# 线性回归分析

Homework 7

*尹秋阳, 2015011468*

*2017 年 5 月 27 日*

## 目录

# 1 Problem 1(KNNL 10.11)

## 1.1 a

- Obtain studentlized deleted residuals
- Bonferroni outlier test procedure, $\alpha = 0.1$(decision rule and conclusion)

Studentlized deleted residuals are stored in variable **stl.del.resid**, the head of which is listed below:

```
regmodel=lm(data = dat,Y~.)
stl.del.resid=rstudent(regmodel)
head(stl.del.resid)
```

```
##           1           2           3           4           5           6
##  0.01155475 -0.93317867  0.40362503  0.21466620  0.59441714 -0.38167082
```

The boferroni outlier test is shown below:

Decision rule:

For each case:

$$t_i = \frac{e_i}{MSE_{(i)}(1 - h_{ii})}$$

If $|ti| < t(1 - \alpha/2n; n - p - 1)$ we conclude that case i is not outlying; otherwise we conclude that case i is an outltying case.(Here n=46 and p=4)

Conclusion:

```
outlierTest(regmodel)
```

```
##
## No Studentized residuals with Bonferonni p < 0.05
## Largest |rstudent|:
##     rstudent unadjusted p-value Bonferonni p
## 27 -1.974202           0.055121           NA
```

Here we found that the largest |rstudent| is 1.974202. And $t(1 - \alpha/2n; n - p - 1)$ is :

```
qt(1-0.1/2/46,df = 46-4-1)
```

```
## [1] 3.271524
```

Since all the absolute studentlized deleted residuals are below this value. We conclude that there is no Y outlier.

## 1.2 b

- Obtain the diagnal elements of the hat matrix
- Identify any outlying X observations

The diagnal elements are stored in hii:

```
hii=hatvalues(regmodel)
head(hii)
```

```
##          1          2          3          4          5          6
## 0.07819669 0.06706793 0.03717097 0.15361084 0.09673692 0.12857668
```

And according to the criteria on the text, the cutoff value is $2p/n$:

```
p=sum(hii)#p=3+1=4
n=nrow(dat)
(cutoff=2*p/n)
```

```
## [1] 0.173913
```

According to this criteria, the outlying X observations are:

```
hii[which(hii>cutoff)]
```

```
##         9        28        39
## 0.1842585 0.1860192 0.1809601
```

Those are case 9, 28 and 39.

## 1.3 d

- Obtain DFFITS, DFBETAS, and Cook's distance values for these cases(11,17,27)
- Assess its influence

I have asked Mr.Zhu about the criteria to identify the size of dataset. He believes that when $n/p = 5$, it is often considered **small**; when $n/p = 10$, it is often considered **medium**; and when $n/p = 15$, it is often considered **large**.

Based on this criteria, the dataset here is considered to be **medium**

The DFFITS are:

```
dffit=dffits(regmodel)
dffit[obj]
```

```
##         11         17         27
##  0.5688200  0.6657370 -0.6087397
```

According to the comments above. When the size of dataset is medium, the cut-off point is **1**. So the dffit result is just okay, and we just conclude that the case 11, 17 and 27 do **not** have much effect on the single fitted value.

The DFBETAS are:

```
dfbeta=dfbetas(regmodel)
dfbeta[obj,]
```

```
##    (Intercept)         X1         X2         X3
## 11  0.09910764 -0.3630892 -0.1899887 0.38998516
## 17 -0.44913479 -0.4711109  0.4432302 0.08926996
## 27 -0.01723432  0.4171827 -0.2498614 0.16136484
```

According to the comments above. When the size of dataset is medium, the cut-off point of dfbeta is also **1**. So the dfbeta result is just okay, and we just conclude that the case 11, 17 and 27 do **not** have much effect on the regression coefficients.

The cook's distances are:

```
(cutoff =4/n)
```

```
## [1] 0.08695652
```

```
cook.d=cooks.distance(regmodel)
cook.d[obj]
```

```
##         11         17         27
## 0.07656783 0.10513344 0.08666240
```

It can be seen above that case 11 does not exceeds the cut off edge, indicating not much effect on the all fitted values; case 17 obviously exceeds the cut off edge, indicating to have much effect on the all fitted values; case 27 is just below the cutting edge, which is in the grey area.

From my perspective, case 27 is just okay with the cook's distance, which means that it **does** not much effect on the all fitted values;

### 1.4 e

- Calculate the average absolute percentage difference in the fitted value with and without each of there cases
- What does this measure indicate about the influence of each of the cases?

