Practical_1_SSR

Soumarya Basak

14/04/2022

```r
#########  Practical 1 ##########
library(readxl)
library(readr)
library(tidyverse)
```

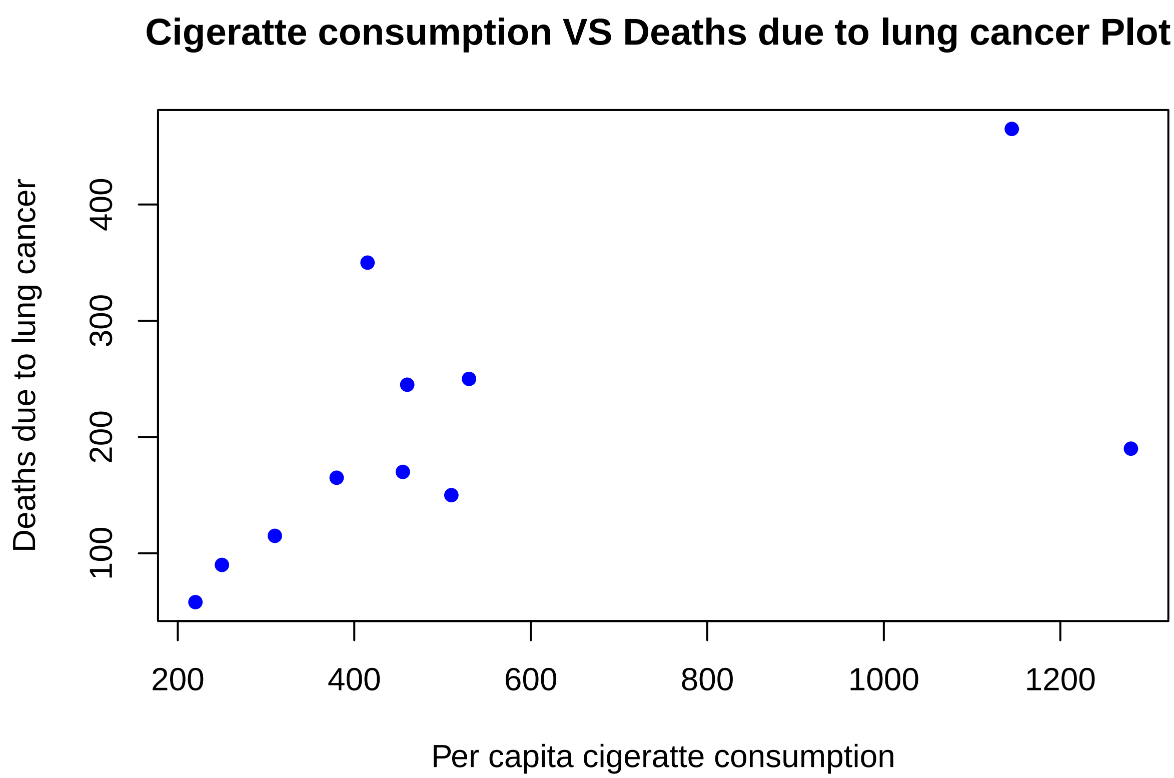
```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v dplyr   1.0.7
## v tibble  3.1.6     v stringr 1.4.0
## v tidyr   1.1.4     v forcats 0.5.1
## v purrr   0.3.4

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(dplyr)
library(ggplot2)
library(olsrr)
```

```
## Warning: package 'olsrr' was built under R version 4.1.3

##
## Attaching package: 'olsrr'

## The following object is masked from 'package:datasets':
##
##     rivers
```

```r
###### Data set #########
df<- read.csv("C:\\Users\\souma\\Dropbox\\Mstat_CU\\Sem 2\\Regression_analysis_1\\Data Sets\\cigaratte_c
colnames(df)<- c("Index","y_var","x_var")
```

```r
plot(df$x_var,df$y_var,pch=16,col="blue",
     xlab="Per capita cigeratte consumption",ylab = "Deaths due to lung cancer",
     main="Cigeratte consumption VS Deaths due to lung cancer Plot ")
```



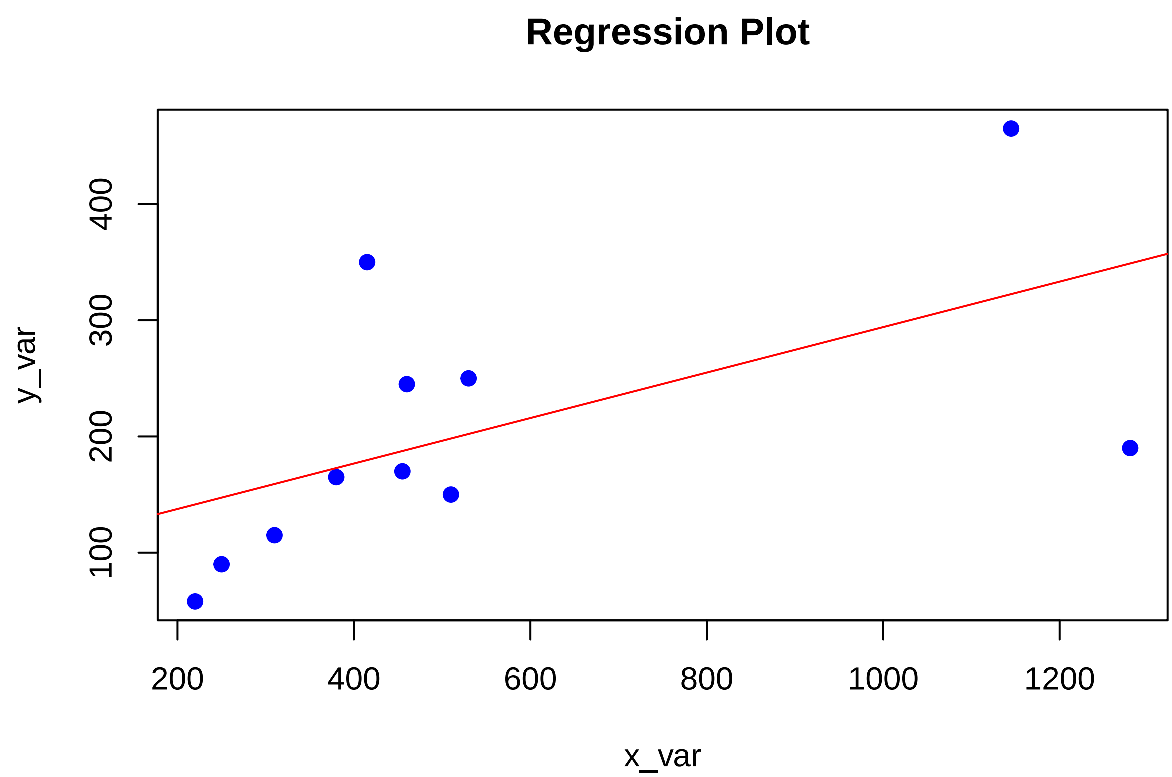

```r
###### (a) ########

ssr1<- lm(y_var~x_var,df)
summary(ssr1)
```

```
##
## Call:
## lm(formula = y_var ~ x_var, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -158.90  -52.79  -17.46   52.21  170.36
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 98.42920   59.30799   1.660   0.1314
## x_var        0.19568    0.09343   2.094   0.0657 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 102.7 on 9 degrees of freedom
## Multiple R-squared:  0.3277, Adjusted R-squared:  0.253
## F-statistic: 4.387 on 1 and 9 DF,  p-value: 0.06571
```

For the F test, the p value is more than 0.05, so the parameters the insignificant.

Both the parameters are not significant

```r
plot(df$x_var,df$y_var,pch=19,col="blue",
     xlab="x_var",ylab = "y_var",
     main=" Regression Plot")
abline(ssr1,col="red")
```



This is due to the influential observation

The residuals

```r
res<- residuals(ssr1)
cbind("Residuals"=res)
```

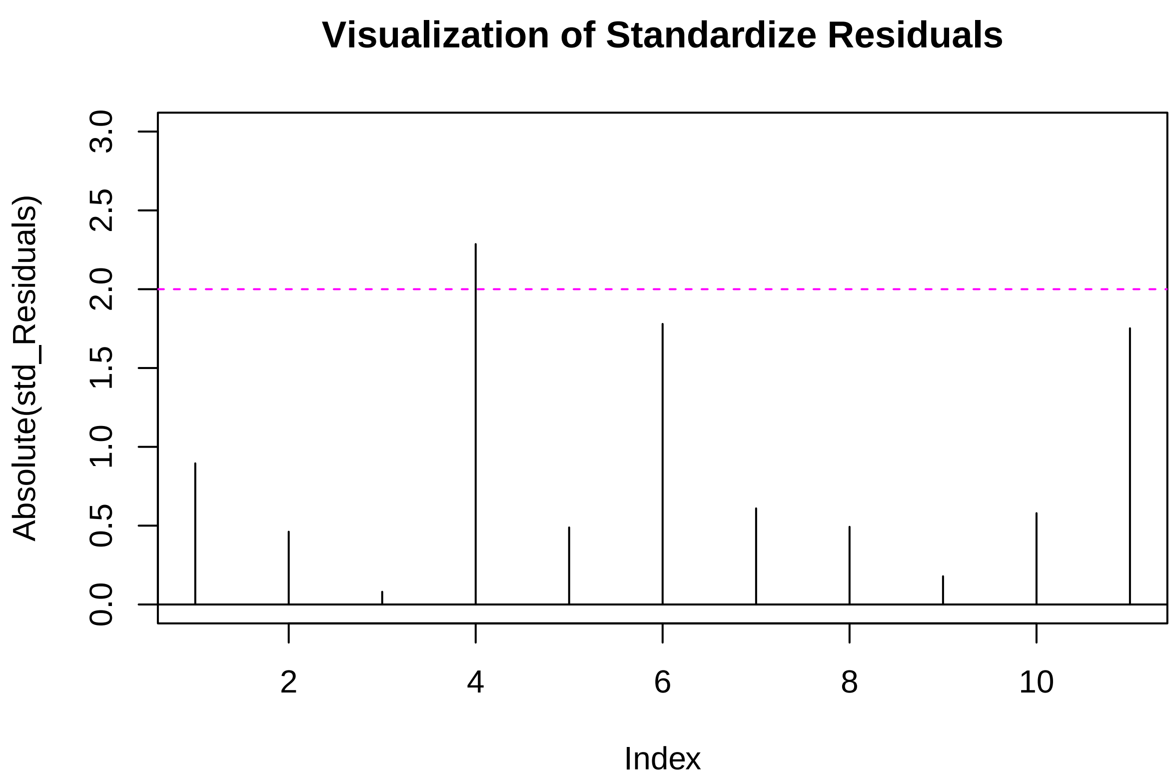
```
##     Residuals
## 1  -83.478964
## 2  -44.090230
## 3   -7.787881
## 4 -158.900543
## 5   47.860008
## 6  142.516357
## 7  -57.349386
## 8  -48.226377
## 9  -17.463937
## 10  56.557660
## 11 170.363293
```

STandardize Residuals

```
res_std<- rstandard(ssr1)
cbind("Standaredize_Residuals"=res_std)
```

```
##    Standaredize_Residuals
## 1             -0.89544653
## 2             -0.46154959
## 3             -0.08047462
## 4             -2.28633046
## 5              0.48868724
## 6              1.77978693
## 7             -0.60956366
## 8             -0.49261973
## 9             -0.17891785
## 10             0.57921016
## 11             1.75221345
```

```
plot(abs(res_std),type='h',
     ylab = "Absolute(std_Residuals)",ylim = c(0,3),
     main="Visualization of Standardize Residuals")
abline(h=0)
abline(h=2,lty=2,col='magenta')
```



