outlier,Influence-Measure.R

Sharad Deshmukh

Outlier = An outlier is an extremum distinct point from dataset.

an outlier is an anomaly that occurs due to measurement errors but in other cases, it can occur because the experiment being observed experiences momentary but drastic turbulence. In either case, it is important to deal with outliers because they can adversely impact the accuracy of your results, especially in regression models.

outliers can be dangerous for your data science activities because most statistical parameters such as mean, standard deviation and correlation are highly sensitive to outliers. Consequently, any statistical calculation based on these parameters is affected by the presence of outliers.

Whether it is good or bad to remove outliers from your dataset depends on whether they affect your model positively or negatively. Remember that outliers aren't always the result of badly recorded observations or poorly conducted experiments. They may also occur due to natural fluctuations in the experiment and might even represent an important finding of the experiment.

Whether you're going to drop or keep the outliers requires some amount of investigation. However, it is not recommended to drop an observation simply because it appears to be an outlier.

Statisticians have devised several ways to locate the outliers in a dataset. The most common methods include the Z-score method and the Interquartile Range (IQR) method. However, I prefer the IQR method because it does not depend on the mean and standard deviation of a dataset and I'll be going over this method throughout the tutorial.

The interquartile range is the central 50% or the area between the 75^{th} and the 25^{th} percentile of a distribution. A point is an outlier if it is above the 75^{th} or below the 25^{th} percentile by a factor of 1.5 times the IQR.

For example, if

Q1= 25^{th} percentile

Q3= 75^{th} percentile

Then, IQR= Q3 – Q1

And an outlier would be a point below [Q1- (1.5)IQR] or above [Q3+(1.5)IQR].

##Q1)

**A soft drink bottler is analyzing the vending machine service routes in his distribution
system. He is interested in predicting the amount of time required by the route driver
to service the vending machines in an outlet. This service activity includes stocking
the machine with beverage products and minor maintenance or housekeeping. The
industrial engineer responsible for the study has suggested that the two most impor-
tant variables affecting the delivery time (Minutes) are the number of cases of product
stocked and the distance (Feet) walked by the route driver. The data are given in the
file \Delivery time data". Identify the leverage points and infuential points, if any.**

Compute MSRes and R2 for the original model and model after removing the influential points. What do you observe? Verify the model assumptions for both the models.

```
### Outlier and inluence measure

d=read.delim("C:\\Users\\Sharad Deshmukh\\Desktop\\MSC=SEM-II\\practical\\influence measure\\DeliveryTimeData.txt",header=TRUE)
names(d)

## [1] "Observation"         "Delivery.Time..y"    "Number.of.Cases..x1"
## [4] "Distance..x2..ft."

y=d$Delivery.Time..y
x1=d$Number.of.Cases..x1
x2=d$Distance..x2..ft.

###BOXPLOT METHOD
boxplot.stats(x2)
```

```
## $stats
## [1]  36 150 330 605 810
##
## $n
## [1] 25
##
## $conf
## [1] 186.22 473.78
##
## $out
## [1] 1460
```

which(x2==1460)

[1] 9

Above points checks the outlier point index in x2 variables which affect ever prediction

##Elimination of outlier

```
Q <- quantile(x2, probs=c(.25, .75), na.rm = FALSE)
iqr <- IQR(x2)
up <-  Q[2]+1.5*iqr # Upper Range
low<- Q[1]-1.5*iqr # Lower Range
eliminated<- subset(d, x2> (Q[1] - 1.5*iqr) & x2< (Q[2]+1.5*iqr))

n=length(y)
p=ncol(d)
p
```

```
## [1] 4
```

```
reg=lm(y~x1+x2)
#inf=influence.measures(reg)
#summary(inf)
summary(reg)
```

