HW2

Mikhail Sidorov

1/25/2018

Contents

Task1

a) Show that assumptions satisfied

We will check criterias for model: $Z_i = u_i * \beta_0 + v_i * \beta_1 + \delta_i$. Assumtions we need to verify described in section 4.2 of the textbook (standard regression assumptions for multiple linear regression).

Assumption about form of the model derived directly from equation for Y_i

1) Error δ_i follow normal curve: if initial $p(\epsilon_i) \simeq N(0, \rho_i^2 * \sigma^2)$ than $p(\delta_i) = p(\frac{\epsilon_i}{\rho_i}) \simeq N(0, \sigma^2)$ This is the result of pdf for normal distribution $p(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$.
2) Mean of δ_i is still 0.
3) Homoscedasticity of errors: $Var(\delta_i) = \sigma^2$
4) Errors δ_i are still independent on each other because trasformation is local (in point i and independemt upon other points). So, if ϵ_i are iid, so are δ_i.
5) Assumptions about predictors also satisfied:

5.1) Predictotrs u_i and v_i are nonrandom

5.2) values u_i and v_i measured without error

5.3) u_i and v_i are independent of each other (because X_i and ρ_i are independent)

b) Find least squares estimation (LSE)

We need to find estimation: $\hat{\beta}_0$ and $\hat{\beta}_1$ for $Z_i = u_i\beta_0 + v_i\beta_1 + \delta_i$.

Below we can try 2 approaches: as a multivariate linear regression or using SSE minimization. Below we tryied both and results should be the same.

Multivariate regression approach

We can treat this as a special case of $Y = b_0 + b_1X_1 + b_2X_2$ multivariance regression where $b_0 = 0$ and $X_1 \equiv u_i$ and $b_1 \equiv \beta_0$ also $X_2 \equiv v_i$ and $b_2 \equiv \beta_1$ and use math for multivariance regression with all theoretical results.

In matrix form it will be:

$$\begin{bmatrix} Z_1 \\ ... \\ ... \\ Z_n \end{bmatrix} = \begin{bmatrix} u_1 & v_1 \\ ... & ... \\ ... & ... \\ u_n & v_n \end{bmatrix} * \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}$$

which we can write (in matrix form) as

$$Z = [UV] * \beta$$

and solution for β will be:

$$[UV]^T * Z = ([UV]^T * [UV]) * \beta$$
$$\beta = ([UV]^T * [UV])^{-1} * [UV]^T * Z$$

Alternative way (for excersice and to revise first principles)

Estimation of β_0 and β_1 will determine: $\hat{Z}_i = u_i\hat{\beta}_0 + v_i\hat{\beta}_1$

Following to LSE approach we need find minimum of $F(\beta_0, \beta_1) = \sum_{i=1}^{n}(Z_i - \hat{Z}_i)^2$ and it will determine $\hat{\beta}_0$ and $\hat{\beta}_1$.

Let us consider u, v and Z as n-dimensional vectors (n - number of observations). I.e. $u = (u_1, ..., u_n)$

Also we will use scalar multiplication of vectors $\langle a, b\rangle = \sum_{i=1}^{n} a_i * b_i$ (also note: $\langle a, b\rangle = \langle b, a\rangle$)

Equations to find min $F(\beta_0, \beta_1)$ are:

$$\frac{\partial F(\beta_0, \beta_1)}{\partial \beta_0} = -2\sum_{i=1}^{n} u_i(Z_i - u_i\beta_0 - v_i\beta_1) = 0$$
$$\frac{\partial F(\beta_0, \beta_1)}{\partial \beta_1} = -2\sum_{i=1}^{n} v_i(Z_i - u_i\beta_0 - v_i\beta_1) = 0$$

Using scalar multiplication notation we will get:

$$\langle u, Z\rangle = \langle u, u\rangle\beta_0 + \langle u, v\rangle\beta_1$$

$$\langle v, Z\rangle = \langle u, v\rangle\beta_0 + \langle v, v\rangle\beta_1$$

Now we need to solve system of 2 linear equations which immidiately lead us to:

$$\hat{\beta}_0 = \frac{\langle v, v\rangle\langle u, Z\rangle - \langle u, v\rangle\langle v, Z\rangle}{\langle u, u\rangle\langle v, v\rangle - \langle u, v\rangle^2}$$

$$\hat{\beta}_1 = \frac{\langle u, u\rangle\langle v, Z\rangle - \langle u, v\rangle\langle u, Z\rangle}{\langle u, u\rangle\langle v, v\rangle - \langle u, v\rangle^2}$$

c) Show the equivalence

Equivalence could be demonstrated by observing that equiation for $\hat{\beta}_0$ and $\hat{\beta}_1$ is the same as we solve in b)

For $S(\beta_0, \beta_1) = \sum_{i=1}^{n}(Y_i - \beta_0 - \beta_1 * Y_i)^2 * \rho_i^{-2}$

Conditions for minimum are $\frac{\partial S(\beta_0,\beta_1)}{\partial \beta_0} = 0$ and $\frac{\partial S(\beta_0,\beta_1)}{\partial \beta_1} = 0$:

$$\frac{\partial S(\beta_0, \beta_1)}{\partial \beta_0} = -2\sum_{i=1}^{n} \rho_i^{-2}(Y_i - \beta_0 - \beta_1 X_i)$$

and

$$\frac{\partial S(\beta_0, \beta_1)}{\partial \beta_1} = -2\sum_{i=1}^{n} X_i \rho_i^{-2}(Y_i - \beta_0 - \beta_1 X_i)$$

Taking into account definition of $u_i = \rho_i^{-1}$, $v_i = \rho_i^{-1} X_i$ and $Z_i = \rho_i^{-1} Y_i$ we get: $\rho_i^{-1} * (Y_i - \beta_0 - \beta_1 * X_i) = (Z_i - u_i * \beta_0 - v_i * \beta_i)$

Which lead us to equations for minimum:

$$\frac{\partial S(\beta_0, \beta_1)}{\partial \beta_0} = -2\sum_{i=1}^{n} u_i(Z_i - u_i\beta_0 - v_i\beta_1)$$

and

$$\frac{\partial S(\beta_0, \beta_1)}{\partial \beta_1} = -2\sum_{i=1}^{n} v_i(Z_i - u_i\beta_0 - v_i\beta_1)$$

which are exactly the same as in section b) which proved the equivalence.

Both methods should give the same results

Task 2

Load data

```
setwd("/Users/sidorovm/projects/Education/Stanford/Stats191/data")

mathData = data.table::fread(file = "math.txt", col.names=c("X1", "X2", "X3", "Y"),
                             header=FALSE, colClasses=c("numeric", "numeric", "numeric", "numeric"), sep

kable(head(mathData), caption="Head of: Annual salaries table.")
```

Table 1: Head of: Annual salaries table.