Firstly, I have not found a proper function in R to calculate such kind of value. So I just write my own function as belows:

```
avr.percent.dif=function(indata){
  temp=NULL
  full=lm(data = indata,Y~.)
  X=indata[,-1]
  for (i in 1:nrow(indata)){
    dat=indata[-i,]
    reduced=lm(data =dat, Y~.)
    fit.values=predict(reduced,X,se.fit = F)
    ans=mean(abs((fit.values-full$fitted.values)/full$fitted.values))*100
    temp=c(temp,ans)
  }
  return (temp)
}
```

And the result has been shown below:

```
ans=avr.percent.dif(dat)
sort(abs(ans),decreasing = T)[1:5]
```

```
## [1] 1.324930 1.261763 1.187468 1.122050 1.100940
```

```
order(abs(ans),decreasing = T)[1:5]
```

```
## [1] 17 31 15 27 11
```

```
ans[obj]
```

```
## [1] 1.10094 1.32493 1.12205
```

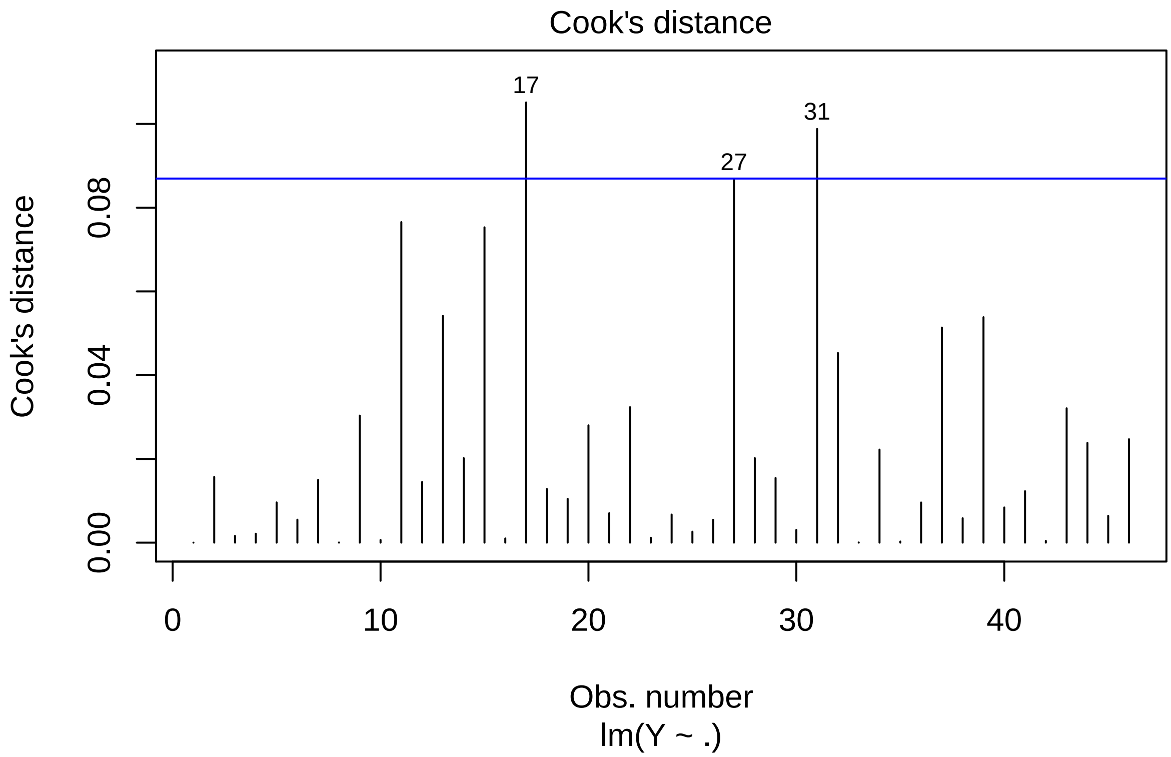
The five largest cases have been listed above.

It appears that all the cases have small impact on the total percentage, because the largest value is just 1.3249%. So based on this measure, we can conclude that the influence of each of these cases is not very much.

### 1.5 f

- Cook's distance plot
- Influential cases?

```
plot(regmodel, which=4, cook.levels=cutoff)
abline(h=cutoff,col="blue")
```