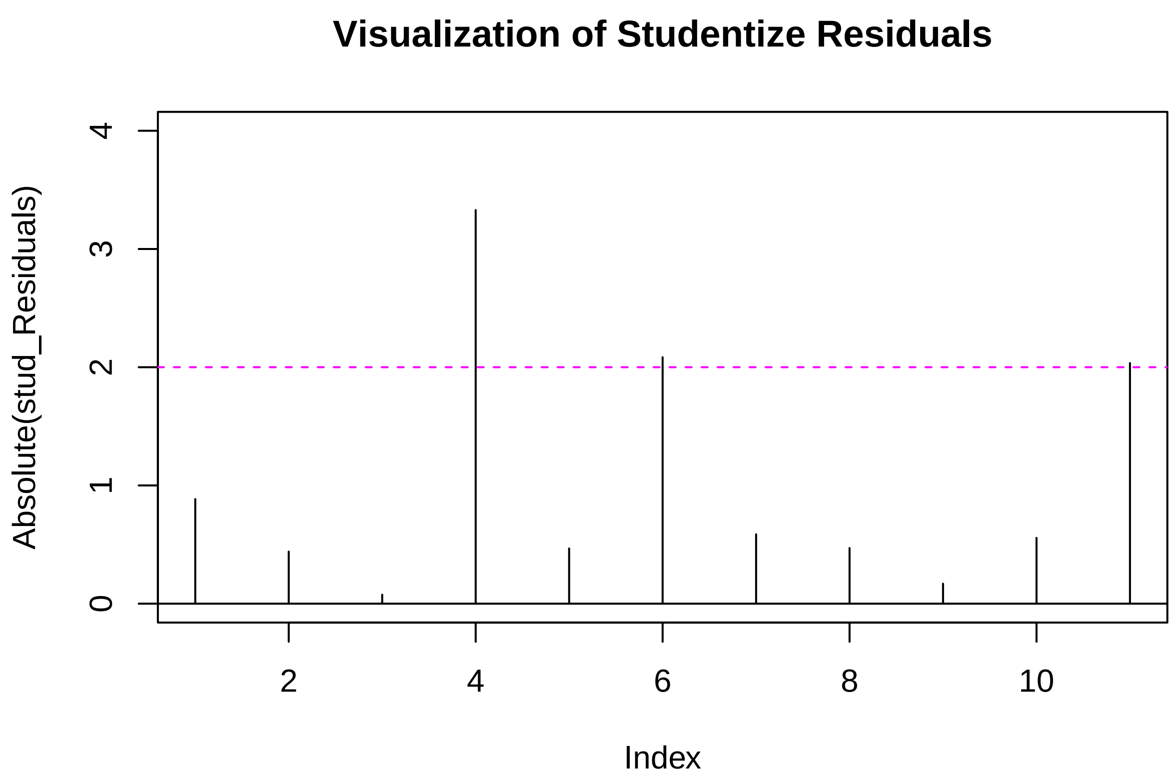
So we can see from the plot that there are 3 oulier values for Y_var

Studentize Residuals

```
res_stu<-rstudent(ssr1)
cbind("Studentize_Residuals"=res_stu)
```

```
##    Studentize_Residuals
## 1           -0.88455745
## 2           -0.44039638
## 3           -0.07589952
## 4           -3.32934087
## 5            0.46697601
## 6            2.08444722
## 7           -0.58694596
## 8           -0.47083750
## 9           -0.16898616
## 10           0.55655620
## 11           2.03523182
```

```
plot(abs(res_stu),type='h',
     ylab = "Absolute(stud_Residuals)",ylim = c(0,4),
     main="Visualization of Studentize Residuals"
     )
abline(h=2,col='magenta',lty=2)
abline(h=0)
```



So studentize residual says that there are three outliers in y_var

```r
df[which(abs(res_stu)>=2),]
```

To find the y outliers

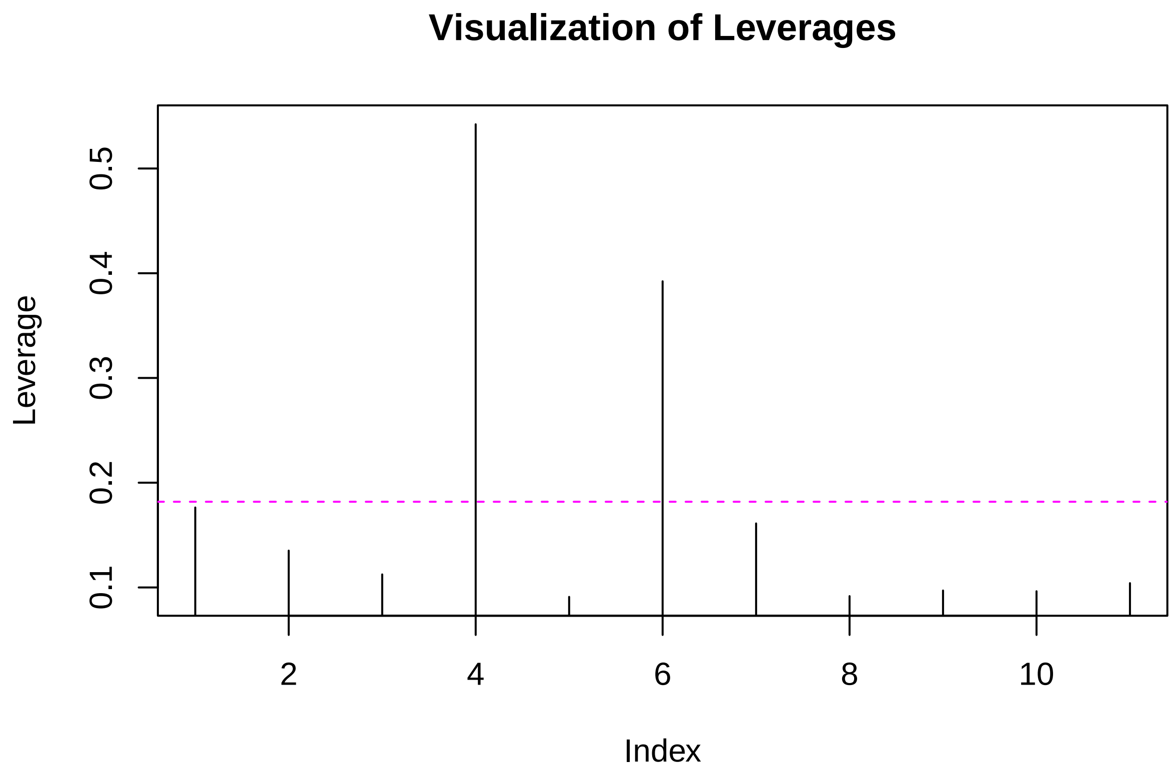
```
##            Index y_var x_var
## 4  United States   190  1280
## 6  Great Britain   465  1145
## 11       Finland   350   415
```

Leverage

```r
lev<-hatvalues(ssr1)
cbind("Leverage"=lev)
```

```
##      Leverage
## 1  0.17634086
## 2  0.13518996
## 3  0.11244867
## 4  0.54223194
## 5  0.09101591
## 6  0.39233234
## 7  0.16113488
## 8  0.09172282
## 9  0.09707913
## 10 0.09638538
## 11 0.10411810
```

```r
plot(lev,type='h',
     ylab="Leverage", main="Visualization of Leverages"
     )
abline(h=(2/11),col='magenta',lty=2)
```



From the image it is clear that there are two 2 x outlier
as two values are more than $\frac{2p}{n} = \frac{2}{11}$

```
# arrange in decreasing order
cbind("Leverage"=lev[order(-lev)])
```

To Identify the x outliers

```
##      Leverage
## 4  0.54223194
## 6  0.39233234
## 1  0.17634086
## 7  0.16113488
## 2  0.13518996
## 3  0.11244867
## 11 0.10411810
## 9  0.09707913
## 10 0.09638538
## 8  0.09172282
## 5  0.09101591
```

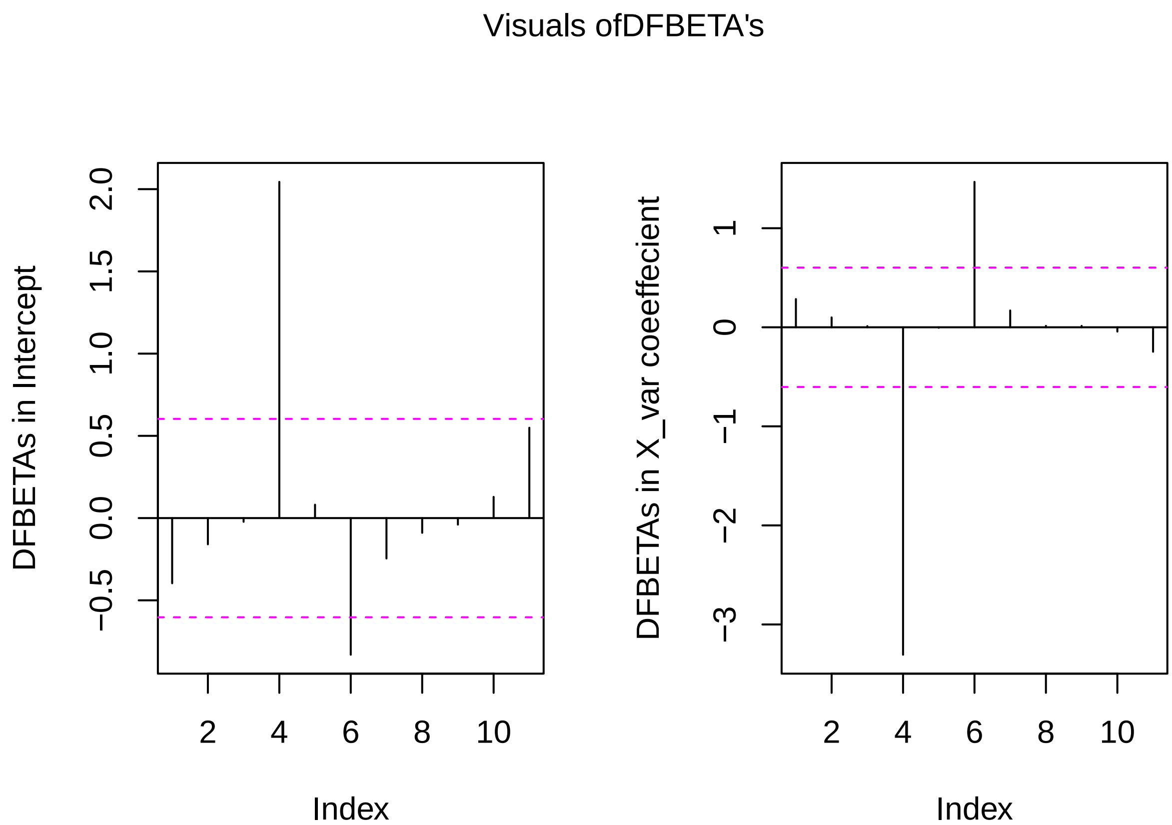
so the 4th and 6th obsns are x-outliers

DFBETA

```
db<-dfbetas(ssr1)
db
```

```
##    (Intercept)        x_var
## 1  -0.39641388  0.284880021
## 2  -0.15955137  0.099653248
## 3  -0.02276883  0.011823905
## 4   2.04443096 -3.305819823
## 5   0.08143844 -0.005062287
## 6  -0.83095128  1.468068127
## 7  -0.24573243  0.169824513
## 8  -0.08980784  0.014092928
## 9  -0.03991472  0.013969157
## 10  0.12913805 -0.043327210
## 11  0.54932238 -0.247128933
```

```
par(mfrow=c(1,2))
plot(db[,1],type='h',
     ylab = "DFBETAs in Intercept")
abline(h=0)
abline(h=2/(sqrt(11)),lty=2, col="magenta")
abline(h=-2/(sqrt(11)),lty=2,col="magenta")

plot(db[,2],type='h',
     ylab = "DFBETAs in X_var coeeffecient")
abline(h=0)
abline(h=2/(sqrt(11)),lty=2,col="magenta")
abline(h=-2/(sqrt(11)),lty=2,col="magenta")

mtext("Visuals ofDFBETA's",side =3,line = -1 ,outer = TRUE)
```



so, is clear that there are 2 influencial observation

```
print(which( abs(db[,1]) >2/(sqrt(11)) ) )
```

```
## 4 6
## 4 6
```

```
paste(" ")
```

```
## [1] " "
```

```
which( abs(db[,2]) >2/(sqrt(11)) )
```

```
## 4 6
## 4 6
```

