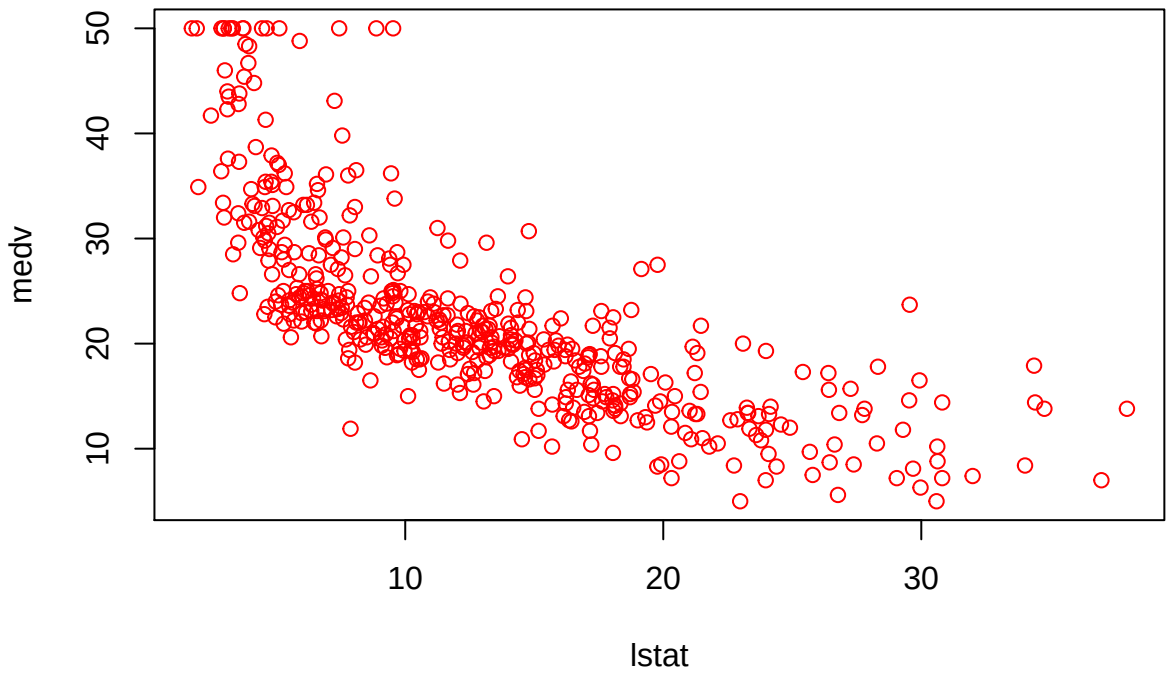
"NICE": normality, independent, constant variance, $E(e) = 0$

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

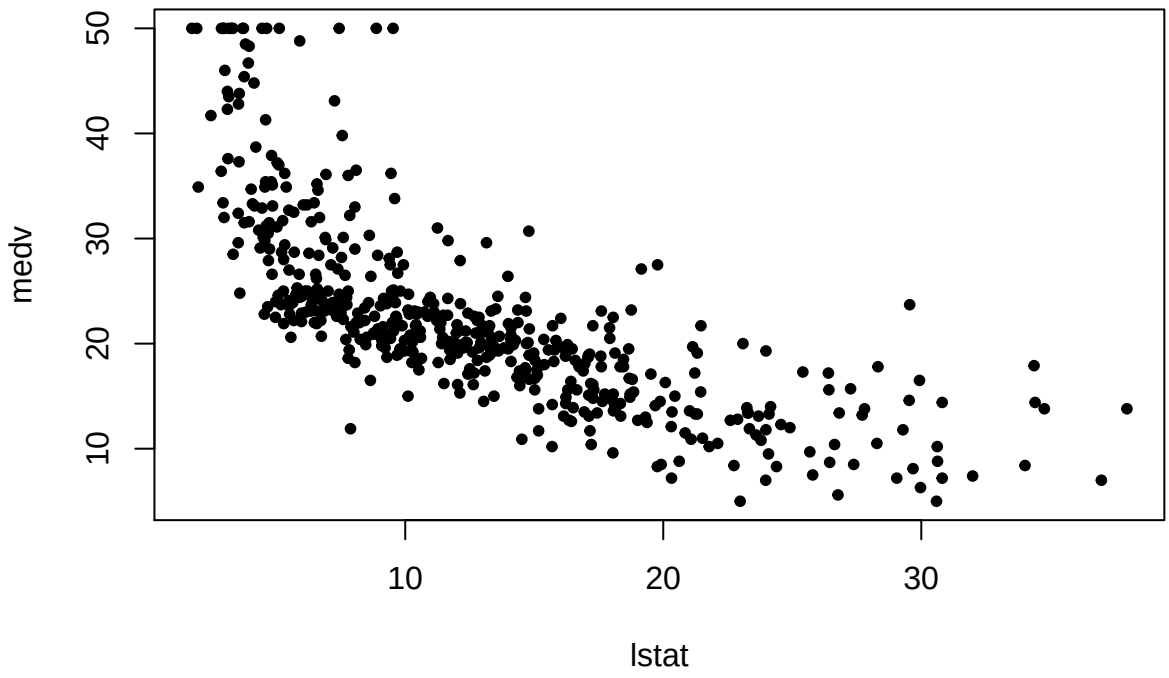
```
abline(lm.fit, lwd = 3)
abline(lm.fit, lwd = 3, col = "red")
```



