Linear Statistical Models, HW4

Qiuyang Yin, 2015011468

April 5th, 2017

Problem 1

```r
dat=read.table("CH01PR27_967407278.txt",header = T)
```

a

- Prepare a stem-and-leaf plot for ages
- Explain whether the plot is consisitent with the random selection of women from each 10-year age group or not.

```r
stem(dat$age,0.5)
```

```
##
##   The decimal point is 1 digit(s) to the right of the |
##
##   4 | 111223345677788
##   5 | 12334456777999
##   6 | 000133345556889
##   7 | 0012235666788888
```

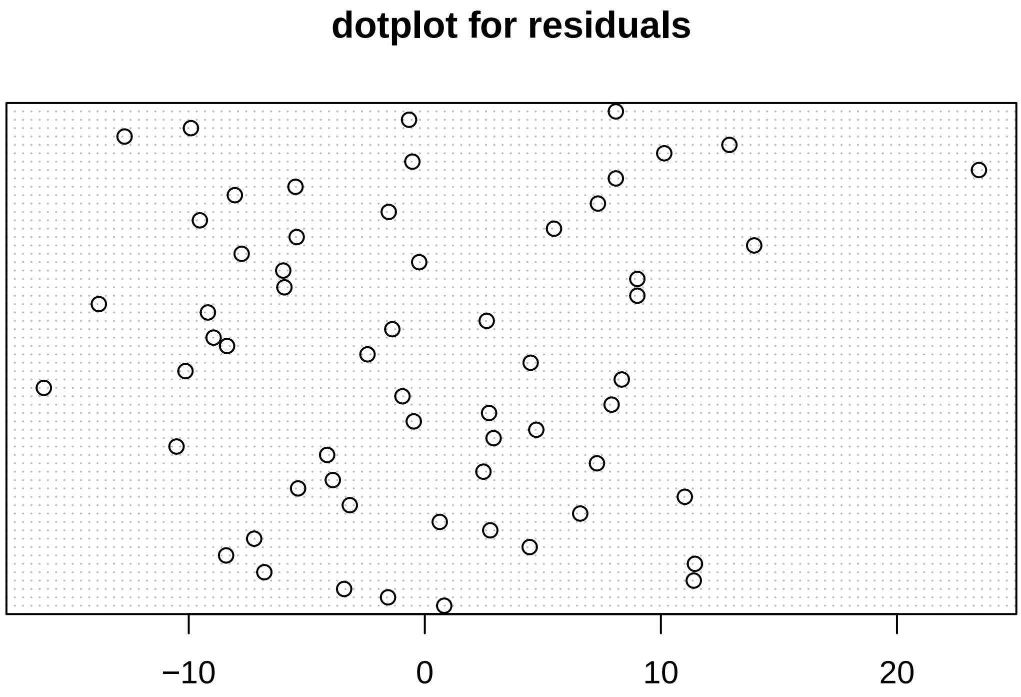
It can be seen that for each group, the distribution of age is approximately random. The prerequisite of random selection is met.

Maybe for senior age(age 70 or elder), the distribution is not so reasonable because there are too many people with age above 75.

b

- Obtain the residuals ei and prepare a dot plot of residuals.

```r
fit=lm(mass~age,data = dat)
diag=cbind(dat,pred=predict(fit),resid=resid(fit))
dotchart(diag$resid,main = "dotplot for residuals")
```



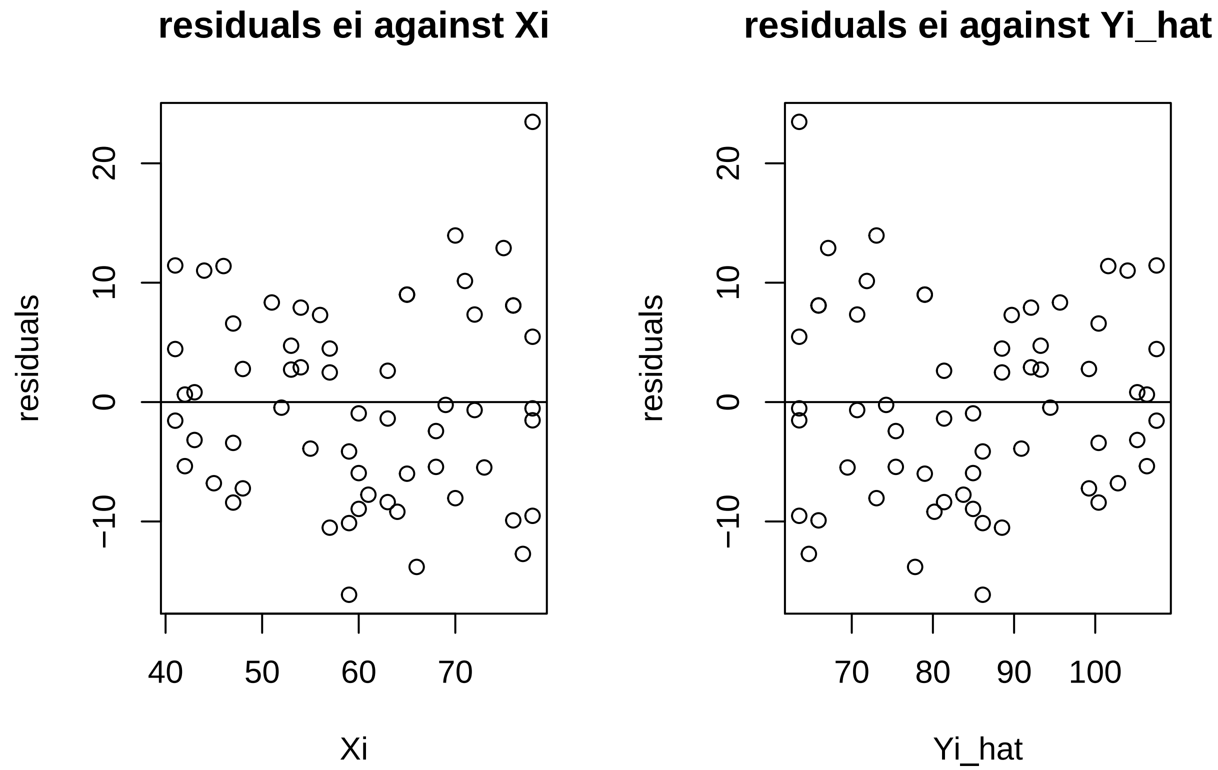
It seems that most of the residuals are between -10 and 10, indicating a good fitting. In addition, the dotplot doesn't reveal any trend against the index of age, which is a good phenomenon.

It can also be shown from the figure that there are a few outliers(with residuals more than 20 or less than -20)

c

- plot residuals against $\hat{Y}_i$ and also against X_i
- ascertain whether any departures from regression model are evident

```r
y=fit$residuals
x1=dat$age
x2=fit$fitted.values
par(mfrow=c(1,2))
plot(y=y,x=x1,main="residuals ei against Xi",xlab = "Xi",ylab = "residuals");abline(h=0)
plot(y=y,x=x2,main="residuals ei against Yi_hat",xlab = "Yi_hat",ylab = "residuals");abline(h=0)
```



Lets check the model assumptions:

- Linear relationship

Residuals are relative small and no obvious trend is spoted. No violation.

- Error variance

It seem that the variance is slightly increasing with the growing of Xi and Yi_hat. But I maintain that the diffrence in variance is so slight that it won't break the model assumptions.

- Error normalty.