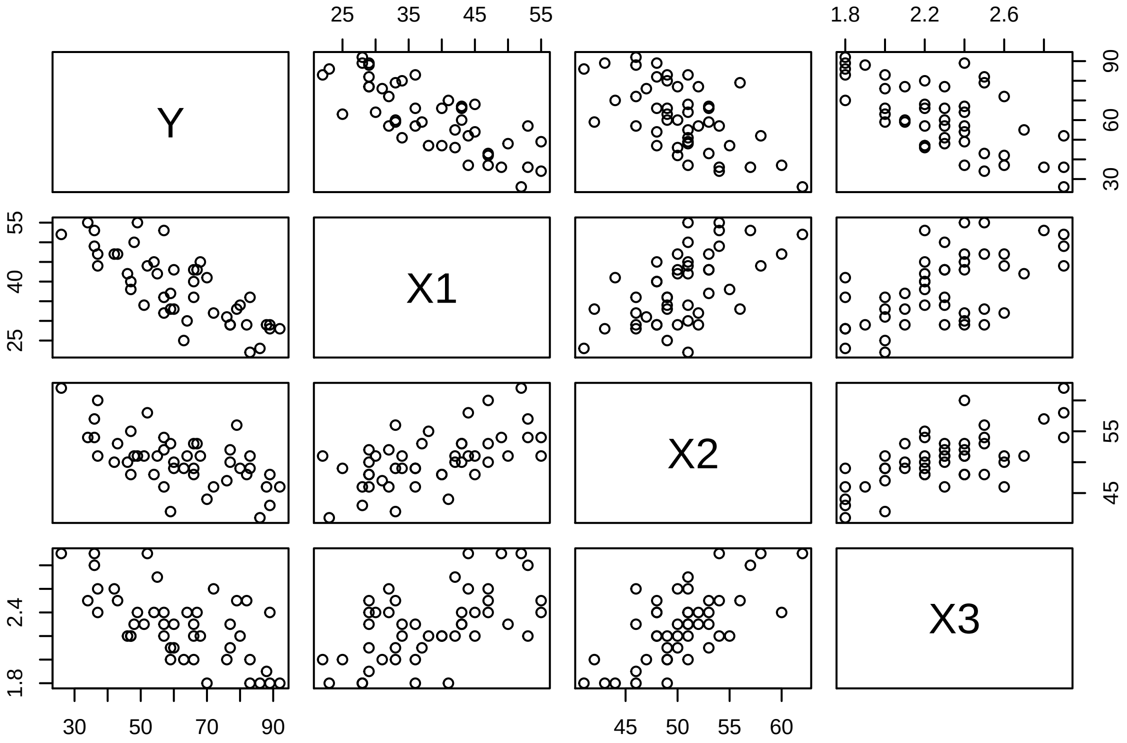


According to the plot, case 17,27 and 31 may be considered as **influential**.

## 2 Problem 2(KNNL 10.17)

### 2.1 a

```
pairs(dat)
```



```r
cor(dat)
```

```
##            Y         X1         X2         X3
## Y   1.0000000 -0.7867555 -0.6029417 -0.6445910
## X1 -0.7867555  1.0000000  0.5679505  0.5696775
## X2 -0.6029417  0.5679505  1.0000000  0.6705287
## X3 -0.6445910  0.5696775  0.6705287  1.0000000
```

It seems that X1, X2 and X3 have strong correlation, espectially for X1 and X2, which is as high as -0.7868. In addition, X1 and X3, as well as X2 and X3, have a large correlation up to 0.56

## 2.2 b

```r
(V=vif(regmodel))
```

```
##       X1       X2       X3
## 1.632296 2.003235 2.009062
```

Since the mean value here :

```r
mean(V)
```

```
## [1] 1.881531
```

largely exceeds 1, we can conclude that there **is** some kind of multicollinearity problem here.

They provide the same evidence with part a.

# 3 Problem 3

## 3.1 KNNL 10.21

### 3.1.1 a

- Variance inflation factors
- explain

```r
regmodel=lm(data = dat,Y~.)
(V=vif(regmodel))
```

```
##       X1       X2       X3
## 1.304608 1.300377 1.023997
```

```r
mean(V)
```

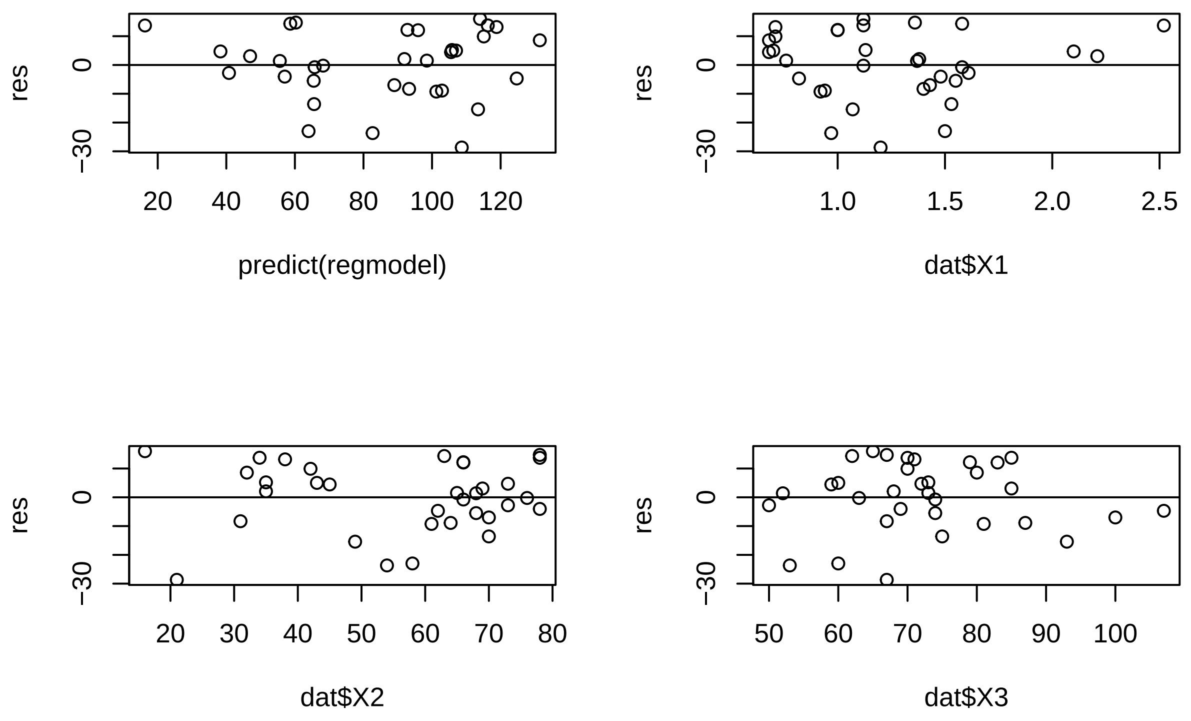
```
## [1] 1.209661
```

The mean value of VIF is only 1.2, we can conclude that there is no indication of serious multicolinearity problem here.