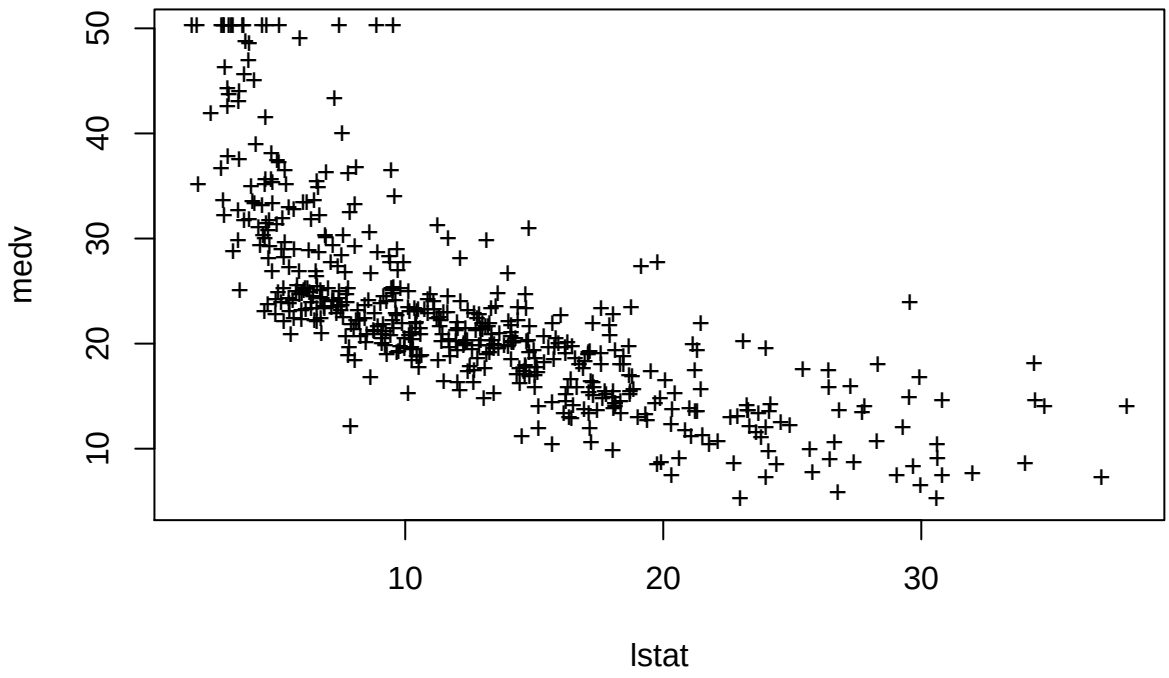

```
plot(lstat, medv, col = "red")
```



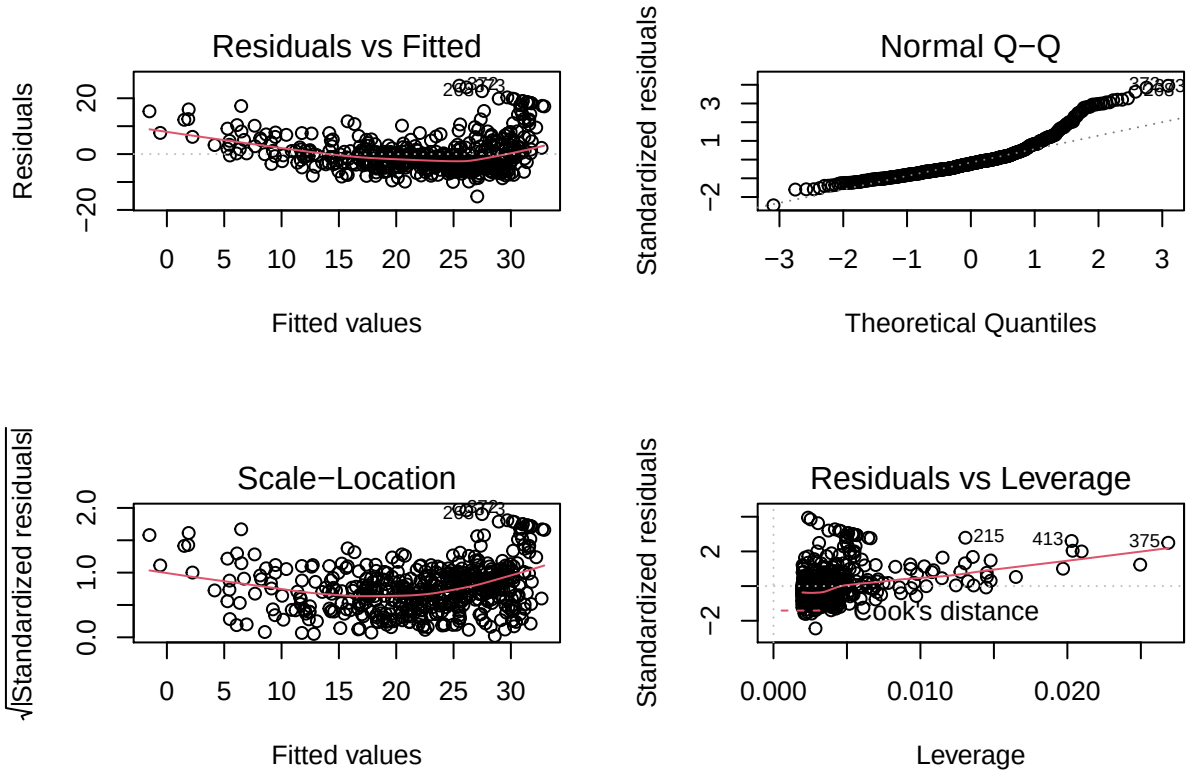

```
plot(lstat, medv, pch = 20)