Discuss in d and e

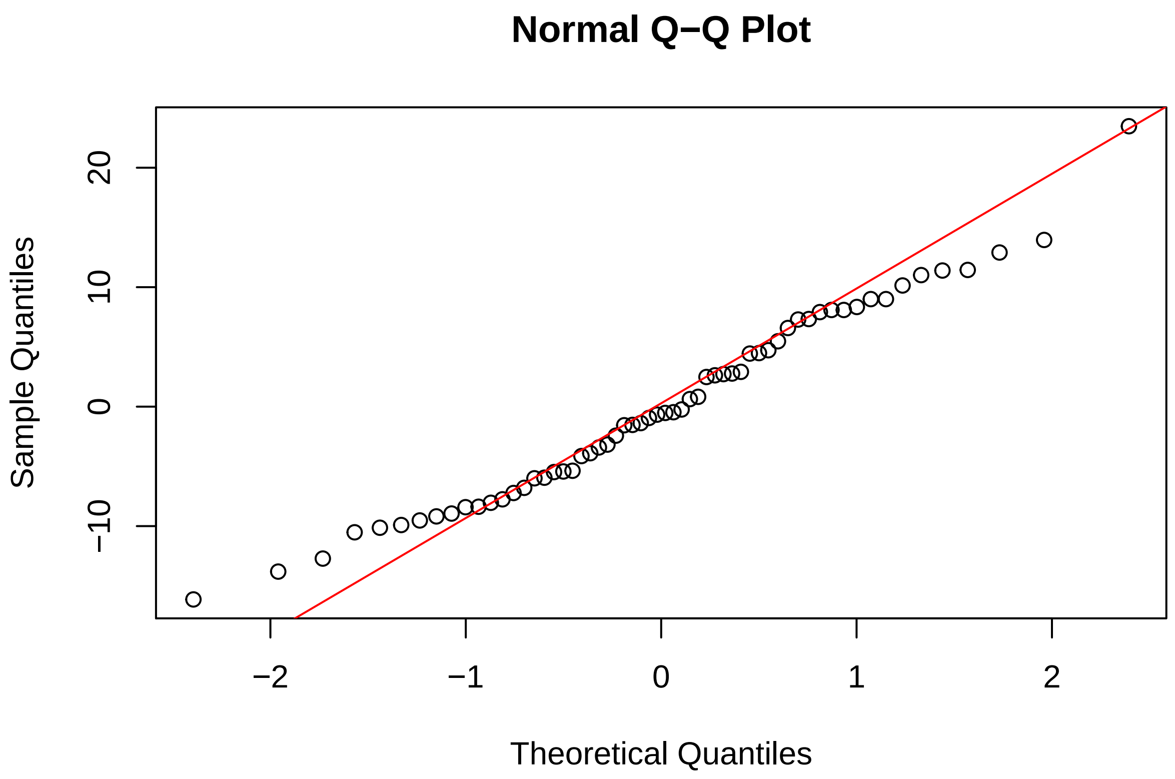
- Outliers

It can be shown in both figures that there **are** a few outliers. For instance, the one with residuals above 20 is evident in both figures.

d

- prepare a normal probability plot for the residuals.
- obtain correlation and state the tenability(not required)

```
par(mfrow=c(1,1))
qqnorm(y)
qqline(y,col=2)
```



Most of the dots follow the line with the slope of 1, indicating that the residuals approximately follow the normality.

An outlier is discovered, without which is normality of the residuals may improve.

```
nscore=qqnorm(diag$resid,plot.it = F)$x
diag=cbind(diag,nscore)
cor(diag$resid,diag$nscore)
```

```
## [1] 0.9896158
```

Although not required, I still do the test in problem 1-d.

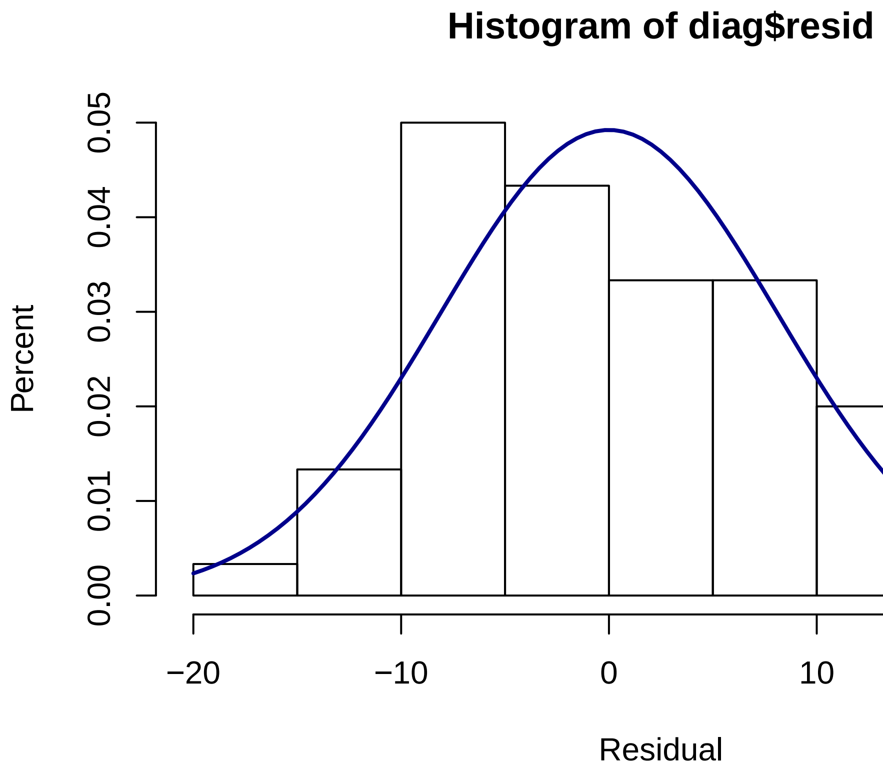
In table B.6, when significance level is .10 and n=60, corelation is 0.971

So the correaltion is larger than the 0.971, indicating the normality of residuals.

e

- generate a histogram of the residuals with added density curves, and comment on it.

```
m=mean(diag$resid)
std<-sqrt(var(diag$resid))
hist(diag$resid, prob=TRUE, xlab="Residual", ylab="Percent")
curve(dnorm(x, mean=m, sd=std),
      col="darkblue", lwd=2, add=TRUE, yaxt="n")
```



Generally the assumption of normality is met, for the histogram follows the trend of the real normal distribution.

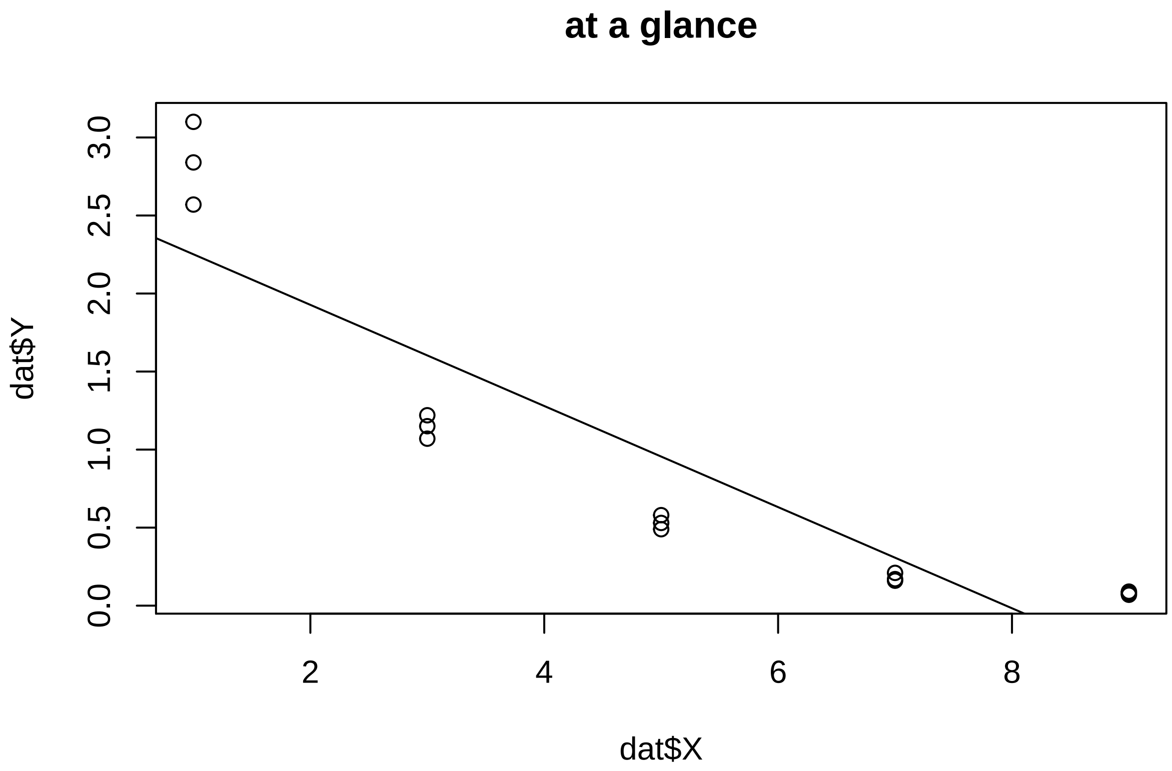
It seems that in the middle of the plot, there is some violation for there are more dots between -10 and -5.