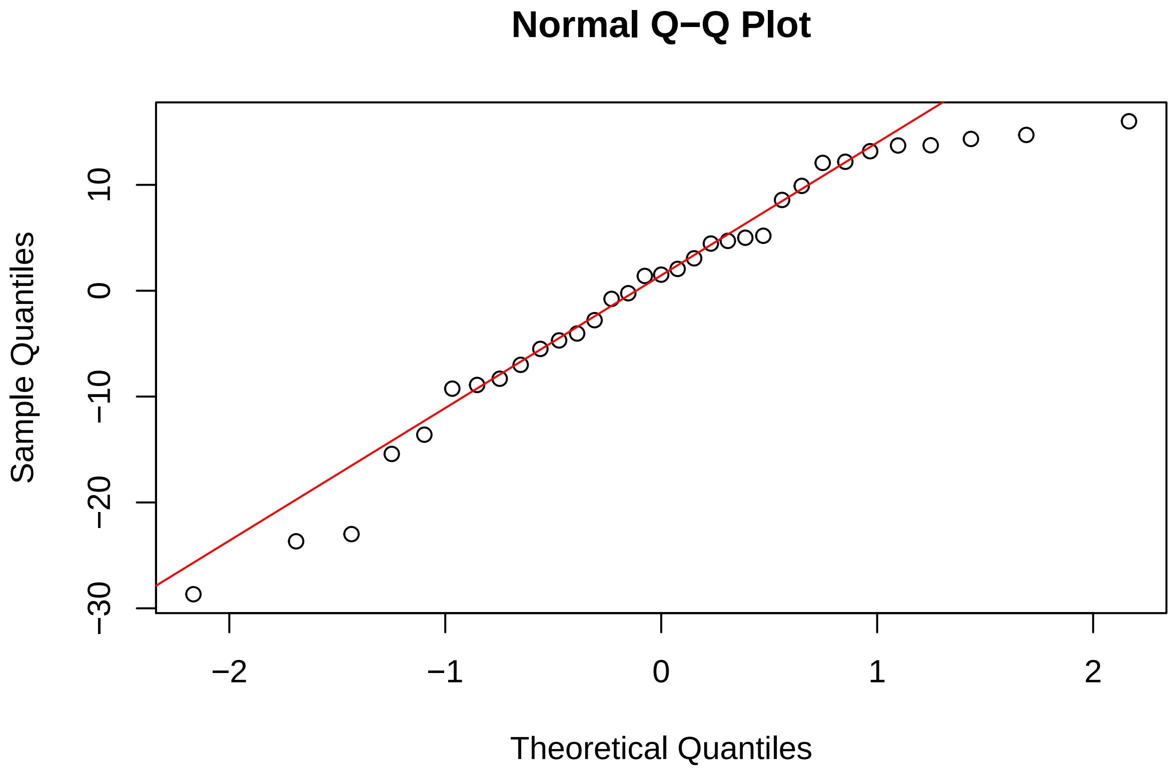
### 3.1.2 b

- residuals against $\hat{Y}$ and each of the variable
- qq plot

```
par(mfrow=c(2,2))
res=regmodel$residuals
plot(y=res,x=predict(regmodel));abline(h=0)
plot(y=res,x=dat$X1);abline(h=0)
plot(y=res,x=dat$X2);abline(h=0)
plot(y=res,x=dat$X3);abline(h=0)
```



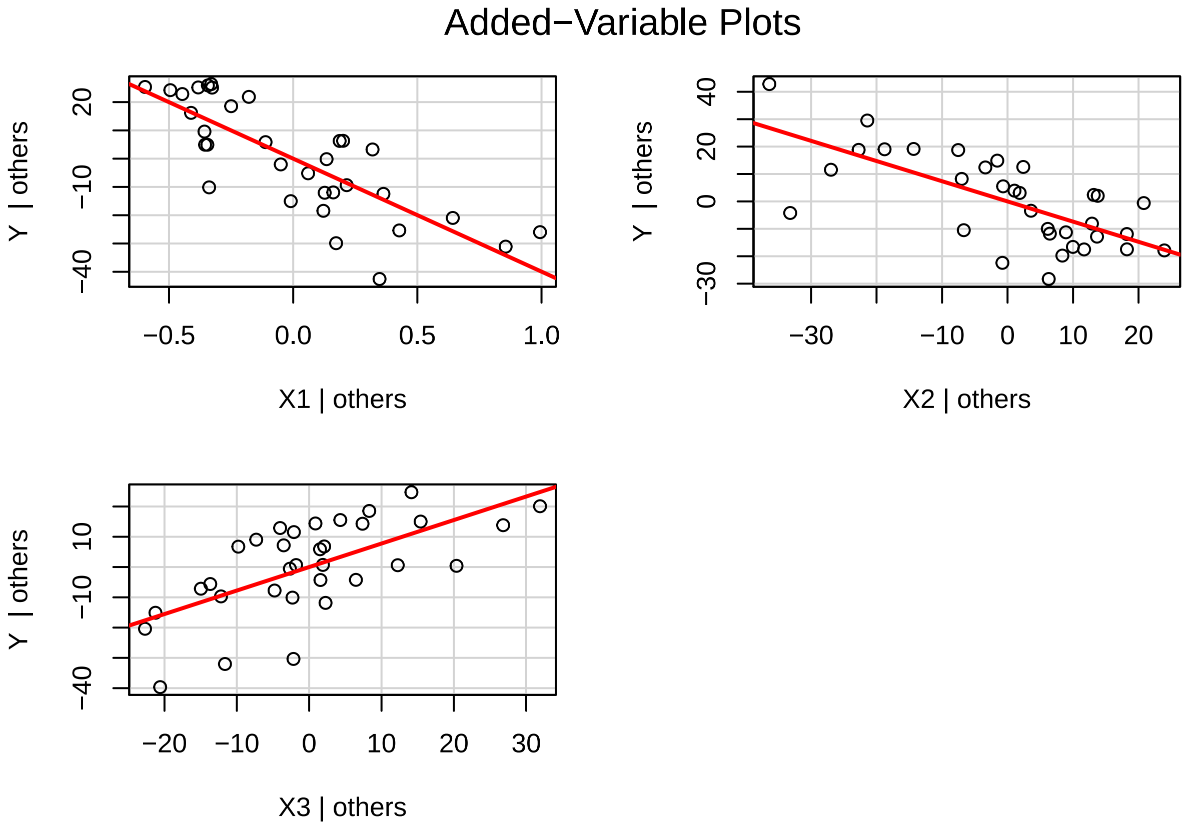
```
par(mfrow=c(1,1))
qqnorm(res)
qqline(res,col="red")
```



