Assignment 3_Group11

Name: Jiamian Liu **VU student number:** 2632301
Name: Xiaoyu Yang **VU student number:** 2640948
Name: Fangzheng Lyu **VU student number:** 2644757

Exercise 1

1.Since the slices are from a same loaf, so we use randomized block design. The codes for the randomization process are shown as figure 1.

```
> I = 2; B = 3 ; N = 3
> for (i in 1:B) print(sample(1:(N*I)))
[1] 4 1 6 3 2 5
[1] 6 1 2 4 3 5
[1] 4 2 3 1 5 6
```

Figure 1: Randomized block design codes

2. The codes and boxplot are shown below as figure 2,3 and 4.

```
> hours = as.vector(as.matrix(bread[,1]))
> environment = as.vector(as.matrix(bread[,2]))
> humidity = as.vector(as.matrix(bread[,3]))
> boxplot(hours~environment,xlab="enviroment",ylab="hours")
> boxplot(hours~humidity,xlab="humidity",ylab="hours")
```

Figure 2: Codes

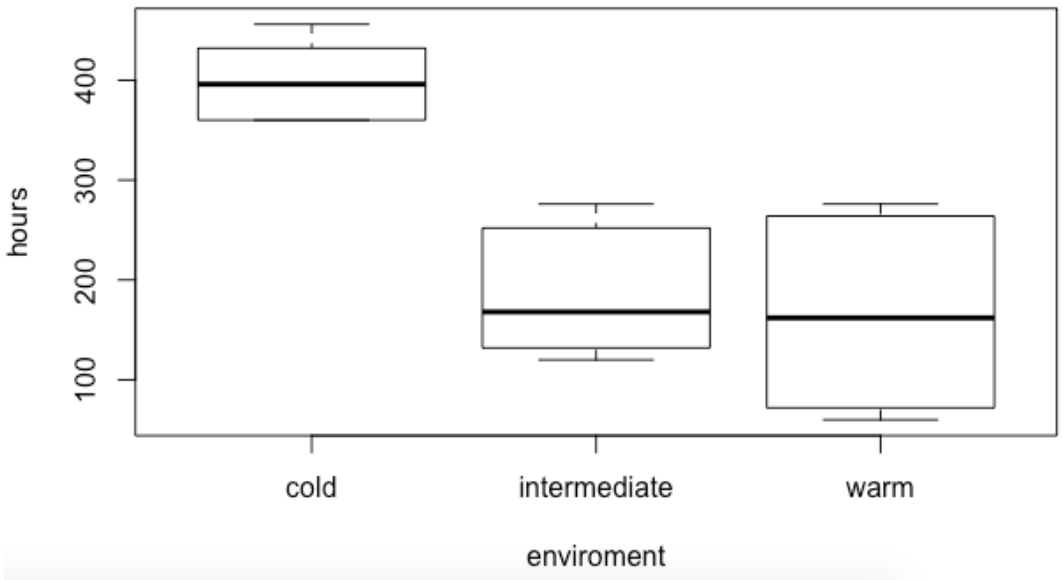

Figure 3: Boxplot of environment and hours

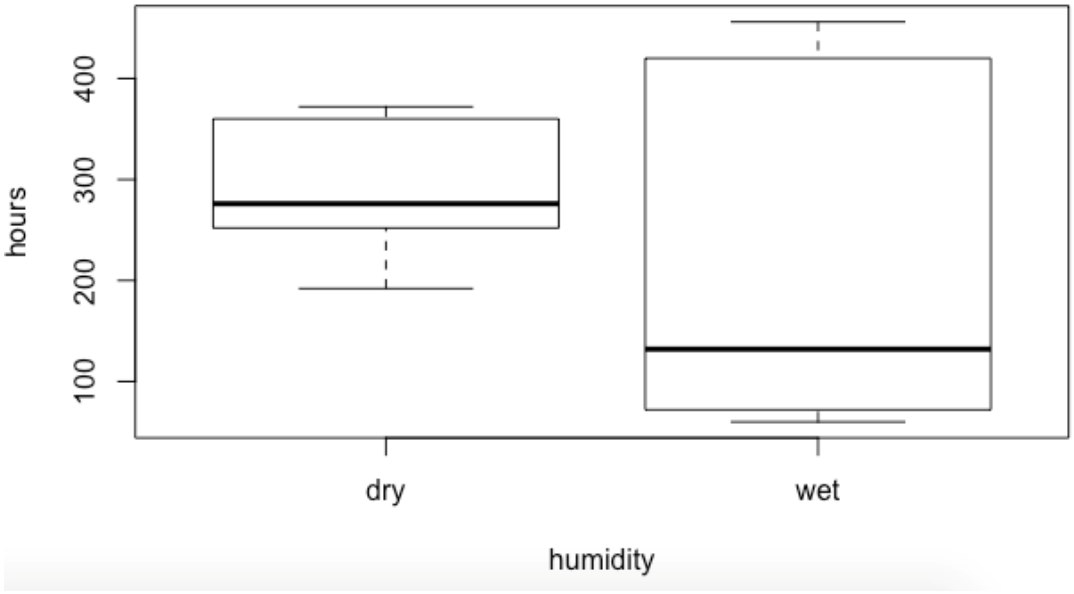

Figure 4: Boxplot of humidity and hours

3.

The analysis code and interaction plot are shown as figure 5,6 and 7.

It can be summarized from the analysis that the cold environment always have the longer store time than intermediate and warm.

From figure 6 we can see that whether the store time of bread in warm environment is less than intermediate depends on the humidity. When in the dry humidity, the decay time in warm environment is a bit more than intermediate; but when it's wet, the time is less than intermediate.

From figure 7 we can see that from the dry humidity to wet, the store time of bread in cold environment have increased, but the trend of those in intermediate and warm is just the opposite.

```
> xtabs(hours~environment+humidity,data=bread)
             humidity
environment     dry  wet
  cold         1092 1308
  intermediate  720  396
  warm          792  204
> interaction.plot(environment,humidity,hours)
> interaction.plot(humidity,environment,hours)
```

Figure 5: Codes

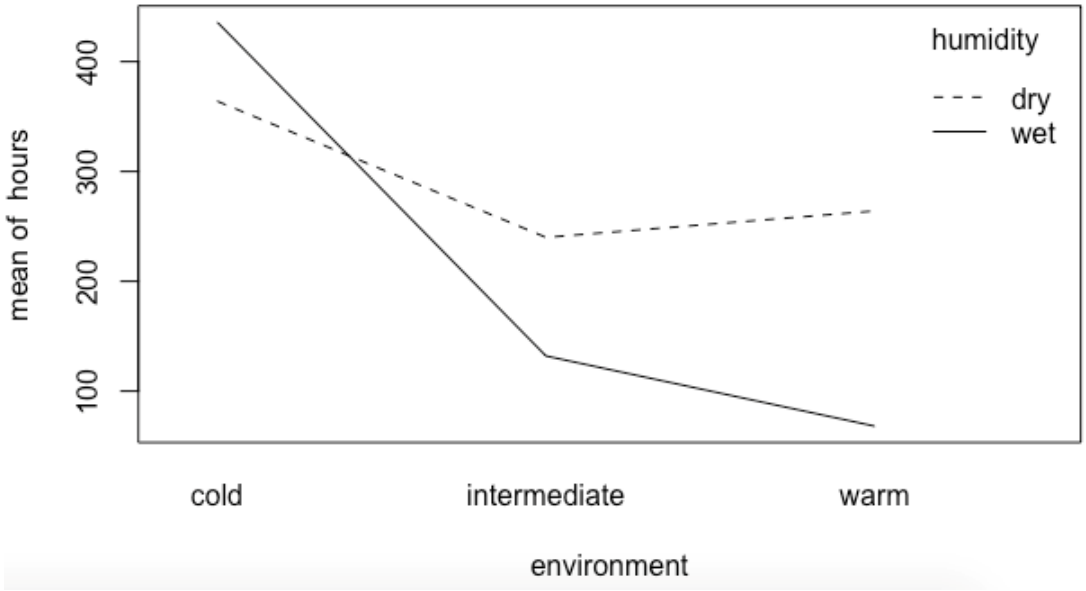

Figure 6: Interaction plot of environment humidity and hours

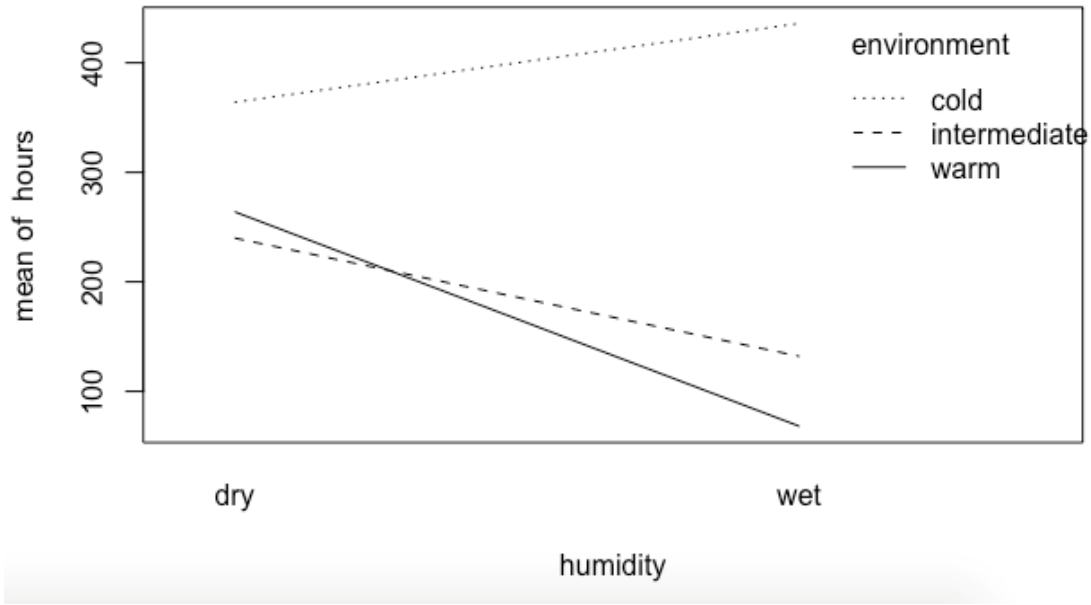

Figure 7: Interaction plot of humidity environment and hours

4.

The environment effects are significantly different from 0 (significant influence on decay hours) ($p<0.05$, reject H0). The humidity is also significantly different from 0 (significant influence on decay hours) ($p<0.05$, reject H0). And environment have the greatest(numerical) effect on the decay time.

But this is not a good question. Because there are interactions between environment and humidity, if we change one factor, the trend is totally different when the other factor changes.

```
> aovpen=lm(hours~environment+humidity,data=bread)
> anova(aovpen)
Analysis of Variance Table

Response: hours
            Df Sum Sq Mean Sq F value    Pr(>F)
environment  2 201904  100952 23.1057 3.674e-05 ***
humidity     1  26912   26912  6.1596   0.02637 *
Residuals   14  61168    4369
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Figure 8: Anova

5.

From the figures below we could know that the value of “Multiple R-squared” is 0.7891. A value close to 1 means that the linear regression model can explain the measured response values very well using a linear function of the explanatory variables.

We can see that there are some curves in the qq-plot, but from the shapiro-test we could know that p-value is greater than 0.05, so it could still be considered normal. There’s no outliers from the boxplot in Figure 3 and 4.

```
> summary(aovpen)

Call:
lm(formula = hours ~ environment + humidity, data = bread)

Residuals:
    Min      1Q  Median      3Q     Max
-78.667 -55.333  -9.333  56.833  94.667

Coefficients:
                         Estimate Std. Error t value Pr(>|t|)
(Intercept)                438.67      31.16  14.078 1.17e-09 ***
environmentintermediate   -214.00      38.16  -5.608 6.46e-05 ***
environmentwarm           -234.00      38.16  -6.132 2.60e-05 ***
humiditywet                -77.33      31.16  -2.482   0.0264 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 66.1 on 14 degrees of freedom
Multiple R-squared:  0.7891,    Adjusted R-squared:  0.7439
F-statistic: 17.46 on 3 and 14 DF,  p-value: 5.271e-05
```

Figure 9: Summary