SO, for the 4th and 6th obs the DFBETa is high

DFFIT

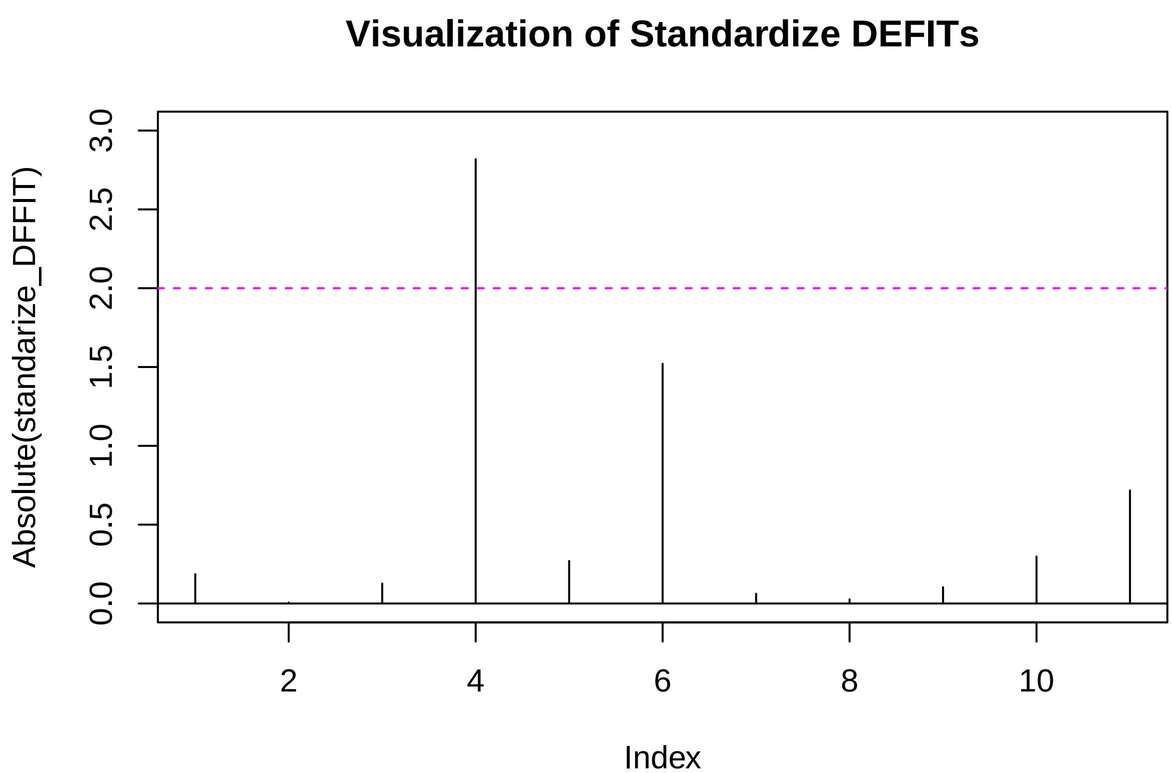
range $= \frac{2}{\sqrt{(11)}}$

```
d_fit<-dffits(ssr1)
cbind("DEFIT"=d_fit)
```

```
##          DEFIT
## 1  -0.40928777
## 2  -0.17412279
## 3  -0.02701590
```

```
## 4  -3.62349834
## 5   0.14776620
## 6   1.67488461
## 7  -0.25724494
## 8  -0.14962375
## 9  -0.05541014
## 10  0.18177033
## 11  0.69382767
```

```
### Standardization of DFFIT

v<- var(d_fit)*( (length(d_fit)-1)/length(d_fit) ) # variance of DFFIT

std_d_fit <- (d_fit - mean(d_fit))/sqrt(v)


plot(abs(std_d_fit),type='h', ylim = c(0,3),
     ylab = "Absolute(standarize_DFFIT)",
     main = "Visualization of Standardize DEFITs")
abline(h=0)
abline(h=2,lty=2,col='magenta')
```



So there are 2 values which act like a outlier as have high DFFIT values

```
cbind("DFFIT"=d_fit[order(-abs(d_fit))])
```

To identify the outlier

```
##          DFFIT
## 4  -3.62349834
## 6   1.67488461
## 11  0.69382767
## 1  -0.40928777
## 7  -0.25724494
## 10  0.18177033
## 2  -0.17412279
## 8  -0.14962375
## 5   0.14776620
## 9  -0.05541014
## 3  -0.02701590
```

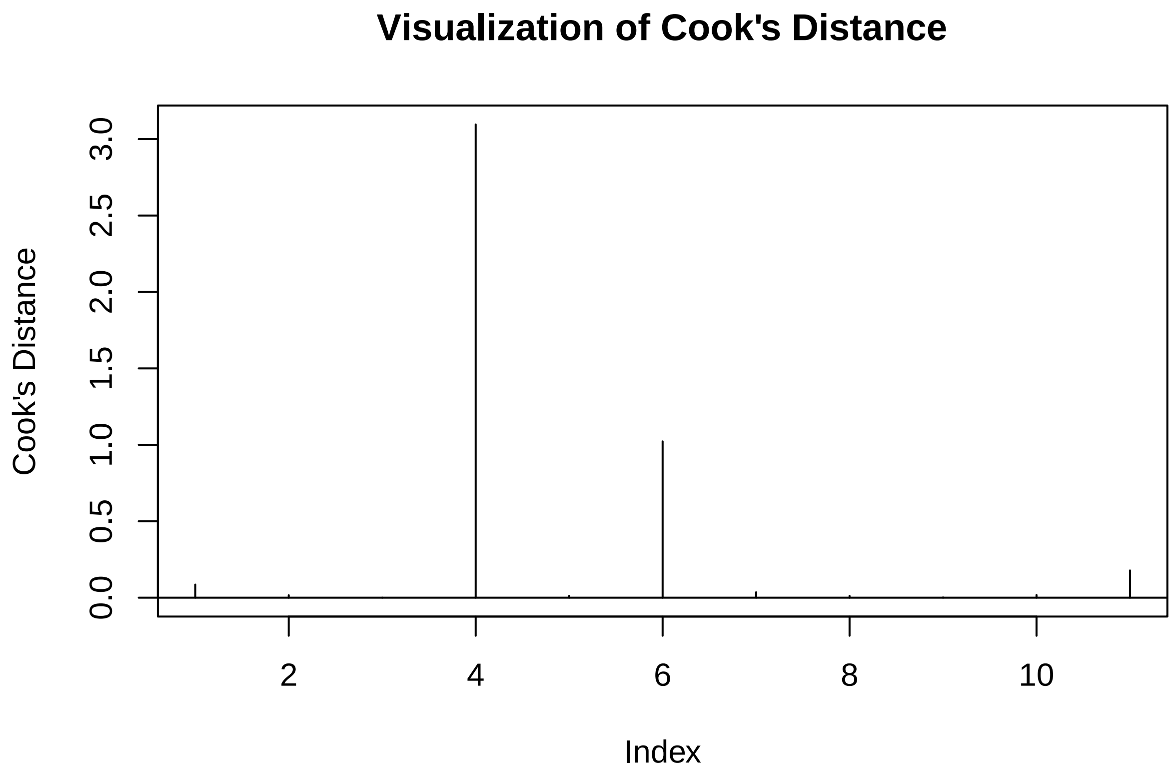
The 4th and 6th observations have higher DFFIT values, so they may be some Influential observation

Cook's Distance

```
cd<- cooks.distance(ssr1)
cbind("Cook_d"=cd)
```

```
##          Cook_d
## 1  0.0858330926
## 2  0.0166506223
## 3  0.0004102502
## 4  3.0959049540
## 5  0.0119561965
## 6  1.0225722645
## 7  0.0356866335
## 8  0.0122532866
## 9  0.0017208916
## 10 0.0178924695
## 11 0.1784101215
```

```
plot(cd,type='h',ylab = "Cook's Distance",
     main = "Visualization of Cook's Distance")
abline(h=0)
```



All the above measures indicates theat the 4^{th} and 6^{th} obsn are influential observation

Next step

so we remove the 6th and 4th obsn.

```
df_ot<-df[-c(4,6),]

ssr1_updated<-lm(y_var~x_var,df_ot)
summary(ssr1_updated)
```

```
##
## Call:
## lm(formula = y_var ~ x_var, data = df_ot)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -90.360 -26.351 -10.490  -1.119 160.746
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -34.0013    94.9690  -0.358    0.731
## x_var         0.5380     0.2339   2.300    0.055 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 73.5 on 7 degrees of freedom
## Multiple R-squared:  0.4304, Adjusted R-squared:  0.349
## F-statistic: 5.288 on 1 and 7 DF,  p-value: 0.05502
```

the parameters are insignificant and even the R^2 value is very low

```
#plot(df_ot$x_var,df_ot$y_var,pch=19,col='blue')
#abline(ssr1_updated,col='red')
```

The fit is again too bad, even the parameters are insignificant. So we don't need to model the response variable with the regrassor.

Now we remove the 11 th obsn with high cooks distance

```
df_ott<-df[-c(4,6,11),]
```

```
ssr1_updated_2<-lm(y_var~x_var,df_ott)
summary(ssr1_updated_2)
```

```
##
## Call:
## lm(formula = y_var ~ x_var, data = df_ott)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -65.203 -14.507   1.373  16.925  54.596
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -37.7487    48.8416  -0.773  0.46894
## x_var         0.4960     0.1207   4.111  0.00628 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 37.79 on 6 degrees of freedom
## Multiple R-squared:  0.738,  Adjusted R-squared:  0.6943
## F-statistic:  16.9 on 1 and 6 DF,  p-value: 0.00628
```

```
#plot(df_ott$x_var,df_ott$y_var,pch=19,col='blue')
#abline(ssr1_updated_2,col='red')
```

So for the last model fits well , have higher R^2 values
but the intercept is statistically insignificant. And so,

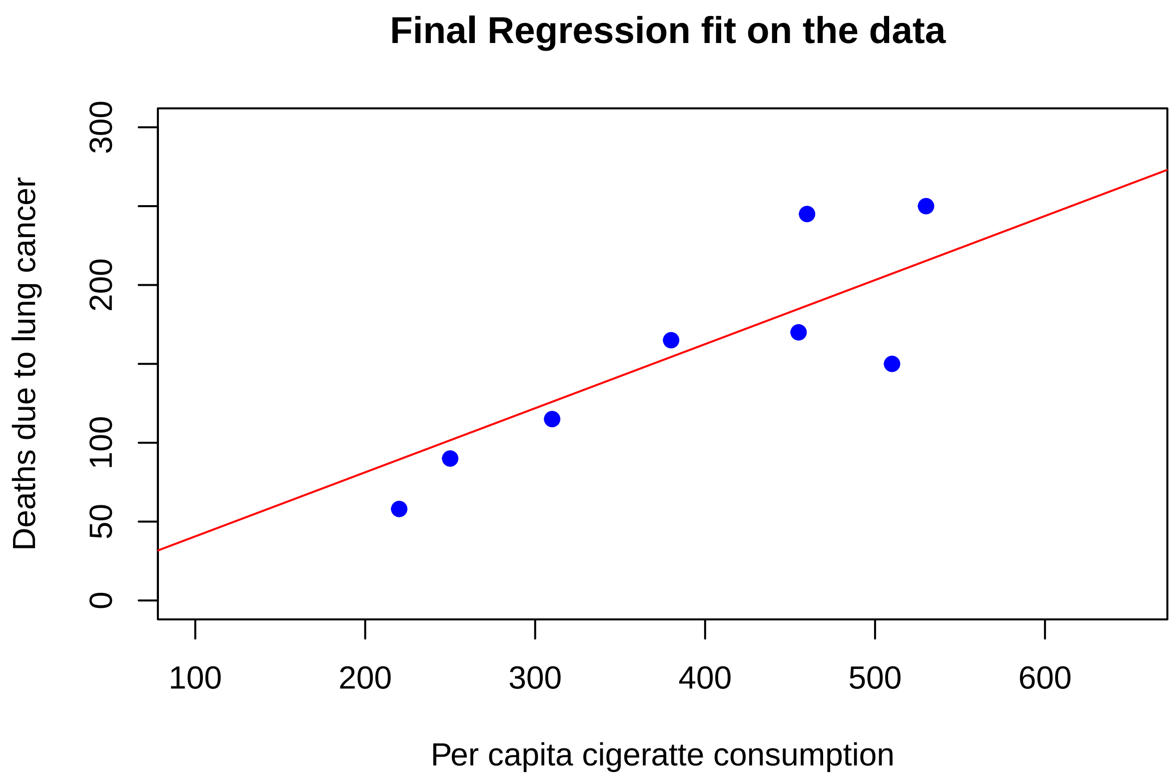
Model without Intercept

```
ssr1_updated_3<- lm(y_var~ 0 +x_var,df_ott)
summary(ssr1_updated_3)
```

```
##
## Call:
## lm(formula = y_var ~ 0 + x_var, data = df_ott)
##
## Residuals:
##    Min     1Q Median     3Q    Max
```

```
## -57.21 -18.99 -11.26  16.62  58.10
##
## Coefficients:
##       Estimate Std. Error t value Pr(>|t|)
## x_var  0.40629    0.03204   12.68 4.39e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 36.69 on 7 degrees of freedom
## Multiple R-squared:  0.9583, Adjusted R-squared:  0.9523
## F-statistic: 160.8 on 1 and 7 DF,  p-value: 4.39e-06
```

```
plot(df_ott$x_var,df_ott$y_var,pch=19,col='blue',xlim = c(100,650),ylim = c(0,300),
     xlab = "Per capita cigeratte consumption",ylab = "Deaths due to lung cancer",
     main = " Final Regression fit on the data")
abline(ssr1_updated_3,col='red')
```