### 3.1.3 c

- prepare separate added-variable plots

```
avPlots(regmodel)
```



### 3.1.4 d

According to the added-variable plots, we can conclude that all the variables are useful, since all the variable shows a strong linear relationship with Y|others.

However, there are something strange.

In the residual plots. I have spotted some kind of trend in residuals against X1 and X2. In addition, the distribution of Xs are not even. The constant in variance is also questioned. Finally,in the QQ plot, there are some points that are too far away from the red line(in the right corner).

In the residual plots. Although the usefulness of the variable is not questioned, there is some kind of curvilinear pattern in Y|other against X3.

All of the evidence above show that the model needs to be modified.

## 3.2 KNNL 10.22

### 3.2.1 a

- fit the theoretical model

```r
regmodel2=lm(data = dat,ln.Y~ln.X1+ln.X2+ln.X3)
summary(regmodel2)
```

```
##
## Call:
## lm(formula = ln.Y ~ ln.X1 + ln.X2 + ln.X3, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.34973 -0.08901 -0.01296  0.11124  0.30979
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.04269    1.01919  -2.004   0.0545 .
## ln.X1       -0.71195    0.09203  -7.736 1.57e-08 ***
## ln.X2        0.74736    0.15696   4.761 4.92e-05 ***
## ln.X3        0.75745    0.15923   4.757 4.99e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1571 on 29 degrees of freedom
## Multiple R-squared:  0.8661, Adjusted R-squared:  0.8523
## F-statistic: 62.54 on 3 and 29 DF,  p-value: 8.963e-13
```

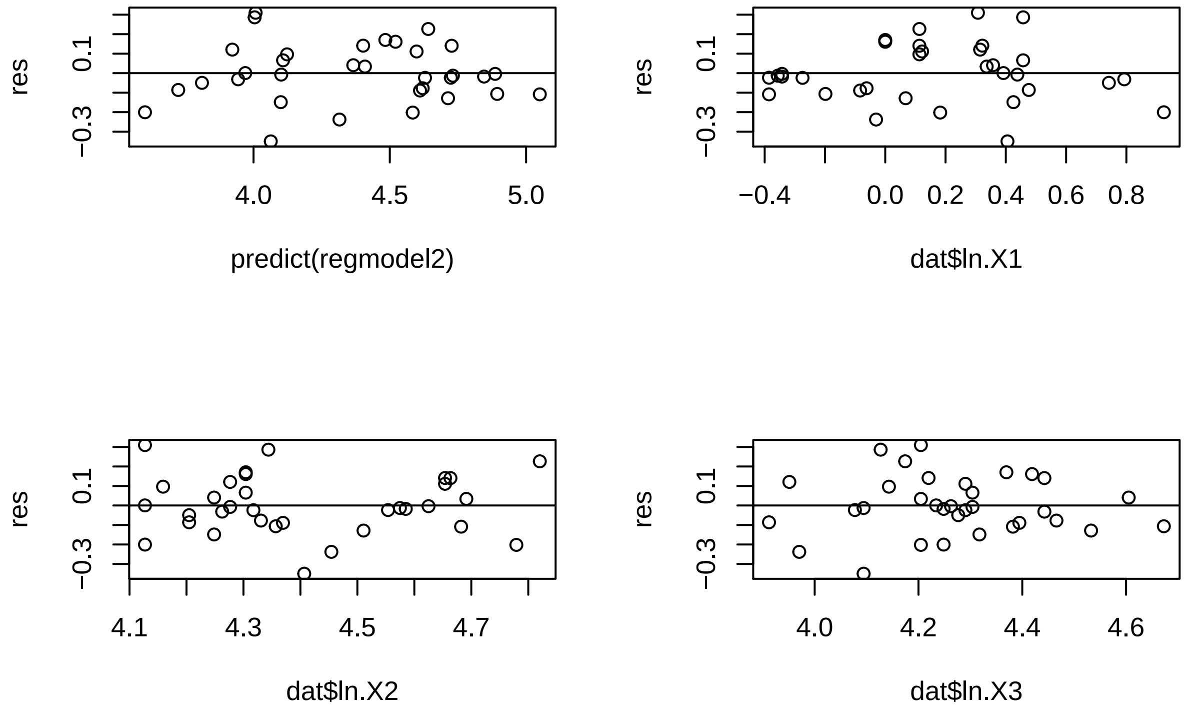
So the fitted model is:

$$\hat{ln}(Y_i) = -2.04269 - 0.71195ln(X_1) + 0.74736ln(140 - X_2) + 0.75745ln(X_3)$$