Problem 2

a

- fit a linear regression function.

```
dat=read.table("CH03PR15_820907781.txt")
colnames(dat)=c("Y","X")
fit=lm(Y~X,data = dat)
plot(y = dat$Y,x=dat$X,main = "at a glance")
abline(fit)
```



```r
summary(fit)
```

```
##
## Call:
## lm(formula = Y ~ X, data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.5333 -0.4043 -0.1373  0.4157  0.8487
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.5753     0.2487  10.354 1.20e-07 ***
## X            -0.3240     0.0433  -7.483 4.61e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4743 on 13 degrees of freedom
## Multiple R-squared:  0.8116, Adjusted R-squared:  0.7971
## F-statistic: 55.99 on 1 and 13 DF,  p-value: 4.611e-06
```

So the regression function is

$$\hat{Y}_i = -0.324X_i + 2.5753$$

b

- Perform a F test to determine whether or not there is lack of fit. $\alpha = 0.025$
- State the alternatives, decision rule, and conclusion'

Null hypothesis and alternatives:

$$H_0 : Y_{ij} = \beta_0 + \beta_1 X_j + \epsilon_{ij}$$
$$H_1 : Y_{ij} = \mu_j + \epsilon_{ij}$$

In other words, H_0 means that reduced model is applied, while H_1 means that full model is applied.

decision rule

If $F* \leq F(1-\alpha, c-2, n-c)$, conclude H_0

If $F* > F(1-\alpha, c-2, n-c)$, conclude H_1

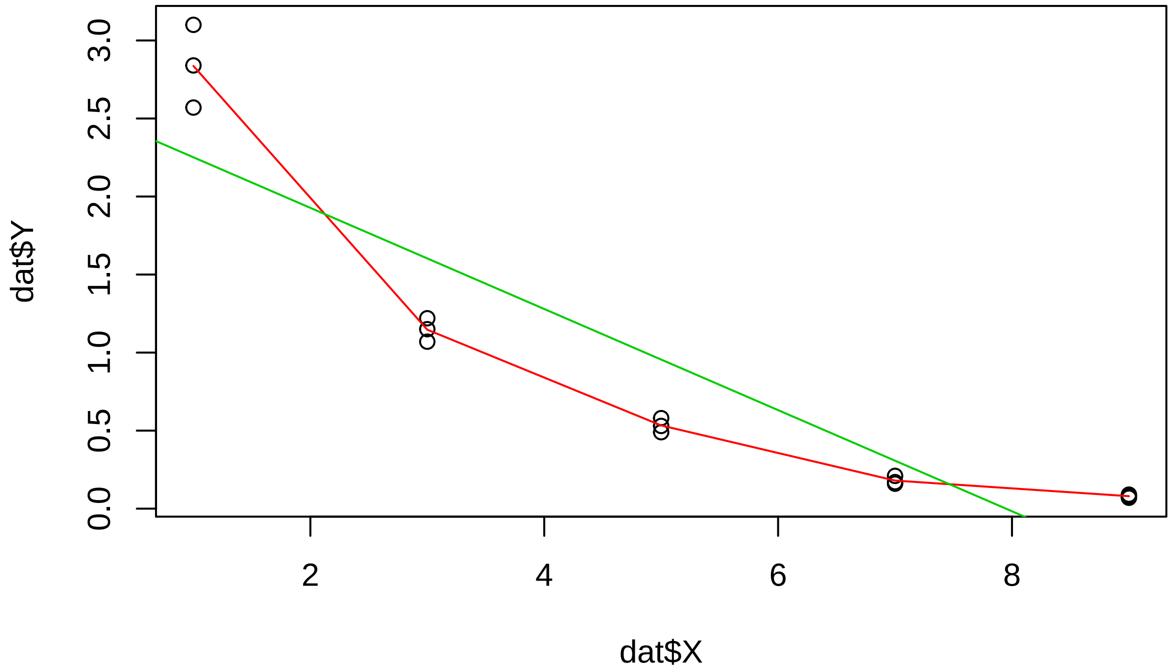
where $F*$:

$$F* = \frac{SSE(R) - SSE(F)}{df_R - df_F} \div \frac{SSE(R)}{df_R}$$

and $df_R = n - 2$,$df_F = n - c$

conclusion

```
Reduced <- lm(Y~X, data = dat)
Full <- lm(Y ~ 0 + as.factor(X), data = dat)
plot(y = dat$Y,x=dat$X)
lines(y=Full$coefficients,x=c(1,3,5,7,9),col=2)
abline(Reduced,col=3)
```



```r
anova(Reduced, Full)
```

```
## Analysis of Variance Table
##
## Model 1: Y ~ X
## Model 2: Y ~ 0 + as.factor(X)
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1     13 2.9247
## 2     10 0.1574  3    2.7673 58.603 1.194e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

It can be seen that p-value is 1.194e-06, so we conclude H_1

c

- Does the test indicate appropriate regression functions?
- How to proceed?

From my perspective the general linear test itself does not indicate appropriate models. It is used to compare two models and to decide whether the reduced models can be applied to substitute for the full models.

However, the general linear test can be used to test various regression functions, and H_1 can be any other funtions. To do this just alter the model and the degree of freedom.

To obtain appropriate regression functions, many other methods can be applied. For instance, box-cox transmation(dicussed in Problem 3)

Problem 3

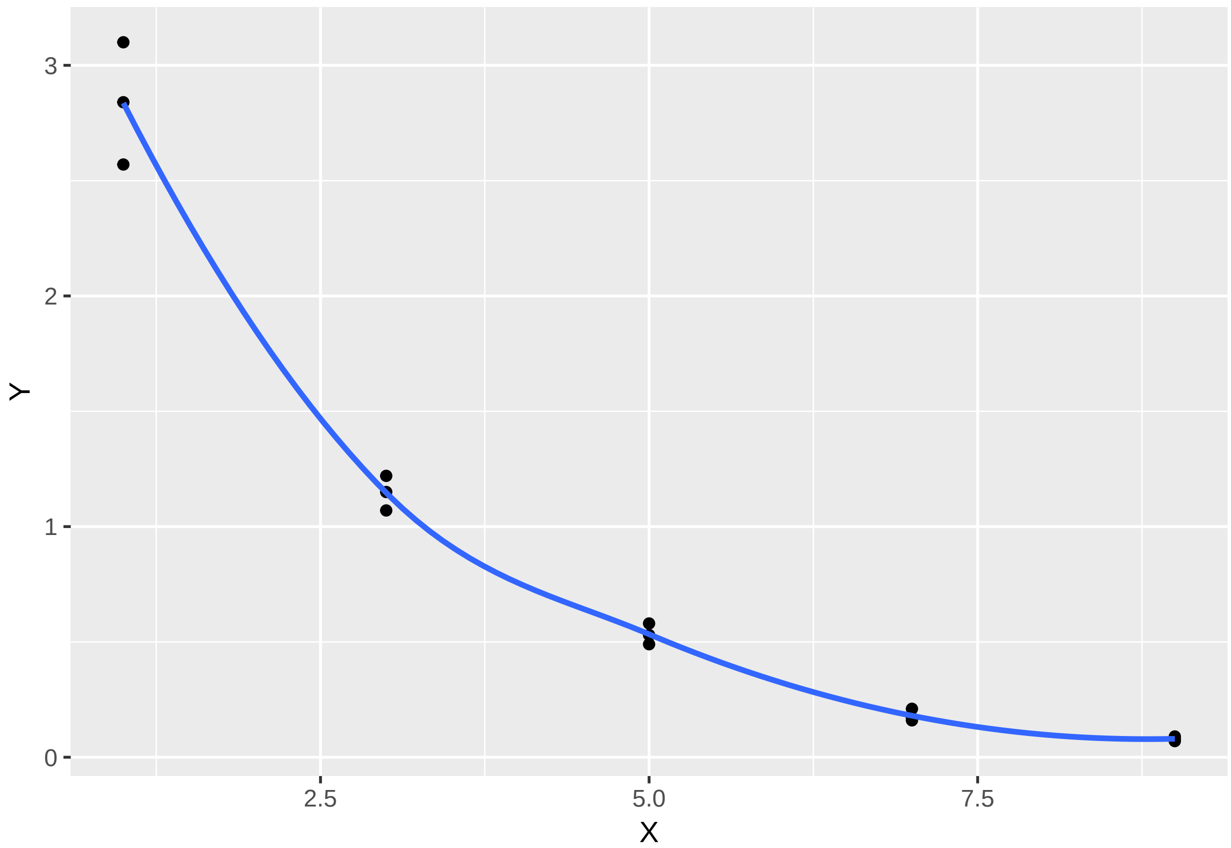
a

- scatter plot of data
- what tranformation is used?