The Model

$$lung\_cancer = (0.40629) \times cigaratte\_consumption$$

Problem2

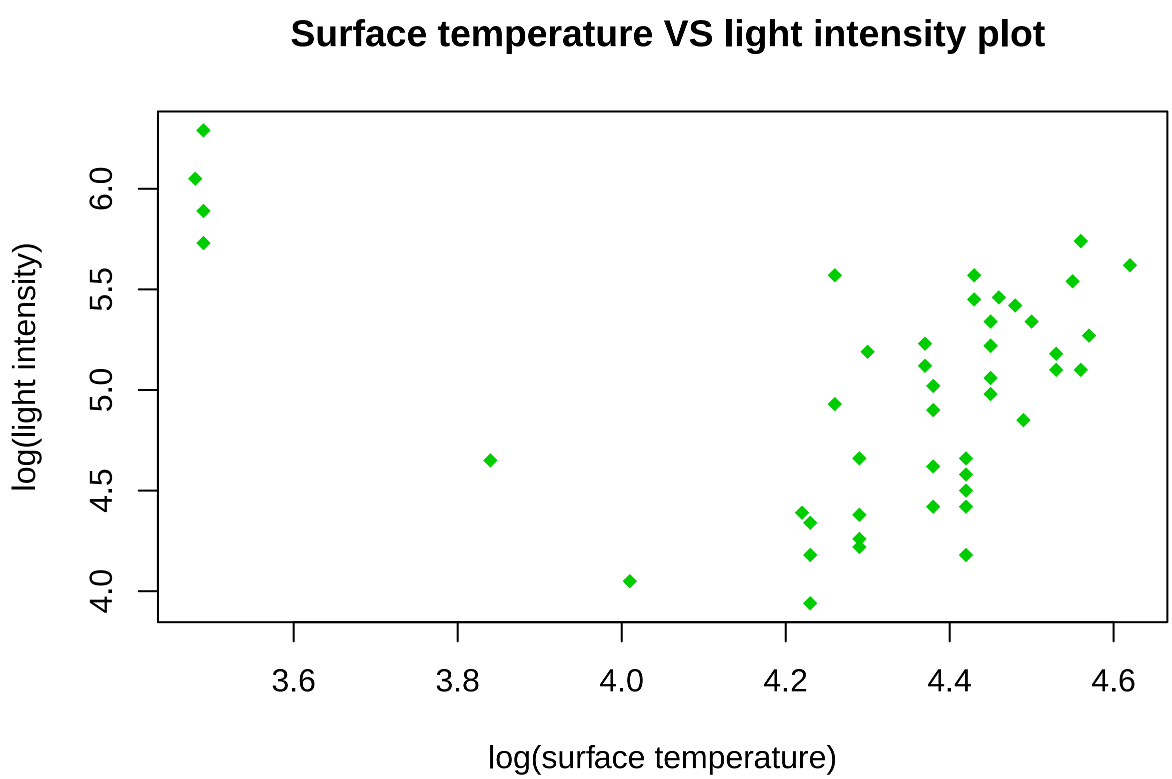
Data Input

```
library(tidyverse)
library(readr)
library(readxl)
library(ggplot2)
library(dplyr)
```

```
df2<- read.csv("C:\\Users\\souma\\Dropbox\\Mstat_CU\\Sem 2\\Regression_analysis_1\\Data Sets\\star_ligh

colnames(df2)<-c("sl no","x_var","y_var")
head(df2)
```

```
##   sl no x_var y_var
## 1     1  4.37  5.23
## 2     2  4.56  5.74
## 3     3  4.26  4.93
## 4     4  4.56  5.74
## 5     5  4.30  5.19
## 6     6  4.46  5.46
```

```
plot(df2$x_var,df2$y_var,col="green3",pch=18,
     xlab = "log(surface temperature)",ylab = "log(light intensity)",
     main = " Surface temperature VS light intensity plot")
```

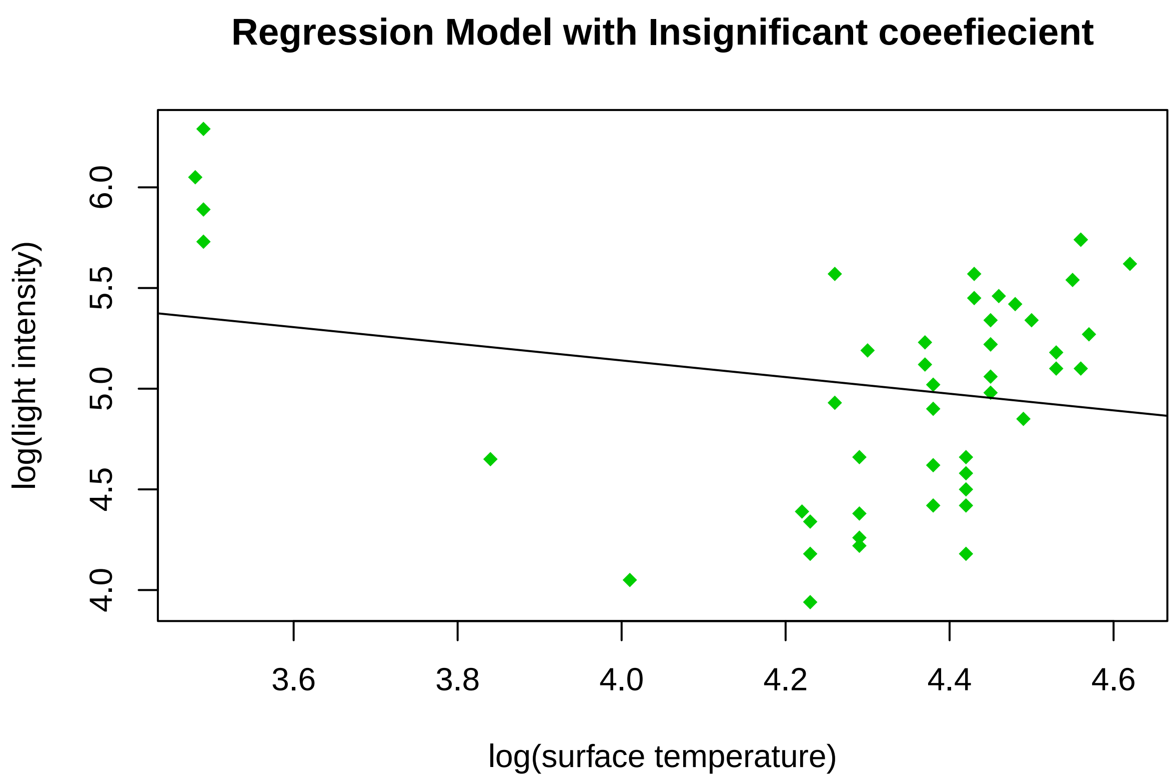


```r
ssr2<- lm(y_var~x_var,df2)
summary(ssr2)
```

```
##
## Call:
## lm(formula = y_var ~ x_var, data = df2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.1052 -0.5067  0.1327  0.4423  0.9390
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   6.7935     1.2365   5.494 1.75e-06 ***
## x_var        -0.4133     0.2863  -1.444    0.156
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5646 on 45 degrees of freedom
## Multiple R-squared:  0.04427,    Adjusted R-squared:  0.02304
## F-statistic: 2.085 on 1 and 45 DF,  p-value: 0.1557
```

Note that, the p-value for f statistics is 0.15 which is larger than 0.05, which implies that both the regression coeffiecients are statistically insignificant. so it says we don;t need to model at all.

```r
plot(df2$x_var,df2$y_var,col="green3",pch=18,
     xlab = "log(surface temperature)",ylab = "log(light intensity)",
     main = "Regression Model with Insignificant coeefiecient")
abline(ssr2)
```