X1	X2	X3	Y
3.5	9	6.1	33.2
5.3	20	6.4	40.3
5.1	18	7.4	38.7
5.8	33	6.7	46.8
4.2	31	7.5	41.4
6.0	13	5.9	37.5

```r
sprintf("#records: %d", nrow(mathData))
```

```
## [1] "#records: 24"
```

a) Fit regression model

```r
lm.fit = lm(Y~X1+X2+X3, data=mathData)
lm.summary = summary(lm.fit)
lm.coef = coef(lm.summary)
print(lm.summary)
```

```
##
## Call:
## lm(formula = Y ~ X1 + X2 + X3, data = mathData)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.2463 -0.9593  0.0377  1.1995  3.3089
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 17.84693    2.00188   8.915 2.10e-08 ***
## X1           1.10313    0.32957   3.347 0.003209 **
## X2           0.32152    0.03711   8.664 3.33e-08 ***
## X3           1.28894    0.29848   4.318 0.000334 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.753 on 20 degrees of freedom
## Multiple R-squared:  0.9109, Adjusted R-squared:  0.8975
## F-statistic: 68.12 on 3 and 20 DF,  p-value: 1.124e-10
```

And estrimated regression function $\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 * X1 + \hat{\beta}_2 * X2 + \hat{\beta}_3 * X3$ is:

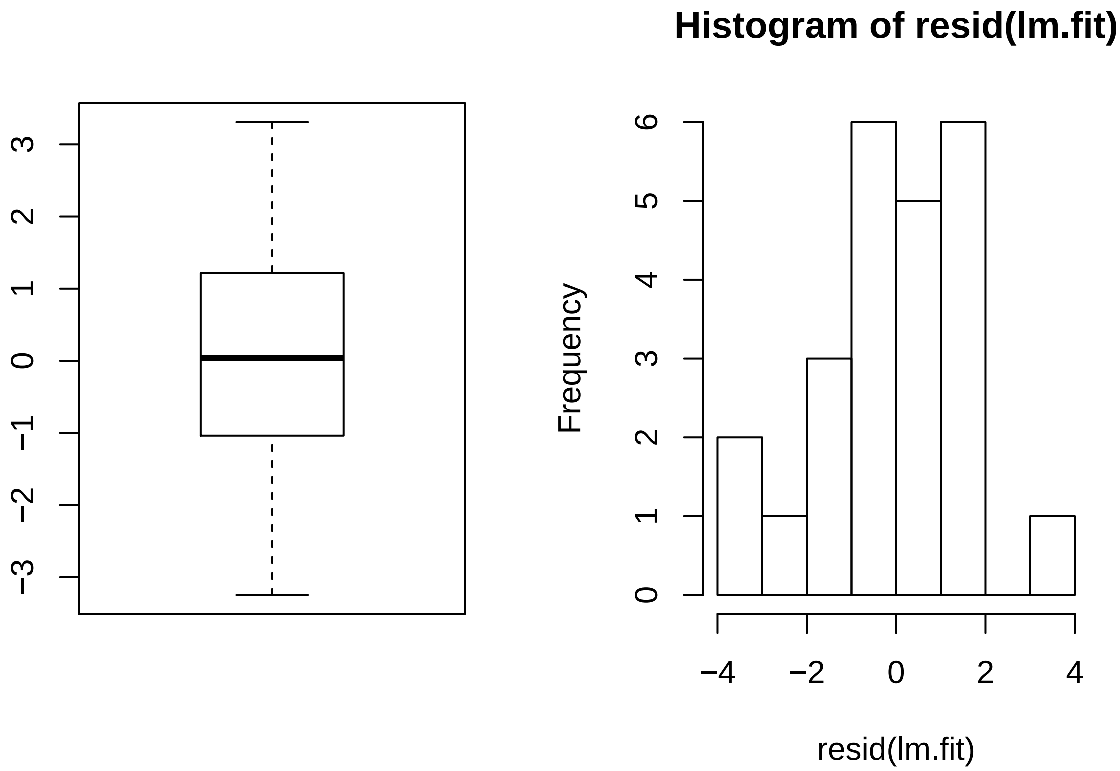
```r
sprintf("%g + %g * X1 + %g * X2 + %g * X3", lm.coef[1,1],lm.coef[2,1],lm.coef[3,1],lm.coef[4,1])
```

```
## [1] "17.8469 + 1.10313 * X1 + 0.32152 * X2 + 1.28894 * X3"
```

b) Fitted values and residuals

Boxplot (also histogram added) for residuals :

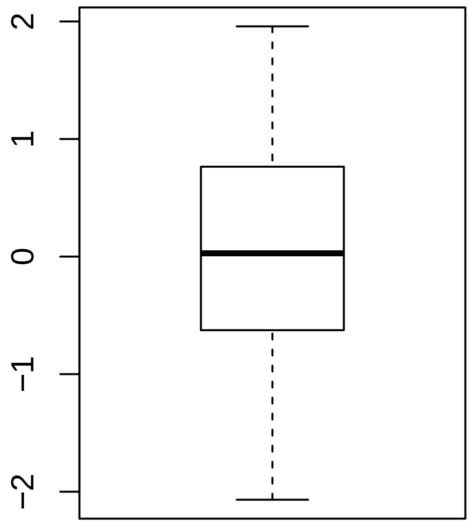
```r
par(mfrow=c(1,2))
boxplot(resid(lm.fit))
hist(resid(lm.fit))
```

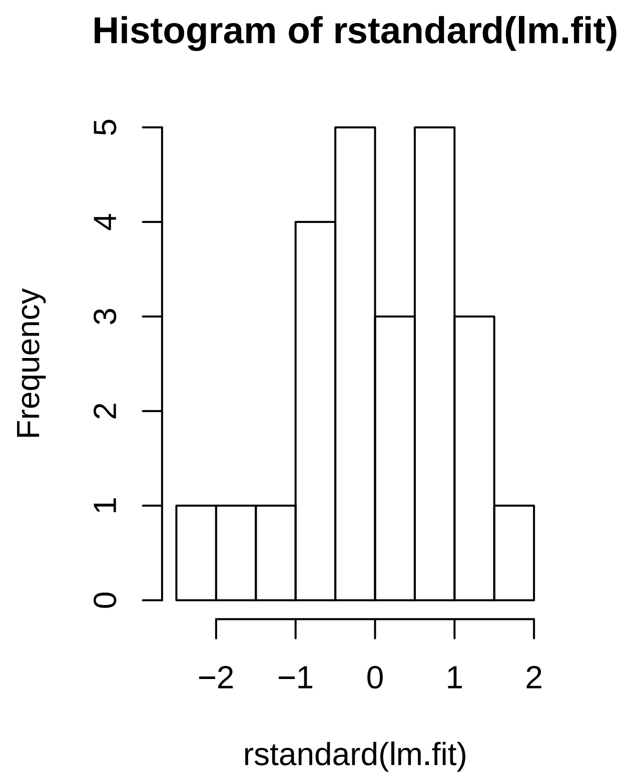



```
par(mfrow=c(1,1))
```

Note: for estimation of normality and homoscedasticity we need to work with standartized residuals (r_i or r_i^*)

```
par(mfrow=c(1,2))
boxplot(rstandard(lm.fit))
hist(rstandard(lm.fit))
```





