Regression Diagnostics

Rakhi Singh

Indian Institute of Technology Bombay

General Preparations

Go to the specified directory and do

```
data=read.table("CH09TA01.txt", header=T)
```

Attach data `attach(data)`

Fit the full model

```
mod= lm(y ~ x1+x2+x3+x4+x5+x6+x7+x8)
```

Find the residuals:

`e = resid(mod)` OR `e = y- predict(mod)`

Calculate the X- Matrix

```
X= matrix(c(rep(1,nrow(data)), x1, x2, x3, x4, x5, x6, x7, x8),
nrow=nrow(data), ncol= 9)
```

Create the Hat-matrix: `H= X%*%solve(t(X)%*%X)%*%t(X)`

Identifying Outlying Y variables

Semi-studentized Residuals:

$$e_i^* = \frac{e_i}{\sqrt{(MSE)}}$$

Studentized Residuals:

$$r_i = \frac{e_i}{\sqrt{(MSE(1-h_{ii}))}}$$

Identifying Outlying Y variables

Semi-studentized Residuals:

$$e_i^* = \frac{e_i}{\sqrt{(MSE)}}$$

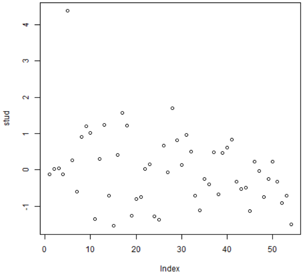
Studentized Residuals:

$$r_i = \frac{e_i}{\sqrt{(MSE(1-h_{ii}))}}$$

```
MSE=tail( anova(mod)[, 3], 1)
h=diag(H)
semistud<-NULL
stud<- NULL
semistud[1:nrow(data)]= 0
stud[1:nrow(data)]=0
for (i in (1:nrow(data))){
semistud[i] = e[i]/(sqrt(MSE))
stud[i] = e[i]/(sqrt(MSE*(1-h[i])))}
```

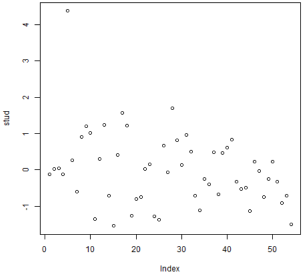
Analyzing

png('stud.png') plot(stud) dev.off()



Analyzing

png('stud.png') plot(stud) dev.off()



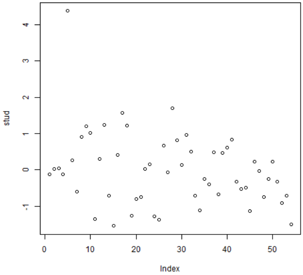
-0.11 0.02 0.05 -0.13 4.38 0.26 -0.61 0.90 1.21 1.01 -1.36 0.30 1.24 -0.72 -1.54 0.41 1.58 1.23 -1.26 -0.80 -0.75

0.03 0.15 -1.29 -1.37 0.67 -0.06 1.69 0.82 0.13 0.96 0.51 -0.72 -1.12 -0.25 -0.41 0.49 -0.67 0.46 0.61 0.83 -0.33

-0.54 -0.49 -1.14 0.22 -0.04 -0.74 -0.26 0.23 -0.33 -0.92 -0.71 -1.50

Analyzing

png('stud.png') plot(stud) dev.off()



-0.11 0.02 0.05 -0.13 4.38 0.26 -0.61 0.90 1.21 1.01 -1.36 0.30 1.24 -0.72 -1.54 0.41 1.58 1.23 -1.26 -0.80 -0.75

0.03 0.15 -1.29 -1.37 0.67 -0.06 1.69 0.82 0.13 0.96 0.51 -0.72 -1.12 -0.25 -0.41 0.49 -0.67 0.46 0.61 0.83 -0.33

-0.54 -0.49 -1.14 0.22 -0.04 -0.74 -0.26 0.23 -0.33 -0.92 -0.71 -1.50

Rule of thumb:

Greater than 3*sd(stud)= 3.065305 OR

Bonferroni test-procedure: $t_{(1-\frac{\alpha}{2n};n-p)} =\sim 3.43$

Identifying Outlying Y variables II

Deleted Residuals:

$$d_i = \frac{e_i}{1-h_{ii}}$$

Studentized deleted Residuals:

$$t_i = e_i \left\{ \frac{n-p-1}{(SSE(1-h_{ii})-e_i^2)} \right\}^{1/2}$$

Identifying Outlying Y variables II

Deleted Residuals:

$$d_i = \frac{e_i}{1-h_{ii}}$$

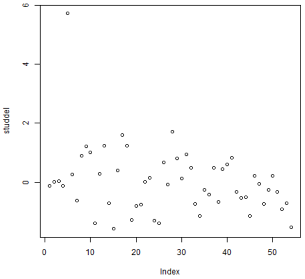
Studentized deleted Residuals:

$$t_i = e_i \left\{ \frac{n-p-1}{(SSE(1-h_{ii})-e_i^2)} \right\}^{1/2}$$

```
SSE=tail( anova(mod)[, 2], 1)
del<-NULL
studdel<- NULL
del[1:nrow(data)]= 0
studdel[1:nrow(data)]=0
for (i in (1:nrow(data))){
del[i] = e[i]/(1-h[i])
studdel[i] = e[i]*sqrt((nrow(data)-9-1)/(SSE*(1-h[i])-e[i]*e[i]))}
```

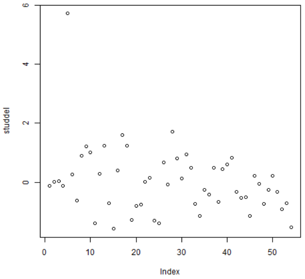
Analyzing

png('studdel.png') plot(studdel) dev.off()



Analyzing

png('studdel.png') plot(studdel) dev.off()



round(studdel,2)

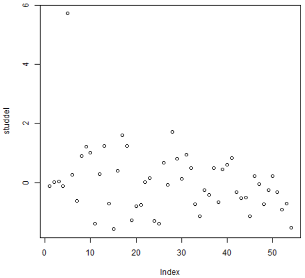
-0.11 0.02 0.05 -0.13 5.71 0.26 -0.60 0.90 1.21 1.01 -1.38 0.30 1.25 -0.72 -1.57 0.40 1.60 1.23 -1.27 -0.80 -0.75

0.03 0.15 -1.29 -1.38 0.67 -0.06 1.73 0.82 0.13 0.96 0.50 -0.72 -1.13 -0.24 -0.40 0.49 -0.66 0.45 0.61 0.83 -0.33

-0.53 -0.49 -1.15 0.22 -0.04 -0.74 -0.26 0.23 -0.32 -0.92 -0.70 -1.52

Analyzing

png('studdel.png') plot(studdel) dev.off()



round(studdel,2)

-0.11 0.02 0.05 -0.13 5.71 0.26 -0.60 0.90 1.21 1.01 -1.38 0.30 1.25 -0.72 -1.57 0.40 1.60 1.23 -1.27 -0.80 -0.75

0.03 0.15 -1.29 -1.38 0.67 -0.06 1.73 0.82 0.13 0.96 0.50 -0.72 -1.13 -0.24 -0.40 0.49 -0.66 0.45 0.61 0.83 -0.33

-0.53 -0.49 -1.15 0.22 -0.04 -0.74 -0.26 0.23 -0.32 -0.92 -0.70 -1.52

Rule of thumb:

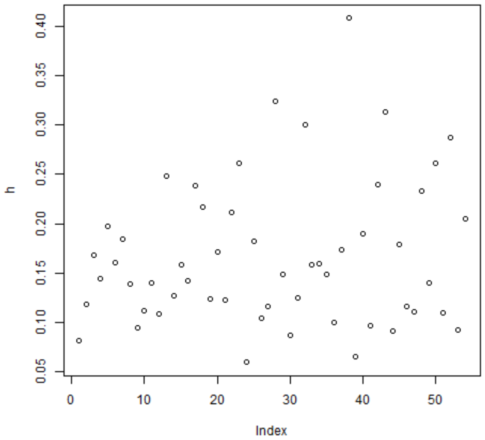
Greater than 3*sd(studdel)= 3.428044 OR

Bonferroni test-procedure: $t_{(1-\frac{\alpha}{2n};n-p-1)} =\sim 3.42$

Identifying outlying X observations

Leverage values h_{ii}

png('leverage.png') plot(h) dev.off()



Identifying outlying X observations

round(h,3)
0.082 0.119 0.168 0.144 0.197 0.160 0.184 0.139 0.095 0.112
0.141 0.109 0.248 0.127 0.159 0.142 0.238 0.217 0.124 0.172
0.123 0.212 0.261 0.060 0.182 0.105 0.116 0.323 0.148 0.088
0.125 0.300 0.159 0.160 0.149 0.100 0.174 0.408 0.066 0.190
0.097 0.239 0.313 0.091 0.179 0.116 0.111 0.233 0.140 0.261
0.110 0.287 0.093 0.205

Identifying outlying X observations

round(h,3)