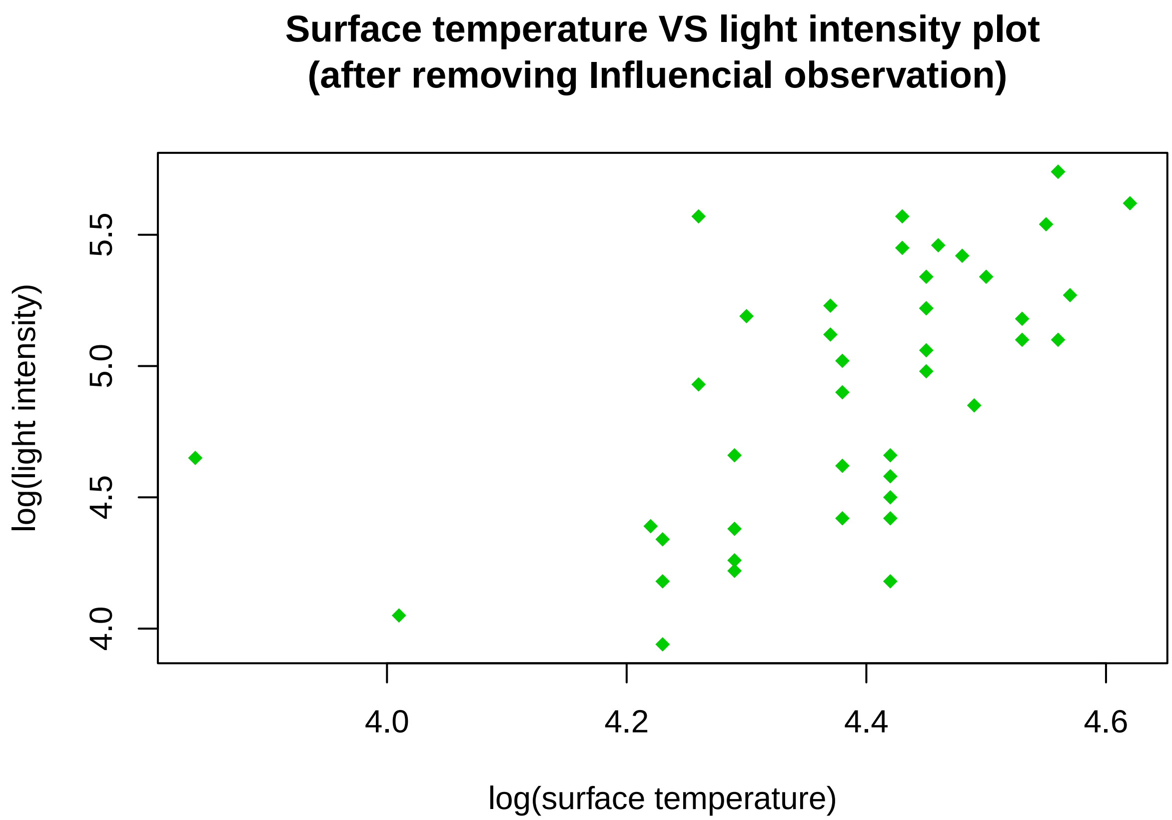

```r
influence.measures(ssr2)
```

```
## Influence measures of
##   lm(formula = y_var ~ x_var, data = df2) :
##
##      dfb.1_ dfb.x_vr    dffit cov.r   cook.d    hat inf
## 1  -0.00899  0.01325  0.06490 1.061 2.14e-03 0.0222
## 2  -0.18113  0.19664  0.29979 0.981 4.37e-02 0.0373
## 3  -0.00645  0.00467 -0.02726 1.068 3.80e-04 0.0219
## 4  -0.18113  0.19664  0.29979 0.981 4.37e-02 0.0373
## 5   0.00460 -0.00158  0.04542 1.064 1.05e-03 0.0213
## 6  -0.06129  0.07045  0.15240 1.035 1.17e-02 0.0271
## 7  -0.26466  0.25483 -0.29879 1.082 4.46e-02 0.0781
## 8  -0.08146  0.08815  0.13148 1.067 8.75e-03 0.0387
## 9   0.03403 -0.02464  0.14390 1.026 1.04e-02 0.0219
## 10 -0.00491  0.00723  0.03543 1.067 6.41e-04 0.0222
## 11  0.35179 -0.34450  0.36509 1.266 6.73e-02 0.1941   *
## 12 -0.04503  0.05373  0.13957 1.037 9.79e-03 0.0250
## 13 -0.06631  0.07492  0.14727 1.042 1.09e-02 0.0287
## 14 -0.33620  0.31667 -0.43877 0.914 9.00e-02 0.0444
## 15 -0.02756  0.01409 -0.20319 0.983 2.02e-02 0.0214
## 16  0.03205 -0.03892 -0.10897 1.049 6.01e-03 0.0244
## 17 -0.10405  0.08409 -0.31389 0.892 4.60e-02 0.0229
## 18  0.06634 -0.08055 -0.22556 0.979 2.49e-02 0.0244
## 19 -0.07999  0.06465 -0.24131 0.959 2.82e-02 0.0229
## 20  0.50356 -0.49313  0.52261 1.233 1.36e-01 0.1941   *
```

```
## 21 -0.02307  0.01179 -0.17007 1.007 1.44e-02 0.0214
## 22 -0.02908  0.01487 -0.21438 0.974 2.24e-02 0.0214
## 23  0.04556 -0.05532 -0.15490 1.027 1.20e-02 0.0244
## 24  0.01288 -0.01445 -0.02725 1.077 3.80e-04 0.0296
## 25 -0.00169  0.00234  0.00990 1.070 5.01e-05 0.0225
## 26  0.02537 -0.03080 -0.08625 1.058 3.78e-03 0.0244
## 27 -0.01285  0.00657 -0.09474 1.049 4.55e-03 0.0214
## 28  0.00383 -0.00529 -0.02238 1.069 2.56e-04 0.0225
## 29 -0.06635  0.05480 -0.18356 1.006 1.67e-02 0.0234
## 30  0.66619 -0.65257  0.69067 1.198 2.34e-01 0.1983   *
## 31  0.02623 -0.03622 -0.15322 1.022 1.17e-02 0.0225
## 32 -0.04066  0.04414  0.06730 1.081 2.31e-03 0.0373
## 33 -0.02928  0.03401  0.07773 1.063 3.07e-03 0.0263
## 34  0.90125 -0.88258  0.93533 1.107 4.13e-01 0.1941   *
## 35 -0.06459  0.05220 -0.19485 0.996 1.87e-02 0.0229
## 36 -0.20279  0.21667  0.29561 1.011 4.29e-02 0.0460
## 37 -0.03295  0.03619  0.05957 1.077 1.81e-03 0.0337
## 38 -0.02928  0.03401  0.07773 1.063 3.07e-03 0.0263
## 39 -0.04776  0.05244  0.08634 1.072 3.79e-03 0.0337
## 40 -0.05638  0.06727  0.17476 1.017 1.52e-02 0.0250
## 41  0.01680 -0.02320 -0.09815 1.050 4.88e-03 0.0225
## 42 -0.01163  0.01350  0.03086 1.073 4.87e-04 0.0263
## 43 -0.06381  0.07114  0.12910 1.053 8.42e-03 0.0306
## 44 -0.04262  0.04951  0.11315 1.051 6.48e-03 0.0263
## 45 -0.12909  0.14064  0.21955 1.024 2.39e-02 0.0361
## 46 -0.00283  0.00329  0.00751 1.074 2.88e-05 0.0263
## 47  0.03878 -0.04708 -0.13184 1.039 8.75e-03 0.0244
```

there are 4 influencial observation 11,20,30,34

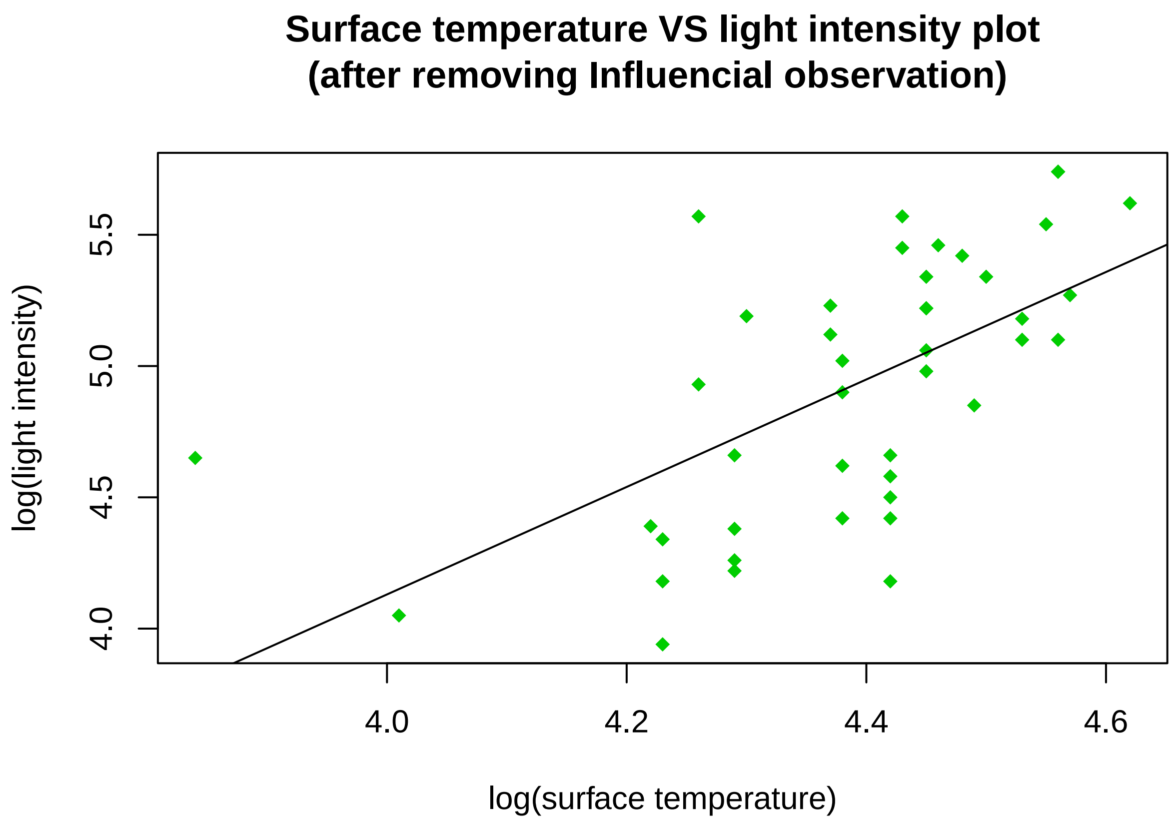
```
df2_updated<-df2[-c(11,20,30,34),]

plot(df2_updated$x_var, df2_updated$y_var,col="green3",pch=18,
     xlab = "log(surface temperature)",ylab = "log(light intensity)",
     main = " Surface temperature VS light intensity plot \n(after removing Influencial observation) \n
```




```r
ssr2_updated<- lm(y_var~x_var,df2_updated)
summary(ssr2_updated)
```

```
##
## Call:
## lm(formula = y_var ~ x_var, data = df2_updated)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.8097 -0.3088 -0.0267  0.2866  0.9078
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -4.0565     1.8441  -2.200   0.0335 *
## x_var         2.0467     0.4202   4.871  1.7e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4058 on 41 degrees of freedom
## Multiple R-squared:  0.3666, Adjusted R-squared:  0.3511
## F-statistic: 23.73 on 1 and 41 DF,  p-value: 1.697e-05
```

```r
plot(df2_updated$x_var,df2_updated$y_var,col="green3",pch=18,
     xlab = "log(surface temperature)",ylab = "log(light intensity)",
     main = " Surface temperature VS light intensity plot \n(after removing Influencial observation) \n
abline(ssr2_updated)
```