```
require(ggplot2)
```

```
## Loading required package: ggplot2
```

```
ggplot(dat,aes(y=Y,x=X))+geom_point()+geom_smooth(method = "loess",se = F)
```

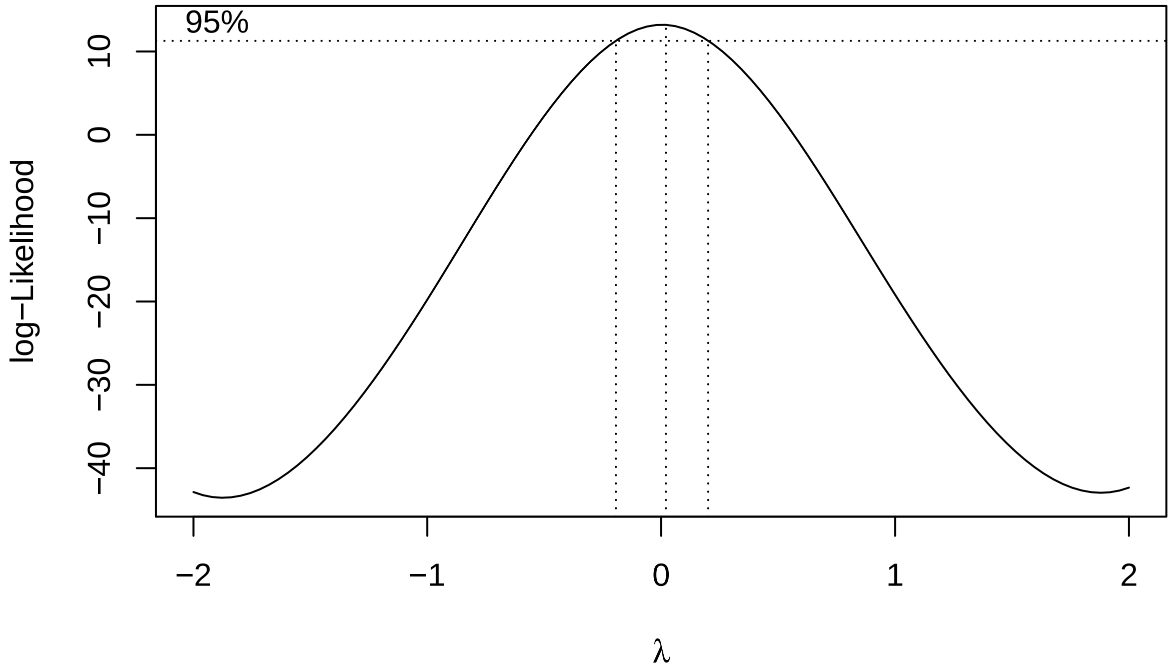


According to the Figure 3.15 on KNNL, $Y' = log_{10}Y$ is recommended.

b

- Box-cox procedure
- Evaluate SSE

```
library(MASS)
bc=boxcox(Y~X, data = dat, lambda = -2:2)
```



then we calculated the sse:

```
k2 = exp(mean(log(dat$Y)))
lambda = seq(-2, 2, by=1)
n=nrow(dat)

transformed = NULL
for(i in 1:length(lambda)){
  k1 =  1/(lambda[i]*k2^(lambda[i]-1))
  trans_y = k1*((dat$Y)^lambda[i]-1)
  if(lambda[i]==0){
    trans_y = k2*(log(dat$Y))
  }
  a2 = cbind(dat, lambda = rep(lambda[i], n), trans_y)
  transformed = rbind(transformed, a2)
}


sse <- by(transformed, transformed[,"lambda"],
          function(x) anova(lm(trans_y ~ dat$X, data = x))[,2][2])
print(sse)
```

```
## transformed[, "lambda"]: -2
## [1] 68.8428
## ---------------------------------------------------------
## transformed[, "lambda"]: -1
## [1] 3.168468
## ----------------------------------------------------------
## transformed[, "lambda"]: 0
## [1] 0.03897303
## ----------------------------------------------------------
## transformed[, "lambda"]: 1
## [1] 2.924653
## ----------------------------------------------------------
## transformed[, "lambda"]: 2
## [1] 64.15599
```

So we can conclude than $\lambda = 0$, i.e. $Y' = logeY$ is recommended.

c

- $Y' = log_{10}Y$ obtain the estimated linear regression function and tranformed data.

```
(dat$new_Y=log10(dat$Y))
```

```
##  [1] -1.15490196 -1.04575749 -1.09691001 -0.79588002 -0.76955108
##  [6] -0.67778071 -0.30980392 -0.23657201 -0.27572413  0.08635983
## [11]  0.06069784  0.02938378  0.45331834  0.40993312  0.49136169
```

```
fit2=lm(dat$new_Y~dat$X)
summary(fit2)
```

```
##
## Call:
## lm(formula = dat$new_Y ~ dat$X)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.082958 -0.044421  0.006813  0.033512  0.085550
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.654880   0.026181   25.01 2.22e-12 ***
## dat$X       -0.195400   0.004557  -42.88 2.19e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.04992 on 13 degrees of freedom
## Multiple R-squared:  0.993,  Adjusted R-squared:  0.9924
## F-statistic:  1838 on 1 and 13 DF,  p-value: 2.188e-15
```

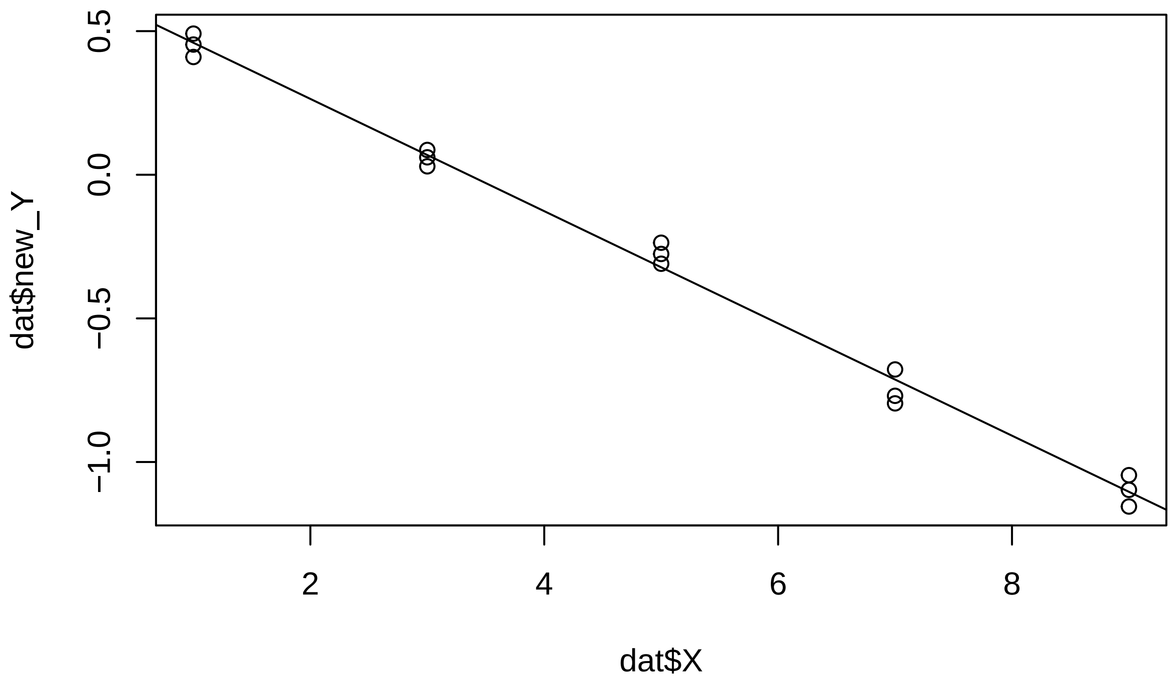
So the new regresion funtion is :

$$\hat{Y}_i' = -0.1954X_i + 0.654880$$

d

- plot the estimated regreesion line and transformed data.
- good fit?