```
> qqnorm(residuals(aovpen))
> plot(fitted(aovpen),residuals(aovpen))
```

Figure 10: Codes

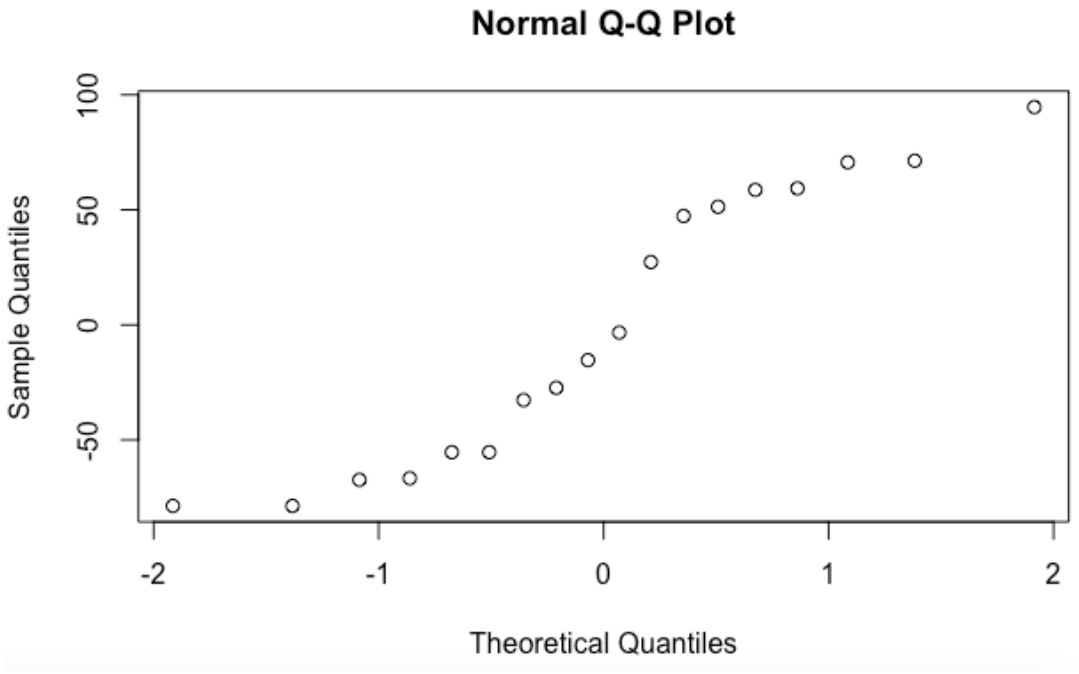

Figure 11: QQ-norm

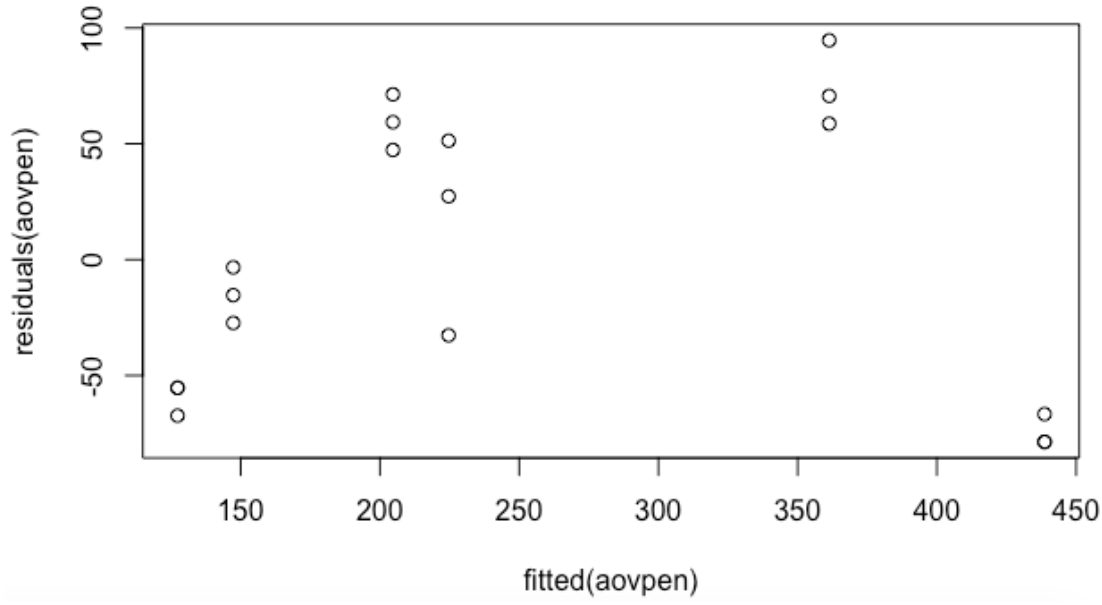

Figure 12: plot between fitted aovpen and residuals aovpen.

```
> shapiro.test(residuals(aovpen))

        Shapiro-Wilk normality test

data:  residuals(aovpen)
W = 0.90064, p-value = 0.05896
```

Figure 13: Shapiro-test

Exercise 2

1. The codes for randomized block design are shown below.

```
> list=sample(1:15)
> block1 = list[1:5]; block2 = list[6:10]; block3 = list[11:15]
> block1;block2;block3
[1] 12 14 15  2 13
[1]  5  7 11  9  1
[1]  6  3  8 10  4
```

Figure 14: Randomized block design codes

2.

```
> skill = as.vector(as.matrix(search[,2]))
> interface = as.vector(as.matrix(search[,3]))
> time = as.vector(as.matrix(search[,1]))
> boxplot(time~skill,xlab="skill",ylab="time")
> boxplot(time~interface,xlab="interface",ylab="time")
> xtabs(time~skill+interface,data=search)
     interface
skill    1    2    3
    1 16.1 15.6 20.5
    2 14.5 20.4 21.2
    3 17.6 19.8 23.9
    4 21.1 24.8 22.2
    5 21.5 23.7 25.3
```

Figure 15: Codes

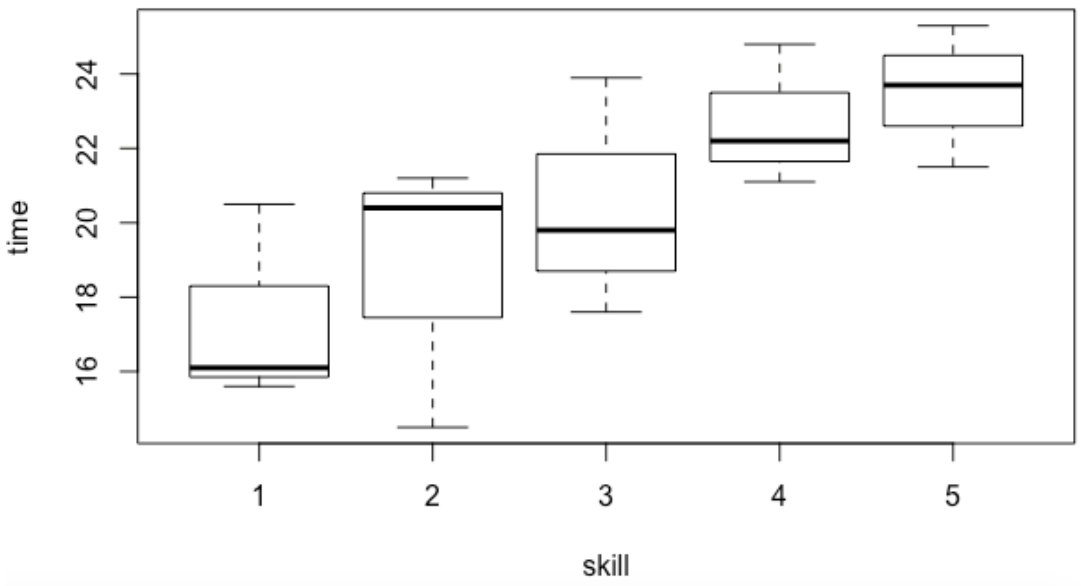

Figure 16: Boxplot of skill and time.

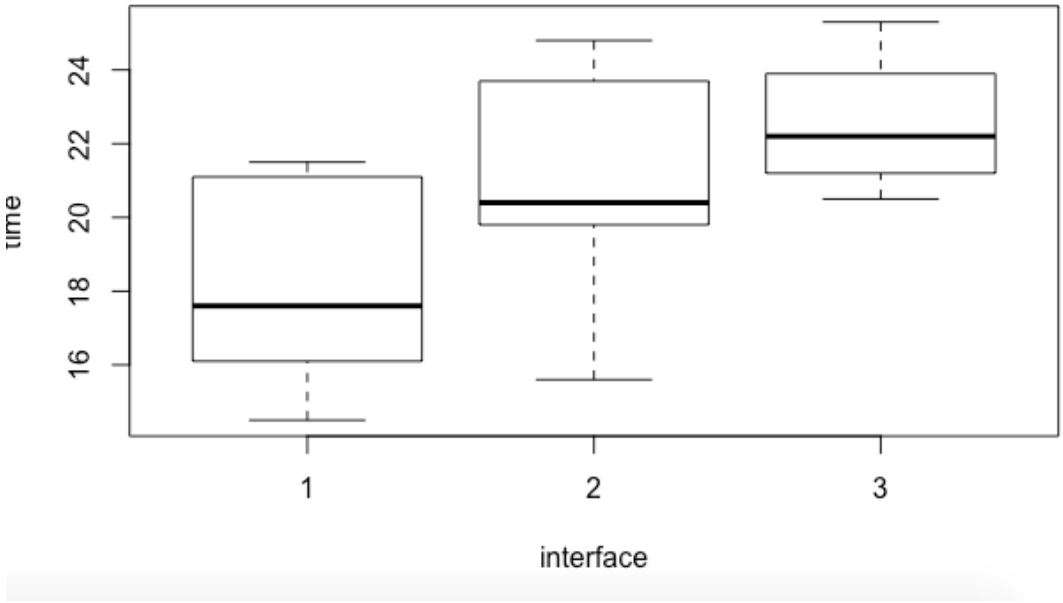

Figure 17: Boxplot of interface and time

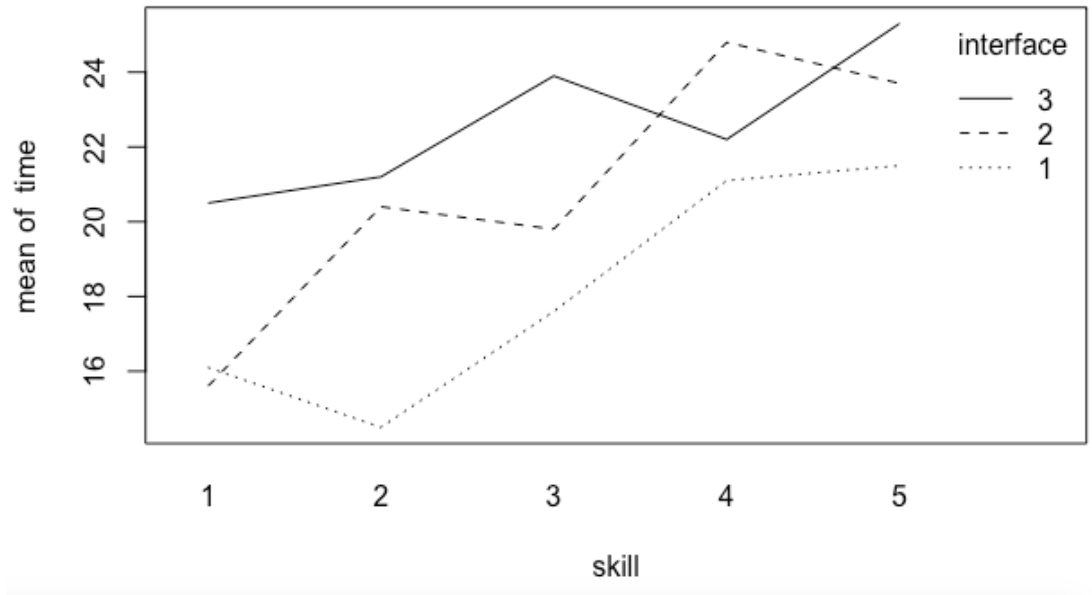

Figure 18: Interaction plot of skill interface and time

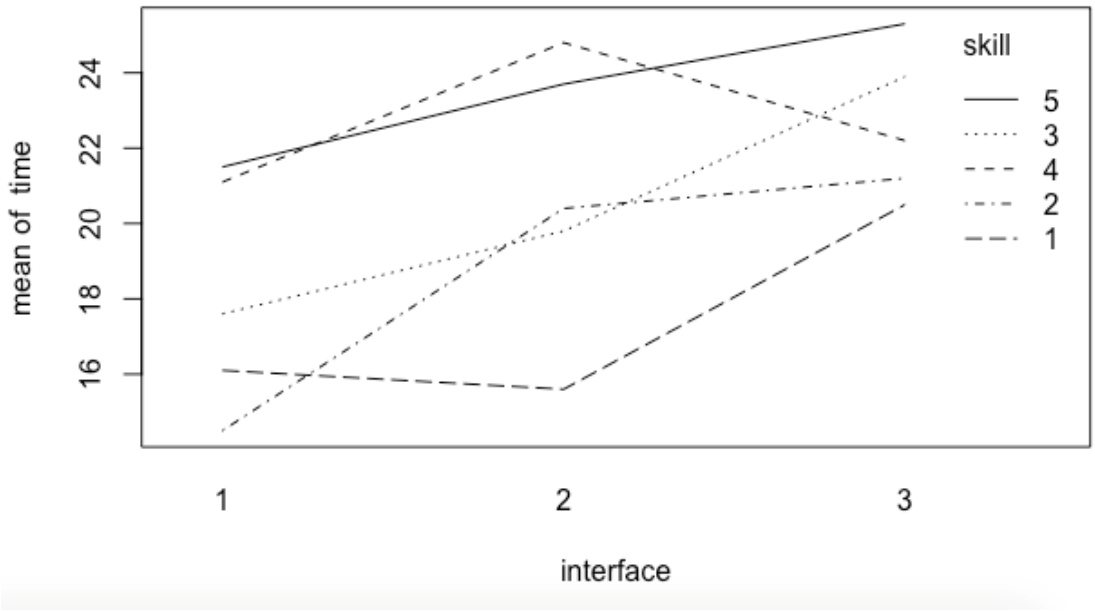

Figure 19: Interaction plot of interface skill and time

3. According to the given question we could know that H0: the search time is the same for all interfaces. And from the Anova and summary we can see that p-values are smaller than 0.05, so H0 is rejected, which means the search time is not the same for all interfaces.