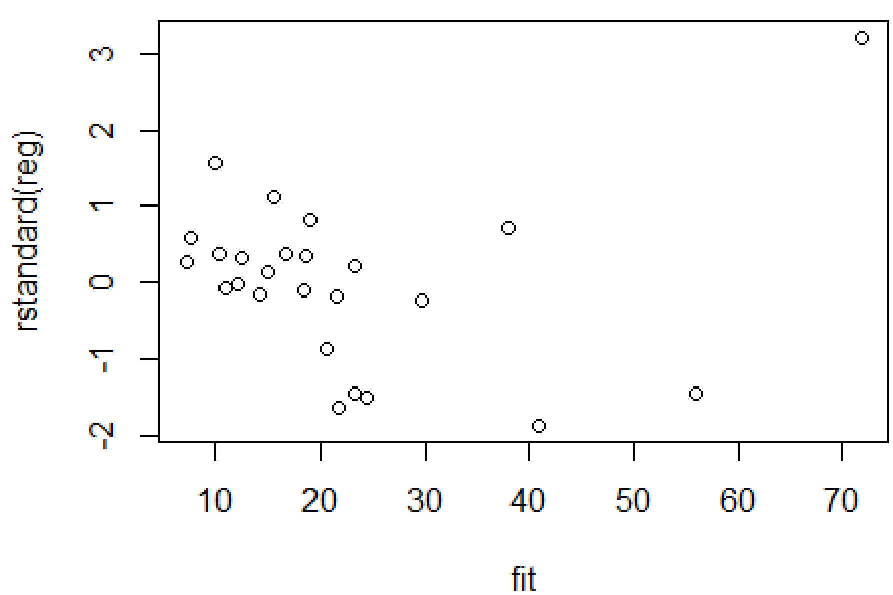
```
##
## Call:
## lm(formula = y ~ x1 + x2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.7880 -0.6629  0.4364  1.1566  7.4197
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2.341231   1.096730   2.135 0.044170 *
## x1          1.615907   0.170735   9.464 3.25e-09 ***
## x2          0.014385   0.003613   3.981 0.000631 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.259 on 22 degrees of freedom
## Multiple R-squared:  0.9596, Adjusted R-squared:  0.9559
## F-statistic: 261.2 on 2 and 22 DF,  p-value: 4.687e-16
```

```
res=reg$residuals
fit=reg$fitted.values
```

2)Second method of outlier detection=Residuals Vs Fitted plot (Y and X)

```
plot(fit,rstandard(reg))     ###we can see outlier
```

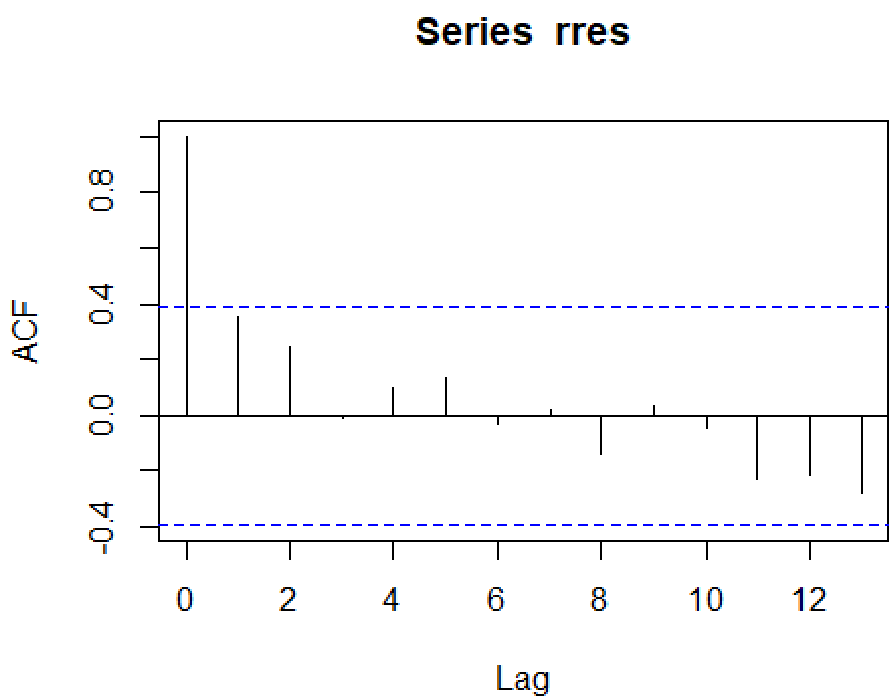



```
rres=rstandard(reg)
shapiro.test(rres)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  rres
## W = 0.92285, p-value = 0.05952
```