```
plot(dat$new_Y~dat$X)
abline(fit2)
```

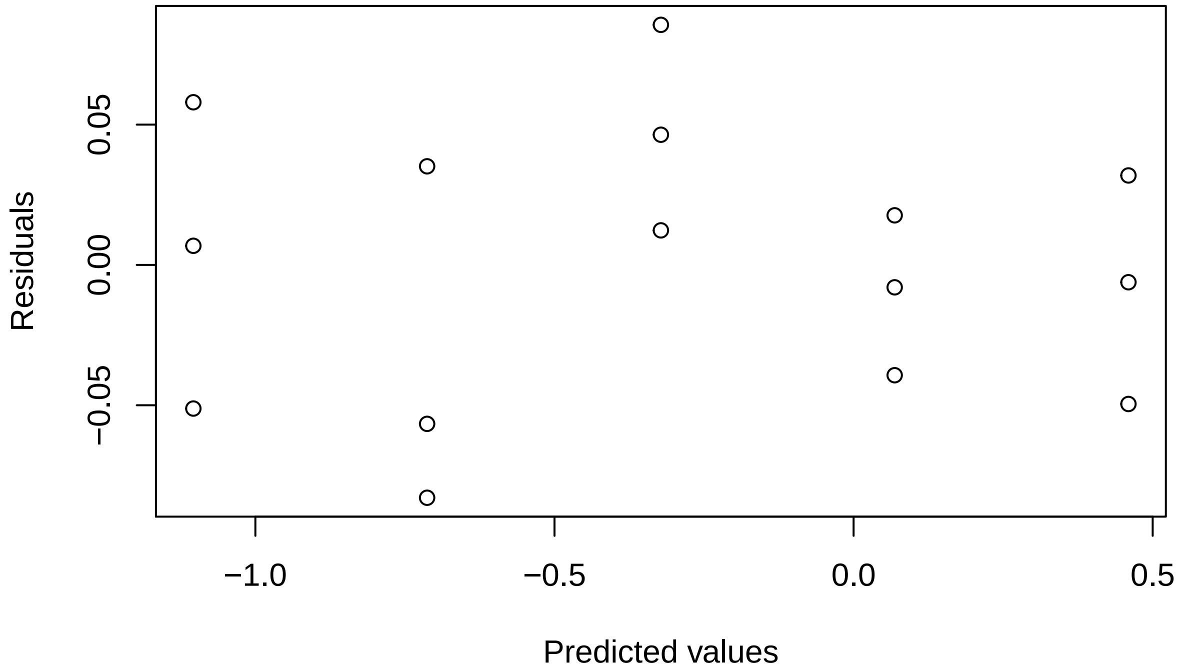


Of course! It **is** a good fit! And R^2 is 0.993!

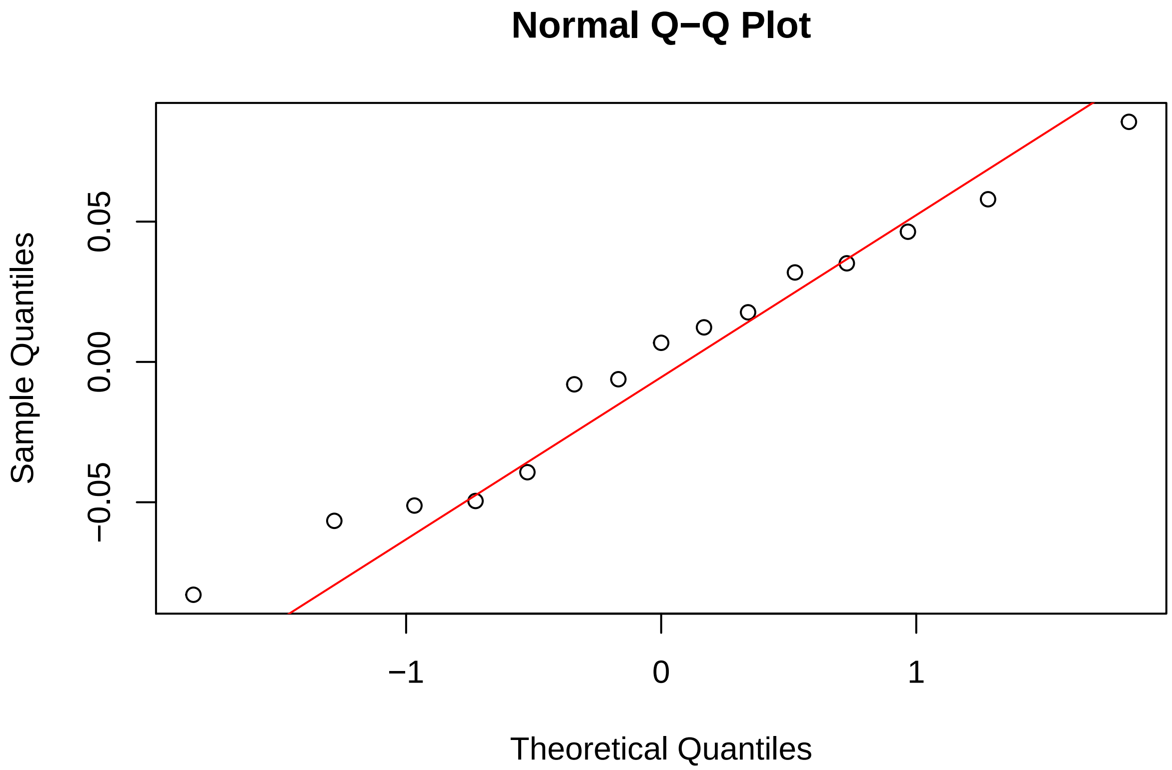
e

- residuals against fitted values
- normal probability plot
- What do your plots show?