```
> aovpen = lm(time~skill+interface,data=search)
> anova(aovpen)
Analysis of Variance Table

Response: time
          Df Sum Sq Mean Sq F value    Pr(>F)
skill      1 78.732  78.732  33.916 8.165e-05 ***
interface  1 49.729  49.729  21.422 0.0005817 ***
Residuals 12 27.856   2.321
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Figure 20: Anova

```
> summary(aovpen)

Call:
lm(formula = time ~ skill + interface, data = search)

Residuals:
     Min       1Q   Median       3Q      Max
-2.19667 -0.73167 -0.05667  1.07333  2.63333

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  11.2267     1.3341   8.415 2.23e-06 ***
skill         1.6200     0.2782   5.824 8.16e-05 ***
interface     2.2300     0.4818   4.628 0.000582 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.524 on 12 degrees of freedom
Multiple R-squared:  0.8218,    Adjusted R-squared:  0.7921
F-statistic: 27.67 on 2 and 12 DF,  p-value: 3.203e-05
```

Figure 21: Summary

4. From the Figure 18 we can see that the approximate search time for a level 4 skill user using interface 3 is 22.
5. From the Figures shown below we could see that residuals are from a normal population, so it’s possible to use Anova test.

```
searchaov=lm(time~skill+interface,data=search)
par(mfrow=c(1,2))
qqnorm(residuals(searchaov))
plot(fitted(searchaov),residuals(searchaov))
qqnorm(residuals(searchaov))
```

Figure 22: Codes

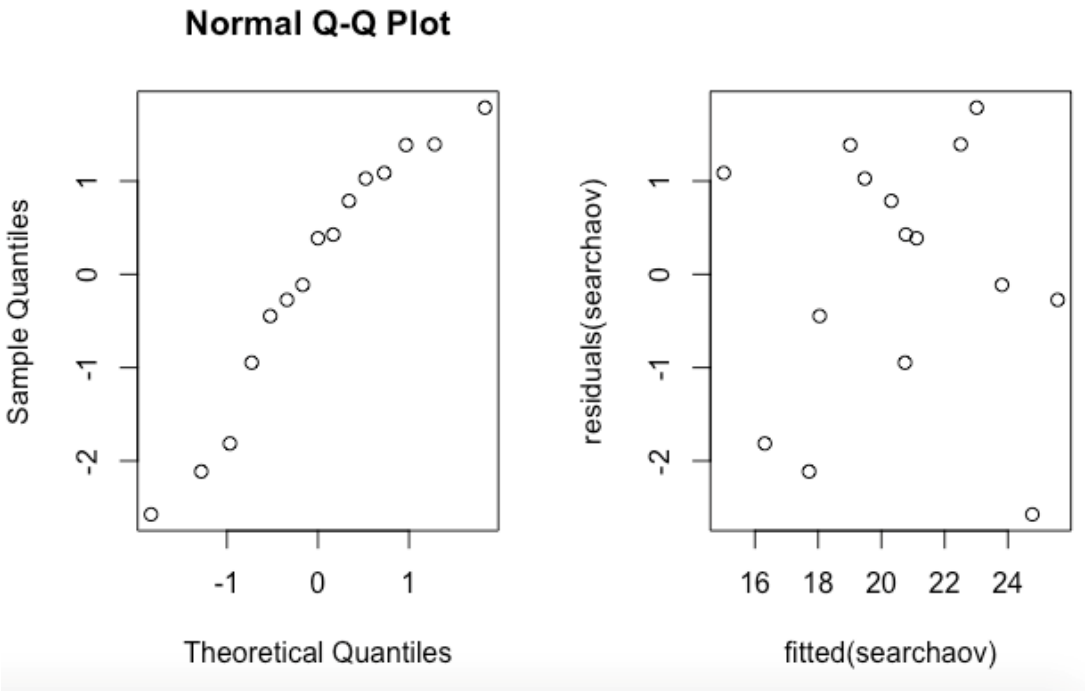

Figure 23: QQ-plot and residuals

6. According to the question we could assume that H0: there is no effect of interfaces. As the p-value of friedman.test is 0.04076<0.05, we reject H0, which means that there are some effects of interfaces.

```
> friedman.test(time,interface, skill)

        Friedman rank sum test

data:  time, interface and skill
Friedman chi-squared = 6.4, df = 2, p-value = 0.04076
```

Figure 24: Friedman test

7. According to the question we can assume H0: search time is the same for all interfaces. From Figure 26: the one-way Avona ignoring skill, we can see that p-value is 0.09642>0.05, so H0 is not rejected, which is not as same as the result we gain in question 2.3. But it is not useful to perform this test on the given dataset, because it cannot test H0 in different blocks. Only when there is no effect of skill we could use this one-way Avona test but the assumption is not met.

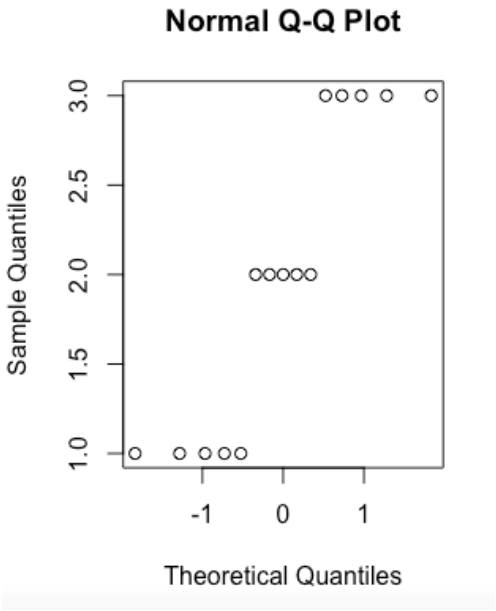

Figure 25: QQ-plot of interface

```
> aovpen2 = lm(time~interface, data = search)
> anova(aovpen2)
Analysis of Variance Table

Response: time
          Df  Sum Sq Mean Sq F value  Pr(>F)
interface  2  50.465  25.233  2.8605 0.09642 .
Residuals 12 105.852   8.821
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Figure 26: Anova

EXERCISE 3

3.1

We use incomplete block design by formula ‘acidity ~ starter + batch + position’, because we do not interest about the batch and position.

First, we study the boxplot of “acidity~batch”, “acidity~position”, “acidity~starter”. We can find starter 4 has a larger acidity than others.

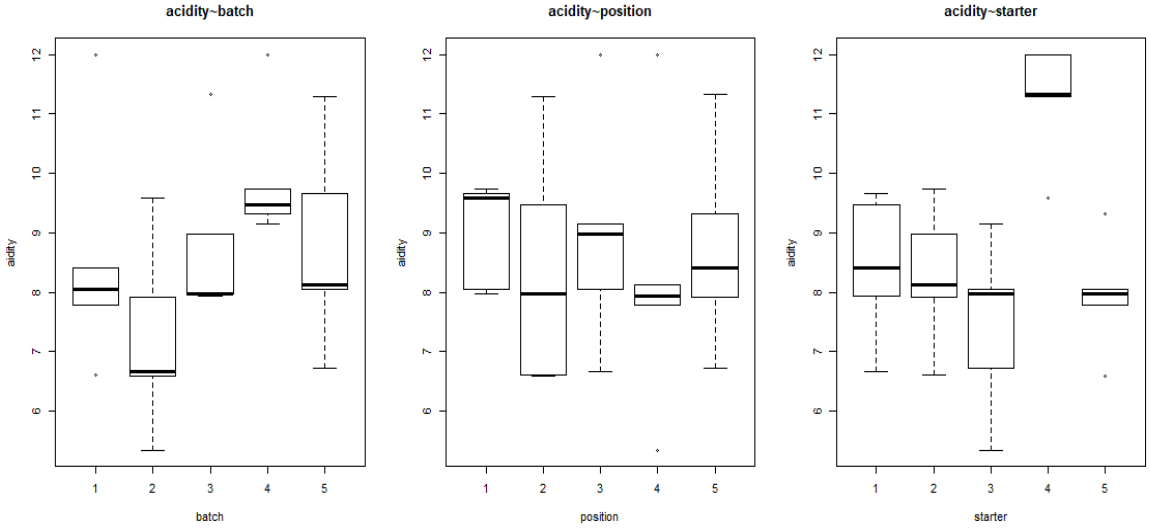

Fig.27 boxplot of "acidity~batch", "acidity~position", "acidity~starter"

Then, we show the interaction plot between starter~position and starter~batch.

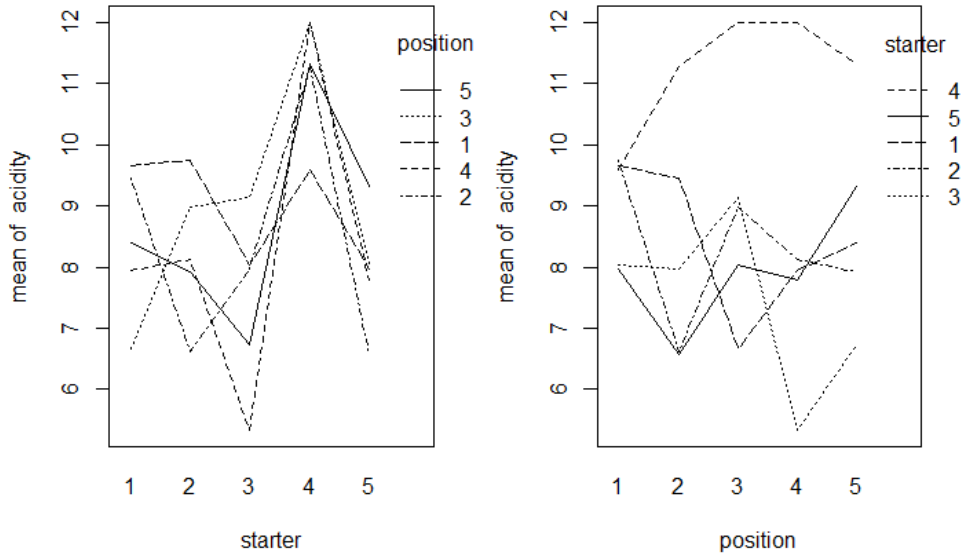

Fig.28 interaction plot between starter~position

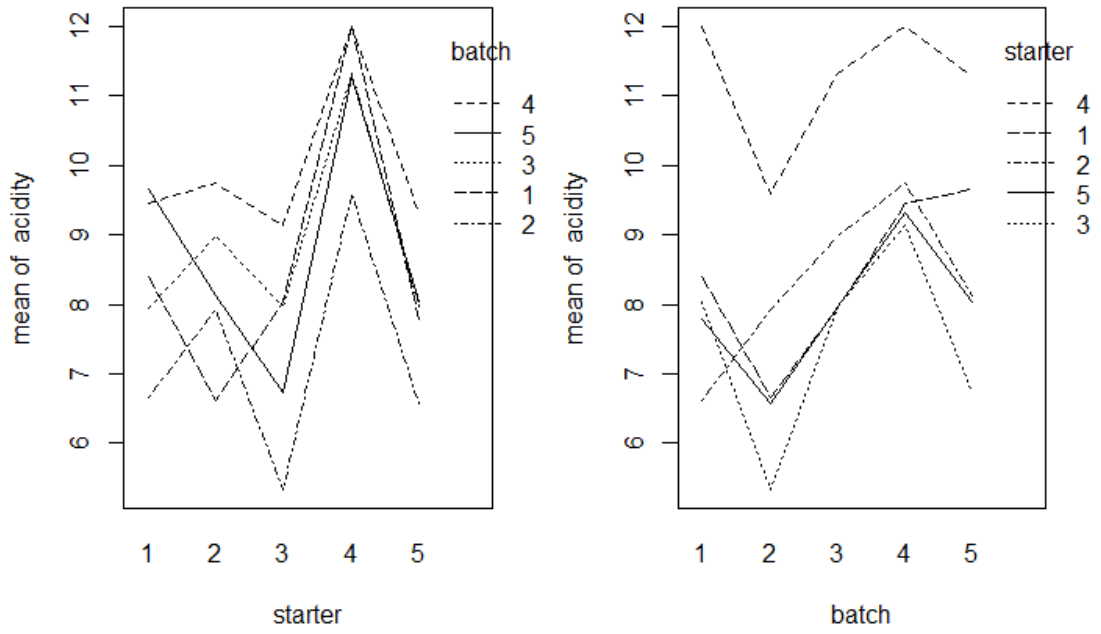

Fig.29 interaction plot between starter~batch

Next, we do the Anova. The starter effects are significantly different from 0 (significant influence on acidity) ($p<0.05$, reject H0). The batch are also significantly different from 0 (significant influence on acidity) ($p<0.05$, reject H0), but this was not the research question. The position effects are not significantly different from 0 ($p>0.05$, cannot reject H0).

Result of Anova:

```
> anova(acidaov)
Analysis of Variance Table

Response: acidity
          Df Sum Sq Mean Sq F value    Pr(>F)
starter    4 44.136 11.0340 20.2080 2.904e-05 ***
batch      4 18.778  4.6944  8.5975  0.001632 **
position   4  2.348  0.5870  1.0750  0.411191
Residuals 12  6.552  0.5460
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Finally, we get the summary. The acidity of starter4 is 2.8 higher than starter1. Also, the p-value of starter2 is much less than 0.05 (we cannot reject H0).

Starter4 has significant difference between starter1 on the acidity. Batch2 and batch4 also have significant difference between batch1 on the acidity, but we do not interest about the batch and position.