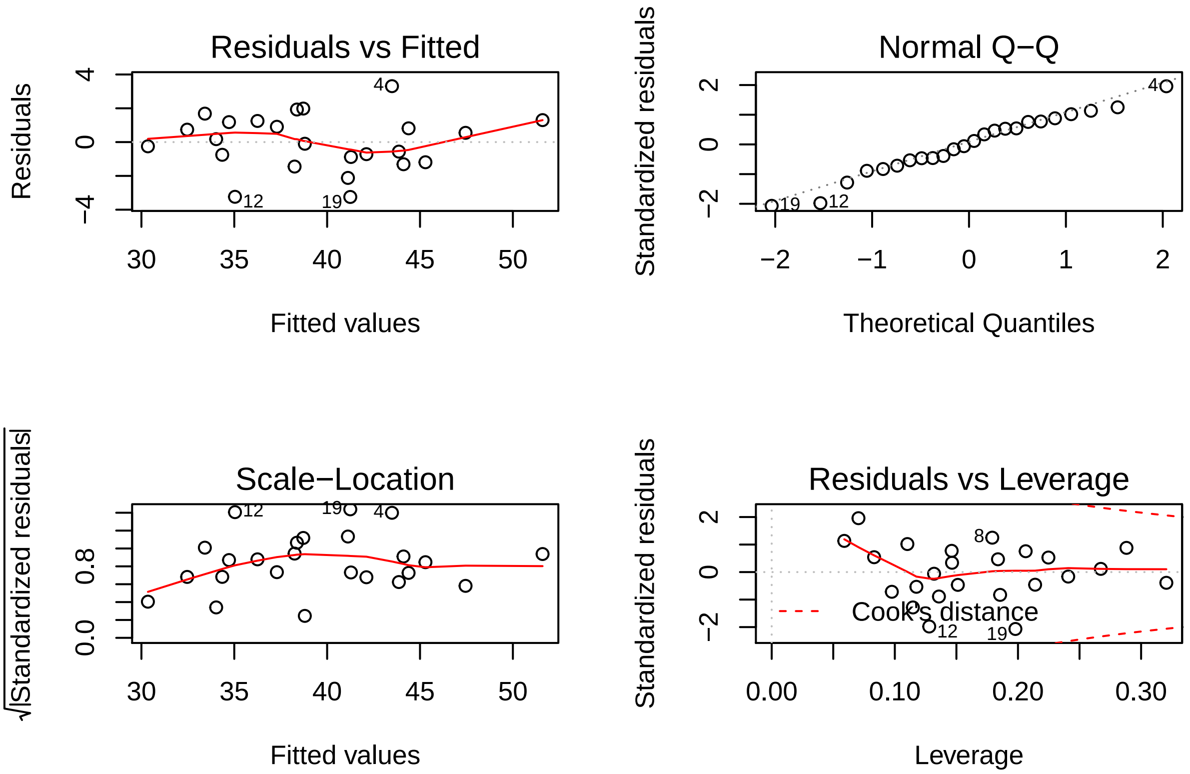

```
par(mfrow=c(1,1))
```

From boxplot we can see that distribution is symmetric and mean is about 0. Also we can see IQR. Also we can see approximately normal distribution of resuduals from histogram plot. We can't make conclusion about homoscedasticity from these graphs. Also it's consider to be good that r_i distributed in range [-2,2].

c) Diagnostic plots

R provides us essential diagnostic plots:

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

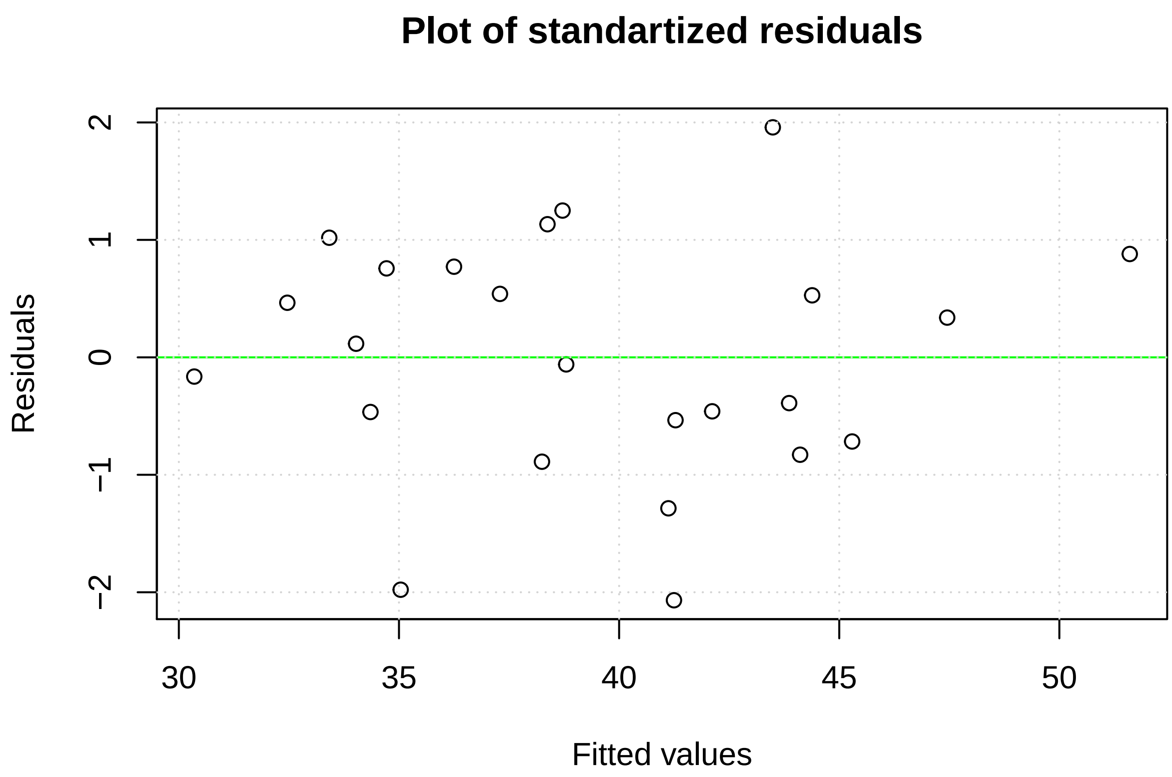



```
par(mfrow=c(1,1))
```

Plot of residuals (standartized)

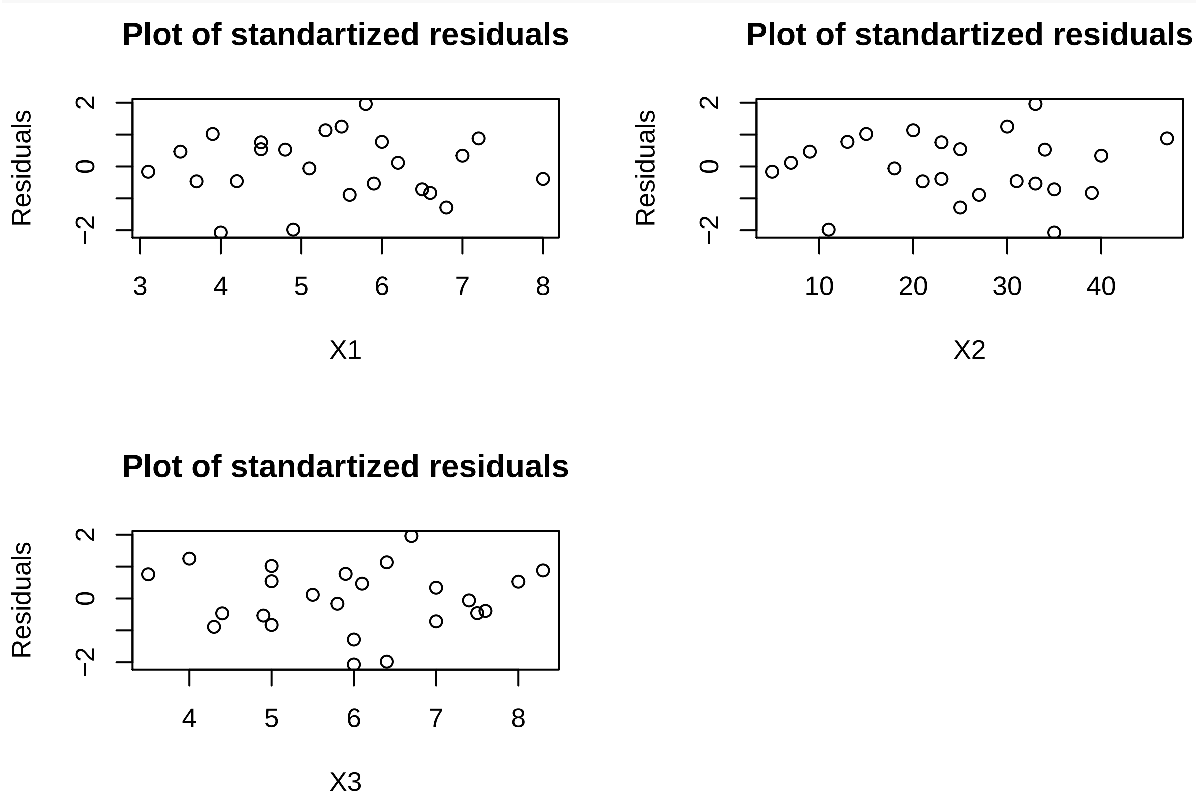
We can see that is about to be the same based on plot below (homoscedasticity)

```
plot(rstandard(lm.fit)~fitted(lm.fit), xlab="Fitted values", ylab="Residuals", main="Plot of standartiz
abline(0,0, col="green")
grid()
```