The R^2 is still too low

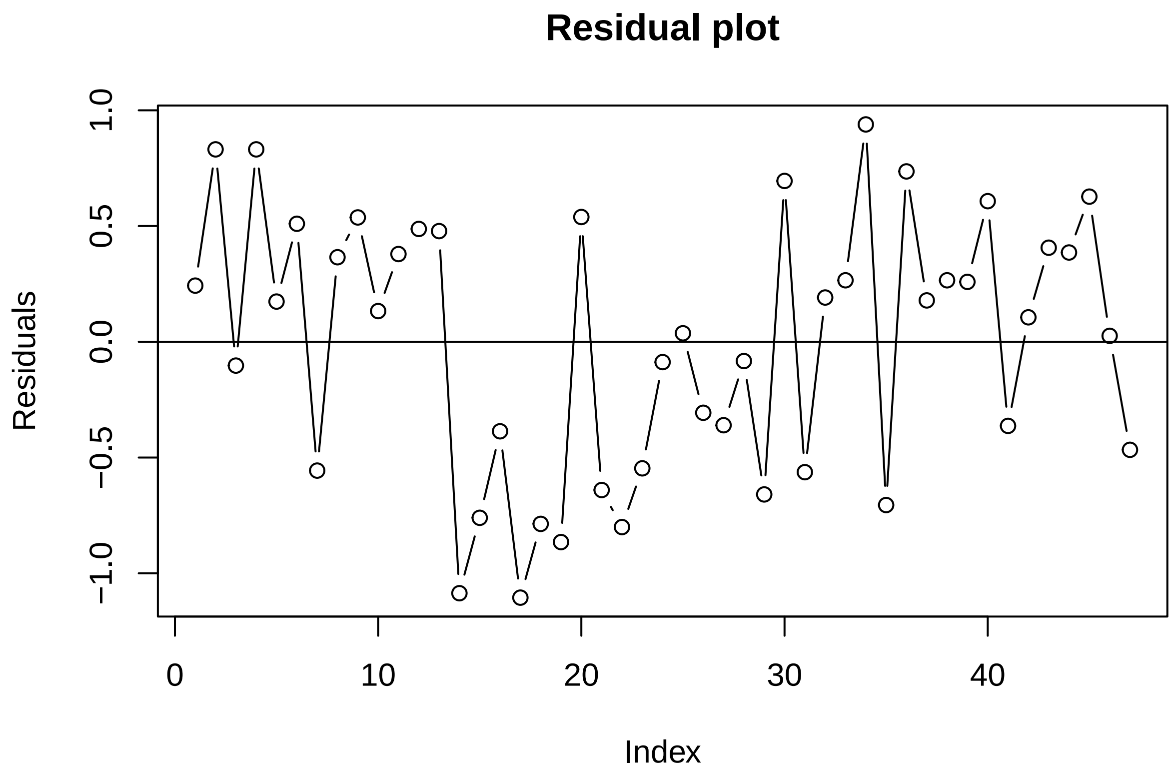
```
influence.measures(ssr2_updated)
```

```
## Influence measures of
##   lm(formula = y_var ~ x_var, data = df2_updated) :
##
##        dfb.1_  dfb.x_vr    dffit cov.r   cook.d    hat inf
## 1   0.019135 -0.014733  0.13224 1.038 8.80e-03 0.0235
## 2  -0.212058  0.218403  0.28649 1.039 4.06e-02 0.0555
## 3   0.092908 -0.089463  0.13730 1.071 9.55e-03 0.0404
## 4  -0.212058  0.218403  0.28649 1.039 4.06e-02 0.0555
## 5   0.107682 -0.101916  0.20123 1.020 2.01e-02 0.0313
## 6  -0.069875  0.074959  0.16789 1.033 1.41e-02 0.0290
## 7   1.958271 -1.941804  2.01108 1.124 1.74e+00 0.3435   *
## 8   0.012766 -0.013127 -0.01683 1.117 1.45e-04 0.0594
## 9   0.335272 -0.322841  0.49547 0.834 1.10e-01 0.0404   *
## 10  0.012929 -0.009955  0.08935 1.058 4.06e-03 0.0235
## 12 -0.044465  0.050198  0.17727 1.015 1.56e-02 0.0253
## 13 -0.071409  0.075440  0.14077 1.055 1.00e-02 0.0326
## 14 -0.117267  0.115812 -0.12436 1.269 7.91e-03 0.1752   *
## 15 -0.124623  0.118616 -0.21642 1.016 2.32e-02 0.0332
## 16  0.030585 -0.035905 -0.16194 1.023 1.31e-02 0.0245
## 17 -0.292884  0.284074 -0.39008 0.960 7.27e-02 0.0495
## 18  0.062883 -0.073819 -0.33295 0.873 5.12e-02 0.0245
## 19 -0.182595  0.177102 -0.24319 1.045 2.95e-02 0.0495
## 21 -0.091667  0.087248 -0.15919 1.048 1.28e-02 0.0332
```

```
## 22 -0.135792  0.129246 -0.23581 1.004 2.74e-02 0.0332
## 23  0.043059 -0.050547 -0.22798 0.973 2.53e-02 0.0245
## 24  0.073213 -0.076931 -0.13381 1.062 9.06e-03 0.0347
## 25  0.003317 -0.001886  0.04271 1.072 9.33e-04 0.0233
## 26  0.024501 -0.028762 -0.12972 1.042 8.48e-03 0.0245
## 27 -0.016826  0.016015 -0.02922 1.085 4.37e-04 0.0332
## 28 -0.000232  0.000132 -0.00298 1.076 4.55e-06 0.0233
## 29 -0.087059  0.084596 -0.11294 1.097 6.50e-03 0.0530
## 31 -0.014681  0.008349 -0.18902 0.999 1.76e-02 0.0233
## 32  0.079406 -0.081782 -0.10728 1.102 5.87e-03 0.0555
## 33 -0.025667  0.027851  0.07036 1.071 2.53e-03 0.0276
## 35 -0.112197  0.108822 -0.14943 1.082 1.13e-02 0.0495
## 36 -0.139064  0.142153  0.16807 1.126 1.44e-02 0.0817
## 37  0.042018 -0.043542 -0.06240 1.096 1.99e-03 0.0453
## 38 -0.025667  0.027851  0.07036 1.071 2.53e-03 0.0276
## 39  0.012734 -0.013196 -0.01891 1.100 1.83e-04 0.0453
## 40 -0.057136  0.064503  0.22778 0.978 2.53e-02 0.0253
## 41 -0.008558  0.004867 -0.11019 1.049 6.14e-03 0.0233
## 42 -0.001349  0.001464  0.00370 1.080 7.01e-06 0.0276
## 43 -0.053111  0.055562  0.09103 1.079 4.22e-03 0.0371
## 44 -0.044089  0.047840  0.12085 1.053 7.39e-03 0.0276
## 45 -0.120480  0.124308  0.16731 1.080 1.42e-02 0.0519
## 46  0.010786 -0.011704 -0.02957 1.079 4.48e-04 0.0276
## 47  0.036764 -0.043158 -0.19466 1.000 1.87e-02 0.0245
```

Residual Plot

```r
res2<- residuals(ssr2)
plot(res2,type='b', ylab="Residuals",main="Residual plot")
abline(h=0)
```




```r
#plot(density(res2))
```

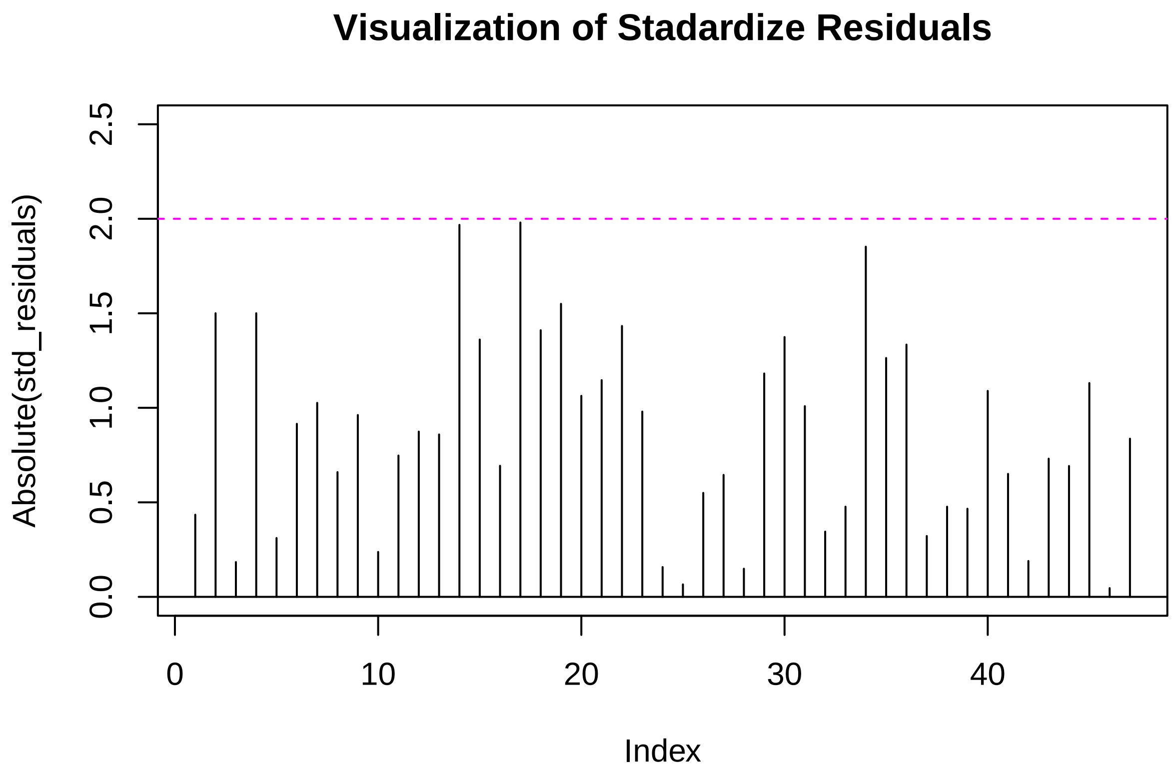
```r
cbind(res2)
```

```
##           res2
## 1   0.24267057
## 2   0.83119831
## 3  -0.10279285
## 4   0.83119831
## 5   0.17373930
## 6   0.50986792
## 7  -0.55638047
## 8   0.36533134
## 9   0.53720715
## 10  0.13267057
## 11  0.37896317
## 12  0.48746880
## 13  0.47813400
## 14 -1.08611882
## 15 -0.76039374
## 16 -0.38666423
## 17 -1.10519197
## 18 -0.78666423
## 19 -0.86519197
## 20  0.53896317
## 21 -0.64039374
## 22 -0.80039374
```

```
## 23 -0.54666423
## 24 -0.08773296
## 25  0.03680361
## 26 -0.30666423
## 27 -0.36039374
## 28 -0.08319639
## 29 -0.65932501
## 30  0.69483014
## 31 -0.56319639
## 32  0.19119831
## 33  0.26573488
## 34  0.93896317
## 35 -0.70519197
## 36  0.73599654
## 37  0.17879919
## 38  0.26573488
## 39  0.25879919
## 40  0.60746880
## 41 -0.36319639
## 42  0.10573488
## 43  0.40640007
## 44  0.38573488
## 45  0.62706527
## 46  0.02573488
## 47 -0.46666423
```

Standardize Residuals

```
res2_std <- rstandard(ssr2)

plot(abs(res2_std),type = 'h', main = "Visualization of Stadardize Residuals",
     ylab="Absolute(std_residuals)",
     ylim = c(0,2.5))
abline(h=2,col='magenta',lty=2)
abline(h=0)
```



ordering in decreasing order

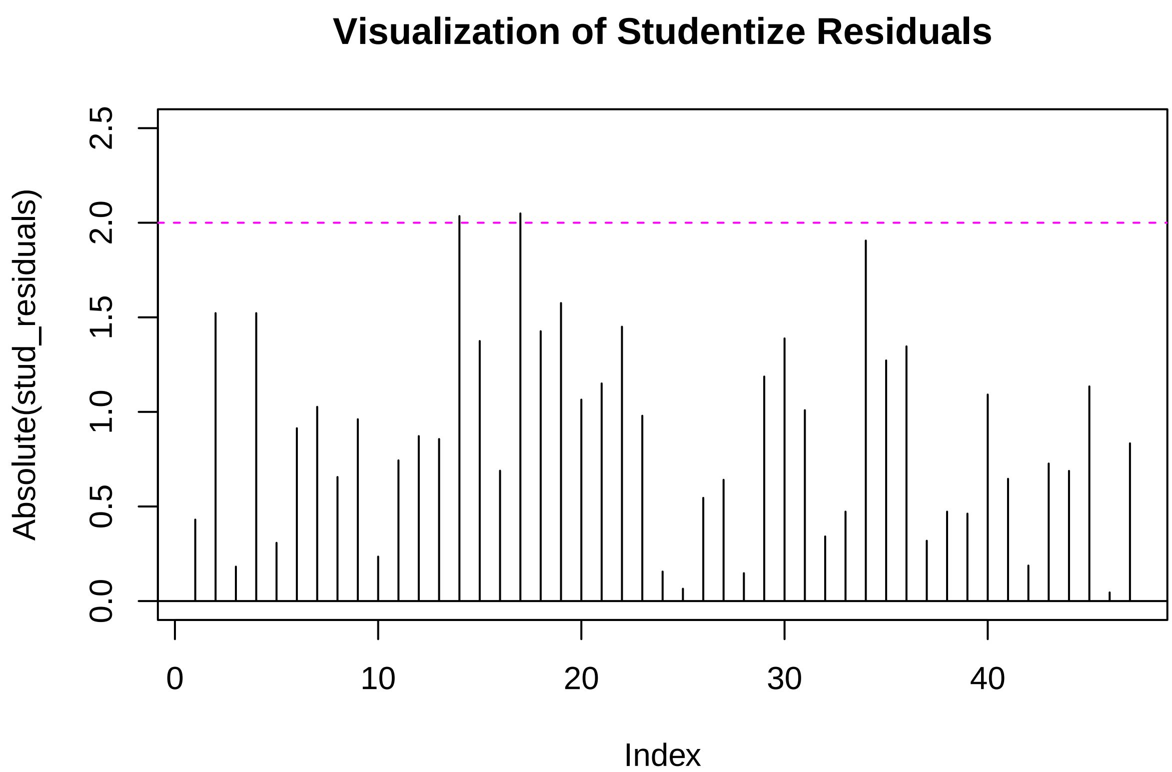
```
# after ordering the standardize residuals, from increasing to decreasig order in absolute values
res2_std[order(-abs(res2_std))]
```

```
##          17          14          34          19           4           2
## -1.98019464 -1.96777866  1.85243838 -1.55018182  1.50038693  1.50038693
##          22          18          30          15          36          35
## -1.43295112 -1.41054001  1.37442043 -1.36133881  1.33453895 -1.26350661
##          29          21          45          40          20           7
## -1.18159009 -1.14650188  1.13116891  1.08956114  1.06329630 -1.02625224
##          31          23           9           6          12          13
## -1.00889125 -0.98020444  0.96203147  0.91548130  0.87432813  0.85922888
##          47          11          43          16          44           8
## -0.83675925  0.74763947  0.73101644 -0.69331406  0.69233200  0.65990504
##          41          27          26          33          38          39
## -0.65061791 -0.64521571 -0.54986887  0.47695132  0.47695132  0.46627870
##           1          32          37           5          10          42
##  0.43463770  0.34512996  0.32214264  0.31103457  0.23762103  0.18977709
##           3          24          28          25          46
## -0.18408161 -0.15773325 -0.14903524  0.06592876  0.04618997
```