```




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



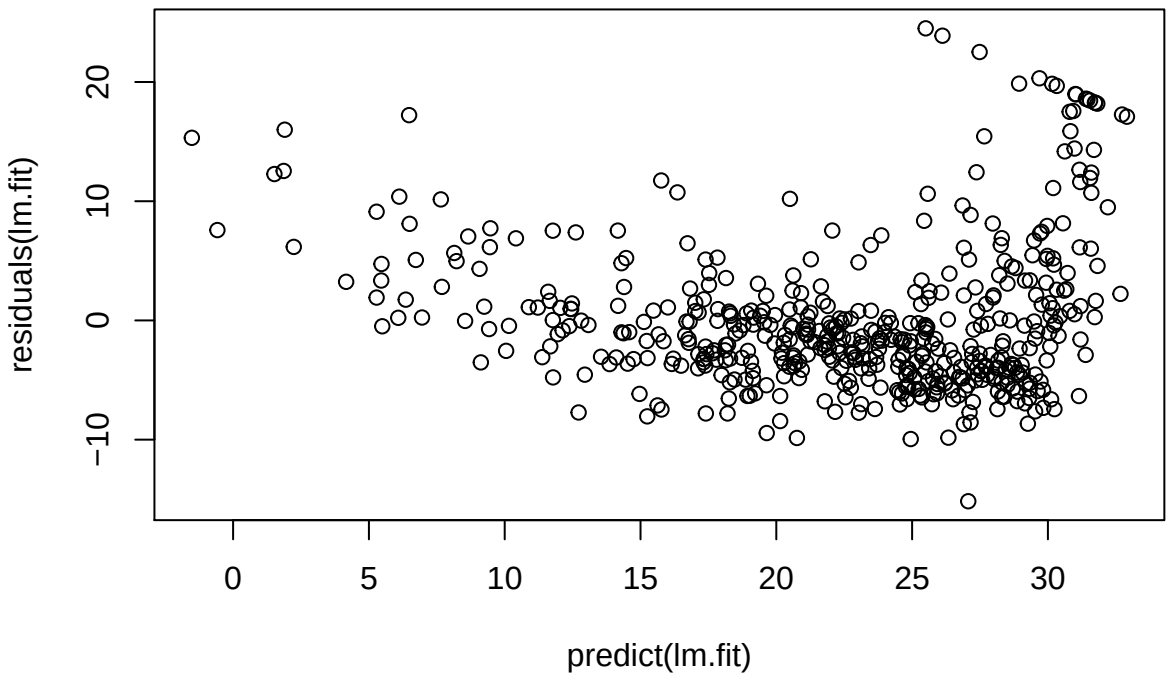

```
par(mfrow = c(2, 2))
plot(lm.fit)
```