Also we can plot standartized residuals againt predictors (and we don't see any patterns):

```
par(mfrow=c(2,2))
plot(rstandard(lm.fit)~X1 + X2 + X3, data=mathData, ylab="Residuals", main="Plot of standartized residu
par(mfrow=c(1,1))
```

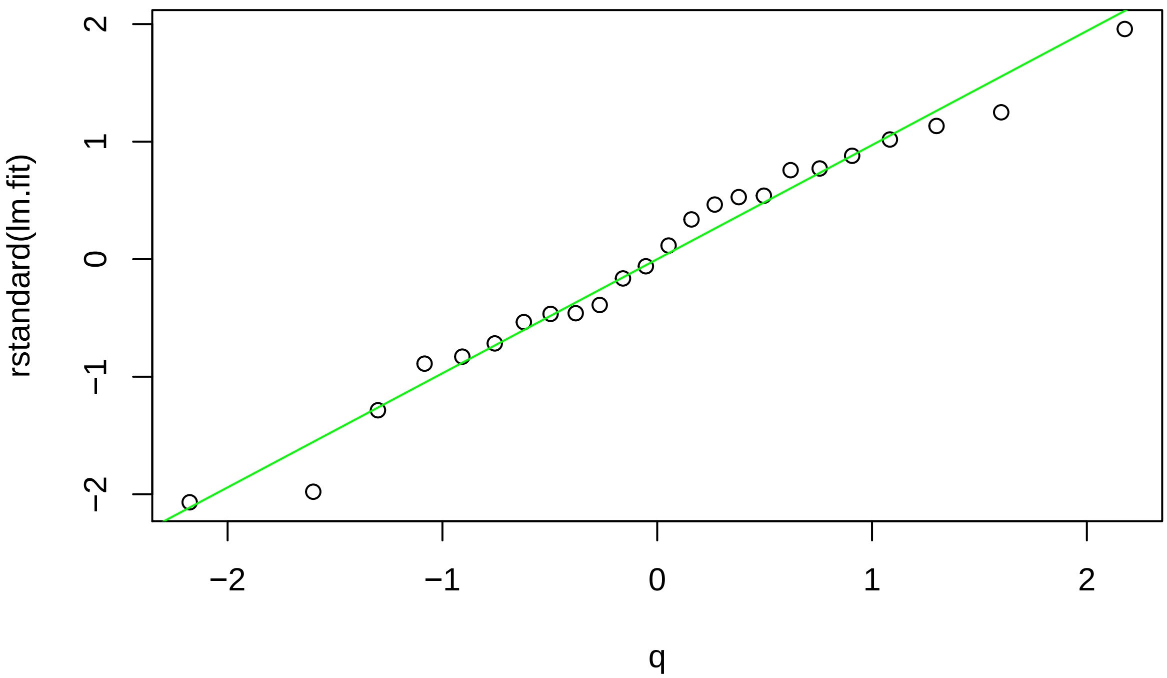


Check normality of distribution of residuals

We will use qqplot to verifydistribution of residuals. Standartized distribution of residuals should follow t distribution with $df = n - p - 1$ (based on 4.11 in textboox) In our case n=24; p=3; so df=20 (=24-3-1)

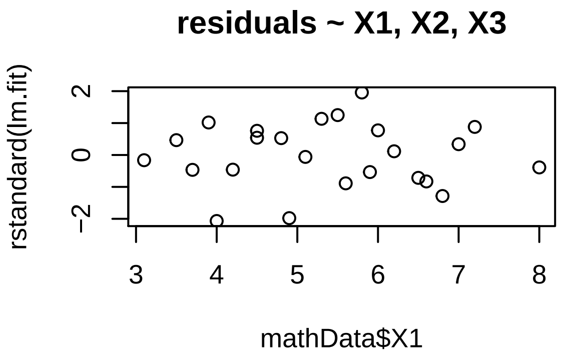
Based on Q-Q plot we can say that residuals close to expected t distribution

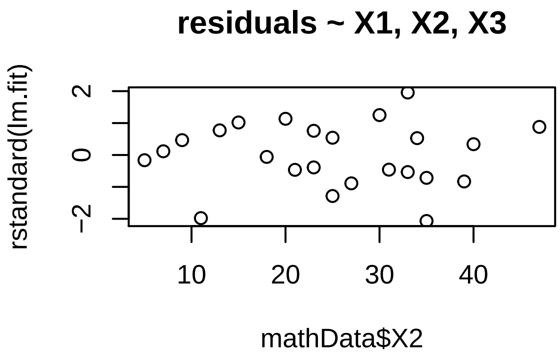
```
par(mfrow=c(1,1))
n = nrow(mathData)
p = 3
q = qt(ppoints(n),df=n - p -1)
qqplot(q, rstandard(lm.fit))
qqline(q, col="green")
```

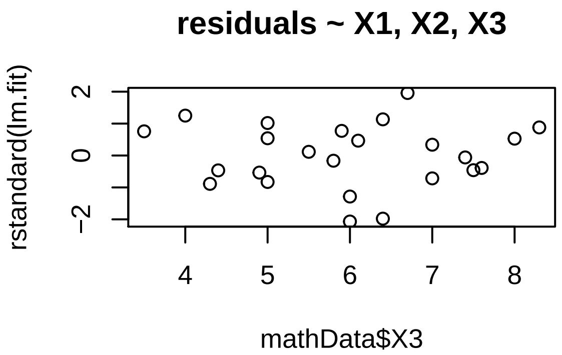


Plot standartized residuals against each of independent variables

```
par(mfrow=c(2,2))
plot(rstandard(lm.fit)~mathData$X1+mathData$X2++mathData$X3, main="residuals ~ X1, X2, X3")
```