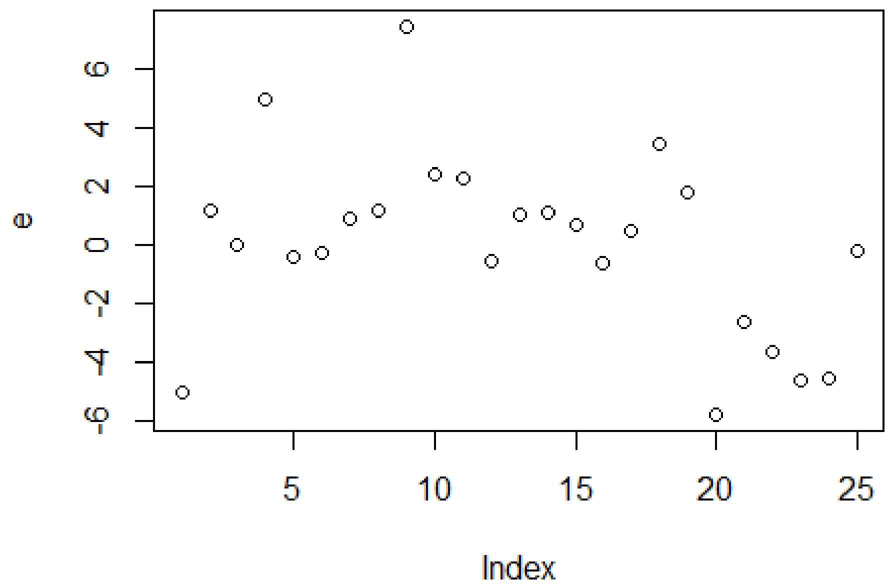
The above data is not normal

```
acf(rres)
```




```
###verification of influenial

yhat=reg$fitted.values
e=y-yhat
plot(e)
```




```
n=length(y)
x0=c(rep(1,n))
X=as.matrix(cbind(x0,x1,x2))
H=X%*%solve(t(X)%*%X)%*%t(X)
h=c()
for(i in 1:25)
{
  h[i]=H[i,i]
}
h

##  [1] 0.10180178 0.07070164 0.09873476 0.08537479 0.07501050 0.04286693
##  [7] 0.08179867 0.06372559 0.49829216 0.19629595 0.08613260 0.11365570
## [13] 0.06112463 0.07824332 0.04111077 0.16594043 0.05943202 0.09626046
## [19] 0.09644857 0.10168486 0.16527689 0.39157522 0.04126005 0.12060826
## [25] 0.06664345

c1=2*p/n

which(h>c1)  ###possible  levrage points

## [1]  9 22

##
anova(reg)

## Analysis of Variance Table
##
```

```
## Response: y
##           Df Sum Sq Mean Sq F value    Pr(>F)
## x1         1 5382.4  5382.4 506.619 < 2.2e-16 ***
## x2         1  168.4   168.4  15.851 0.0006312 ***
## Residuals 22  233.7    10.6
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

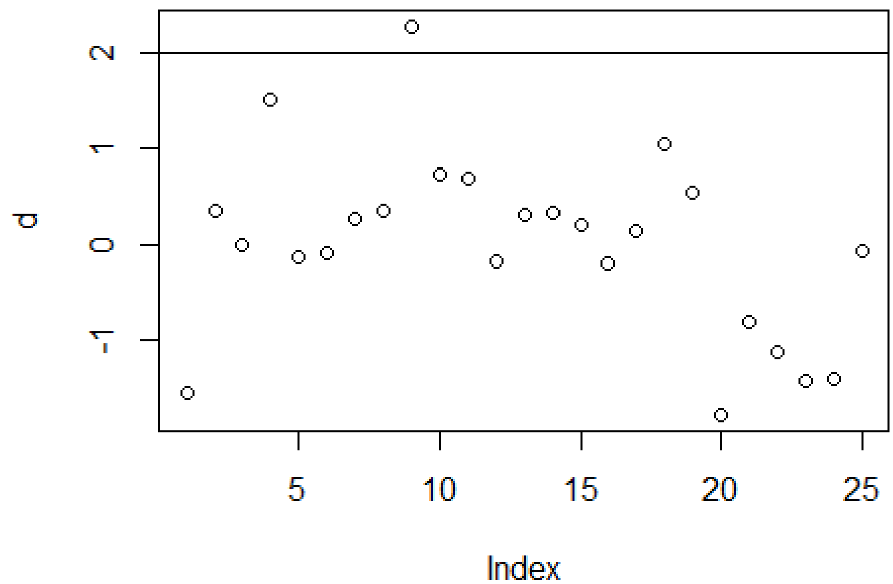
```
sig2=anova(reg)$"Mean Sq"[3]
sig2
```