0.082 0.119 0.168 0.144 0.197 0.160 0.184 0.139 0.095 0.112
0.141 0.109 0.248 0.127 0.159 0.142 0.238 0.217 0.124 0.172
0.123 0.212 0.261 0.060 0.182 0.105 0.116 0.323 0.148 0.088
0.125 0.300 0.159 0.160 0.149 0.100 0.174 0.408 0.066 0.190
0.097 0.239 0.313 0.091 0.179 0.116 0.111 0.233 0.140 0.261
0.110 0.287 0.093 0.205

Rule of thumb:

Greater than $\frac{2p}{n} = 0.337$

Point 28 and 38 needs to be seen further

Identifying Influential observations: DFFits

DFFits:

$$(DFFits)_i = \frac{\hat{Y}_i - \hat{Y}_{i(i)}}{\sqrt{(MSE_{(i)} h_{ii})}} = t_i \left\{ \frac{h_{ii}}{(1 - h_{ii})} \right\}^{1/2}$$

```
dffits<-NULL
dffits[1:nrow(data)]=0
for (i in (1:nrow(data))){
dffits[i] = studdel[i]*sqrt(h[i]/(1-h[i]))
}
```

Identifying Influential observations: DFFits

DFFits:

$$(DFFits)_i = \frac{\hat{Y}_i - \hat{Y}_{i(i)}}{\sqrt{(MSE_{(i)} h_{ii})}} = t_i \left\{ \frac{h_{ii}}{(1 - h_{ii})} \right\}^{1/2}$$

```
dffits<-NULL
dffits[1:nrow(data)]=0
for (i in (1:nrow(data))){
dffits[i] = studdel[i]*sqrt(h[i]/(1-h[i]))
}
```

Rule of thumb:

Influential if the value exceeds 1 (for small-medium datasets ($n \leq 60$) or if it exceeds $2\sqrt{\frac{p}{n}}$ (for large datasets)

Identifying Influential observations: DFBetas

DFBetas:

$$(DFBetas)_{k(i)} = \frac{b_k - b_{k(i)}}{\sqrt{(MSE_{(i)} c_{kk})}}$$

where c_{kk} is the kth diagonal of $(X^T X)^{-1}$.

```
c= diag(solve(t(X)%*%X))
mse=0
dfbetas1<-NULL
dfbetas1[1:54]=0
for (i in (1:nrow(data))){
fit<-lm(y[-i]~ x1[-i]+x2[-i]+x3[-i]+x4[-i]+x5[-i]+x6[-i]+x7[-i]+x8[-i])
mse<-tail( anova(fit)[, 3], 1)
dfbetas1[i] = (mod$coeff[2]-fit$coeff[2])/sqrt(mse* c[2]) }
```

Identifying Influential observations: DFBetas

DFBetas:

$$(DFBetas)_{k(i)} = \frac{b_k - b_{k(i)}}{\sqrt{(MSE_{(i)} c_{kk})}}$$

where c_{kk} is the kth diagonal of $(X^T X)^{-1}$.

```
c= diag(solve(t(X)%*%X))
mse=0
dfbetas1<-NULL
dfbetas1[1:54]=0
for (i in (1:nrow(data))){
fit<-lm(y[-i]~ x1[-i]+x2[-i]+x3[-i]+x4[-i]+x5[-i]+x6[-i]+x7[-i]+x8[-i])
mse<-tail( anova(fit)[, 3], 1)
dfbetas1[i] = (mod$coeff[2]-fit$coeff[2])/sqrt(mse* c[2]) }
```

```
ifelse(dfbetas(mod)[,]>1,1,0)
```

Identifying Influential observations: DFBetas

DFBetas:

$$(DFBetas)_{k(i)} = \frac{b_k - b_{k(i)}}{\sqrt{(MSE_{(i)} c_{kk})}}$$

where c_{kk} is the kth diagonal of $(X^T X)^{-1}$.

```
c= diag(solve(t(X)%*%X))
mse=0
dfbetas1<-NULL
dfbetas1[1:54]=0
for (i in (1:nrow(data))){
fit<-lm(y[-i]~ x1[-i]+x2[-i]+x3[-i]+x4[-i]+x5[-i]+x6[-i]+x7[-i]+x8[-i])
mse<-tail( anova(fit)[, 3], 1)
dfbetas1[i] = (mod$coeff[2]-fit$coeff[2])/sqrt(mse* c[2]) }
```

```
ifelse(dfbetas(mod)[,]>1,1,0)
```

Rule of thumb: Influential if the value exceeds 1 (for small-medium datasets ($n \leq 60$) or if it exceeds $2\sqrt{\frac{p}{n}}$ (for large datasets)