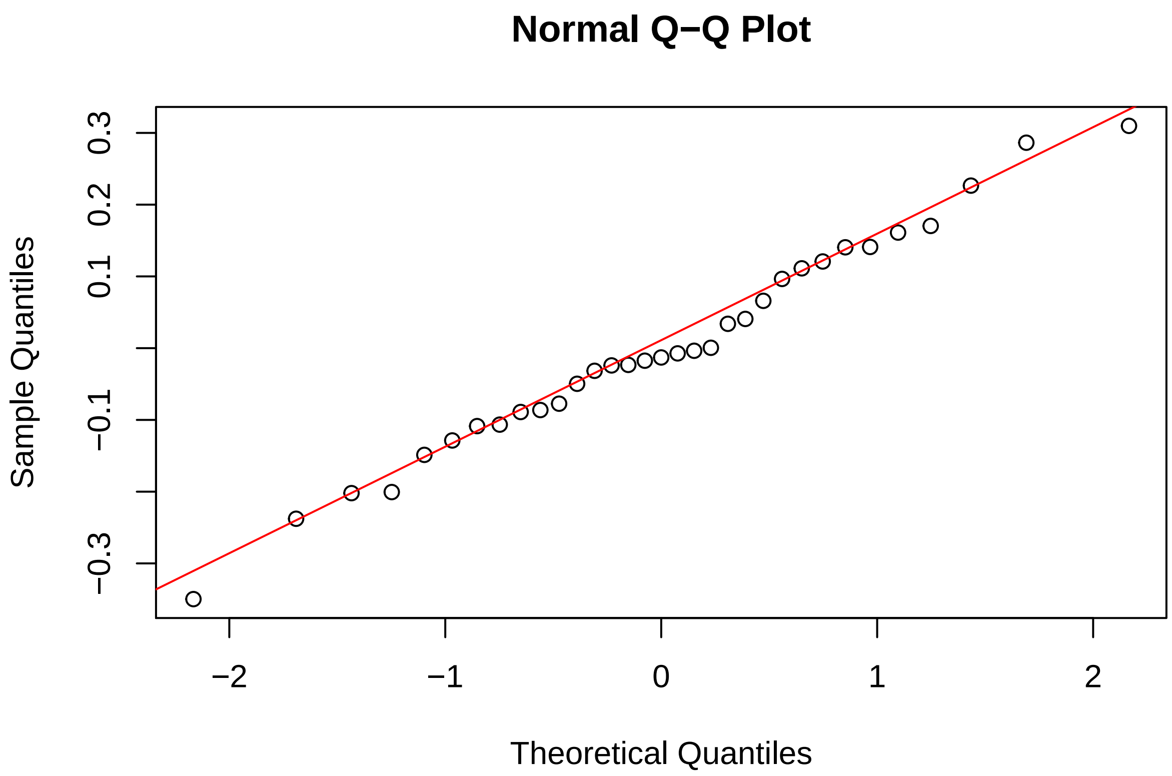
### 3.2.2 b

- residuals vs $\hat{Y}$ and each of the predictors
- difficulties solved?

```r
par(mfrow=c(2,2))
res=regmodel2$residuals
plot(y=res,x=predict(regmodel2));abline(h=0)
plot(y=res,x=dat$ln.X1);abline(h=0)
plot(y=res,x=dat$ln.X2);abline(h=0)
plot(y=res,x=dat$ln.X3);abline(h=0)
```



```r
par(mfrow=c(1,1))
qqnorm(res)
qqline(res,col="red")# Yes
```



Yes! According to the new plots, all of the problems mentioned above have been solved.

### 3.2.3 c

```
(V=vif(regmodel2))
```

```
##     ln.X1     ln.X2     ln.X3
## 1.339318 1.330109 1.016032
```

```
mean(V)
```

```
## [1] 1.228486
```

$V\bar{I}F = 1.23$,which is relatively small. So we can conclude that there is no serious multicolinearity problems here.

### 3.2.4 d

- studentlized deleted residuals

- Bonferroni outlier test, $\alpha = 0.1$

Decision rule:

For each case:

$$t_i = \frac{e_i}{MSE_{(i)}(1 - h_{ii})}$$

If $|ti| < t(1 - \alpha/2n; n - p - 1)$ we conclude that case i is not outlying; otherwise we conclude that case i is an outltying case.(Here n=33 and p=4)

Conclusion:

```
n=nrow(dat)
r.student=rstudent(regmodel2)
outlierTest(regmodel)
```

```
##
## No Studentized residuals with Bonferonni p < 0.05
## Largest |rstudent|:
##     rstudent unadjusted p-value Bonferonni p
## 26 -2.829414          0.0085276      0.28141
```

Since the extreme point has the p-value of 0.28, which is larger than 0.1, we fail to reject $H_0$. So we can conclude that there is no outlying cases based on the Boferonni test.