Result of summary:

```
> summary(acidaov)

Call:
lm(formula = acidity ~ starter + batch + position, data = cream)

Residuals:
    Min      1Q  Median      3Q     Max
-1.2836 -0.2336  0.0384  0.3584  1.0204

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   8.6616     0.5329  16.255 1.55e-09 ***
starter2     -0.1500     0.4673  -0.321   0.7538
starter3     -0.9800     0.4673  -2.097   0.0579 .
starter4      2.8100     0.4673   6.013 6.10e-05 ***
starter5     -0.4840     0.4673  -1.036   0.3208
batch2       -1.3480     0.4673  -2.884   0.0137 *
batch3        0.2760     0.4673   0.591   0.5658
batch4        1.3680     0.4673   2.927   0.0127 *
batch5        0.2000     0.4673   0.428   0.6763
position2    -0.6180     0.4673  -1.322   0.2107
position3    -0.0380     0.4673  -0.081   0.9365
position4    -0.7640     0.4673  -1.635   0.1280
position5    -0.2640     0.4673  -0.565   0.5825
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7389 on 12 degrees of freedom
Multiple R-squared:  0.9088,    Adjusted R-squared:  0.8175
F-statistic:  9.96 on 12 and 12 DF,  p-value: 0.0001777
```

Code of 3.1:

```
cream = read.table("cream.txt",header=TRUE)
cream$batch = factor(cream$batch)
cream$position = factor(cream$position)
cream$starter = factor(cream$starter)
attach(cream)
acidaov = lm(acidity~starter+batch+position,data=cream)
anova(acidaov)
summary(acidaov)

cream = read.table("cream.txt",header=TRUE)
cream$batch = factor(cream$batch)
cream$position = factor(cream$position)
cream$starter = factor(cream$starter)
acidaov = lm(acidity~starter+batch+position,data=cream)
anova(acidaov)
summary(acidaov)
```

```
attach(cream)
par(mfrow=c(1,3))
boxplot(acidity~batch,main="acidity~batch",ylab="aidity", xlab="batch");
boxplot(acidity~position,main="acidity~position",ylab="aidity", xlab="position");
boxplot(acidity~starter,main="acidity~starter",ylab="aidity", xlab="starter")

attach(cream)
par(mfrow=c(1,2))
interaction.plot(starter,position,acidity); interaction.plot(position,starter,acidity)
interaction.plot(starter,batch,acidity); interaction.plot(batch,starter,acidity)
```

3.2

We use multiple testing and comparisons to get the table of p-value. We can find that starter4 leads to significantly different acidity, because the p-values of "4-1", "4-2", "4-3", "5-4" all less than level 5%. We reject the H0, so starter4 is significantly different from all other starters. Starter4 leads to significantly different acidity.

```
> summary(creammult)

         Simultaneous Tests for General Linear Hypotheses

Multiple Comparisons of Means: Tukey Contrasts


Fit: lm(formula = acidity ~ starter + batch + position, data = cream)

Linear Hypotheses:
           Estimate Std. Error t value Pr(>|t|)
2 - 1 == 0  -0.1500     0.4673  -0.321    0.997
3 - 1 == 0  -0.9800     0.4673  -2.097    0.282
4 - 1 == 0   2.8100     0.4673   6.013   <0.001 ***
5 - 1 == 0  -0.4840     0.4673  -1.036    0.834
3 - 2 == 0  -0.8300     0.4673  -1.776    0.429
4 - 2 == 0   2.9600     0.4673   6.334   <0.001 ***
5 - 2 == 0  -0.3340     0.4673  -0.715    0.949
4 - 3 == 0   3.7900     0.4673   8.110   <0.001 ***
5 - 3 == 0   0.4960     0.4673   1.061    0.822
5 - 4 == 0  -3.2940     0.4673  -7.048   <0.001 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
(Adjusted p values reported -- single-step method)
```

Code of 3.2:

```
creammult=glht(acidaov,linfct=mcp(starter="Tukey"))
summary(creammult)
```

3.3

It means there is no significant difference between p-value (p-value=0.997) of "2-1" in question (2) and p-value (p-value=0.754) of starter2 in question (1). We can find p-value of '2-1' is smaller than the simultaneous p-value (0.997), and it is not a coincidence. The reason is that simultaneous confidence intervals have confidence level of 95%.

3.4

From the table of confidence intervals, we can find the intervals of [1.3204, 4.2996], [1.4704 , 4.4496], [2.3004 , 5.2796], [-4.7836 ,-1.8044] (4-1, 4-2, 4-3, 5-4) are not contain the number 0.

Therefore, the starter4 lead to significantly different between other starters.

```
> confint(creammult)

         Simultaneous Confidence Intervals

Multiple Comparisons of Means: Tukey Contrasts


Fit: lm(formula = acidity ~ starter + batch + position, data = cream)

Quantile = 3.1874
95% family-wise confidence level


Linear Hypotheses:
           Estimate lwr     upr
2 - 1 == 0 -0.1500  -1.6396  1.3396
3 - 1 == 0 -0.9800  -2.4696  0.5096
4 - 1 == 0  2.8100   1.3204  4.2996
5 - 1 == 0 -0.4840  -1.9736  1.0056
3 - 2 == 0 -0.8300  -2.3196  0.6596
4 - 2 == 0  2.9600   1.4704  4.4496
5 - 2 == 0 -0.3340  -1.8236  1.1556
4 - 3 == 0  3.7900   2.3004  5.2796
5 - 3 == 0  0.4960  -0.9936  1.9856
5 - 4 == 0 -3.2940  -4.7836 -1.8044
```

Code of 3.4:

```
creammult=glht(acidaov,linfct=mcp(starter="Tukey"))
confint(creammult)
```

EXERCISE 4

4.1

Fixed Effects: There is not a significant treatment (feeding stuff) effect, because the treatment p-value=0.75>0.05. There is no significant influences milk production by the type of feeding stuffs.

```
> anova(cowlm)
Analysis of Variance Table

Response: milk
          Df  Sum Sq Mean Sq  F value    Pr(>F)
treatment  1    0.27   0.269   0.1085   0.75147
per        1   25.39  25.387  10.2462   0.01505 *
id         8 2467.47 308.434 124.4832 7.494e-07 ***
Residuals  7   17.34   2.478
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Code of 4.1:

```
cow = read.table("cow.txt",header=TRUE)
cow$id=factor(cow$id)
cow$per=factor(cow$per)
attach(cow)
cowlm=lm(milk~treatment+per+id,data=cow)
anova(cowlm)
```

4.2

As the results shown on 4.1, There is no statistically significant feeding stuff effect on milk. The feeding stuff B gives 0.51 less than feeding stuff A. Also, the p-value of treatment is 0.51>0.05 (cannot reject H0). There is a statistically significant period effect. Period 2 gives 2.39 less than period 1. There is also a statistically significant cow(=id) effect. For example, id2-cow gives 23 more than id1-cow, but we do not interest on the id effects.

```
> summary(cowlm)

Call:
lm(formula = milk ~ treatment + per + id, data = cow)

Residuals:
    Min      1Q  Median      3Q     Max
-2.2600 -0.4375  0.0000  0.4375  2.2600

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  30.3000     1.2444  24.349 5.02e-08 ***
treatmentB   -0.5100     0.7466  -0.683 0.516536
per2         -2.3900     0.7466  -3.201 0.015046 *
id2          23.0000     1.5741  14.612 1.68e-06 ***
id3          11.1500     1.5741   7.084 0.000196 ***
id4          -1.3500     1.5741  -0.858 0.419480
id5          -7.0500     1.5741  -4.479 0.002870 **
id6          23.4500     1.5741  14.898 1.47e-06 ***
id7          13.5500     1.5741   8.608 5.69e-05 ***
id8           4.9000     1.5741   3.113 0.017011 *
id9         -11.2000     1.5741  -7.115 0.000191 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.574 on 7 degrees of freedom
Multiple R-squared:  0.9931,    Adjusted R-squared:  0.9832
F-statistic: 100.6 on 10 and 7 DF,  p-value: 1.349e-06
```

Code of 4.2:

```
summary(cowlm)
```

4.3 crossover design with random effects

The number 133.14 under Random effects is the estimated variance of the normal population of the “individual effects” (bn).

```
> summary(cowlmer)
Linear mixed model fit by maximum likelihood  ['lmerMod']
Formula: milk ~ treatment + order + per + (1 | id)
   Data: cow

     AIC      BIC   logLik deviance df.resid
   119.3    124.7    -53.7    107.3       12

Scaled residuals:
     Min       1Q   Median       3Q      Max
-1.53112 -0.37104  0.02686  0.26748  1.72489

Random effects:
 Groups   Name        Variance Std.Dev.
 id       (Intercept) 133.145  11.539
 Residual               1.927   1.388
Number of obs: 18, groups:  id, 9

Fixed effects:
            Estimate Std. Error t value
(Intercept)  38.5000     5.8110   6.625
treatmentB   -0.5100     0.6585  -0.775
orderBA      -3.4700     7.7685  -0.447
per2         -2.3900     0.6585  -3.630

Correlation of Fixed Effects:
           (Intr) trtmnB ordrBA
treatmentB -0.063
orderBA    -0.743  0.000
per2       -0.063  0.111  0.000
```

By applying Anova with 2 arguments, we found that there is no significant effects by treatment (feeding stuff).

```
> cowlmer1=lmer(milk~order+per+(1|id),data=cow,REML=FALSE)

> anova(cowlmer1,cowlmer)
Data: cow
Models:
cowlmer1: milk ~ order + per + (1 | id)
cowlmer: milk ~ treatment + order + per + (1 | id)
         Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
cowlmer1  5 117.89 122.34 -53.946   107.89
cowlmer   6 119.31 124.65 -53.656   107.31 0.5807      1      0.446
>
```

The estimated treatment and period effects under Fixed effects are identical to those in the result of 4.1. The difference between the “fixed effects” and “mixed effects” is minor. Also, we got the similar result: There is no significant influences milk production by the type of feeding stuffs.

Code of 4.3:

```
cowlmer=lmer(milk~treatment+order+per+(1|id),data=cow,REML=FALSE)
summary(cowlmer)
cowlmer1=lmer(milk~order+per+(1|id),data=cow,REML=FALSE)
anova(cowlmer1,cowlmer)
```

4.4

From paired t-test, we can find the p-value=0.82>0.05, so we cannot reject H0 which means feeding stuff A and B do not have significant influence on the milk production. It is not a valid test for a difference in milk production, because this test cannot consider the period effects. When we delete the period effects on 4.1, we can get a similar p-value with the paired t-test. It has the similar result with 4.1.

```
> t.test(milk[treatment=="A"],milk[treatment=="B"],paired=TRUE)

        Paired t-test

data:  milk[treatment == "A"] and milk[treatment == "B"]
t = 0.22437, df = 8, p-value = 0.8281
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -2.267910  2.756799
sample estimates:
mean of the differences
              0.2444444
```

Code of 4.4:

```
attach(cow)
t.test(milk[treatment=="A"],milk[treatment=="B"],paired=TRUE)
```

EXERCISE 5

5.1

We a nausea vector which 0 means incidence of no nausea and 1 means incidence of nausea. We also build a medicin vector. Finally, we build a data.frame by combine the two vectors.

Code of 5.1:

```
nauseatable=read.table("nauseatable.txt", header = TRUE)
nausea=c(rep(0,sum(nauseatable$Incidence.of.no.nausea)),
         rep(1,sum(nauseatable$Incidence.of.Nausea)))
medicin=c(rep("Chlorpromazine",100),
          rep("Pentobarbital(100mg)",32),
          rep("Pentobarbital(150mg)",48),
          rep("Chlorpromazine", 52),
          rep("Pentobarbital(100mg)",35),
          rep("Pentobarbital(150mg)",37))
nausea.frame=data.frame(nausea,medicin)
```

5.2

We can find the xtabs has the same result with the original file, which the rows show the 3

different medicins and 2 columns show the nausea.

```
> xtabs(~medicin+nausea)
                     nausea
medicin                0   1
  Chlorpromazine     100  52
  Pentobarbital(100mg)  32  35
  Pentobarbital(150mg)  48  37
```

Code of 5.2:

```
attach(nausea.frame)
xtabs(~medicin+nausea)
```

5.3

Permutation test results show that the pr=0.041<0.05 (reject H0), and the pl=0.959>0.05 (cannot reject H0). Therefore, different medicins do not work equally well against nausea.