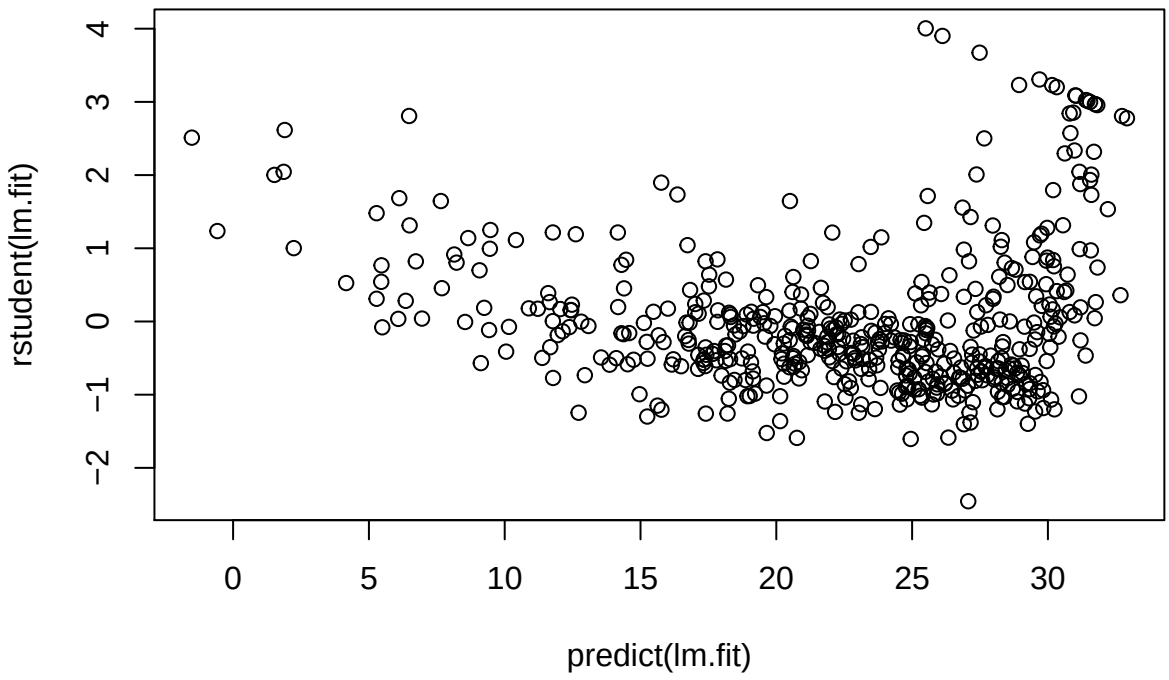

```
par(mfrow = c(1, 1))
```

plot of fitted values vs (standardized) residuals

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

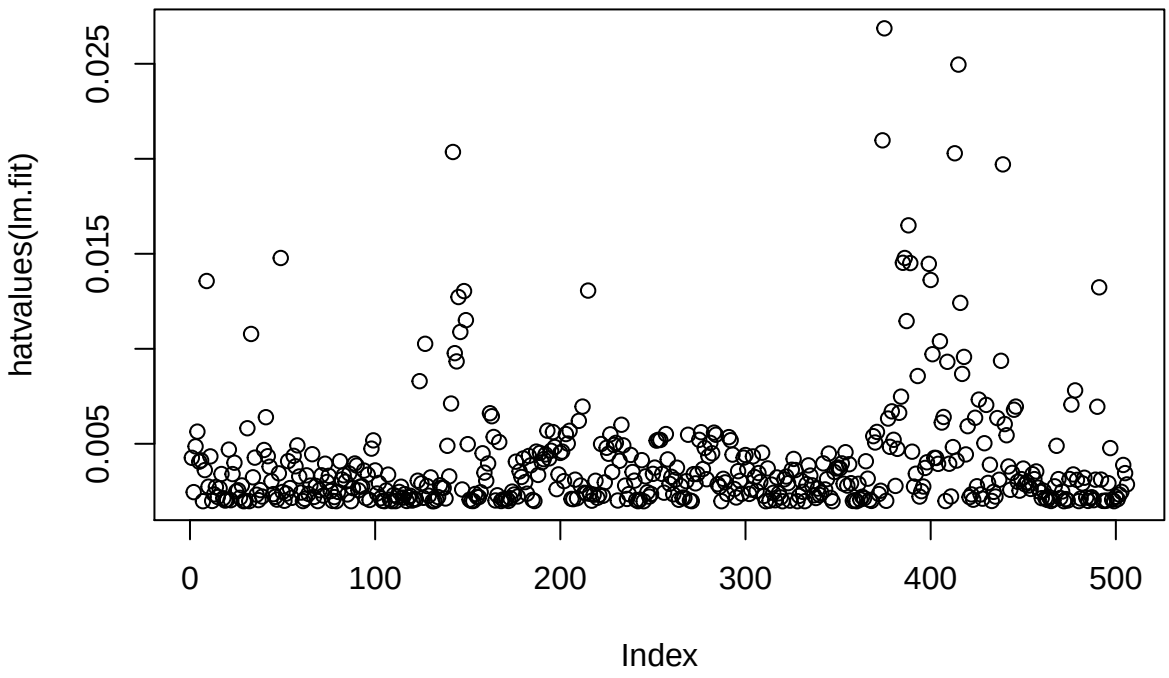



```
plot(predict(lm.fit), rstudent(lm.fit))
```



plot of leverage statistics

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




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

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

lab 2.5 Multiple Linear Regression

```
lm.fit1 = lm(medv ~ lstat + age, data = Boston)
summary(lm.fit1)
```

```
##
## 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.fit2 = lm(medv ~ ., data = Boston)
summary(lm.fit2)
```

```
##
## 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
```

```
lm.fit3 = lm(medv ~ . - age, data = Boston)
summary(lm.fit3)
```

```
##
## 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
```

```
lm.fit4 <- lm(medv ~ lstat + age + tax + rad, data = Boston)
summary(lm.fit4)
```

```
##
## Call:
## lm(formula = medv ~ lstat + age + tax + rad, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -16.251  -3.685  -1.096   1.745  24.266
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 36.926839   1.071869  34.451  < 2e-16 ***
## lstat       -0.966093   0.050274 -19.217  < 2e-16 ***
## age          0.046880   0.012410   3.778 0.000177 ***
## tax         -0.019054   0.004042  -4.715 3.14e-06 ***
## rad          0.250685   0.074542   3.363 0.000830 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.039 on 501 degrees of freedom
## Multiple R-squared:  0.5723, Adjusted R-squared:  0.5689
## F-statistic: 167.6 on 4 and 501 DF,  p-value: < 2.2e-16
```

```
anova(lm.fit1, lm.fit4)  ## F test, anova() function performs a hypothesis test comparing the two model
```

```
## Analysis of Variance Table
##
## Model 1: medv ~ lstat + age
## Model 2: medv ~ lstat + age + tax + rad
##   Res.Df   RSS Df Sum of Sq      F    Pr(>F)
## 1    503 19168
## 2    501 18271  2    897.33 12.303 6.082e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Interaction term :.

eg.: `ptratio` and `ptratio` are not excluded from the model if using `*`. Instead, `:` should be used:

```r
summary(lm(medv ~ . - age + ptratio*tax, data = Boston))
```

```
##
## Call:
## lm(formula = medv ~ . - age + ptratio * tax, data = Boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.5027  -2.7485  -0.5454   1.7144  25.9285
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  58.431446  11.188593   5.222 2.61e-07 ***
## crim         -0.106156   0.032715  -3.245 0.001255 **
## zn            0.039447   0.013916   2.835 0.004775 **
## indus         0.021373   0.061196   0.349 0.727045
## chas          2.760781   0.856872   3.222 0.001358 **
## nox         -15.803052   3.766099  -4.196 3.22e-05 ***
## rm            3.639830   0.414527   8.781  < 2e-16 ***
## dis          -1.375158   0.195585  -7.031 6.89e-12 ***
## rad           0.230368   0.074193   3.105 0.002013 **
## tax          -0.075807   0.029042  -2.610 0.009323 **
## ptratio      -2.098685   0.536099  -3.915 0.000103 ***
## black         0.008764   0.002679   3.271 0.001147 **
## lstat        -0.543317   0.048255 -11.259  < 2e-16 ***
## tax:ptratio   0.003369   0.001528   2.204 0.027979 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.722 on 492 degrees of freedom
## Multiple R-squared:  0.7432, Adjusted R-squared:  0.7364
## F-statistic: 109.5 on 13 and 492 DF,  p-value: < 2.2e-16
```

```r
summary(lm(medv ~ . - age - ptratio - tax + ptratio*tax, data = Boston)) # ptratio and tax are not excl
```

```
##
## Call:
## lm(formula = medv ~ . - age - ptratio - tax + ptratio * tax,
##     data = Boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.5027  -2.7485  -0.5454   1.7144  25.9285
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  58.431446  11.188593   5.222 2.61e-07 ***
## crim         -0.106156   0.032715  -3.245 0.001255 **
## zn            0.039447   0.013916   2.835 0.004775 **
## indus         0.021373   0.061196   0.349 0.727045
## chas          2.760781   0.856872   3.222 0.001358 **
## nox         -15.803052   3.766099  -4.196 3.22e-05 ***
## rm            3.639830   0.414527   8.781  < 2e-16 ***
## dis          -1.375158   0.195585  -7.031 6.89e-12 ***
```

```
## rad           0.230368   0.074193   3.105 0.002013 **
## black         0.008764   0.002679   3.271 0.001147 **
## lstat        -0.543317   0.048255 -11.259  < 2e-16 ***
## ptratio      -2.098685   0.536099  -3.915 0.000103 ***
## tax          -0.075807   0.029042  -2.610 0.009323 **
## tax:ptratio   0.003369   0.001528   2.204 0.027979 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.722 on 492 degrees of freedom
## Multiple R-squared:  0.7432, Adjusted R-squared:  0.7364
## F-statistic: 109.5 on 13 and 492 DF,  p-value: < 2.2e-16
```

```
summary(lm(medv ~ . - age - ptratio - tax + ptratio:tax, data = Boston)) # ptratio and tax are excluded
```

```
##
## Call:
## lm(formula = medv ~ . - age - ptratio - tax + ptratio:tax, data = Boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.7970  -2.8864  -0.7284   1.8605  26.5237
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.725e+01  4.213e+00   4.095 4.94e-05 ***
## crim        -1.064e-01  3.368e-02  -3.159 0.001682 **
## zn           7.318e-02  1.295e-02   5.652 2.69e-08 ***
## indus        3.598e-02  6.270e-02   0.574 0.566289
## chas         2.807e+00  8.821e-01   3.183 0.001552 **
## nox         -1.265e+01  3.593e+00  -3.521 0.000469 ***
## rm           4.077e+00  4.160e-01   9.800  < 2e-16 ***
## dis         -1.569e+00  1.950e-01  -8.046 6.40e-15 ***
## rad          3.750e-01  6.945e-02   5.399 1.04e-07 ***
## black        8.931e-03  2.744e-03   3.254 0.001216 **
## lstat       -5.338e-01  4.879e-02 -10.941  < 2e-16 ***
## tax:ptratio -1.093e-03  1.742e-04  -6.274 7.70e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.863 on 494 degrees of freedom
## Multiple R-squared:  0.7265, Adjusted R-squared:  0.7204
## F-statistic: 119.3 on 11 and 494 DF,  p-value: < 2.2e-16
```

Variables Transformation

Square transformation of `rm` is not recognized