```
resid=resid(fit2)
pred=predict(fit2)
plot(resid~pred, xlab="Predicted values", ylab="Residuals")
```



```
qqnorm(resid)
qqline(resid, col=2)
```



It shows that the normality of residuals is good and there no obvious trend with predicted values. So the regression is good.

f

- Express the estimeated regression funtion in original units

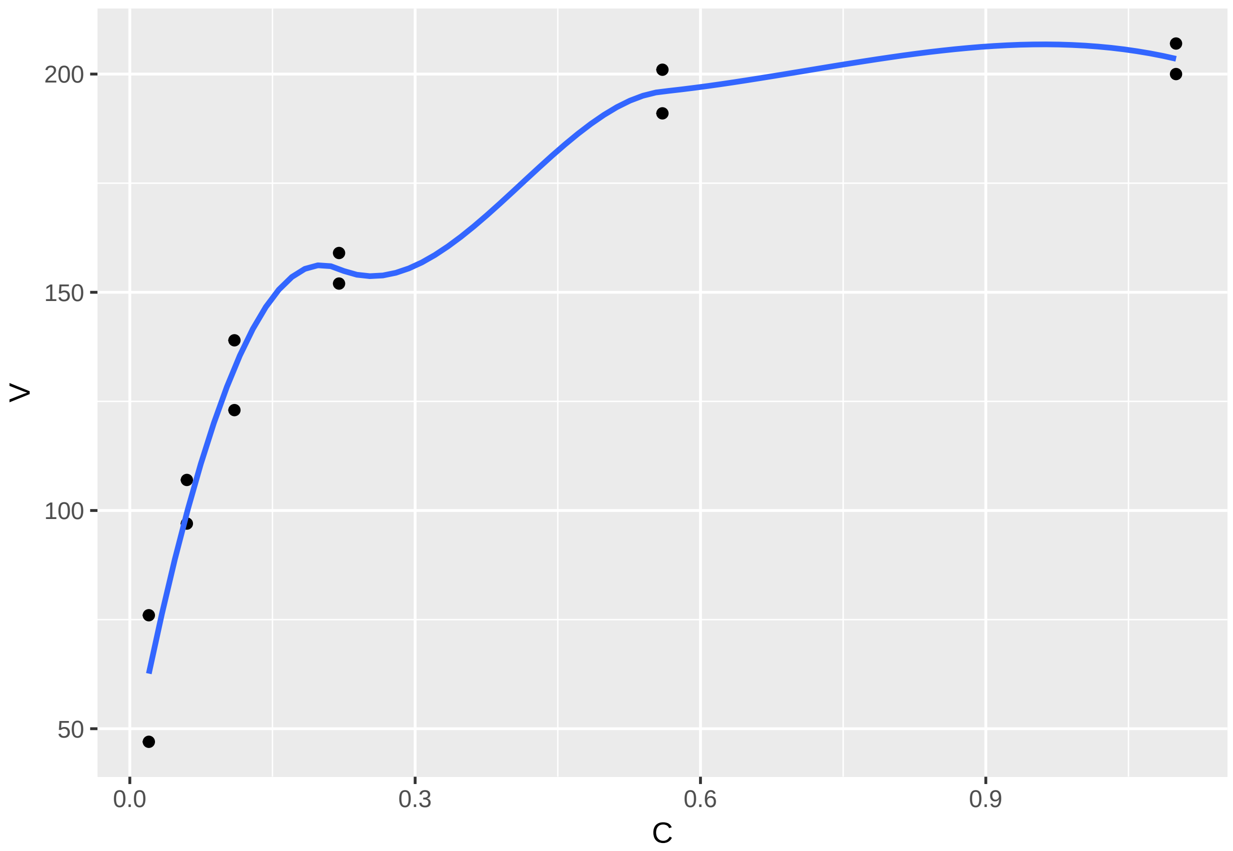
$$\hat{Y} = 10^{0.654880*X-0.195}$$

Problem 4

a

- generate a scatter plot
- comment

```
dat4=read.table("data4.txt",header = T)
colnames(dat4)=c("C","V")
require(ggplot2)
ggplot(dat4,aes(y=V,x=C))+geom_point()+geom_smooth(method = "loess",se = F)
```

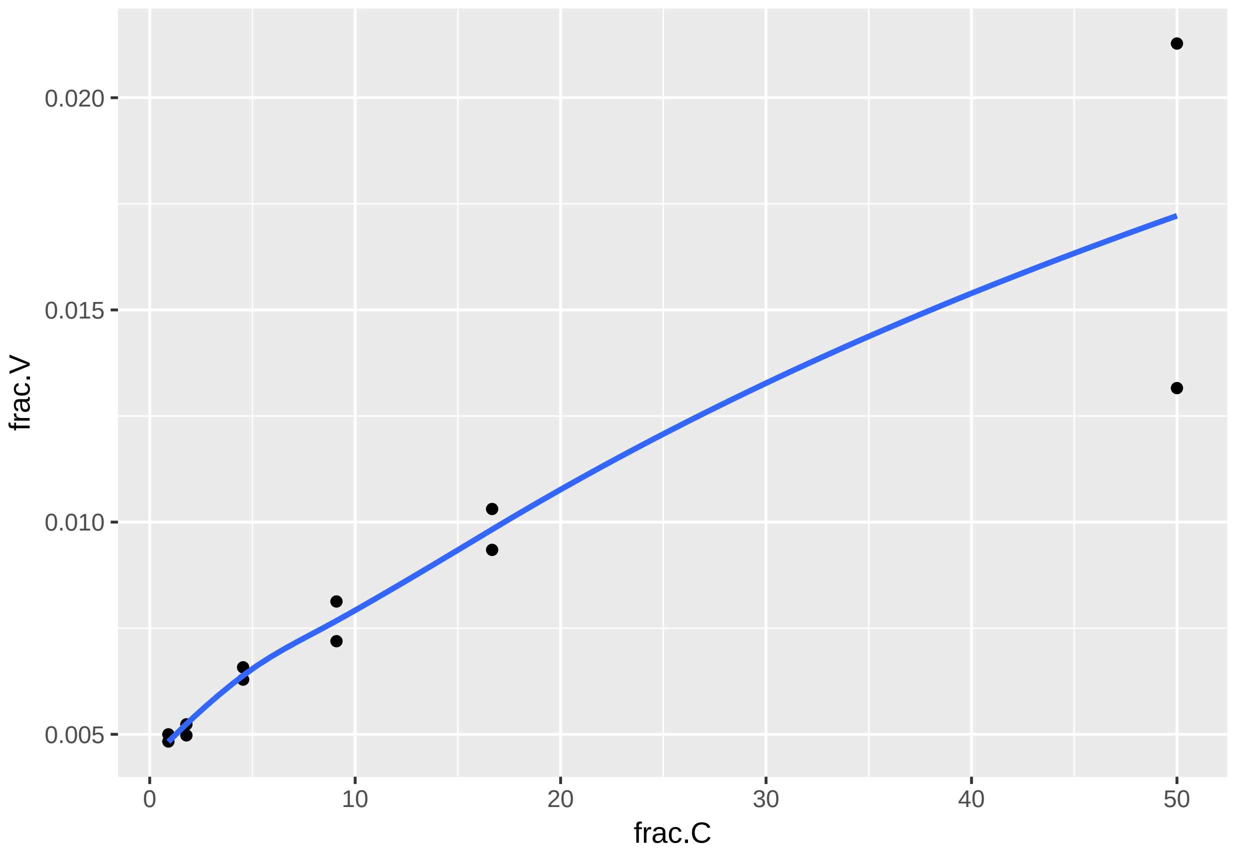


Obviously some transmation is needed. The shape of the scatter plot may follow a 1/X relationship.

b

- define new variables for $1/V$ and $1/C$
- generate a scatter plot
- what does the fit appear?Any violation of assumptions?