```
> print(myt)
X-squared
 6.624765

> print(pl)
[1] 0.959

> print(pr)
[1] 0.041
```

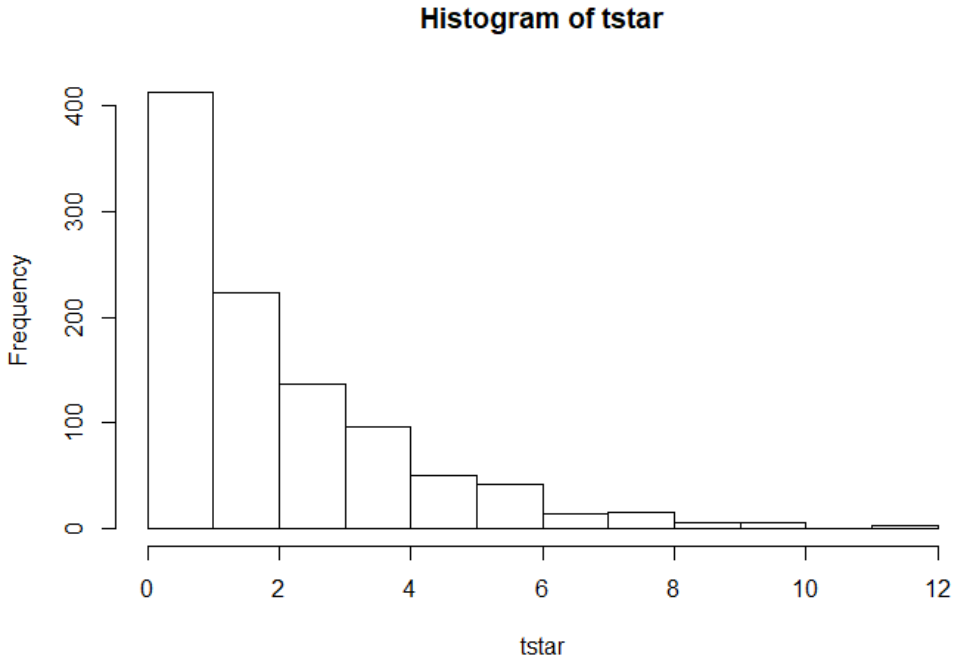

Fig.30 Histogram of tstar

Code of 5.3:

```
attach(nausea.frame)
B = 1000
tstar = numeric(B)
for (i in 1:B) {
  treatstar=sample(medicin)
  tstar[i]=chisq.test(xtabs(~treatstar+nausea))[[1]]
}
myt = chisq.test(xtabs(~medicin+nausea))[[1]]
hist(tstar)
pl = sum(tstar<myt)/B
pr = sum(tstar>myt)/B
```

5.4

The p-value found by the permutation test and found from the chisquare test for contingency tables both smaller than 0.05 (reject H0). Different kinds of medicine have different effect on nausea.

Result:

```
> print(pnew)

        Pearson's Chi-squared test

data:  xtabs(~medicin + nausea)
X-squared = 6.6248, df = 2, p-value = 0.03643
```

Code of 5.4:

```
pnew = chisq.test(xtabs(~medicin+nausea))
```

Exercise 6

1. The scatter plots of the candidate explanatory variables against each other and against the response variable is shown in Fig.31. The code of generating this scatter plots is shown in Fig.32.

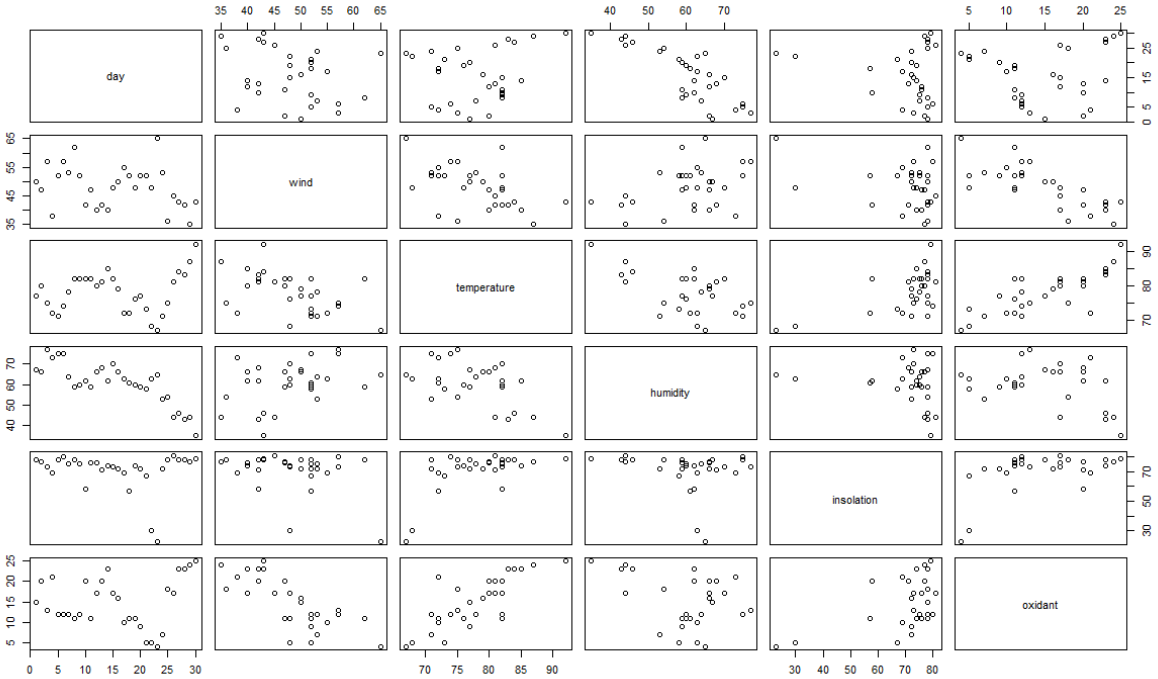

Figure 31: Scatter plots

```
airpollution=read.table("airpollution.txt",header=TRUE)
factors <- names(airpollution)
pairs(airpollution[,factors])
```

Figure 32: Code for scatter plots

It is useful to judge linear model here because scatterplot matrices are a great way to roughly determine if there are linear correlations between multiple variables. From the Scatter plots we could there may be linear correlations between such as wind and oxidant or temperature and oxidant. We could also observe the collinearity, the outliers and trend of each plot.

2. We first create a data-frame for each explanatory variables: “day”, “wind”, “temperature”, “humidity”, “insolation” and the response variable: “oxidant”, the code is as shown in Fig.33.

```
pairs(airpollution[,c(1:5,6)])
```

Figure 33: Create separate dataframe

Then we calculate the results via linear regression, the code is as shown in Fig.4.

```
summary(lm(oxidant~day,data=airpollution))
summary(lm(oxidant~wind,data=airpollution))
summary(lm(oxidant~temperature,data=airpollution))
summary(lm(oxidant~humidity,data=airpollution))
summary(lm(oxidant~insolation,data=airpollution))
```

Figure 34: Codes for step-up method linear regression model 1

The outputs are shown as Fig.35 to Fig.39.

```
Call:
lm(formula = oxidant ~ day, data = airpollution)

Residuals:
     Min       1Q   Median       3Q      Max
-11.3373  -3.8537   0.1298   5.5403   9.1613

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 13.68966    2.28580   5.989 1.89e-06 ***
day          0.07164    0.12876   0.556    0.582
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.104 on 28 degrees of freedom
Multiple R-squared:  0.01093,   Adjusted R-squared:  -0.02439
F-statistic: 0.3095 on 1 and 28 DF,  p-value: 0.5824
```

Figure 35: Output of linear regression model: oxidant and day

```
Call:
lm(formula = oxidant ~ wind, data = airpollution)

Residuals:
    Min      1Q  Median      3Q     Max
-9.9266 -2.5923  0.2065  2.6636  6.9077

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  45.3171     4.8976   9.253 5.19e-10 ***
wind         -0.6331     0.1005  -6.300 8.20e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.948 on 28 degrees of freedom
Multiple R-squared:  0.5863,    Adjusted R-squared:  0.5715
F-statistic: 39.68 on 1 and 28 DF,  p-value: 8.205e-07
```

Figure 36: Output of linear regression model: oxidant and wind

```
Call:
lm(formula = oxidant ~ temperature, data = airpollution)

Residuals:
    Min      1Q  Median      3Q     Max
-6.9400 -2.2138  0.3775  2.5550 10.9099

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -46.4292     9.9542  -4.664 6.94e-05 ***
temperature   0.7850     0.1273   6.168 1.17e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.997 on 28 degrees of freedom
Multiple R-squared:  0.576,     Adjusted R-squared:  0.5609
F-statistic: 38.04 on 1 and 28 DF,  p-value: 1.167e-06
```

Figure 37: Output of linear regression model: oxidant and temperature

```
Call:
lm(formula = oxidant ~ humidity, data = airpollution)

Residuals:
     Min       1Q   Median       3Q      Max
-10.3358  -4.0749   0.8782   4.7800   8.7957

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  27.4446     6.4368   4.264 0.000206 ***
humidity     -0.2088     0.1049  -1.991 0.056317 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.745 on 28 degrees of freedom
Multiple R-squared:  0.124,     Adjusted R-squared:  0.09273
F-statistic: 3.964 on 1 and 28 DF,  p-value: 0.05632
```

Figure 38: Output of linear regression model: oxidant and humidity

```
Call:
lm(formula = oxidant ~ insolation, data = airpollution)

Residuals:
    Min      1Q  Median      3Q     Max
-8.9723 -4.4841 -0.3281  4.7631  8.2686

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.43279    5.32967  -0.269  0.79003
insolation   0.22993    0.07424   3.097  0.00441 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.297 on 28 degrees of freedom
Multiple R-squared:  0.2552,    Adjusted R-squared:  0.2286
F-statistic: 9.592 on 1 and 28 DF,  p-value: 0.004411
```

Figure 39: Output of linear regression model: oxidant and insolation

From the figures above we could know that the largest value of "Multiple R-squared" is 0.5863, which exists in the wind and oxidant linear regression model. A value close to 1 means that the linear regression model can explain the measured response values very well using a linear function of the explanatory variables. So the best model is the oxidant~wind model.

We use Person's correlation test to check whether the extension is useful. The code and output is as shown in Fig.40.

```
> cor.test(airpollution$oxidant,airpollution$wind)

        Pearson's product-moment correlation

data:  airpollution$oxidant and airpollution$wind
t = -6.2996, df = 28, p-value = 8.205e-07
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.8825259 -0.5598984
sample estimates:
       cor
-0.7657126
```

Figure 40: Correlation test for oxidant and wind

From the output of correlation test we could know that the absolute p-value is 8.205e-07, so the extension is useful. Based on oxidant~wind, we add other explanatory variables, the code is as shown in Fig.41.

```
summary(lm(oxidant~wind+day,data=airpollution))
summary(lm(oxidant~wind+temperature,data=airpollution))
summary(lm(oxidant~wind+humidity,data=airpollution))
summary(lm(oxidant~wind+insolation,data=airpollution))
```

Figure 41: Codes for step-up method linear regression model 2

The output is as shown in Fig.42 to Fig.45.

```
Call:
lm(formula = oxidant ~ wind + day, data = airpollution)

Residuals:
    Min      1Q  Median      3Q     Max
-9.4129 -2.5621  0.4498  2.3827  7.9267

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 47.84224    5.62785   8.501 4.10e-09 ***
wind        -0.65984    0.10489  -6.291 9.87e-07 ***
day         -0.07986    0.08691  -0.919    0.366
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.959 on 27 degrees of freedom
Multiple R-squared:  0.5989,    Adjusted R-squared:  0.5691
F-statistic: 20.15 on 2 and 27 DF,  p-value: 4.411e-06
```

Figure 42: Output of linear regression model: oxidant+wind and day

```
Call:
lm(formula = oxidant ~ wind + temperature, data = airpollution)

Residuals:
    Min      1Q  Median      3Q     Max
-6.3939 -1.8608  0.5826  1.9461  4.9661

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -5.20334   11.11810  -0.468    0.644
wind        -0.42706    0.08645  -4.940 3.58e-05 ***
temperature  0.52035    0.10813   4.812 5.05e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.95 on 27 degrees of freedom
Multiple R-squared:  0.7773,    Adjusted R-squared:  0.7608
F-statistic: 47.12 on 2 and 27 DF,  p-value: 1.563e-09
```

Figure 43: Output of linear regression model: oxidant+wind and temperature

```
Call:
lm(formula = oxidant ~ wind + humidity, data = airpollution)

Residuals:
    Min      1Q  Median      3Q     Max
-9.8120 -2.2808  0.3433  3.0476  5.8757

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 46.91570    5.68573   8.251 7.38e-09 ***
wind        -0.60955    0.10971  -5.556 6.86e-06 ***
humidity    -0.04516    0.07866  -0.574    0.571
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.996 on 27 degrees of freedom
Multiple R-squared:  0.5913,    Adjusted R-squared:  0.561
F-statistic: 19.53 on 2 and 27 DF,  p-value: 5.674e-06
```

Figure 44: Output of linear regression model: oxidant+wind and humidity

```
Call:
lm(formula = oxidant ~ wind + insolation, data = airpollution)

Residuals:
    Min      1Q  Median      3Q     Max
-7.2119 -2.7198  0.4815  2.8733  6.2012

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 32.32615    6.97098   4.637 8.07e-05 ***
wind        -0.55639    0.09778  -5.690 4.81e-06 ***
insolation   0.13161    0.05383   2.445   0.0213 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.638 on 27 degrees of freedom
Multiple R-squared:  0.6613,    Adjusted R-squared:  0.6362
F-statistic: 26.36 on 2 and 27 DF,  p-value: 4.491e-07
```

Figure 45: Output of linear regression model: oxidant+wind and insolation

From Fig.42 to Fig.45, we could know the best R-squared value is 0.7773, which exist in the oxidant~wind+temperature. We use Person's correlation test to check whether the extension is useful. The code and output is as shown in Fig.46.

```
> cor.test(airpollution$oxidant,airpollution$temperature)

        Pearson's product-moment correlation

data:  airpollution$oxidant and airpollution$temperature
t = 6.1677, df = 28, p-value = 1.167e-06
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.5487258 0.8789078
sample estimates:
      cor
0.7589575
```

Figure 46: Correlation test for oxidant and temperature

From the output of correlation test we could know that the absolute value of cor is 0.7589575,

so the extension is useful.So we do further calculation based on this step. The code is as shown in Fig.47.

```
summary(lm(oxidant~wind+temperature+day,data=airpollution))
summary(lm(oxidant~wind+temperature+humidity,data=airpollution))
summary(lm(oxidant~wind+temperature+insolation,data=airpollution))
```

Figure 47: Codes for step-up method linear regression model 3

The output is as shown in Fig.48 to Fig.50.