```
summary(lm(medv ~ . - age + rm ^ 2,    data = Boston))
```

```
##
## Call:
## lm(formula = medv ~ . - age + rm^2, 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
```

You need `I()` to isolate the transformation. In other words, `I()` inhibits the interpretation of operators such as "+", "-", "*" and "^" as formula operators, so they are used as arithmetical operators:

```
summary(lm(medv ~ . - age + I(rm ^ 2), data = Boston))
```

```
##
## Call:
## lm(formula = medv ~ . - age + I(rm^2), data = Boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -28.2492  -2.2673  -0.3692   1.5237  26.8484
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 136.274604   9.119606  14.943  < 2e-16 ***
## crim         -0.129963   0.028672  -4.533 7.32e-06 ***
## zn            0.034566   0.011904   2.904 0.003852 **
## indus         0.069857   0.053695   1.301 0.193872
## chas          2.340211   0.749815   3.121 0.001908 **
## nox         -17.926116   3.207239  -5.589 3.78e-08 ***
## rm          -29.003535   2.644680 -10.967  < 2e-16 ***
## dis          -1.126323   0.168517  -6.684 6.33e-11 ***
## rad           0.261065   0.057719   4.523 7.65e-06 ***
## tax          -0.011396   0.003274  -3.481 0.000545 ***
## ptratio      -0.757759   0.114632  -6.610 1.00e-10 ***
## black         0.007306   0.002340   3.123 0.001898 **
## lstat        -0.548148   0.041560 -13.189  < 2e-16 ***
## I(rm^2)       2.559777   0.204405  12.523  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 4.132 on 492 degrees of freedom
## Multiple R-squared:  0.8033, Adjusted R-squared:  0.7981
## F-statistic: 154.6 on 13 and 492 DF,  p-value: < 2.2e-16
```

It is OK to do log transformation without `I()`:

```
summary(lm(medv ~ . - age + log(rm), data = Boston))
```

```
##
## Call:
## lm(formula = medv ~ . - age + log(rm), data = Boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -25.9642  -2.3553  -0.2786   1.6199  27.0151
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.858e+02  1.292e+01  14.383  < 2e-16 ***
## crim        -1.291e-01  2.879e-02  -4.486 9.06e-06 ***
## zn           3.156e-02  1.197e-02   2.636 0.008657 **
## indus        7.350e-02  5.394e-02   1.363 0.173610
## chas         2.423e+00  7.526e-01   3.220 0.001367 **
## nox         -1.760e+01  3.220e+00  -5.464 7.40e-08 ***
## rm           3.366e+01  2.450e+00  13.740  < 2e-16 ***
## dis         -1.100e+00  1.696e-01  -6.487 2.14e-10 ***
## rad          2.496e-01  5.802e-02   4.301 2.05e-05 ***
## tax         -1.132e-02  3.287e-03  -3.443 0.000625 ***
## ptratio     -7.616e-01  1.151e-01  -6.618 9.54e-11 ***
## black        7.753e-03  2.347e-03   3.303 0.001025 **
## lstat       -5.246e-01  4.168e-02 -12.586  < 2e-16 ***
## log(rm)     -1.865e+02  1.514e+01 -12.315  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.149 on 492 degrees of freedom
## Multiple R-squared:  0.8017, Adjusted R-squared:  0.7965
## F-statistic: 153.1 on 13 and 492 DF,  p-value: < 2.2e-16
```

lab 2.6 Categorical Predictors

```
# ?Carseats
names(Carseats)
```

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

Predict sales (child car seat sales) in 400 locations on a number of predictors

```
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
```

ShelveLoc: an indicator of the quality of the shelving location, i.e. the space within a store in which the car seat is displayed at each location

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

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

```
contrasts(Urban)
```

```
##     Yes
## No    0
## Yes   1
```

lab 2.7 Writing Functions

```
f1 = function(x){
  d = median(x) - mean(x)
  return(d)
}
f1(x = c(1,2,6))
```

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

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

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

```r
LoadLibraries()
```

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