### 3.2.5 e

- identify outlying X observation based on hat matrix

```
hii=hatvalues(regmodel2)
p=sum(hii)
n=nrow(dat)
h=2*p/n
which(hii>h)
```

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

Here the cut-off point is $2p/n$. It can be seen above that none of the case is beyond the cut-off edge. We can conclude that there is no outlying X observations.

### 3.2.6 f

- DFFITS, DFBETAS and Cook's distance for case 28 and 29.
- Assess the influence

```
obj=c(28,29)
dffit=dffits(regmodel)
dffit[obj]
```

```
##         28          29
##  0.4013746 -0.5363874
```

```
dfbeta=dfbetas(regmodel2)
dfbeta[obj,]
```

```
##    (Intercept)      ln.X1      ln.X2      ln.X3
## 28   0.5298824 -0.1505673 -0.5768082 -0.1873520
## 29  -0.1973071 -0.3099959 -0.1334312  0.4202176
```

```
cook.d=cooks.distance(regmodel2)
cook.d[obj]
```

```
##        28        29
## 0.1201784 0.1091149
```

```
(cutoff=c(1,1,4/n))
```

```
## [1] 1.0000000 1.0000000 0.1212121
```

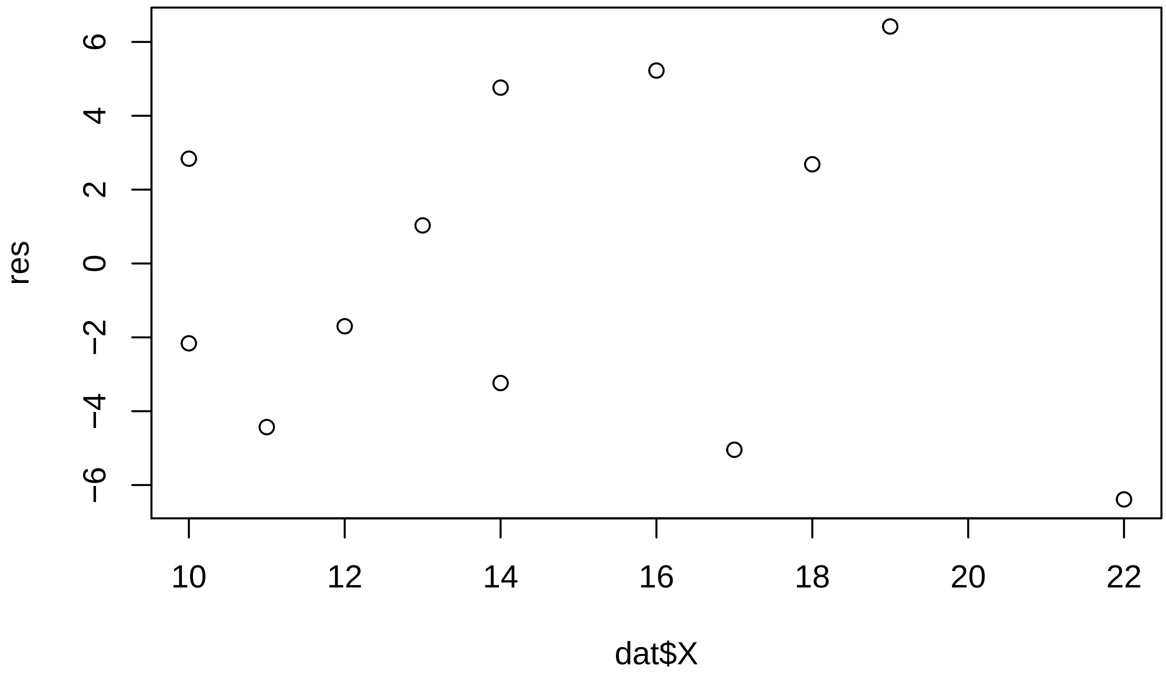
The cut off value is listed at the end of the code. Here we use 1 for DFFIT and DFBETAS since the size of dataset is small.

It can be seen that all the DFFIT, DFBETAS and Cook's distance are below the cut-off edge. We can concude although case 28, 29 are relatively far outlying with repect to their Y values, they do not have much impact on fitted values and coeffients.