Studentize Residual

```
res2_stu<- rstudent(ssr2)

plot(abs(res2_stu),type = 'h',
```

```r
      main = "Visualization of Studentize Residuals",
     ylab="Absolute(stud_residuals)" ,
     ylim=c(0,2.5))
abline(h=2,col='magenta',lty=2)
abline(h=0)
```

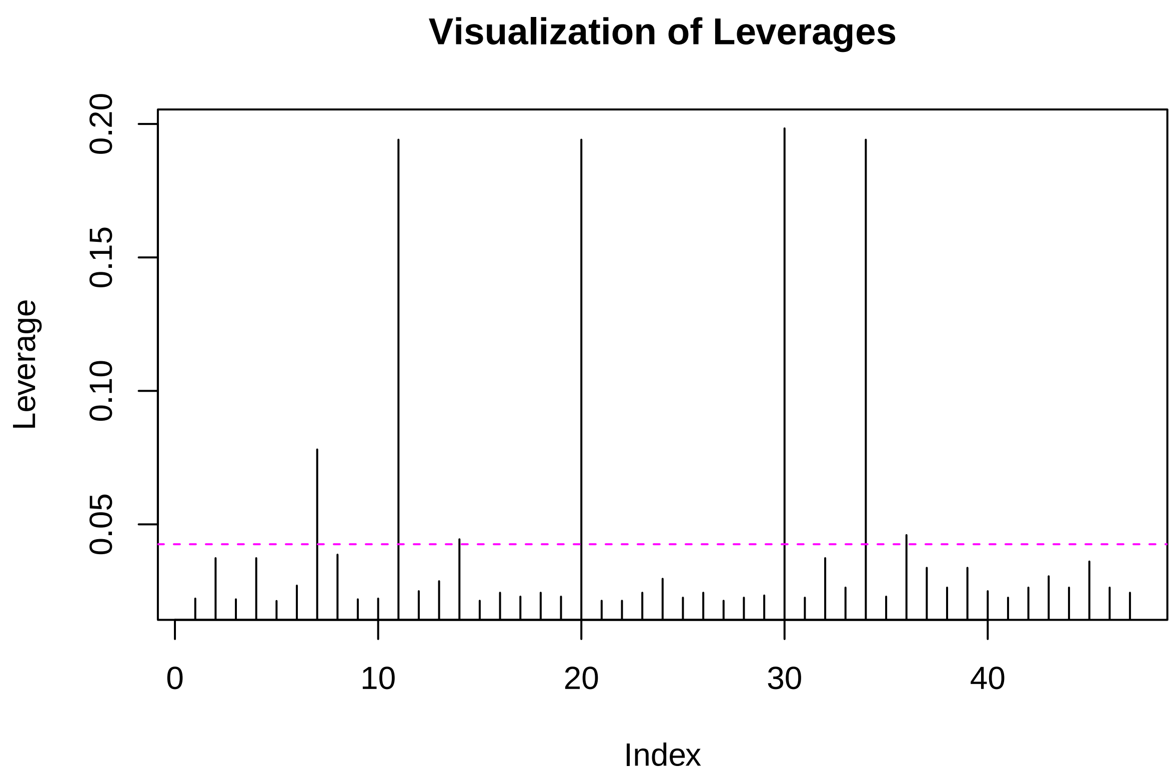



```r
# arranged in decreasing order in absolute values
res2_stu[order(-abs(res2_stu))]
```

```
##          17          14          34          19           4           2
## -2.04939273 -2.03532888  1.90584720 -1.57550504  1.52218503  1.52218503
##          22          18          30          15          36          35
## -1.45041761 -1.42667524  1.38851971 -1.37473341  1.34654294 -1.27215850
##          29          21          45          40          20           7
## -1.18694552 -1.15062098  1.13477943  1.09188565  1.06487784 -1.02687315
##          31          23           9           6          12          13
## -1.00909609 -0.97976810  0.96121811  0.91380172  0.87199716  0.85668474
##          47          11          43          16          44           8
## -0.83392273  0.74392041  0.72717900 -0.68925846  0.68827163  0.65571202
##          41          27          26          33          38          39
## -0.64639564 -0.64097816 -0.54556080  0.47281869  0.47281869  0.46218659
##           1          32          37           5          10          42
##  0.43068622  0.34172621  0.31891110  0.30789035  0.23511352  0.18773174
##           3          24          28          25          46
## -0.18209335 -0.15601395 -0.14740637  0.06519525  0.04567494
```

Leverage

```
lev2<- hatvalues(ssr2)

plot(lev2,type='h',
     main = "Visualization of Leverages",
     ylab="Leverage")
abline(h=2/47,col='magenta',lty=2)
```




```
lev2[order(-lev2)]
```

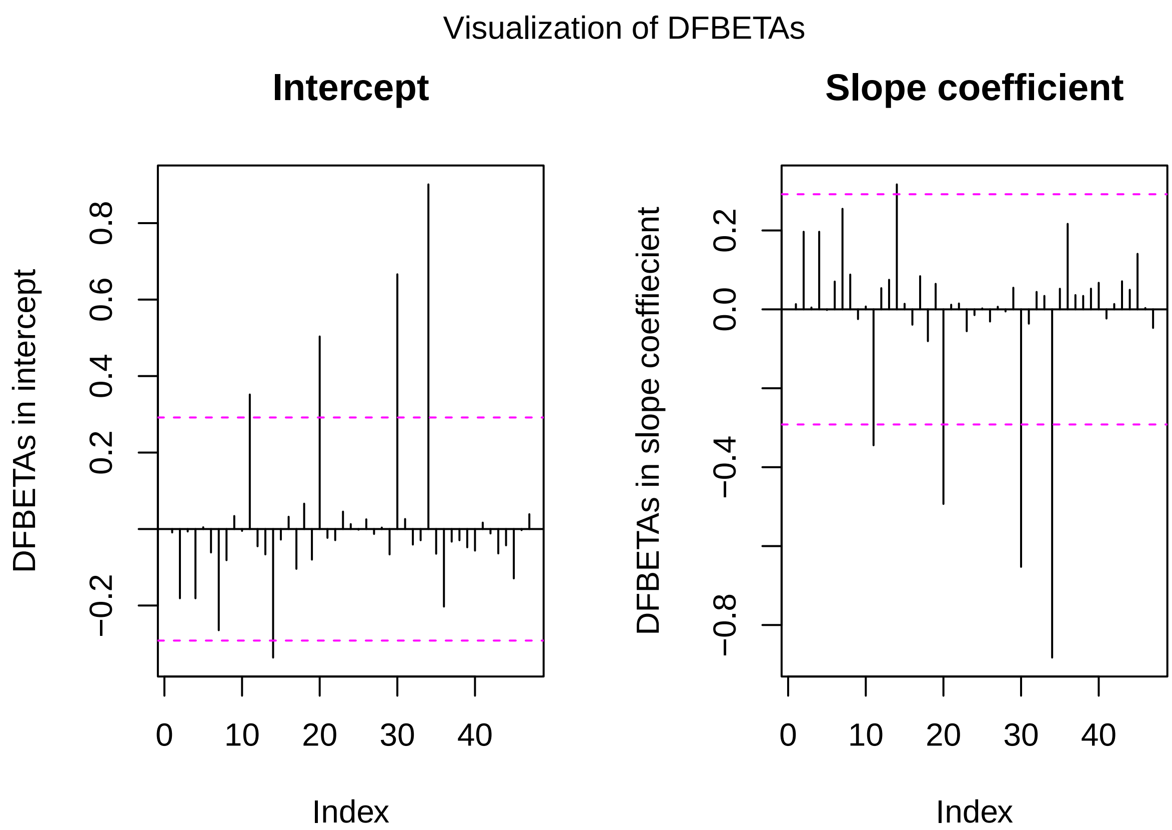
```
##         30         11         20         34          7         36         14 
## 0.19834440 0.19410341 0.19410341 0.19410341 0.07805447 0.04597716 0.04440927 
##          8          4         32          2         45         37         39 
## 0.03865181 0.03734096 0.03734096 0.03734096 0.03608151 0.03371684 0.03371684 
##         43         24         13          6         33         38         42 
## 0.03055537 0.02960436 0.02870476 0.02705977 0.02631438 0.02631438 0.02631438 
##         44         46         12         40         16         18         23 
## 0.02631438 0.02631438 0.02497782 0.02497782 0.02438666 0.02438666 0.02438666 
##         26         47         29         17         19         35         25 
## 0.02438666 0.02438666 0.02335854 0.02292159 0.02292159 0.02292159 0.02253604 
##         28         31         41         10          1          3          9 
## 0.02253604 0.02253604 0.02253604 0.02220190 0.02220190 0.02191917 0.02191917 
##         15         21         22         27          5 
## 0.02137941 0.02137941 0.02137941 0.02137941 0.02130230
```

```
which(lev2>(2/47))
```

```
##  7 11 14 20 30 34 36
##  7 11 14 20 30 34 36
```

Dfbeta

```
b<- dfbetas(ssr2)

par(mfrow=c(1,2))
plot(b[,1],type='h',
     main="Intercept",
     ylab="DFBETAs in intercept")
abline(h=c((-2/sqrt(47)),(2/sqrt(47))) , col='magenta',lty=2)
abline(h=0)

plot(b[,2],type='h',
     main="Slope coefficient",
     ylab="DFBETAs in slope coeffiecient")
abline(h=c((-2/sqrt(47)),(2/sqrt(47))) ,col='magenta',lty=2)
abline(h=0)
mtext("Visualization of DFBETAs", side = 3,line = -1, outer = TRUE)
```




```
which( abs(b[,1])> 2/sqrt(47))
```

```
## 11 14 20 30 34
```

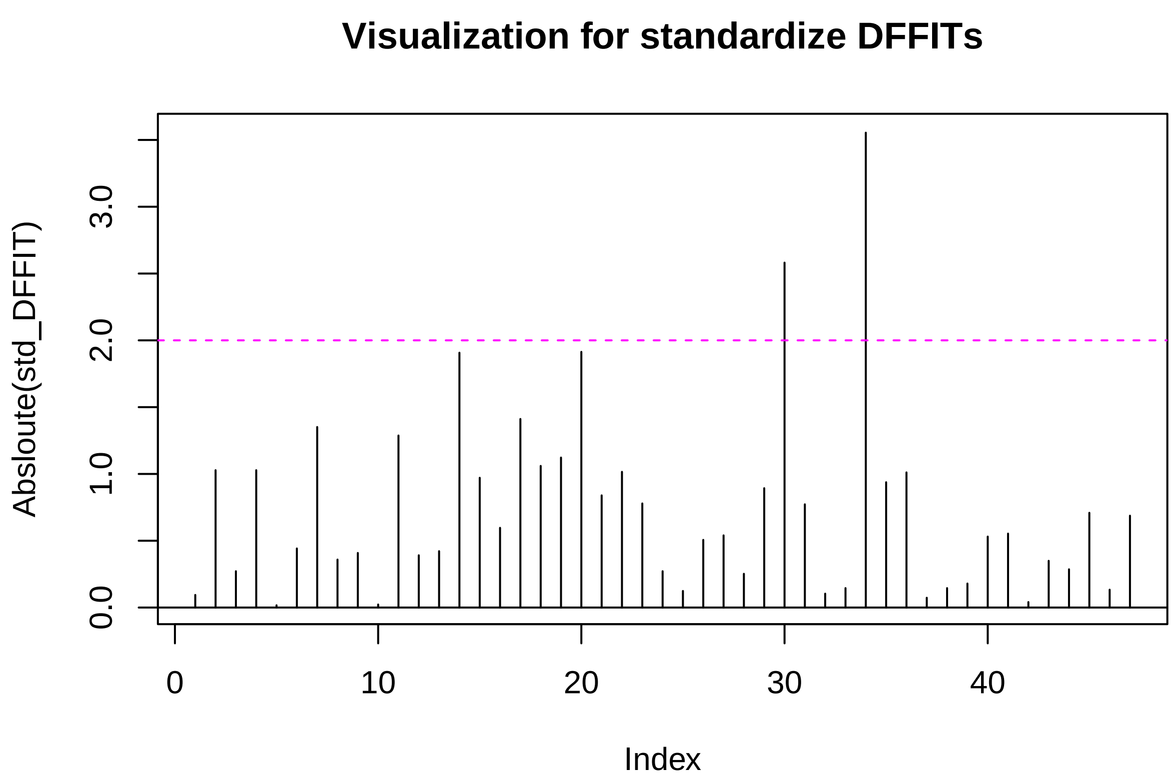
```
## 11 14 20 30 34
```

```
which(abs(b[,2]) > 2/sqrt(47))
```

```
## 11 14 20 30 34
## 11 14 20 30 34
```