```
## [1] 10.62417
```

```
###
###standardised residual
d=c()
for (i in 1:n)
{
  d[i]=e[i]/sqrt(sig2)
}
d
```

```
##  [1] -1.54260631  0.35170879 -0.01527661  1.51078203 -0.13634053 -
0.08884082
##  [7]  0.25912883  0.35484408  2.27635117  0.72907878  0.68645843 -
0.18194377
## [13]  0.31508443  0.32751789  0.20592338 -0.20338513  0.13387449
1.05803019
## [19]  0.55014821 -1.77573772 -0.80202492 -1.13101946 -1.41359270 -
1.40294240
## [25] -0.06522033
```

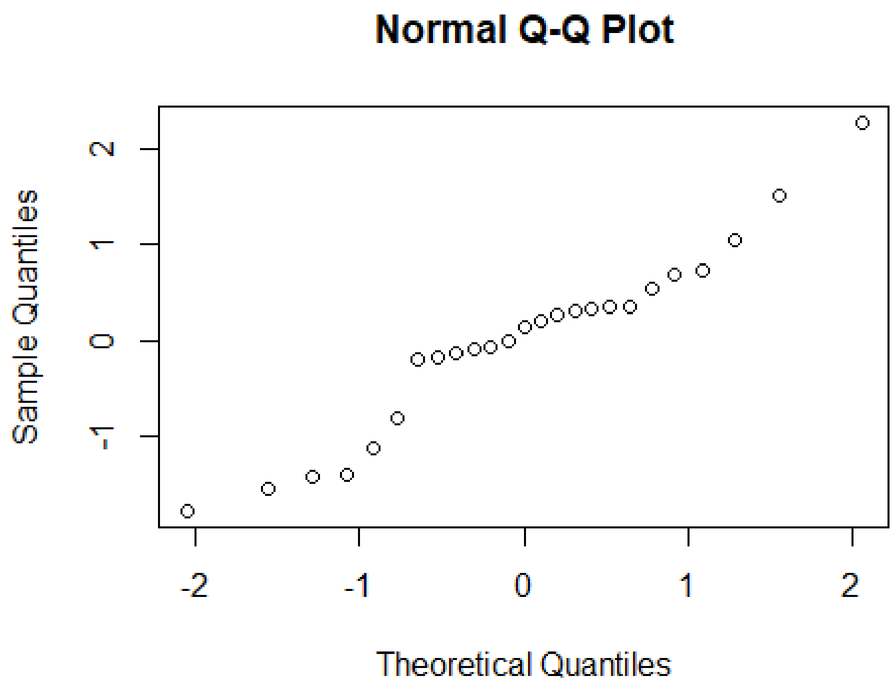
```
plot(d)
abline(h=2)
```