```
Call:
lm(formula = oxidant ~ wind + temperature + day, data = airpollution)

Residuals:
    Min      1Q  Median      3Q     Max
-6.9010 -1.3477  0.1596  1.7766  3.9405

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -2.98987   10.94466  -0.273    0.787
wind        -0.45604    0.08644  -5.276 1.63e-05 ***
temperature  0.52918    0.10568   5.008 3.29e-05 ***
day         -0.09711    0.06328  -1.535    0.137
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.878 on 26 degrees of freedom
Multiple R-squared:  0.7958,    Adjusted R-squared:  0.7722
F-statistic: 33.78 on 3 and 26 DF,  p-value: 4.042e-09
```

Figure 48: Output of linear regression model: oxidant+wind+temperature and day

```
Call:
lm(formula = oxidant ~ wind + temperature + humidity, data = airpollution)

Residuals:
    Min      1Q  Median      3Q     Max
-6.5887 -1.1686  0.1978  1.9004  4.1544

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -16.60697   13.07154  -1.270    0.215
wind         -0.44620    0.08513  -5.241 1.78e-05 ***
temperature   0.60190    0.11764   5.117 2.47e-05 ***
humidity      0.09850    0.06316   1.559    0.131
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.874 on 26 degrees of freedom
Multiple R-squared:  0.7964,    Adjusted R-squared:  0.7729
F-statistic: 33.89 on 3 and 26 DF,  p-value: 3.904e-09
```

Figure 49: Output of linear regression model: oxidant+wind+temperature and humidity

```
Call:
lm(formula = oxidant ~ wind + temperature + insolation, data = airpollution)

Residuals:
   Min     1Q Median     3Q    Max
-6.407 -2.056  1.012  1.760  4.792

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -4.45496   11.26714  -0.395 0.695778
wind        -0.42353    0.08737  -4.848 5.02e-05 ***
temperature  0.47558    0.12564   3.785 0.000816 ***
insolation   0.03646    0.05071   0.719 0.478636
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.976 on 26 degrees of freedom
Multiple R-squared:  0.7816,    Adjusted R-squared:  0.7565
F-statistic: 31.02 on 3 and 26 DF,  p-value: 9.583e-09
```

Figure 50: Output of linear regression model: oxidant+wind+temperature and insolation

From Fig.48 to Fig.50, we could know the best R-squared value is 0.7964, which exist in the oxidant~wind+temperature+humidity. However, we could know that adding the third explanatory variable will not get a significantly increase in R-squared value, therefore we should stop the step.

Adding either day or insolation yields insignificant explanatory variables. Therefore, we should stop at the previous step as shown in Fig.51, which is the same as Fig.43.

```
Call:
lm(formula = oxidant ~ wind + temperature, data = airpollution)

Residuals:
    Min      1Q  Median      3Q     Max
-6.3939 -1.8608  0.5826  1.9461  4.9661

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -5.20334   11.11810  -0.468    0.644
wind        -0.42706    0.08645  -4.940 3.58e-05 ***
temperature  0.52035    0.10813   4.812 5.05e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.95 on 27 degrees of freedom
Multiple R-squared:  0.7773,    Adjusted R-squared:  0.7608
F-statistic: 47.12 on 2 and 27 DF,  p-value: 1.563e-09
```

Figure 51: Output: oxidant+wind+temperature

The resulting model of the step-up methods is:

Oxidant= -5.20334 - 0.42706*wind + 0.52035*temperature + error

3. We use the step-down method to realize. The code and first step output is shown in Fig.22.

```
> summary(lm(oxidant~day+wind+temperature+humidity+insolation,data=airpollution))

Call:
lm(formula = oxidant ~ day + wind + temperature + humidity +
    insolation, data = airpollution)

Residuals:
    Min      1Q  Median      3Q     Max
-6.6920 -1.1675  0.2582  1.8289  4.0773

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -12.04010   21.20961  -0.568  0.57553
day          -0.02997    0.13995  -0.214  0.83227
wind         -0.44749    0.09103  -4.916 5.14e-05 ***
temperature   0.55714    0.15347   3.630  0.00133 **
humidity      0.06818    0.13336   0.511  0.61384
insolation    0.01822    0.05583   0.326  0.74694
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.977 on 24 degrees of freedom
Multiple R-squared:  0.7984,    Adjusted R-squared:  0.7564
F-statistic: 19.01 on 5 and 24 DF,  p-value: 1.203e-07
```

Figure 52: First step of step-down method

We could know that the p-value of day is highest and larger than 0.05, so we remove it. The next step code and output is shown in Fig.53.

```
> summary(lm(oxidant~wind+temperature+humidity+insolation,data=airpollution))

Call:
lm(formula = oxidant ~ wind + temperature + humidity + insolation,
    data = airpollution)

Residuals:
    Min      1Q  Median      3Q     Max
-6.5861 -1.0961  0.3512  1.7570  4.0712

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -15.49370   13.50647  -1.147  0.26219
wind         -0.44291    0.08678  -5.104 2.85e-05 ***
temperature   0.56933    0.13977   4.073  0.00041 ***
humidity      0.09292    0.06535   1.422  0.16743
insolation    0.02275    0.05067   0.449  0.65728
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.92 on 25 degrees of freedom
Multiple R-squared:  0.798,     Adjusted R-squared:  0.7657
F-statistic: 24.69 on 4 and 25 DF,  p-value: 2.279e-08
```

Figure 53: Second step of step-down method

We could know that the p-value of insolation is highest and larger than 0.05, so we remove it. The next step code and output is shown in Fig.54.

```
> summary(lm(oxidant~wind+temperature+humidity,data=airpollution))

Call:
lm(formula = oxidant ~ wind + temperature + humidity, data = airpollution)

Residuals:
    Min      1Q  Median      3Q     Max
-6.5887 -1.1686  0.1978  1.9004  4.1544

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -16.60697   13.07154  -1.270    0.215
wind         -0.44620    0.08513  -5.241 1.78e-05 ***
temperature   0.60190    0.11764   5.117 2.47e-05 ***
humidity      0.09850    0.06316   1.559    0.131
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.874 on 26 degrees of freedom
Multiple R-squared:  0.7964,    Adjusted R-squared:  0.7729
F-statistic: 33.89 on 3 and 26 DF,  p-value: 3.904e-09
```

Figure 54: Third step of step-down method

We could know that the p-value of humidity is highest and larger than 0.05, so we remove it. The next step code and output is shown in Fig.55.

```
> summary(lm(oxidant~wind+temperature,data=airpollution))

Call:
lm(formula = oxidant ~ wind + temperature, data = airpollution)

Residuals:
    Min      1Q  Median      3Q     Max
-6.3939 -1.8608  0.5826  1.9461  4.9661

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -5.20334   11.11810  -0.468    0.644
wind        -0.42706    0.08645  -4.940 3.58e-05 ***
temperature  0.52035    0.10813   4.812 5.05e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.95 on 27 degrees of freedom
Multiple R-squared:  0.7773,    Adjusted R-squared:  0.7608
F-statistic: 47.12 on 2 and 27 DF,  p-value: 1.563e-09
```

Figure 55: Fourth step of step-down method

We could know the p-value of temperature is highest and smaller than 0.05. So all explanatory variables in the model are significant.

The resulting of the step-up method is:

Oxidant = -5.20334+0.52035*temperature-0.42706*wind+error

4. From 2 and 3 we conclude that the final estimates of the parameters of the model is:
   Oxidant = -5.20334+0.52035*temperature-0.42706*wind+error

5. We use normal QQ-plot to investigate the normality of the residuals. The code is as shown in Fig.56 and the QQ plot is as shown in Fig.57.

```
odtlm=lm(oxidant~wind+temperature,data=airpollution)
qqnorm(residuals(odtlm))
```

Figure 56: Code for QQ-plot

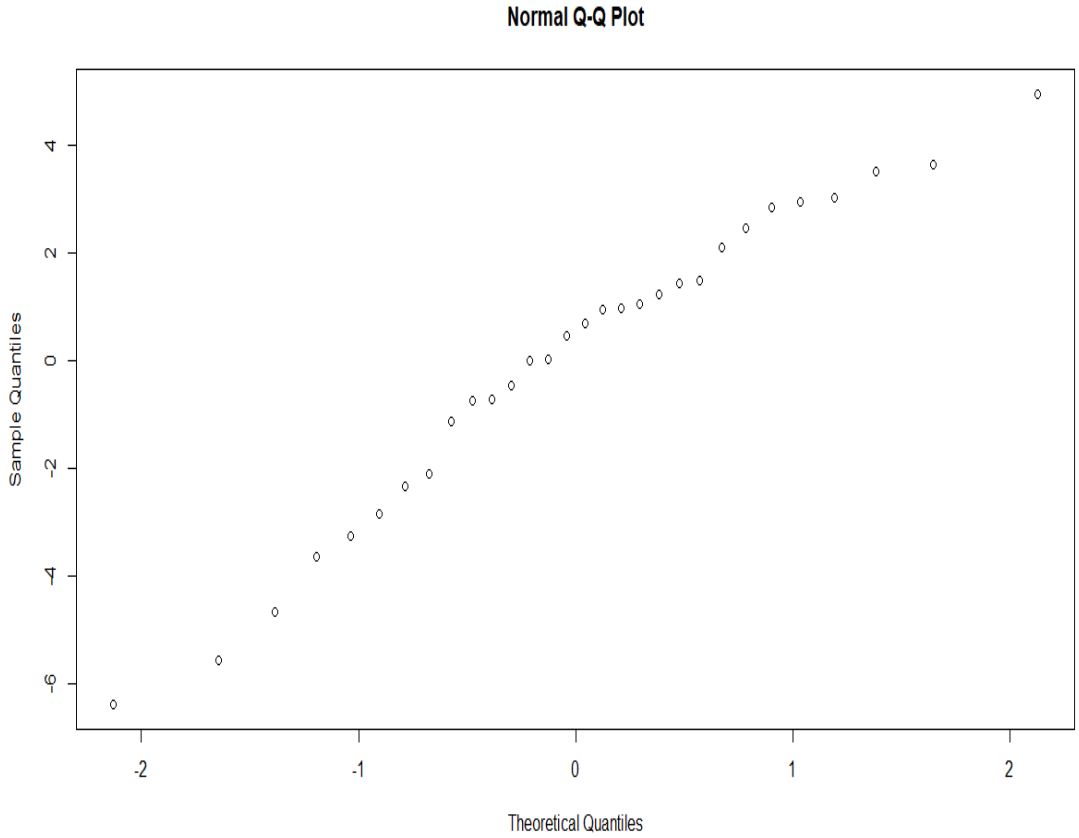

Figure 57: QQ-plot

We do the shapiro test to check the result and the code and results are shown as below:

```
> shapiro.test(residuals(odtlm))

        Shapiro-Wilk normality test

data:  residuals(odtlm)
W = 0.96591, p-value = 0.4342
```

Figure 58: Test result

From the QQ-plot we could know that residuals are conform to normal population and for the linear regression model. Therefore, the chosen linear model is appropriate.

Exercise 7

We use the step-down to calculate the linear regression model. The first step and code is as shown in Fig.59.

```
> crime=read.table("expensescrime.txt",header=TRUE)
> summary(lm(expend~bad+crime+lawyers+employ+pop,data=crime))

Call:
lm(formula = expend ~ bad + crime + lawyers + employ + pop, data = crime)

Residuals:
    Min      1Q  Median      3Q     Max
-638.41  -87.42   22.15  114.96  804.98

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -2.991e+02  1.401e+02  -2.136  0.03817 *
bad         -2.832e+00  1.240e+00  -2.283  0.02719 *
crime        3.241e-02  2.813e-02   1.152  0.25534
lawyers      2.324e-02  8.044e-03   2.890  0.00592 **
employ       2.297e-02  7.462e-03   3.078  0.00354 **
pop          7.787e-02  3.515e-02   2.215  0.03184 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 225.6 on 45 degrees of freedom
Multiple R-squared:  0.9675,    Adjusted R-squared:  0.9639
F-statistic: 268.2 on 5 and 45 DF,  p-value: < 2.2e-16
```

Figure 59: First step of step-down method

We could know that the p-value of crime is 0.25534 which is the highest and larger than 0.05, so we remove it. The code of next step is shown in Fig.60.

```
> summary(lm(expend~bad+lawyers+employ+pop,data=crime))

Call:
lm(formula = expend ~ bad + lawyers + employ + pop, data = crime)

Residuals:
    Min      1Q  Median      3Q     Max
-635.62  -80.18   18.77  114.54  809.66

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.464e+02  4.541e+01  -3.224  0.00232 **
bad         -2.241e+00  1.133e+00  -1.977  0.05402 .
lawyers      2.646e-02  7.571e-03   3.495  0.00106 **
employ       2.283e-02  7.487e-03   3.049  0.00380 **
pop          6.368e-02  3.304e-02   1.927  0.06012 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 226.4 on 46 degrees of freedom
Multiple R-squared:  0.9666,    Adjusted R-squared:  0.9637
F-statistic: 332.5 on 4 and 46 DF,  p-value: < 2.2e-16
```

Figure 60: Second step of step-down method

We could know that the p-value of pop is 0.06012 which is the highest and larger than 0.05, so we remove it. The code of next step is shown in Fig.61.