```
dat4$frac.V=1/dat4$V
dat4$frac.C=1/dat4$C
ggplot(dat4,aes(y=frac.V,x=frac.C))+geom_point()+geom_smooth(method = "loess",se = F)
```



It appears that the linear relation is good but there is a growing variance.

c

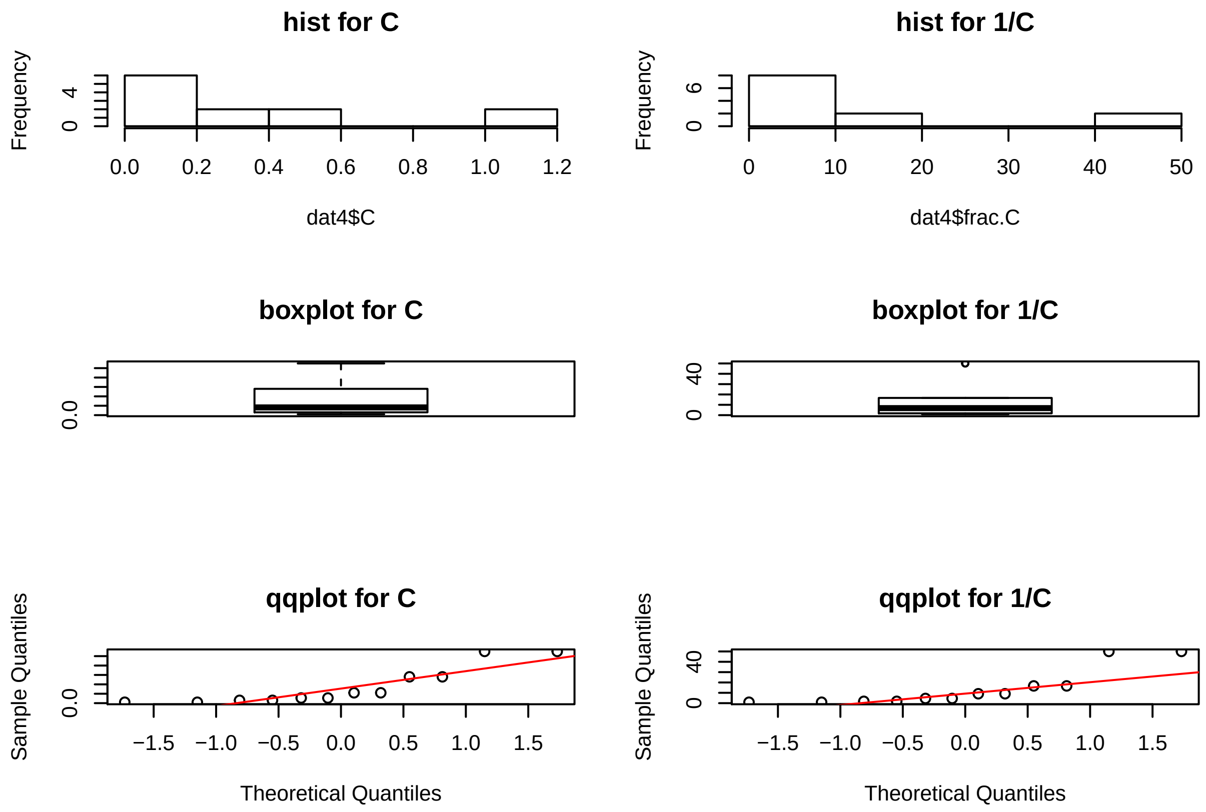
- The diffrence in distribution between $1/C$ and C
- Are there influential points?

I obtain three kinds of plots for $1/C$ and C

```
par(mfrow=c(3,2))
hist(dat4$C,main = "hist for C")
hist(dat4$frac.C,main = "hist for 1/C")

boxplot(dat4$C,main="boxplot for C")
boxplot(dat4$frac.C,main="boxplot for 1/C")

qqnorm(dat4$C,main="qqplot for C")
qqline(dat4$C, col = 2)
qqnorm(dat4$frac.C,main="qqplot for 1/C")
qqline(dat4$frac.C, col = 2)
```



The figures show that the $1/C$ tends to be denser than C, but the distribution is appropriately similar to each other.

Yes from all the plots we can spot outliers. there are two points that are too large in both C and $1/C$ distribution.

d

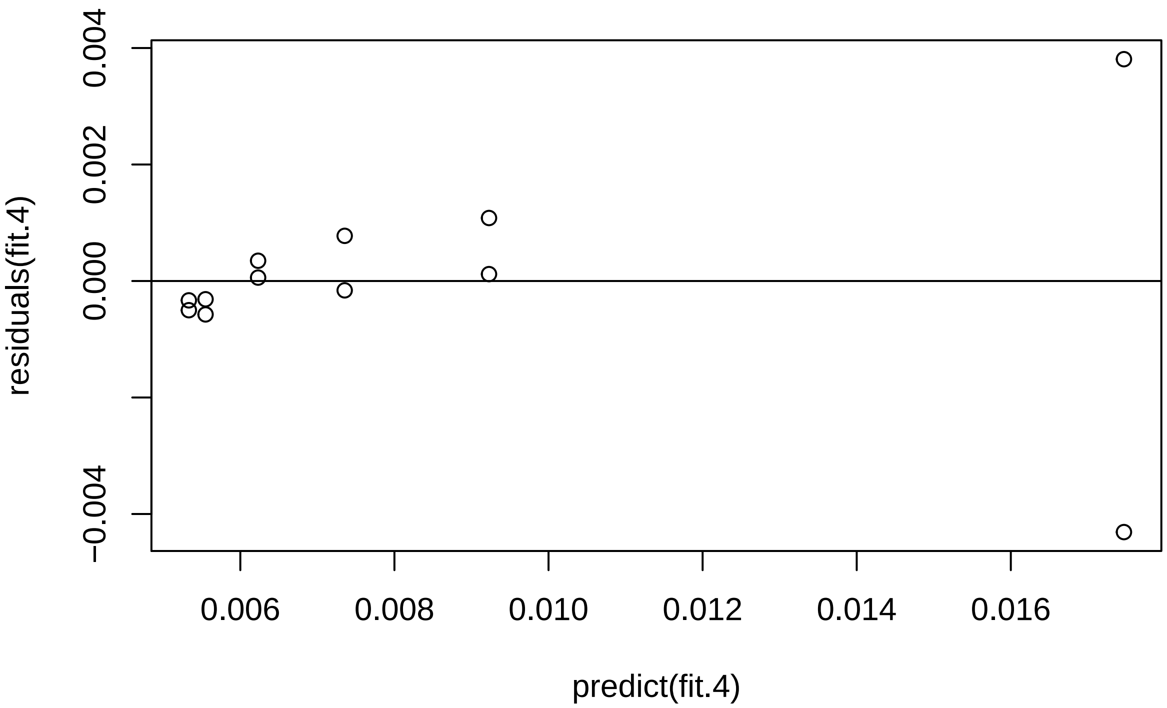
- regression line for $1/V$ and $1/C$
- residual plot
- problem with assumption?

```
require(ggplot2)
par(mfrow=c(1,1))
fit.4=lm(frac.V~frac.C,data = dat4)
summary(fit.4)
```

```
##
## Call:
## lm(formula = frac.V ~ frac.C, data = dat4)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0043103 -0.0003742 -0.0000510  0.0004549  0.0038084
##
```

```
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.0051072  0.0007040   7.255 2.74e-05 ***
## frac.C      0.0002472  0.0000321   7.700 1.64e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.001892 on 10 degrees of freedom
## Multiple R-squared:  0.8557, Adjusted R-squared:  0.8413
## F-statistic:  59.3 on 1 and 10 DF,  p-value: 1.642e-05
```