```
which(d>2)

## [1] 9

qqnorm(d)
```



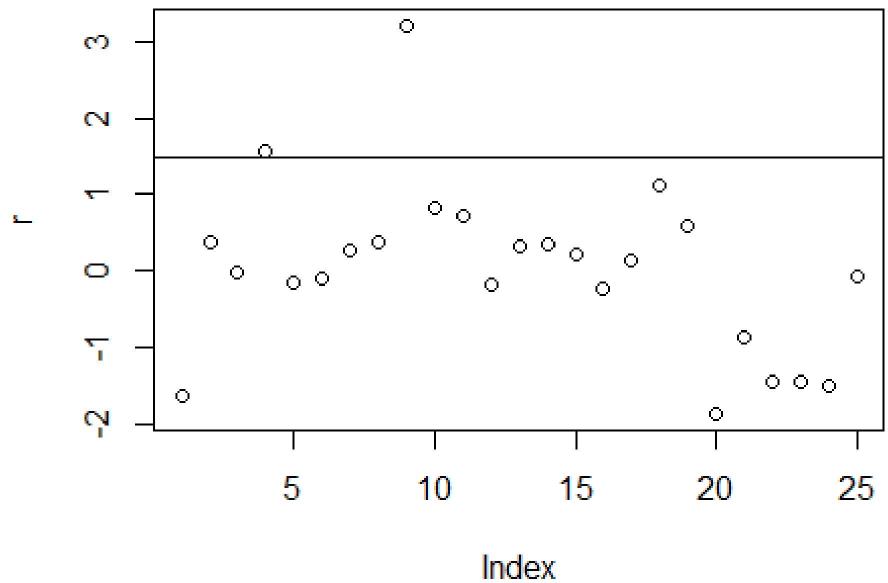

```
library(nortest)
ad.test(d)

##
##  Anderson-Darling normality test
##
## data:  d
## A = 0.59454, p-value = 0.11

###studentized rasidual
r=c()
for(i in 1:25)
{
  r[i]=e[i]/sqrt(sig2*(1-h[i]))
}
r

##  [1] -1.62767993  0.36484267 -0.01609165  1.57972040 -0.14176094 -
0.09080847
##  [7]  0.27042496  0.36672118  3.21376278  0.81325432  0.71807970 -
0.19325733
## [13]  0.32517935  0.34113547  0.21029137 -0.22270023  0.13803929
1.11295196
## [19]  0.57876634 -1.87354643 -0.87784258 -1.44999541 -1.44368977 -
1.49605875
## [25] -0.06750861
```

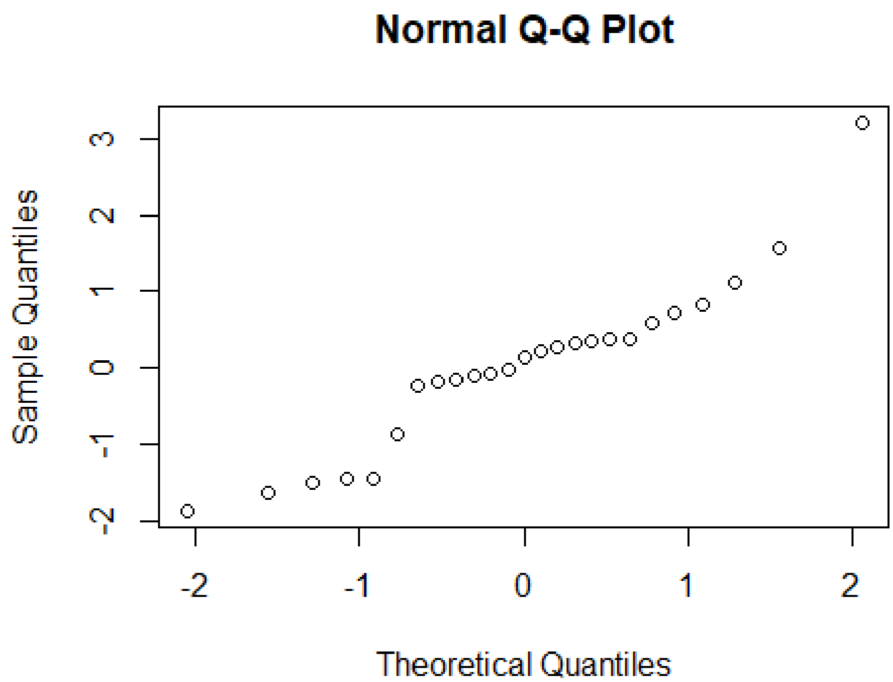
```r
plot(r)
abline(h=1.5)
```