# 4 Problem 4(KNNL 11.6)

## 4.1 a

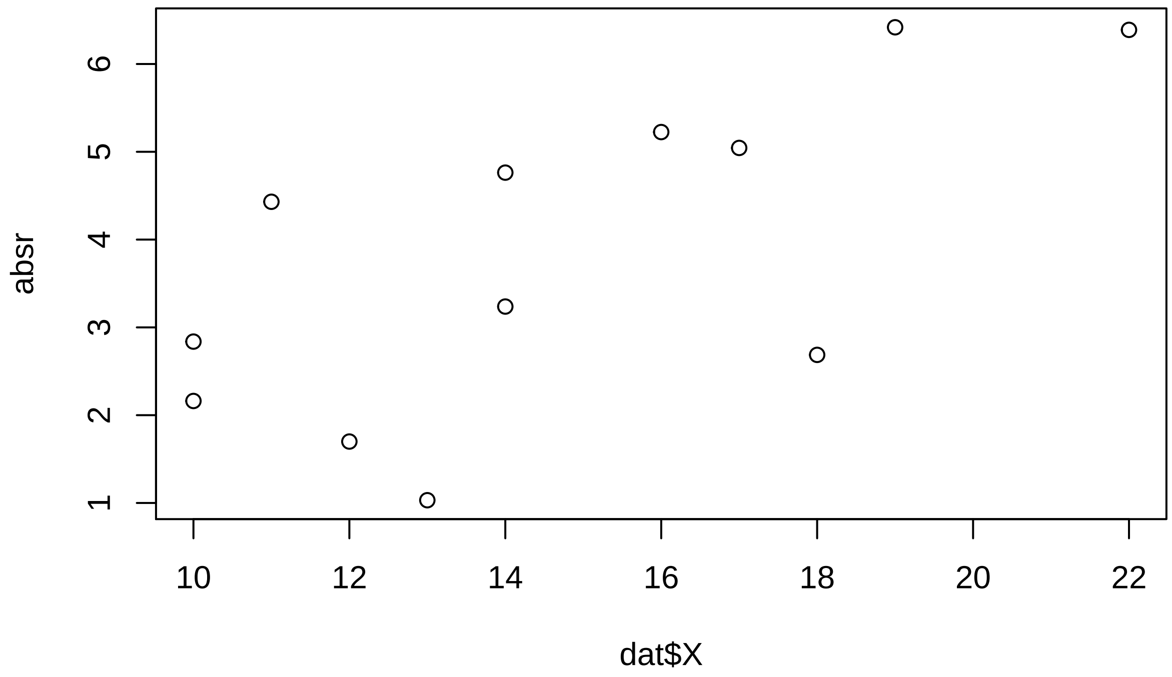
```
regmodel=lm(data = dat,Y~.)
res=regmodel$residuals
plot(res~dat$X)
```



It seems that the residuals does not have a constant variance with the growth of X.

### 4.2 c

```
absr=abs(res)
plot(absr~dat$X)
```



It seems that the absolute standard devition grows with the growth of X, leading to inconstant variance.

### 4.3 d

```
fit=lm(absr~dat$X)
#summary(fit)
dat$shat=fit$fitted.values
dat$wt=1/(dat$shat*dat$shat)

wt=dat$wt
s=sort(wt,decreasing = T)
o=order(wt,decreasing = T)
dat[o,]
```

```
##     Y  X     shat         wt
## 4  50 10 2.321429 0.18556213
## 7  55 10 2.321429 0.18556213
## 12 51 11 2.644058 0.14304022
## 10 57 12 2.966687 0.11362048
## 8  63 13 3.289316 0.09242488
## 5  62 14 3.611945 0.07665099
## 2  70 14 3.611945 0.07665099
## 1  77 16 4.257203 0.05517614
## 6  70 17 4.579832 0.04767612
## 11 81 18 4.902461 0.04160751
## 9  88 19 5.225090 0.03662794
## 3  85 22 6.192977 0.02607360
```

Case 4 and Case 7 receive the largest weight, while case 3 receives the smallest weight.

## 4.4 e

So the cofficients for weighted least squares estimates are:

```
fit2=lm(data = dat,weights = wt,Y~X)
fit2$coefficients
```

```
## (Intercept)           X
##   17.300637    3.421106
```

In constract to the previous least squars estimate:

```
fit$coefficients
```

```
## (Intercept)       dat$X
##  -0.9048619   0.3226291
```

They have been varied a lot, not similar at all.

## 4.5 f

- compare the estimated standard deviation of $b_{w0}$ and $b_{w1}$

```
ans2=summary(fit2)
ans1=summary(fit)
ans1$coefficients
```

```
##                Estimate Std. Error    t value   Pr(>|t|)
## (Intercept) -0.9048619  1.6610989 -0.5447369 0.59787180
## dat$X        0.3226291  0.1099286  2.9348954 0.01491526
```

```
ans2$coefficients
```

```
##               Estimate Std. Error  t value     Pr(>|t|)
## (Intercept) 17.300637   4.827736 3.583592 4.981868e-03
## X            3.421106   0.370310 9.238492 3.268919e-06
```

It seems that the std.errors of $b_1$ and $b_0$ are nearly three times bigger than the previous ordinary least squared estimates.Let's give some explanation to this phenomenonn.

Let's look back to the expression of $s(b)$:

$$s^2(b_w) = MSE_w(X'WX)^{-1}$$

As a special case for weightes linear regresion, the ordinary least squared estimates are:

$$s^2(b) = MSE(X'X)^{-1}$$

Here we can get $MSE = 1.385$ and $MSE_w = 1.159$

Since W is a dignal matrix, we can divide $X'WX$ into $(\sqrt{(W)}X)'(\sqrt{(W)}X)$

So **approximately**, the standard error of weighted least squared estimate should be $\sqrt{(\bar{W})} * MSE_w/MSE$:

```
W=diag(wt)
T=solve(sqrt(W))
mean(T)*1.159/1.385
```

```
## [1] 0.266879
```

which is similar to the practical value 3.