```r
plot(predict(fit.4),residuals(fit.4))
abline(h=0)
```



So the regression funtion is

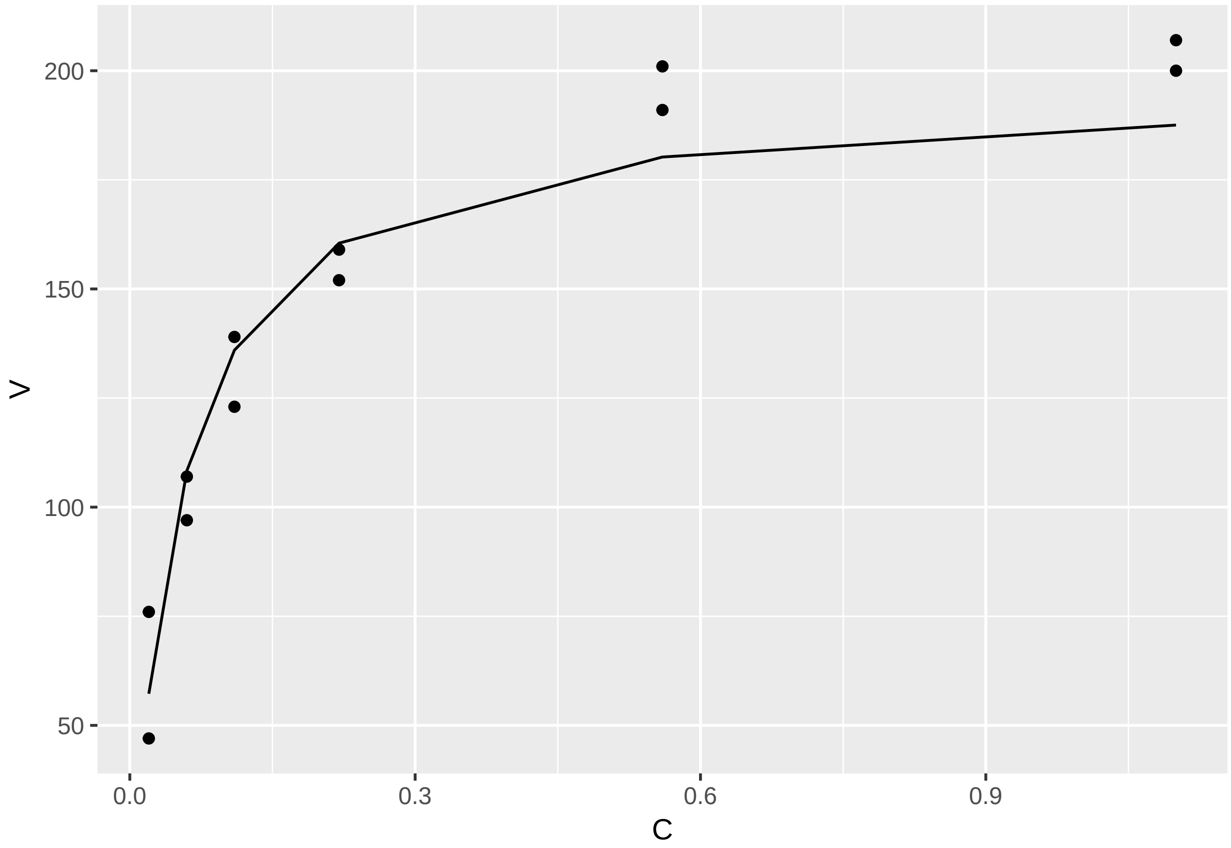
$$1\hat{/}V = 0.0002472 * 1/C + 0.0051$$

We can see from the residual plot that the variance is not constant; in contrast to the constant, the variance grows as the predicted value grows.

e

- scatter plot of V versus C and predicted values.
- comment on the fit

```
dat4$predict.y=1/predict(fit.4)
ggplot(dat4,aes(x=C))+geom_point(aes(y=V))+geom_line(aes(y=predict.y))
```



The model is good when C is relatively small. For larger C, there is a large gap between real values and predicted values.

Problem 5

a

- obtain Bonferroni joint confidence intervals for β_1 and β_0, $\alpha = 0.99$
- interpret

```
rm(list=ls())
dat=read.table("CH01PR27_967407278.txt",header = T)
```

```
muscle.fit=lm(mass~age,data = dat)
confint(muscle.fit,level = 0.995)
```

```
##                 0.25 %     99.75 %
## (Intercept) 140.259608 172.4335205
## age          -1.453227  -0.9267644
```

For Bonferroni joint confidence. $1-\alpha/2$ is appiled for $1-\alpha$, so 0.995 is used.

We conclude that β_0 is between 140.26 and 172.43 and β_1 is between -1.453 and -0.9268. The familt confidence is at least 0.99.

b

- Will b_0 and b_1 tend to err in the same or opposite direction

According to fomula (4.5)

$$\sigma b_0, b_1 = -\bar{X}\sigma^2 b_1$$

```
mean(dat$age)
```

```
## [1] 59.98333
```

Since $\bar{X}$ is positive. b_0 and b_1 are negatively correlate, implying they tend to err in **opposite** directions.

c

- Does the interval support expectation?

Partly.

For intercept β_0 the confidence interval support the researcher for they correspond with each other.

For slope β_1, however, even -1.5 is not in the interval, indicating the wrong expectation.

Problem 6

a

- Working-Hotelling procedure, family confidence coeffecient of 0.95.

I use the same method in confidence band:

```
n = 60
alpha = 0.05
dfn = 2
dfd = n-2
w2 = 2 * qf(1-alpha, dfn, dfd)
w = sqrt(w2)
alphat = 2 * (1-pt(w, dfd))
conf_band=predict(muscle.fit,se.fit = T, data.frame(age=c(45,55,65)),
                  interval="confidence", level = 1-alphat)
(conf_band_ans=conf_band$fit)
```

```
##         fit      lwr       upr
## 1 102.79677 98.48916 107.10437
## 2  90.89681 88.01540  93.77822
## 3  78.99686 76.11248  81.88123
```

To justify the correctness, I recalculate the upr of age 45 from the definition:

```
W=sqrt(2*qf(0.95,2,58))
conf=predict(muscle.fit,se.fit = T,data.frame(age=45),
             interval = "confidence",level = 0.95)
se=conf$se.fit
conf$fit[1]+W*se
```

```
## [1] 107.1044
```

So the previous method is valid.

b

- Any other efficient method?

Yes, Boferroni method is quicker and more direct.

```
conf=predict(muscle.fit,se.fit = T,data.frame(age=c(45,55,65)),
             interval = "confidence",level = 1-0.05/3)
conf$fit
```

```
##         fit      lwr       upr
## 1 102.79677 98.56965 107.02388
## 2  90.89681 88.06924  93.72438
## 3  78.99686 76.16637  81.82734
```

Here we apply 1-0.05/3 as substitute for α

Of course the answer is similar to the method in part (a)

c

- predict the muscle mass for 48,59,74 using Bonferroni procedure for $\alpha = 0.95$(family confidence)

```
conf=predict(muscle.fit,se.fit = T,data.frame(age=c(48,59,74)),
             interval = "prediction",level = 1-0.05/3)
conf$fit
```

```
##        fit      lwr       upr
## 1 99.22678 78.73541 119.71815
## 2 86.13683 65.81829 106.45537
## 3 68.28690 47.73184  88.84195
```

d

From my perspective, intervals have to be recalculated for both procedures.

For Bonferroni procedure. We substitute $1-\alpha$ for $1-\alpha/g$, where g stands for the number of people. In the procedure, we have to recalculate the $B = t(1-\alpha/2g, n-2)$

For Sch procedure, we have to recalculate $S^2 = gF_{\alpha,g,n-2}$ for diffrent g

For both procedure, mean response and standard deviation do **not** need to be recalculated.

Maybe for *Working-Hotelling* procedure,the interval need not be recalculated.

Problem 7

According to our model assumptions:

$$b_1 \sim N(\beta_1, \frac{\sigma^2}{S_{xx}})$$

$$b_0 \sim N(\beta_0, \sigma^2(\frac{1}{n} + \frac{\bar{X}^2}{S_{xx}}))$$

So b_1 and b_0 follow a **Multivariate normal distribution**

Next let's deduce the corvariance of b_0 and b_1

$$Cov(b_0, b_1) = E(b_0 - \beta_0)(b_1 - \beta_1)$$

$$b_0 = \bar{Y} - b_1\bar{X}, \beta_0 = \bar{Y} - \beta_1\bar{X} - \frac{\sum \epsilon_i}{n}$$

$$Cov(b_0, b_1) = -\bar{X}E(b_1 - \beta_1)^2 + \frac{1}{n}E((b_1 - \beta_1)\sum \epsilon_i)$$

$$E(b_1 - \beta_1)^2 = \sigma^2(b_1), E((b_1 - \beta_1)\sum \epsilon_i) = 0$$

$$Cov(b_0, b_1) = -\bar{X}\sigma^2(b_1)$$

So when $\bar{X}$ is zero, $Cov(b_0, b_1) = 0$. According to the property of *Multivariate normal distribution* ,b_0 and b_1 are independent.

When b_0 and b_1 are independent, their joint confidences are then independent.i.e. we can calculate the whole probability by the multiply of the individual confidence interval.