```r
ad.test(r)

##
##  Anderson-Darling normality test
##
## data:  r
## A = 0.74447, p-value = 0.04558

qqnorm(r)
```




```
yn=rnorm(1000)
ks.test(r,yn)

##
##  Two-sample Kolmogorov-Smirnov test
##
## data:  r and yn
## D = 0.171, p-value = 0.4737
## alternative hypothesis: two-sided

shapiro.test(r)

##
##  Shapiro-Wilk normality test
##
## data:  r
## W = 0.92285, p-value = 0.05952

cvm.test(r)

##
##  Cramer-von Mises normality test
##
## data:  r
## W = 0.12683, p-value = 0.04487

ad.test(d)
```

```
##
##  Anderson-Darling normality test
##
## data:  d
## A = 0.59454, p-value = 0.11
```

```
###studentized resials
R=solve(t(X)%*%(X))%*%t(X)
dim(R)
```

```
## [1]  3 25
```

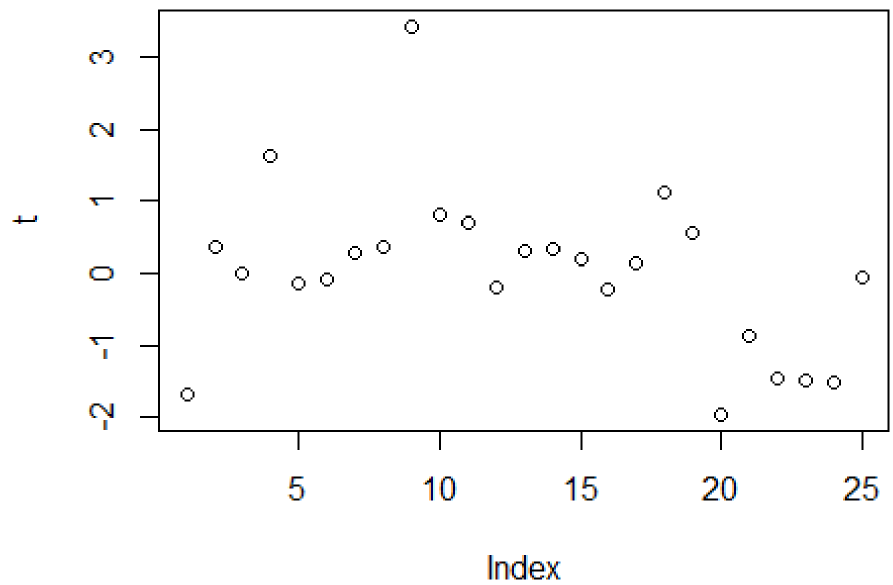
```
p=dim(X)[2]
p
```

```
## [1] 3
```

```
Si2=((n-p)*sig2-e^2/(1--h))/(n-p-1)
Si2
```

```
##         1         2         3         4         5         6         7
8
## 10.037427 11.071631 11.129972 10.066183 11.121332 11.126251 11.098678
11.070194
##         9        10        11        12        13        14        15
16
##  9.380404 10.905285 10.910587 11.115042 11.082747 11.079750 11.109474
11.112131
##        17        18        19        20        21        22        23
24
## 11.121521 10.613476 10.990428  9.682056 10.850811 10.665020 10.159201
10.241490
##        25
## 11.128062
```