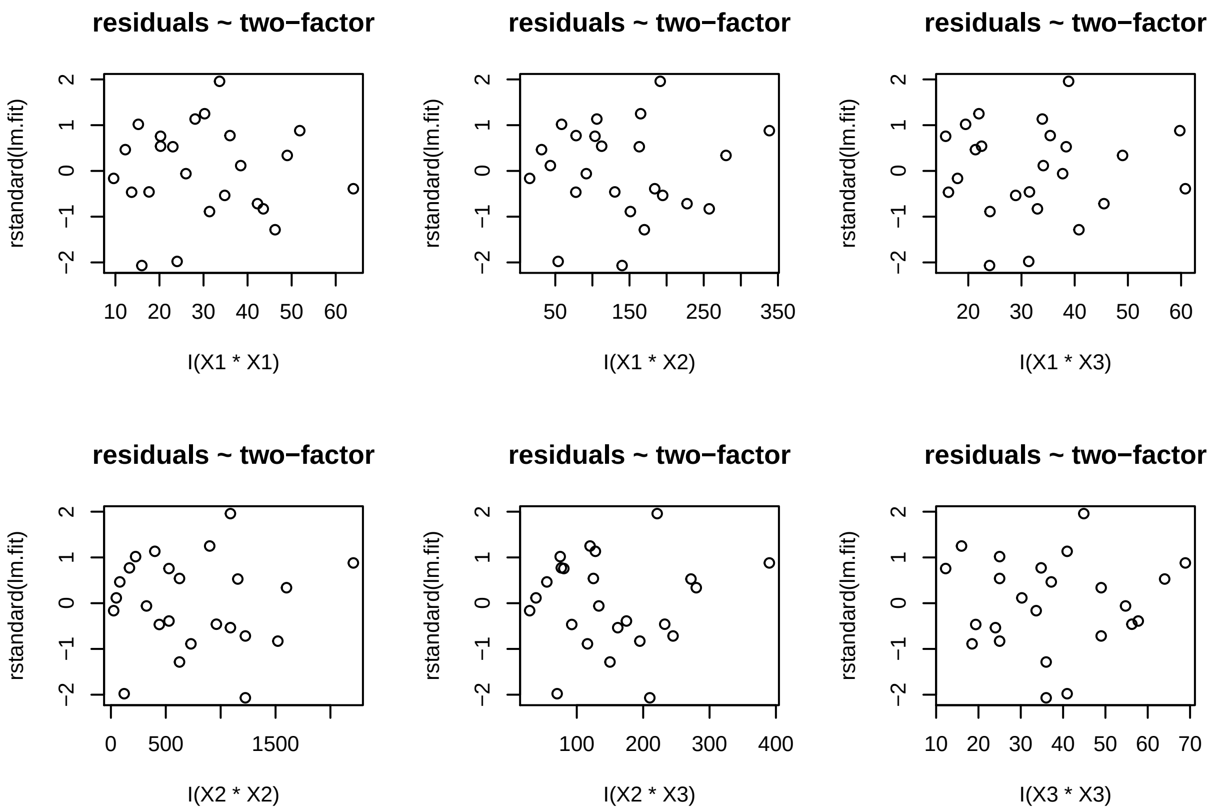






Plot standartized residuals against each of two-factotr independent variables

```
par(mfrow=c(2,3))
plot(rstandard(lm.fit)~I(X1*X1) + I(X1*X2)+I(X1*X3) + I(X2*X2) + I(X2*X3) + I(X3*X3), data=mathData, ma
```



We don’t observe strong evidence of nonlinear pattern

Summarize

1) Residuals follow normality
2) Residuals expose homoscedasticity
3) No parrens observe in graphs residuals ~ X1+X2+X3 and two-phactor iteraction graph, except: 3.1) residuals ~ (X2*X2) demonstrates some nonlinearity 3.1) residual ~ (X2*X3) demonstrates some nonlinearity

d) Test significance of the regression

According 3.10.1 section of the textbook:

$H0 : \beta_1 = \beta_2 = \beta_3 = 0$ i.e. model is $Y = \beta_0 + \epsilon$

$H1 : Y = \beta_0 + \beta_1 * X_1 + \beta_2 * X_2 + \beta_3 * X_3 + \epsilon$

We need to find $F = \frac{MSR}{MSE}$

F statistics already calculated by R (value is):

```
lm.fit.s = summary(lm.fit)

lm.fit.s["fstatistic"]
```

```
## $fstatistic
##    value    numdf    dendf
## 68.11917  3.00000 20.00000
```

Based on that we see F-Test=68.11917 and we need to check significance for $\alpha = 0.05$ with degrees of freedom: $df1 = p = 3$ and $df2 = n - p - 1 = 20$

95th percentile for F-distribution with df1=3 and df2=20 is provided below:

```
alpha = 0.05
qf(1 - alpha, df1=3, df2=20)
```

```
## [1] 3.098391
```

And so we confirm that F-test is significant (greated than 95th percentile) and H0 should be rejected. It means that X1, X2 and X3 have explonatory power.

e) Test whether beta1 = beta3 with alpha=0.01

H0: $\beta_1 = \beta_3$ (H0: Reduce model is adequate)

H1: Full model (H1: Full model is adequate)

To test hypothesis we will use approach described in 3.10.3 in textbook

F in th following way:

$$F = \frac{(SSE(RM) - SSE(FM))/(p + 1 - k)}{SSE(FM)/(n - p - 1)}$$

where $k = 3$ - count of df in reduced model

```
lm.fit.rm = lm(Y~X2++I(X1+X3), data=mathData)
summary(lm.fit.rm)
```

```
##
## Call:
## lm(formula = Y ~ X2 + +I(X1 + X3), data = mathData)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.1970 -1.0522  0.0122  1.0875  3.3485
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 17.89290    1.95684   9.144 9.07e-09 ***
## X2           0.31865    0.03556   8.960 1.28e-08 ***
## I(X1 + X3)   1.20345    0.18912   6.363 2.62e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.717 on 21 degrees of freedom
## Multiple R-squared:  0.9102, Adjusted R-squared:  0.9017
## F-statistic: 106.5 on 2 and 21 DF,  p-value: 1.019e-11
```

Comparing models using F-test

```
alpha = 0.01
n = nrow(mathData)
p = 3
k = 3
hat.Y.RF = predict(lm.fit, newdata=mathData)
hat.Y.RM = predict(lm.fit.rm, newdata=mathData)
```

```r
SSE.RF = sum((hat.Y.RF - mathData$Y)^2)
SSE.RM = sum((hat.Y.RM - mathData$Y)^2)
fTest.1 =  (SSE.RM - SSE.RF) / (p + 1 - k) / SSE.RF * (n - p - 1)

criticalValue = qf(1 - alpha, df1=p + 1 - k, df2=n - p -1)

sprintf("F-Test=%g | F(df1=%d, df2=%d, alpha=%g)=%g", fTest.1, p+1-k, n - p -1, 1-alpha, criticalValue)
```

```
## [1] "F-Test=0.141097 | F(df1=1, df2=20, alpha=0.99)=8.09596"
```

Decision rule: $H0$ is rejected if $FTest \geq F_{(p+1-k,n-p-1,\alpha)}$ Based on that H0 could not be rejected $FTest < F_{(p+1-k,n-p-1,\alpha)}$ i.e. we can conlude that $\beta_1 = \beta_3$

Comapring models via anova (just for verivication/practice)

Same results confirmed by comaring models by using anova:

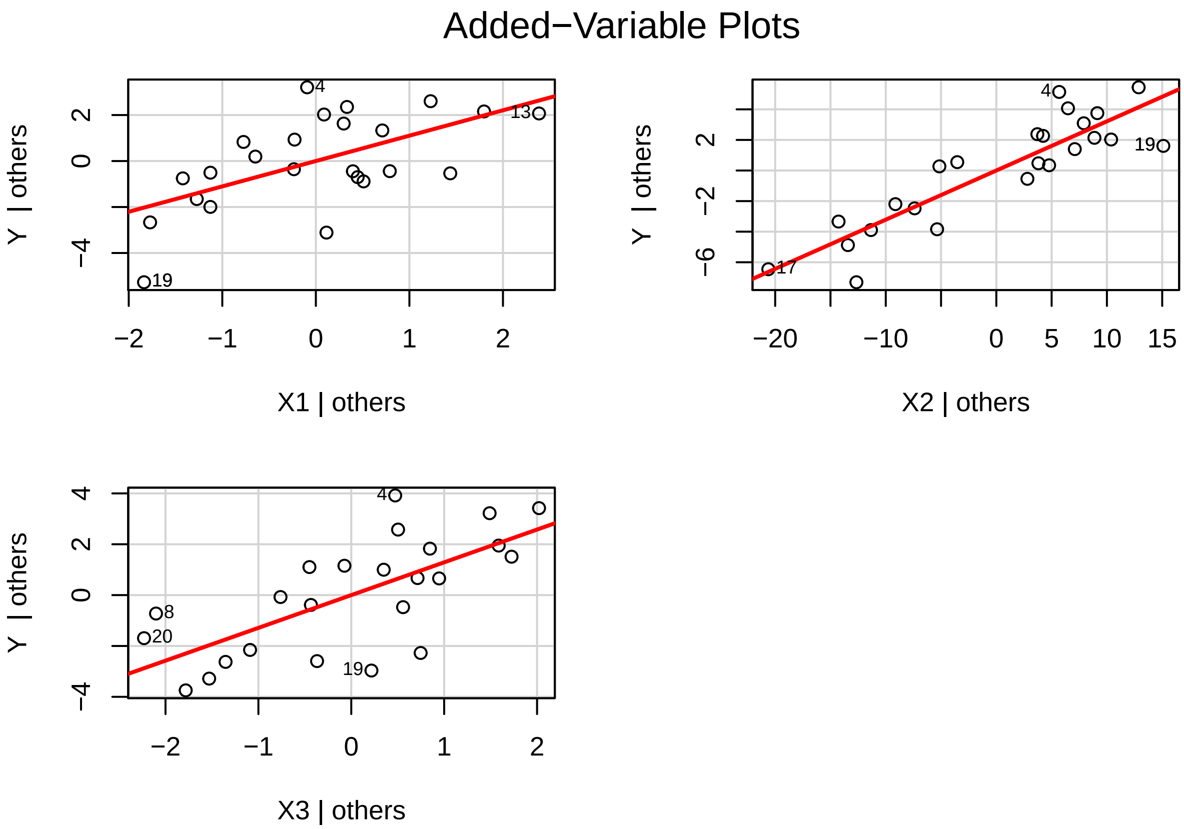
```r
anova(lm.fit, lm.fit.rm)
```

```
## Analysis of Variance Table
##
## Model 1: Y ~ X1 + X2 + X3
## Model 2: Y ~ X2 + +I(X1 + X3)
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     20 61.443
## 2     21 61.876 -1  -0.43347 0.1411 0.7111
```

f) Partial regression plot

Let us use avPlots from car:

```r
avPlots(lm.fit, id.n=2, id.cex=0.7)
```



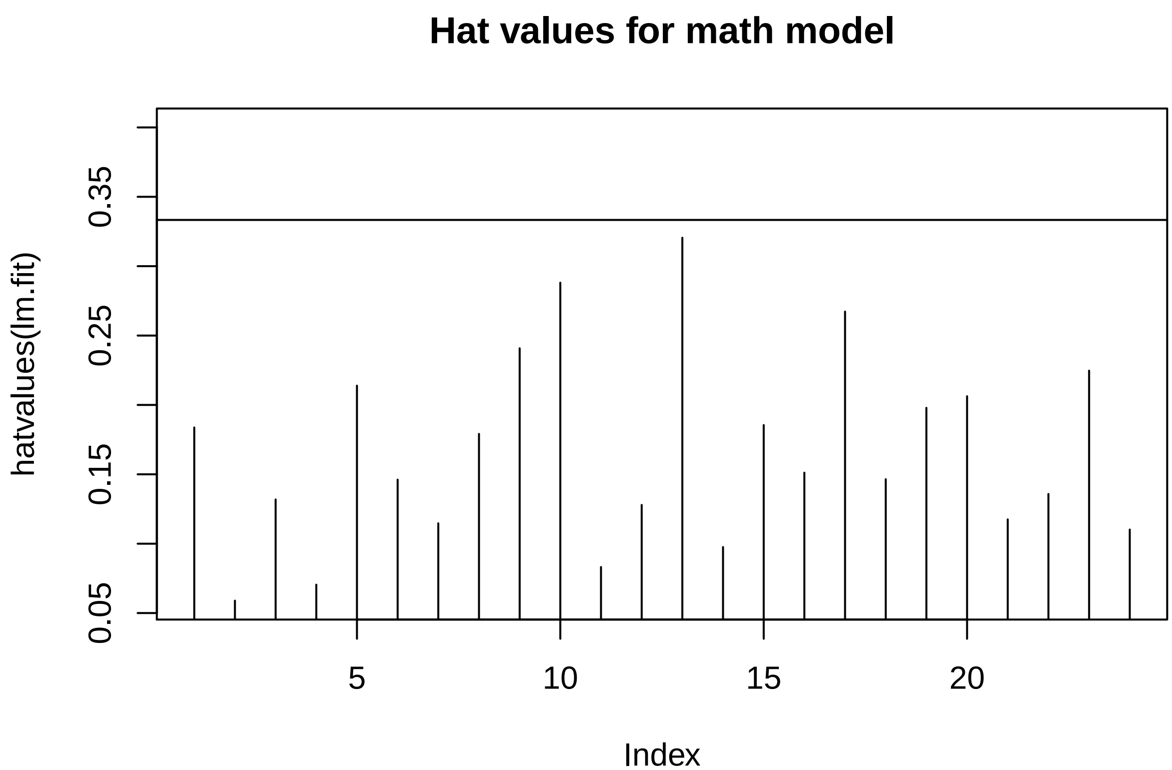
May be Y~X1 graph demostrate non-linearity (like inverted U shape - parabolic), but effect does not look large. But in general we see that each predictor provides contribution to explain Y.

g) Identify outlying X values

We will do detection of X outlyiers based on method §4.8.2 from textbook. Outlying X_i values are detected using $p_{ii} = \frac{1}{n} + \frac{(x_i-\bar{x})^2}{\sum(x_i-\bar{x})^2}$ (also named Hat-values h_i)

Points with $p_{ii} > \frac{2(p+1)}{n}$ should be considered as outliers (the same results could be observed from: influence.measures(lm.fit))

```
plot(hatvalues(lm.fit), main="Hat values for math model", type="h", ylim=c(min(hatvalues(lm.fit)), 0.4)
p = 3
n = 24
abline(2*(p+1)/n,0)
```



Based on this plot we can say that there are no outliers (all values < threshold)

h) Obtain externaly studentized residuals and identify outlying observations

Externaly studentized residuals $r_i^* = \frac{\hat{\epsilon_i}}{\hat{\sigma_{(i)}}\sqrt{1-h_{ii}}}$

```
rstudent(lm.fit)
```

```
##           1           2           3           4           5           6
##  0.45541537  1.14182069 -0.05874067  2.12277227 -0.45036166  0.76357492
##           7           8           9          10          11          12
## -1.30803527  1.26855205 -0.15971914  0.87447286  0.53021865 -2.15012187
##          13          14          15          16          17          18
## -0.38116412 -0.70752805 -0.82239887 -0.45622775  0.11297641  0.33066219
##          19          20          21          22          23          24
## -2.27334231  0.74904846 -0.52549919 -0.88376539  0.51824556  1.01912996
```

To identify outliers let us use $t(n-p-1, 0.025)$ which is 5% tails.