```
> summary(lm(expend~bad+lawyers+employ,data=crime))

Call:
lm(formula = expend ~ bad + lawyers + employ, data = crime)

Residuals:
    Min      1Q  Median      3Q     Max
-631.75  -93.69   30.34   89.68  963.67

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.106e+02  4.261e+01  -2.595  0.01257 *
bad         -8.627e-01  9.042e-01  -0.954  0.34496
lawyers      2.631e-02  7.786e-03   3.379  0.00147 **
employ       3.232e-02  5.803e-03   5.569  1.2e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 232.8 on 47 degrees of freedom
Multiple R-squared:  0.9639,    Adjusted R-squared:  0.9616
F-statistic:   418 on 3 and 47 DF,  p-value: < 2.2e-16
```

Figure 61: Third step of step-down method

We could know that the p-value of bad is 0.34496 which is the highest and larger than 0.05, so we remove it. The code of next step is shown in Fig.62.

```
> summary(lm(expend~lawyers+employ,data=crime))

Call:
lm(formula = expend ~ lawyers + employ, data = crime)

Residuals:
    Min      1Q  Median      3Q     Max
-599.47  -94.43   36.01   91.98  936.55

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.107e+02  4.257e+01  -2.600  0.01236 *
lawyers      2.686e-02  7.757e-03   3.463  0.00113 **
employ       2.971e-02  5.114e-03   5.810 4.89e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 232.6 on 48 degrees of freedom
Multiple R-squared:  0.9632,    Adjusted R-squared:  0.9616
F-statistic: 627.7 on 2 and 48 DF,  p-value: < 2.2e-16
```

Figure 62: Fourth step of step-down method

We could know that all the p value is less than 0.05, so the resulting model of the step-down method is: expend=-1.107e+02 + 2.686e-02*lawyers + 2.971e-02*employ + error

This is an **initial** result, after analyzing in different aspects, such as influence points, collinearity and residuals, the result could be modified.

a. For the potential or influence points, according to the definition: A potential point (or leverage point) is an observation with an outlying value in an explanatory variable Xi. So we first plot the scatter plot for each lawyers and employ, where expend is response variable. The code is shown in Fig.63 and the result is as shown in Fig.64.

```
lawyerslm=lm(expend~lawyers,data=crime)
employlm=lm(expend~employ,data=crime)
par(mfrow=c(2,3))
plot(crime$expend~crime$lawyers)
plot(residuals(lawyerslm)~crime$lawyers)
plot(residuals(lawyerslm)~fitted(lawyerslm))
plot(crime$expend~crime$employ)
plot(residuals(employlm)~crime$employ)
plot(residuals(employlm)~fitted(employlm))
```

Figure 63: Code for scatter plot

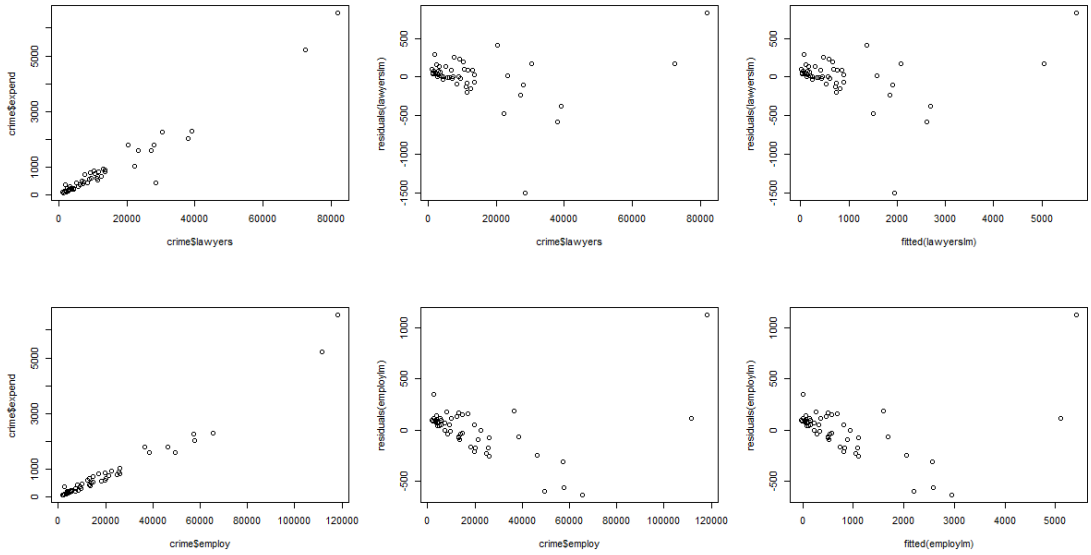

Figure 64: Scatter plot for lawyers and employ

We could in the linear model there is some potential points here and the linear regression in the first graph of two explanatories perform well. So we perform Cook's distance to check the influence points for lawyers and employ. The code and results are shown in Fig.65.

```
> lawyerslm=lm(expend~lawyers,data=crime)
> round(cooks.distance(lawyerslm),2)
   1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17
0.01 0.00 0.00 0.01 3.69 0.00 0.00 0.51 0.00 0.01 0.00 0.00 0.00 0.00 0.14 0.00 0.00
  18   19   20   21   22   23   24   25   26   27   28   29   30   31   32   33   34
0.00 0.00 0.03 0.00 0.00 0.02 0.00 0.00 0.00 0.00 0.01 0.00 0.00 0.00 0.00 0.00 0.00
  35   36   37   38   39   40   41   42   43   44   45   46   47   48   49   50   51
0.09 0.01 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.06 0.00 0.00 0.00 0.00 0.00 0.00 0.00
> employlm=lm(expend~employ,data=crime)
> round(cooks.distance(employlm),2)
   1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17
0.03 0.00 0.00 0.00 7.00 0.00 0.00 0.01 0.00 0.05 0.01 0.00 0.00 0.00 0.18 0.01 0.00
  18   19   20   21   22   23   24   25   26   27   28   29   30   31   32   33   34
0.00 0.00 0.00 0.00 0.00 0.01 0.00 0.00 0.00 0.00 0.01 0.00 0.00 0.00 0.14 0.00 0.00
  35   36   37   38   39   40   41   42   43   44   45   46   47   48   49   50   51
0.06 0.00 0.00 0.00 0.02 0.00 0.00 0.00 0.00 0.31 0.00 0.00 0.00 0.00 0.00 0.00 0.00
```

Figure 65: Code and result for Cook's distances

From the output we could know that both the 5th point in lawyers and expend is larger than 1, so there is influence points in this model. The code is as shown in Fig.66 and plots are as shown in Fig.67.

```
par(mfrow=c(1,2))
plot(1:51,cooks.distance(lawyerslm))
plot(1:51,cooks.distance(employlm))
```

Figure 66: Scatter plot code for Cook's distances

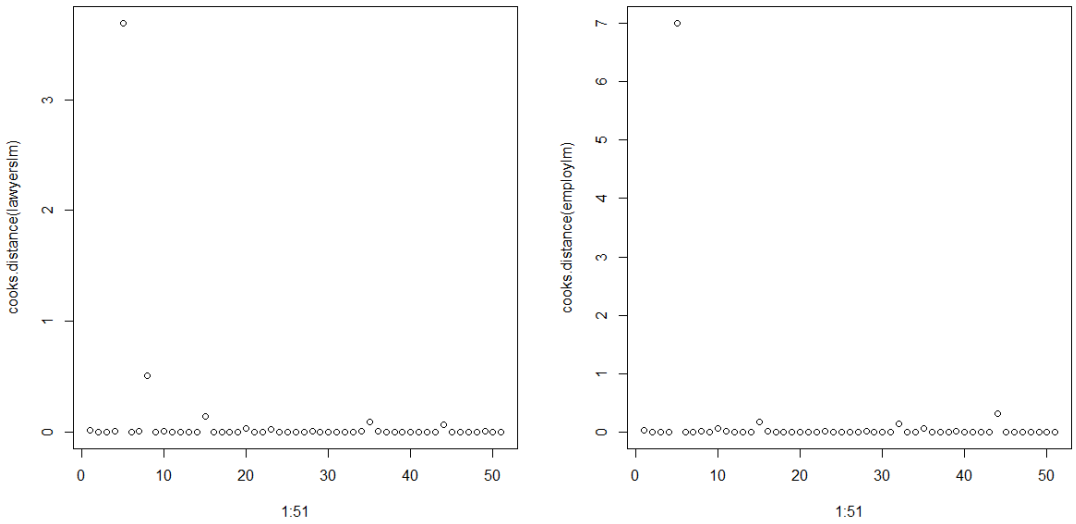

Figure 67: Scatter plot for Cook's distances

Because of some duplicated work, we pause our influence points check work there. After checking collinearity, we will continue the influence points check work then.

b. For the problems due to collinearity, we first draw the scatter plot of Xj against Xk for all combinations j, k, the plot is as shown in Fig.68.

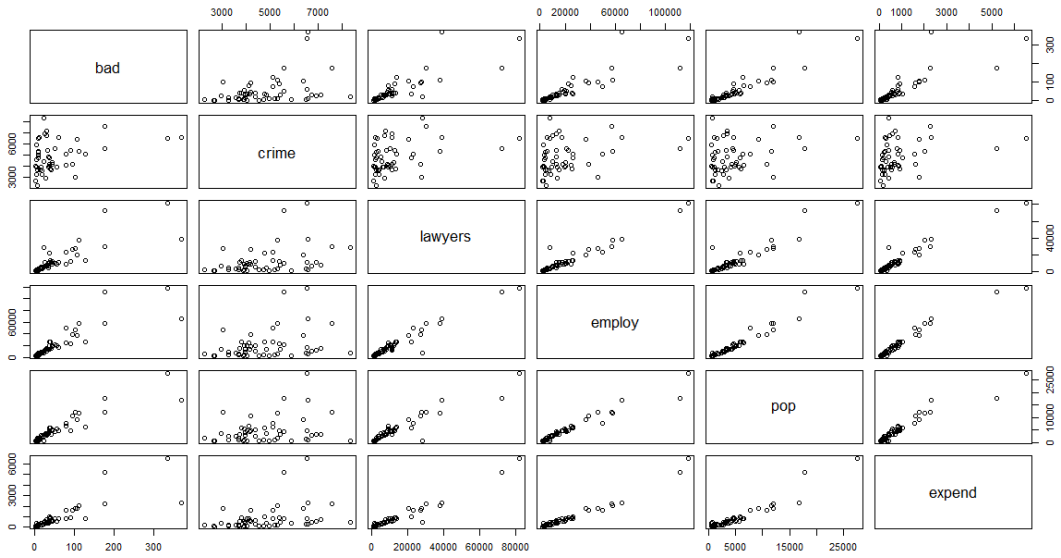

Figure 68: Scatter plot for each variables

From the plot we could know that there may be collinearity in bad and lawyers, bad and employ, bad and population, lawyers and employ, lawyers and pop, employ and pop.

We compute the pairewise correlations of the crime data. The code and the result are as shown in Fig.69.

```
> round(cor(crime[,3:7]),2)
         bad crime lawyers employ  pop
bad     1.00  0.37    0.83   0.87 0.92
crime   0.37  1.00    0.38   0.31 0.28
lawyers 0.83  0.38    1.00   0.97 0.93
employ  0.87  0.31    0.97   1.00 0.97
pop     0.92  0.28    0.93   0.97 1.00
```

Figure 69: correlations of crime data

We could know that the correlation between bad and lawyers (0.83), bad and employ (0.87), bad and population (0.92), lawyers and employ (0.97), lawyers and pop (0.93), employ and pop (0.97) are very high. This is in agreement with the scatter plots.

Therefore, in order to avoiding having two collinear explanatory variables in the model, we should modify the model as shown in Fig.70.

```
> summary(lm(expend~employ,data=crime))

Call:
lm(formula = expend ~ employ, data = crime)

Residuals:
    Min      1Q  Median      3Q     Max
-636.04  -84.35   47.60  107.99 1124.70

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.167e+02  4.706e+01   -2.48   0.0166 *
employ       4.681e-02  1.469e-03   31.87   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 257.4 on 49 degrees of freedom
Multiple R-squared:  0.954,     Adjusted R-squared:  0.953
F-statistic:  1016 on 1 and 49 DF,  p-value: < 2.2e-16
```

Figure 70: Modified result

The R-squared does not change too much, only slightly lower (from 0.9632 to 0.954) and the collinearity is eliminated. The result is as follows:

expend=-1.167e+02 + 4.681e-02*employ + error

a. Influence points check **continue**:

From the updated results above, we should only check the influence points of employ now. Because the 5th point is the influence point, we should remove this point and re-do the Cook's distance calculation. The code and result is shown in Fig.71.