```
###R-STudent RESIDUAL
t=e/(sqrt(Si2*(1-h)))
plot(t)
```




```
###Cooks D

D=(r^2*h)/(p*(1-h))
which(D>1)  ####influencial points

## [1] 9

#### DFITS
DEFT=t*(h/(1-h))^0.5
DEFT

##            1            2            3            4            5
6
## -0.563763359  0.098579002 -0.005203671  0.495836518 -0.039456472 -
0.018779085
##            7            8            9           10           11
12
##  0.078970117  0.093726236  3.408529655  0.396700636  0.217539557 -
0.067658482
##           13           14           15           16           17
18
##  0.081236436  0.097325093  0.042580987 -0.097128572  0.033914329
0.363410412
##           19           20           21           22           23
24
##  0.185915191 -0.660300936 -0.386516311 -1.161013836 -0.306271097 -
0.564302358
```

```
##           25
## -0.017625921
```

```
c3=2*sqrt(p/n)
which(DEFT>c3)
```

```
## 9
## 9
```

```
##DFBETAS
C=(R)%*%t(R)   ###same AS (X'X)^(-1)
solve(t(X)%*%X)
```

```
##               x0            x1            x2
## x0  1.132152e-01 -4.448593e-03 -8.367257e-05
## x1 -4.448593e-03  2.743783e-03 -4.785709e-05
## x2 -8.367257e-05 -4.785709e-05  1.228745e-06
```

```
DFB=matrix(rep(0,n*p),nrow=n,ncol=p)
for(j in 1:p)
{
  for(i in 1:n)
  {
    DFB[i,j]=(t(R[j,i])*t[i])/(sqrt(C[j,j]*(1-h[i])))
  }
}
DFB
```

```
##               [,1]          [,2]         [,3]
##  [1,] -0.184942352  0.4062054300 -0.429463273
##  [2,]  0.089757228 -0.0477450548  0.014408364
##  [3,] -0.003515173  0.0039483483 -0.002846465
##  [4,]  0.447483649  0.0874050196 -0.270662685
##  [5,] -0.031672081 -0.0132992645  0.024238911
##  [6,] -0.014681254  0.0017920792  0.001078969
##  [7,]  0.078051604 -0.0222727033 -0.011016025
##  [8,]  0.071176586  0.0333700389 -0.053804139
##  [9,] -2.043603382  0.7368690854  1.196097344
## [10,]  0.107374664 -0.3364560522  0.339603958
## [11,] -0.034209486  0.0923515481 -0.002681153
## [12,] -0.030263683 -0.0486580049  0.053964025
## [13,]  0.072346348 -0.0356113167  0.011331953
## [14,]  0.049497602 -0.0670609871  0.061792937
## [15,]  0.022277322 -0.0047891216  0.006837692
## [16,] -0.002692320  0.0644001282 -0.084160493
## [17,]  0.028854152  0.0064873345 -0.015695744
## [18,]  0.247266018  0.1887468741 -0.271014465
## [19,]  0.172324296  0.0235417382 -0.098834513
## [20,]  0.165167332 -0.2113239930 -0.091328559
## [21,] -0.161101383 -0.2956568093  0.334687530
## [22,]  0.387219262 -0.9962208835  0.556823134
```

```
## [23,] -0.159193455  0.0371393445 -0.052434967
## [24,] -0.118287013  0.3997787484 -0.459874957
## [25,] -0.016815806  0.0008498869  0.005592120
```

```
c4=2/sqrt(n)
which(abs(DFB[,1])>c4)   ###FOr B1
```

```
## [1] 4 9
```

```
which(abs(DFB[,2])>c4)   ###for B2
```

```
## [1]  1  9 22
```

```
#####COVRATIO
CVR=(Si2^p)/(Si2^p*(1-h))
CVR          ##high CV###9th observation is most influencer which is smaller
than 1
```