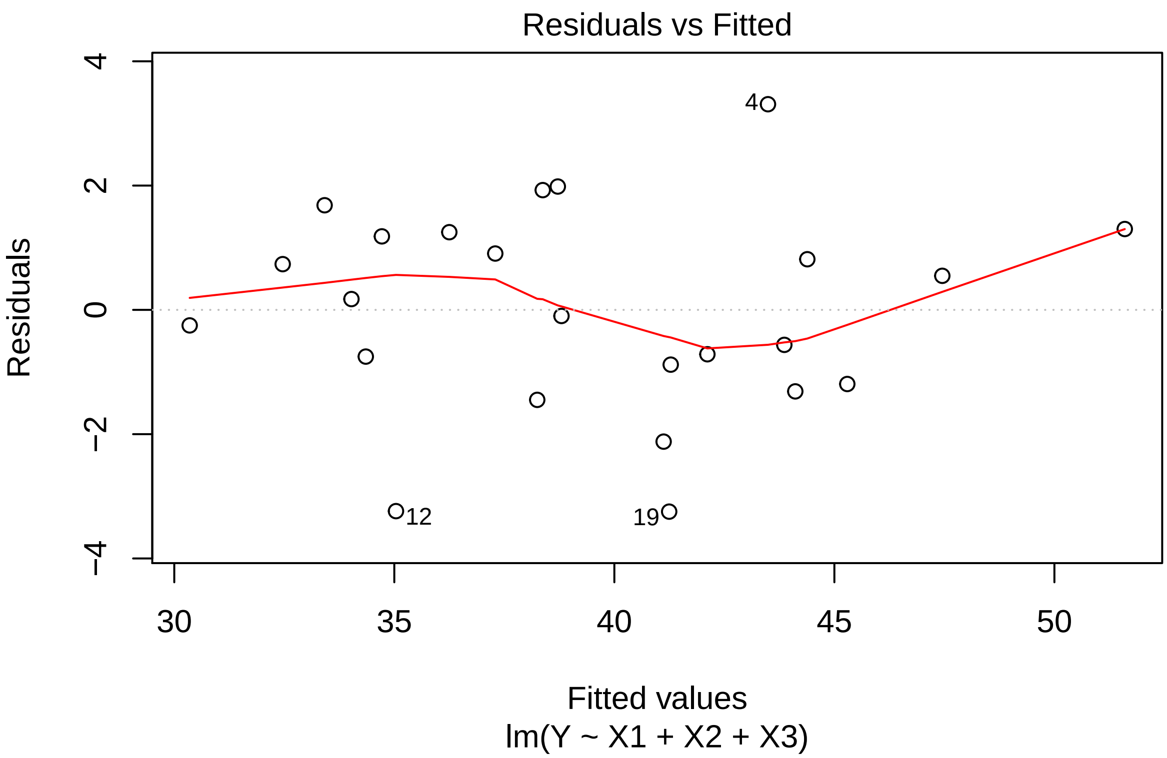
```
pt = abs(qt(0.025, df=24 -3 -1))

which(abs(rstudent(lm.fit)) > pt )
```

```
##  4 12 19
##  4 12 19
```

We get points: 4, 12, 19

The same points we can see on graph "Residuals vs Fitted" below:

```
plot(lm.fit, which=1)
```



i) Obtain for data point 19 DFFITS, DFBETAS and Cook's distance

Required information provided below:

```
influence.measures(lm.fit)$infmat[19,]
```

```
##       dfb.1_      dfb.X1      dfb.X2      dfb.X3       dffit       cov.r
## -0.40663693  0.87659247 -0.81167123 -0.09379799 -1.12930336  0.58470552
##       cook.d         hat
##  0.26384508  0.19792716
```

Interpretation: 1) Cook's distance is the higest one. We identgified this point as potential outlier in j)

2) **Based on DFFIT value it's Outlier:** DFFIT is a mesure similar to Cook's distance. Is outlier if it's abs(DFFIT) > than $2 * sqrt\frac{p+1}{n-p-1}$

```
p = 3
n = nrow(mathData)

2*sqrt((p+1)/(n - p -1))
```

```
## [1] 0.8944272
```

So, it's an outlier based on DFFIT value

j) Cook's distance for each point

R calculated it for us

```
influence.measures(lm.fit)$infmat[,"cook.d"]
```

```
##            1            2            3            4            5 
## 0.0121554183 0.0201145771 0.0001379079 0.0726768960 0.0143721927 
##            6            7            8            9           10 
## 0.0254846657 0.0535044376 0.0851963254 0.0021266581 0.0782936585 
##           11           12           13           14           15 
## 0.0066148947 0.1435558586 0.0178961739 0.0138793949 0.0391448170 
##           16           17           18           19           20 
## 0.0096500265 0.0012244182 0.0049091478 0.2638450790 0.0372684944 
##           21           22           23           24 
## 0.0095379470 0.0310288231 0.0202011318 0.0320969399
```

or

```r
cooks.distance(lm.fit)
```

```
##            1            2            3            4            5 
## 0.0121554183 0.0201145771 0.0001379079 0.0726768960 0.0143721927 
##            6            7            8            9           10 
## 0.0254846657 0.0535044376 0.0851963254 0.0021266581 0.0782936585 
##           11           12           13           14           15 
## 0.0066148947 0.1435558586 0.0178961739 0.0138793949 0.0391448170 
##           16           17           18           19           20 
## 0.0096500265 0.0012244182 0.0049091478 0.2638450790 0.0372684944 
##           21           22           23           24 
## 0.0095379470 0.0310288231 0.0202011318 0.0320969399
```

3) DFBETAS - not outlier. n <= 30, so cutoff is 1 (based on concpect lectures) and for point 19 DFBETAS < 1

Calculation Cook's distance

This should match to what R provided for us

$C_i = \frac{\sum_{j=1}^{n}(\hat{y}_j - \hat{y}_{j(i)})^2}{\hat{\sigma}^2(p+1)}$ Where $\hat{\sigma}^2 = \frac{SSE}{n-p-1}$

```r
p = 3
n = nrow(mathData)

lm.fit = lm(Y~X1+X2+X3, data=mathData)
lm.summary = summary(lm.fit)
Y.pred = predict(lm.fit, newdata=mathData)

SSE = sum((Y.pred - mathData$Y)^2)
sigma2 = SSE / (n - p - 1)

cooks.d = c()

for (i in 1:n) {
  subset = (1:n) != i
  lm.fit.i = lm(Y~X1+X2+X3, data=mathData, subset = subset)
  Y.pred.i = predict(lm.fit.i, newdata=mathData)

  Ci = sum((Y.pred.i - Y.pred)^2) / sigma2/(p+1)

  cooks.d = c(cooks.d, Ci)
}
```

```
cooks.d
```

```
##  [1] 0.0121554183 0.0201145771 0.0001379079 0.0726768960 0.0143721927
##  [6] 0.0254846657 0.0535044376 0.0851963254 0.0021266581 0.0782936585
## [11] 0.0066148947 0.1435558586 0.0178961739 0.0138793949 0.0391448170
## [16] 0.0096500265 0.0012244182 0.0049091478 0.2638450790 0.0372684944
## [21] 0.0095379470 0.0310288231 0.0202011318 0.0320969399
```