DFFITS

```
bb<- dffits(ssr2)

v<- var(bb)*( (length(bb)-1)/length(bb) )

std_bb<- ( bb- mean(bb)) /sqrt(v)




plot(abs(std_bb),type='h',
     main="Visualization for standardize DFFITs",
     ylab="Absloute(std_DFFIT)")
abline(h=2,col="magenta",lty=2)
abline(h=0)
```



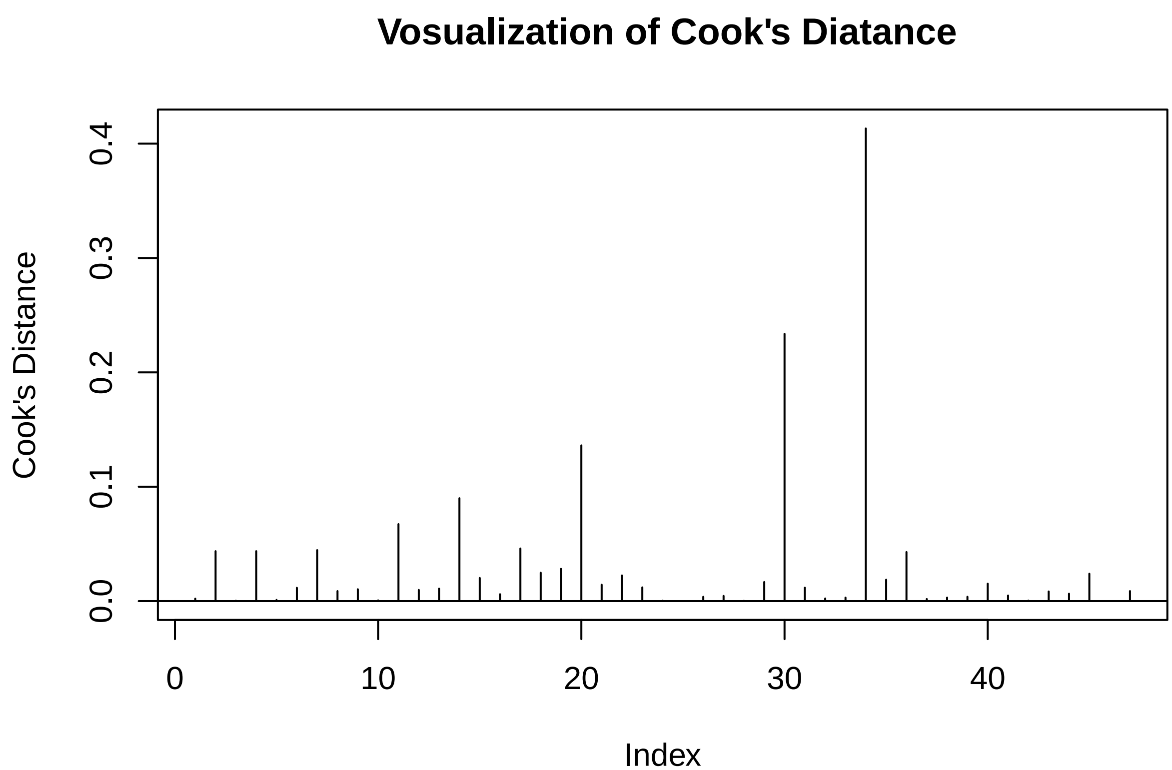

```
std_bb[order(-abs(std_bb))]
```

```
##            34            30            20            14            17             7
```

```
##  3.55430797  2.58179878  1.91379261 -1.90755967 -1.41119877 -1.35115466
##          11          19          18           4           2          22
##  1.28768706 -1.12269183 -1.06008356  1.02813491  1.02813491 -1.01564281
##          36          15          35          29          21          23
##  1.01148174 -0.97117773 -0.93801183 -0.89315170 -0.83950986 -0.77923141
##          31          45          47          16          41          27
## -0.77254839  0.70917297 -0.68757705 -0.59666494 -0.55364135 -0.54009057
##          40          26           6          13           9          12
##  0.53114630 -0.50636028  0.44224025  0.42187961  0.40845479  0.39125337
##           8          43          44           3          24          28
##  0.35910312  0.34964221  0.28623834 -0.27186396 -0.27182607 -0.25247749
##          39          33          38          46          25          32
##  0.17966122  0.14545127  0.14545127 -0.13366455 -0.12416226  0.10401035
##           1          37          42          10           5
##  0.09445067  0.07327967 -0.04083795 -0.02268845  0.01704376
```

Cook's Distance

```
cd<- cooks.distance(ssr2)

plot(cd,type='h', main=" Vosualization of Cook's Diatance",
     ylab="Cook's Distance")
abline(h=0)
```



```r
cd[order(-cd)]
```

```
##           34           30           20           14           11           17 
## 4.132486e-01 2.336906e-01 1.361546e-01 8.997550e-02 6.731445e-02 4.599398e-02 
##            7            4            2           36           19           18 
## 4.458315e-02 4.366058e-02 4.366058e-02 4.291566e-02 2.818711e-02 2.486654e-02 
##           45           22           15           35           29           40 
## 2.394800e-02 2.242922e-02 2.024341e-02 1.872580e-02 1.669607e-02 1.520594e-02 
##           21           23           31            6           13            9 
## 1.435823e-02 1.200820e-02 1.173372e-02 1.165485e-02 1.090914e-02 1.037046e-02 
##           12            8           47           43           44           16 
## 9.791721e-03 8.754312e-03 8.750756e-03 8.421488e-03 6.476989e-03 6.007649e-03 
##           41           27           39           26           33           38 
## 4.879765e-03 4.547379e-03 3.793181e-03 3.778877e-03 3.073919e-03 3.073919e-03 
##           32            1           37            5           10           42 
## 2.310193e-03 2.144696e-03 1.810543e-03 1.052847e-03 6.410336e-04 4.866670e-04 
##            3           24           28           25           46 
## 3.796996e-04 3.795101e-04 2.560500e-04 5.010681e-05 2.882967e-05
```

The more model

```r
df2_ott<- df2[-c(11,14,20,30,34),]


ssr3<- lm(y_var~x_var,df2_ott)
summary(ssr3)
```

```
## 
## Call:
## lm(formula = y_var ~ x_var, data = df2_ott)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -0.81089 -0.32092 -0.01575  0.28938  0.89870 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  -3.8378     2.0339  -1.887 0.066451 .  
## x_var         1.9974     0.4625   4.319 0.000101 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.4105 on 40 degrees of freedom
## Multiple R-squared:  0.318,  Adjusted R-squared:  0.3009 
## F-statistic: 18.65 on 1 and 40 DF,  p-value: 0.0001006
```

Here the p value fro f statistics is less than 0.05, so the coefficients are not equals. But the p value for intercept is 0.066 which is larger than 0.05, so the intercept term is statistically insignificant, so we need to drop it from the model.

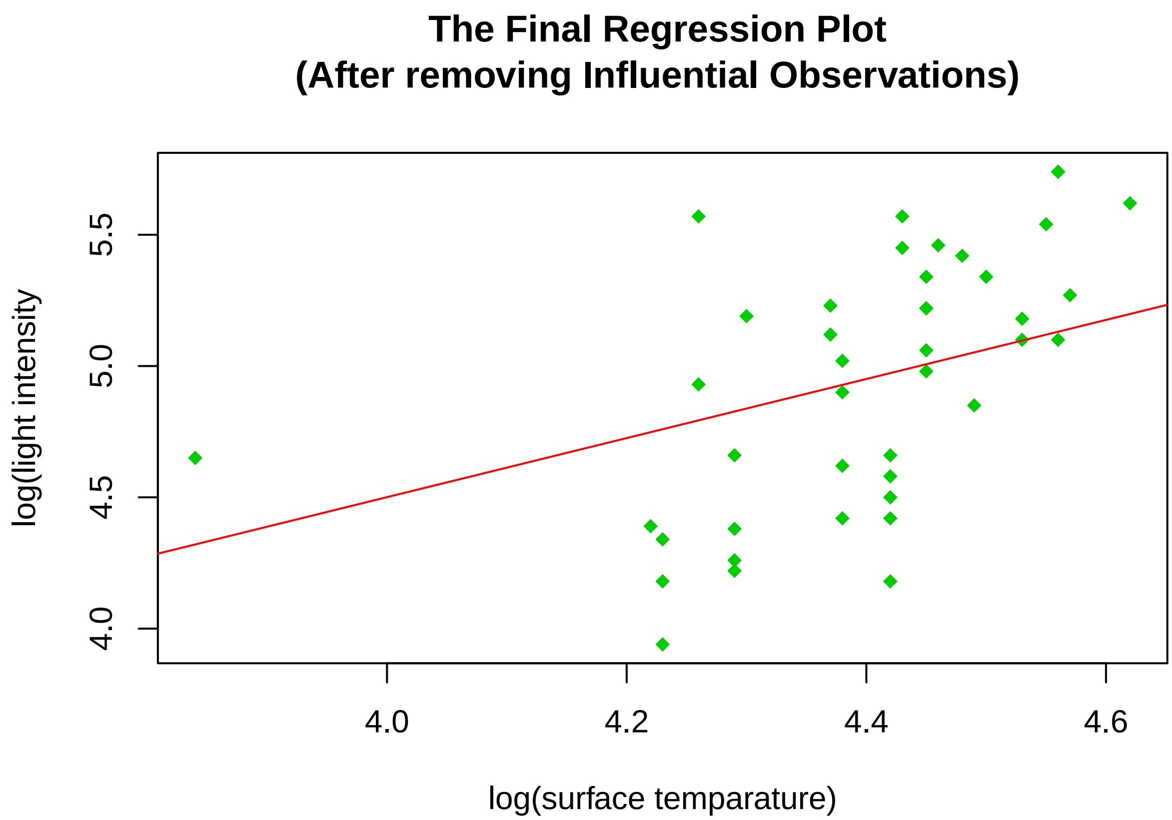
Model without intercept

```r
ssr3_updated<- lm(y_var~ 0+x_var, data = df2_ott)
summary(ssr3_updated)
```

```
##
## Call:
## lm(formula = y_var ~ 0 + x_var, data = df2_ott)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.81944 -0.38446  0.06802  0.33211  0.77681
##
## Coefficients:
##       Estimate Std. Error t value Pr(>|t|)
## x_var  1.12516    0.01485   75.79   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4231 on 41 degrees of freedom
## Multiple R-squared:  0.9929, Adjusted R-squared:  0.9927
## F-statistic:  5744 on 1 and 41 DF,  p-value: < 2.2e-16
```

The regression plot

```r
plot(df2_ott$x_var,df2_ott$y_var, pch=18,col="green3",
     main="The Final Regression Plot \n(After removing Influential Observations) \n",
     xlab="log(surface temparature)",
     ylab = "log(light intensity")

abline(ssr3_updated,col="red")
```



The residual standard deviation is very low for the model and the adjusted R-squared is 0.99 which is too good, so our models fits well here.

After deleting 4 obsn the model is good