```
##        1        2        3        4        5        6        7        8
## 1.113340 1.076081 1.109551 1.093344 1.081093 1.044787 1.089086 1.068063
##        9       10       11       12       13       14       15       16
## 1.993192 1.244239 1.094251 1.128230 1.065104 1.084885 1.042873 1.198955
##       17       18       19       20       21       22       23       24
## 1.063187 1.106513 1.106744 1.113195 1.198002 1.643589 1.043036 1.137150
##       25
## 1.071402
```

```
r=cbind(D,DEFT,DFB,CVR)
r
```

```
##               D         DEFT
CVR
## 1  1.000921e-01 -0.563763359 -0.184942352  0.4062054300 -0.429463273
1.113340
## 2  3.375704e-03  0.098579002  0.089757228 -0.0477450548  0.014408364
1.076081
## 3  9.455785e-06 -0.005203671 -0.003515173  0.0039483483 -0.002846465
1.109551
## 4  7.764718e-02  0.495836518  0.447483649  0.0874050196 -0.270662685
1.093344
## 5  5.432217e-04 -0.039456472 -0.031672081 -0.0132992645  0.024238911
1.081093
## 6  1.231067e-04 -0.018779085 -0.014681254  0.0017920792  0.001078969
1.044787
## 7  2.171604e-03  0.078970117  0.078051604 -0.0222727033 -0.011016025
1.089086
## 8  3.051135e-03  0.093726236  0.071176586  0.0333700389 -0.053804139
1.068063
## 9  3.419318e+00  3.408529655 -2.043603382  0.7368690854  1.196097344
1.993192
## 10 5.384516e-02  0.396700636  0.107374664 -0.3364560522  0.339603958
1.244239
```

```
## 11 1.619975e-02  0.217539557 -0.034209486  0.0923515481 -0.002681153
1.094251
## 12 1.596392e-03 -0.067658482 -0.030263683 -0.0486580049  0.053964025
1.128230
## 13 2.294737e-03  0.081236436  0.072346348 -0.0356113167  0.011331953
1.065104
## 14 3.292786e-03  0.097325093  0.049497602 -0.0670609871  0.061792937
1.084885
## 15 6.319880e-04  0.042580987  0.022277322 -0.0047891216  0.006837692
1.042873
## 16 3.289086e-03 -0.097128572 -0.002692320  0.0644001282 -0.084160493
1.198955
## 17 4.013419e-04  0.033914329  0.028854152  0.0064873345 -0.015695744
1.063187
## 18 4.397807e-02  0.363410412  0.247266018  0.1887468741 -0.271014465
1.106513
## 19 1.191868e-02  0.185915191  0.172324296  0.0235417382 -0.098834513
1.106744
## 20 1.324449e-01 -0.660300936  0.165167332 -0.2113239930 -0.091328559
1.113195
## 21 5.086063e-02 -0.386516311 -0.161101383 -0.2956568093  0.334687530
1.198002
## 22 4.510455e-01 -1.161013836  0.387219262 -0.9962208835  0.556823134
1.643589
## 23 2.989892e-02 -0.306271097 -0.159193455  0.0371393445 -0.052434967
1.043036
## 24 1.023224e-01 -0.564302358 -0.118287013  0.3997787484 -0.459874957
1.137150
## 25 1.084694e-04 -0.017625921 -0.016815806  0.0008498869  0.005592120
1.071402

which(abs(DFB[,1])>c4) ###FOR VERIFICATION PROCESS ONLY

## [1] 4 9

###REMOVE this Observation
Y=y[-9]
X1=x1[-9]
X2=x2[-9]
length(Y)

## [1] 24

REG=lm(Y~X1+X2)
av=anova(REG)
newMse=av$"Mean Sq" [3];newMse

## [1] 5.904876

sig2=anova(reg)$"Mean Sq"[3]
sig2
```

```
## [1] 10.62417
```