Find potential point of influence

Based on§4.1.1 from textbook: Ci which is greater that F(df1=p+1, df2=n - p -1, 50%)

```
criticalValue = qf(0.5, df1=p + 1, df2=n - p -1)

sprintf("Threshold value=%g", criticalValue)
```

```
## [1] "Threshold value=0.868293"
```

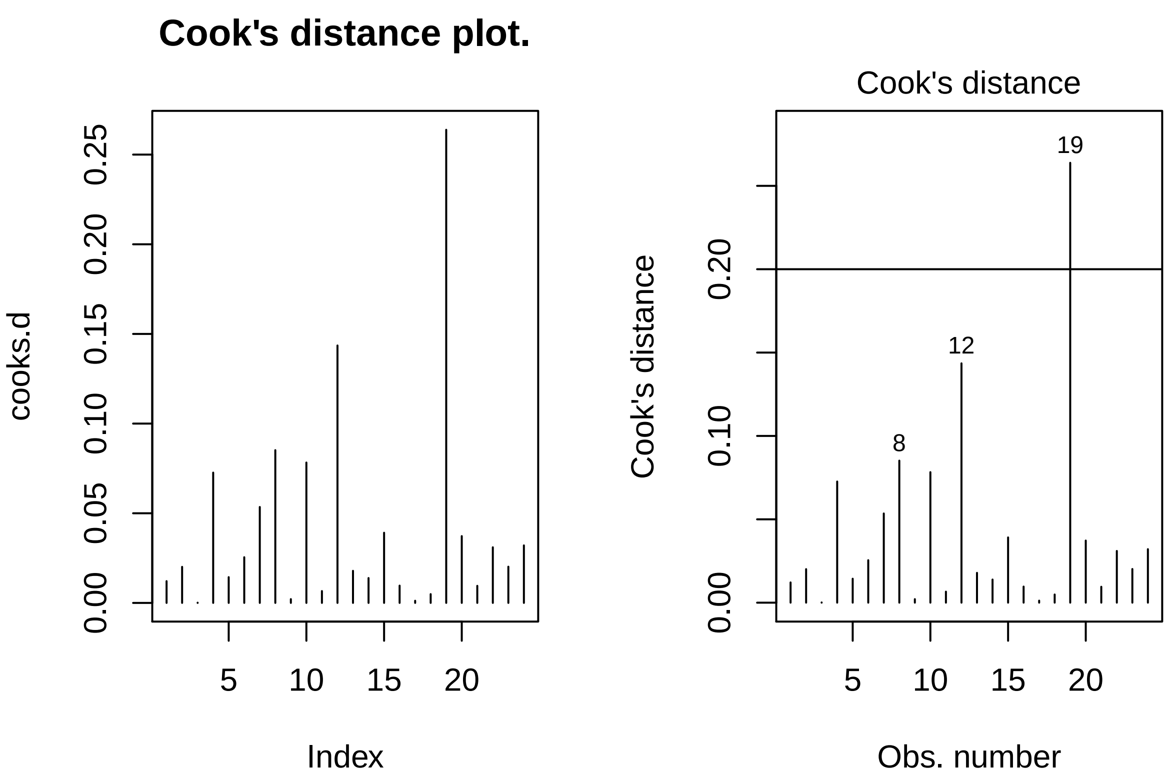
```
which(cooks.d > criticalValue)
```

```
## integer(0)
```

Nothing has been identified by F-test. But based on other literature it was suggested to identify outliers as those, for whom $C_i > \frac{4}{n-p-1}$ and point=19 could be identified as outlier

Compare our cooks distance graph with R (should be the same):

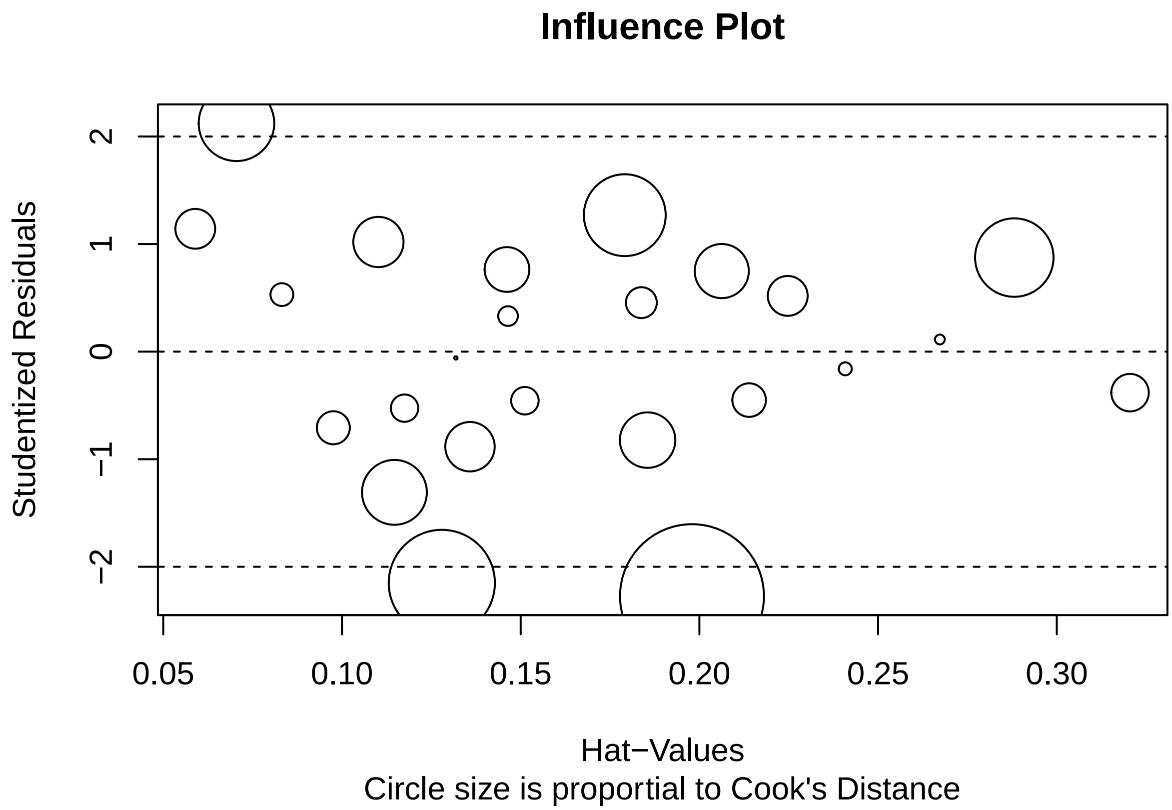
```
par(mfrow=c(1,2))
plot(cooks.d, main="Cook's distance plot.", type="h")
cutoff <- 4/(n -p -1)
plot(lm.fit, which=4, cook.levels=cutoff)
abline(cutoff,0)
```



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

Let us build influence plot:

```r
influencePlot(lm.fit,   id.method="identify", main="Influence Plot", sub="Circle size is proportial to C
```



References

LATEX greek latters

Official book site (Regression Analysis by Example)

R Companion to Linear regression

Regression diagnostics in R