```
> crimenew=crime[-5,]
> employlm1=lm(expend~employ,data=crimenew)
> round(cooks.distance(employlm1),2)
   1    2    3    4    6    7    8    9   10   11   12   13   14   15   16   17   18
0.05 0.00 0.00 0.01 0.01 0.00 0.01 0.00 0.01 0.02 0.00 0.00 0.00 0.22 0.01 0.00 0.00
  19   20   21   22   23   24   25   26   27   28   29   30   31   32   33   34   35
0.00 0.00 0.00 0.00 0.06 0.01 0.01 0.00 0.00 0.01 0.00 0.00 0.00 0.23 0.00 0.00 9.43
  36   37   38   39   40   41   42   43   44   45   46   47   48   49   50   51
0.00 0.00 0.00 0.01 0.00 0.00 0.00 0.01 0.38 0.00 0.01 0.00 0.01 0.00 0.00 0.00
```

Figure 71: Code and result for Cook's distances2

After deleting 5th point we could know the 35th point of the origin dataset is still the influence

point. We draw the scatter plot as follow:
plot(1:50,cooks.distance(employlm1))

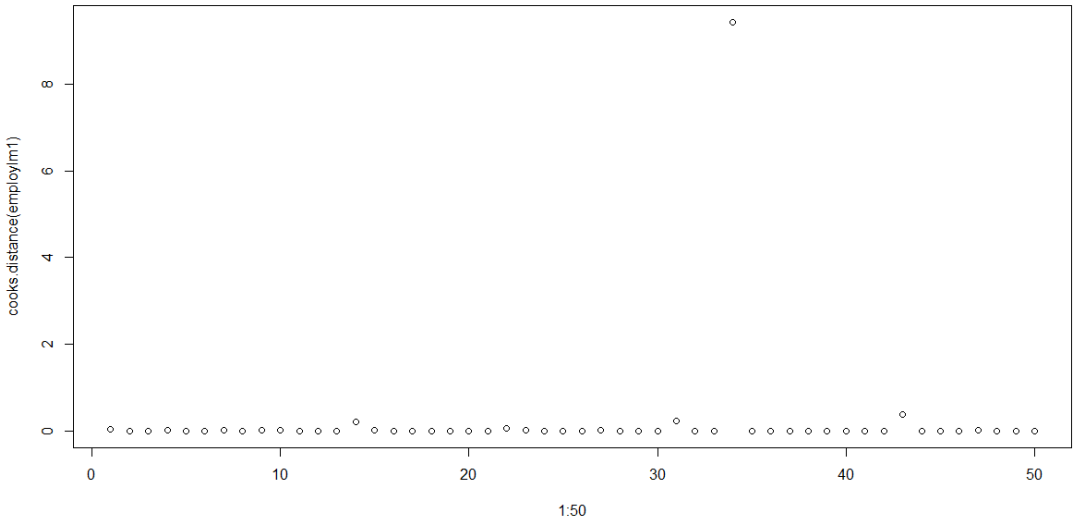

Figure 72: Scatter plot for Cook's distances 2

So we should remove 34th point and re-do the Cook's distance calculation. The code and result is shown in Fig.73.

```
> crimenew1=crime[-c(5,35),]
> employlm2=lm(expend~employ,data=crimenew1)
> round(cooks.distance(employlm2),2)
   1    2    3    4    6    7    8    9   10   11   12   13   14   15   16   17   18
0.07 0.00 0.01 0.02 0.01 0.00 0.01 0.00 0.13 0.02 0.00 0.00 0.00 0.11 0.02 0.00 0.01
  19   20   21   22   23   24   25   26   27   28   29   30   31   32   33   34   36
0.00 0.00 0.01 0.00 0.37 0.02 0.01 0.01 0.00 0.01 0.00 0.00 0.00 0.28 0.00 0.00 0.08
  37   38   39   40   41   42   43   44   45   46   47   48   49   50   51
0.01 0.00 0.02 0.00 0.01 0.00 0.01 0.15 0.00 0.00 0.00 0.03 0.01 0.00 0.00
```

Figure 73: Code and result for Cook's distances3

We could know there is no influence points now. There are 2 influence points in the origin data, which is 5th and 35th. We draw the scatter plot as follow:
plot(1:49,cooks.distance(employlm2))

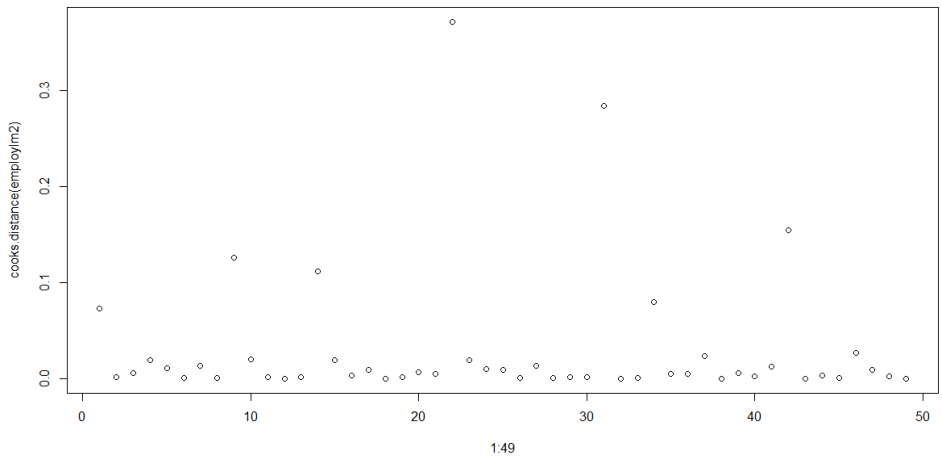

Figure 74: Scatter plot for Cook's distances 3

c. For the residuals, we use graphic checks to check the residuals.

From b: investigation of problems due to collinearity we get Scatter plot of Y against each Xk separately.

We then get the Scatter plot of residuals against each Xk in the model separately. The code is shown in Fig.75 and results is shown in Fig.76.

```
employlm=lm(expend~employ,data=crime)
plot(residuals(employlm),crime$employ)
```

Figure 75: Code for residuals check

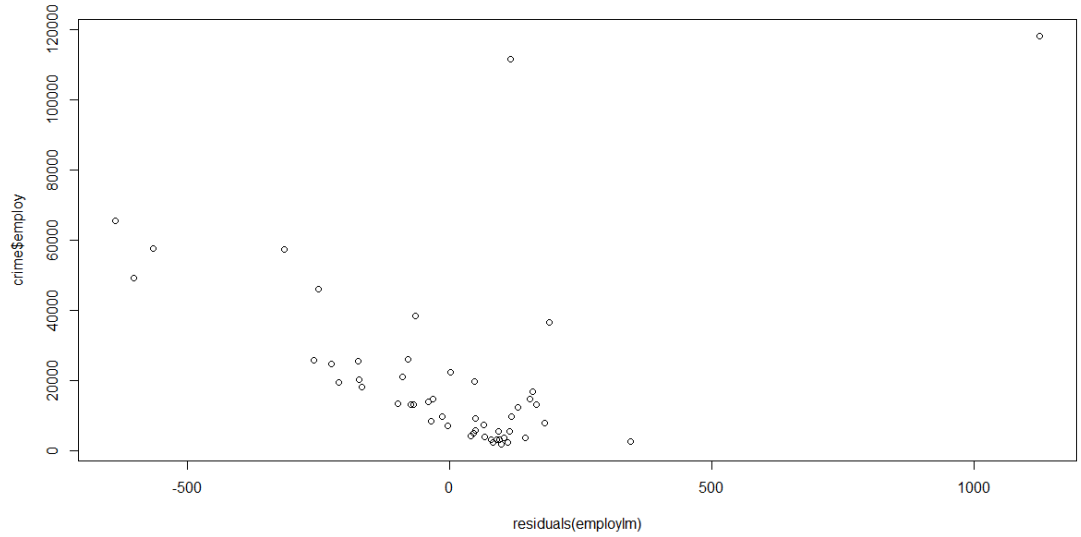

Figure 76: Scatter plot for the residuals(employlm) and employ

Then scatter plot of residuals against each Xk not in the model separately. Because employ and bad, lawyers, pop are collinearity, we do not need this step to add these variables, so the variables we need to check is crime.

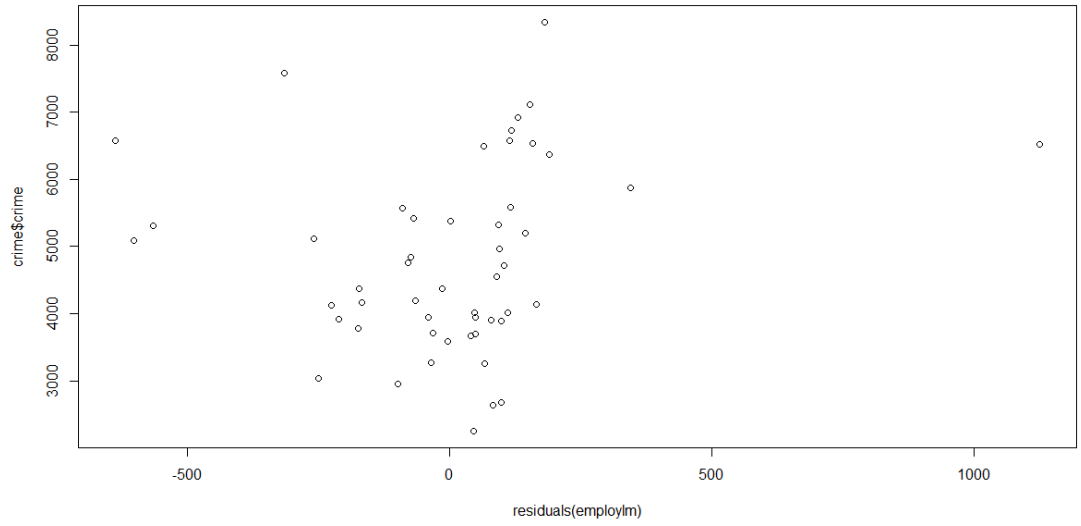

Figure 77: Scatter plot for the residuals(employlm) and crime

The result is not liner so we do not need to add crime.

Then we do Scatter plot of residuals against Y (and ˆY). The code is as shown in Fig.78 and the result are shown in Fig.79 and Fig.80.

```
plot(residuals(employlm),crime$expend)
plot(residuals(employlm),fitted(employlm))
```

Figure 78: Code for scatter plot of residuals against Y (and ˆY).

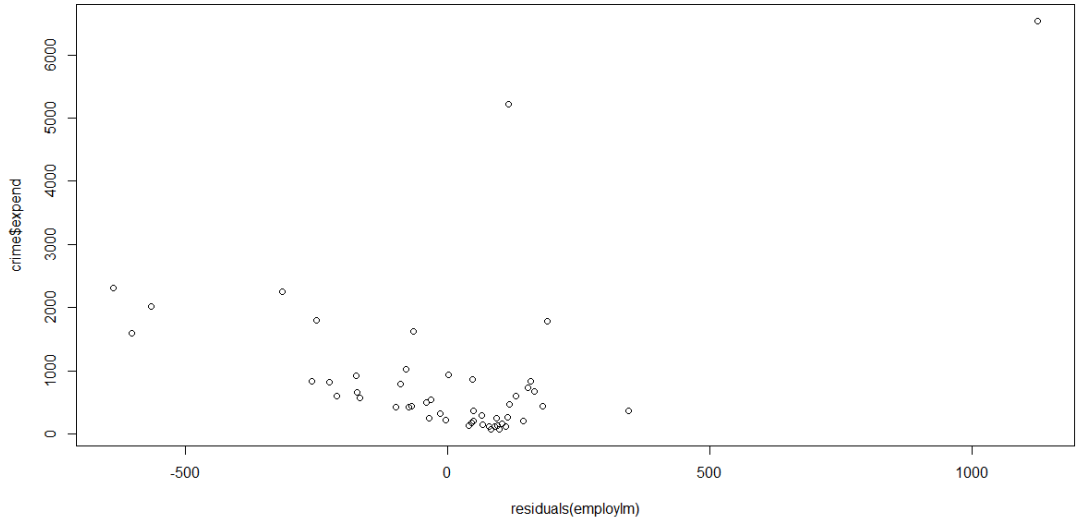

Figure 79: Scatter plot for the residuals(employlm) and expend

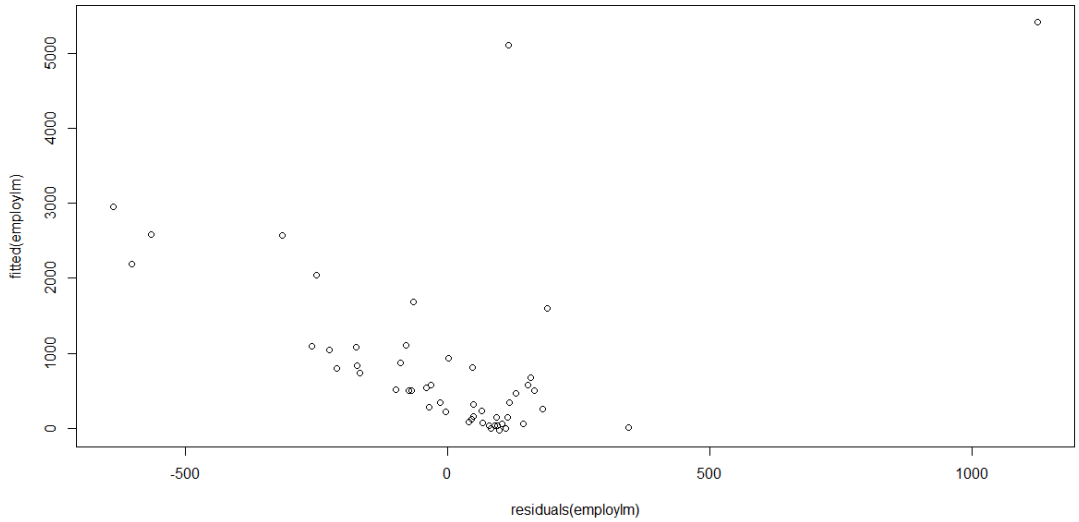

Figure 80: Scatter plot for the residuals(employlm) and fitted(employlm)

Final, we do the Normal QQ-plot of the residuals to check the normality assumption. The code is:

qqnorm(residuals(employlm))

and the result is as shown in Fig.81.

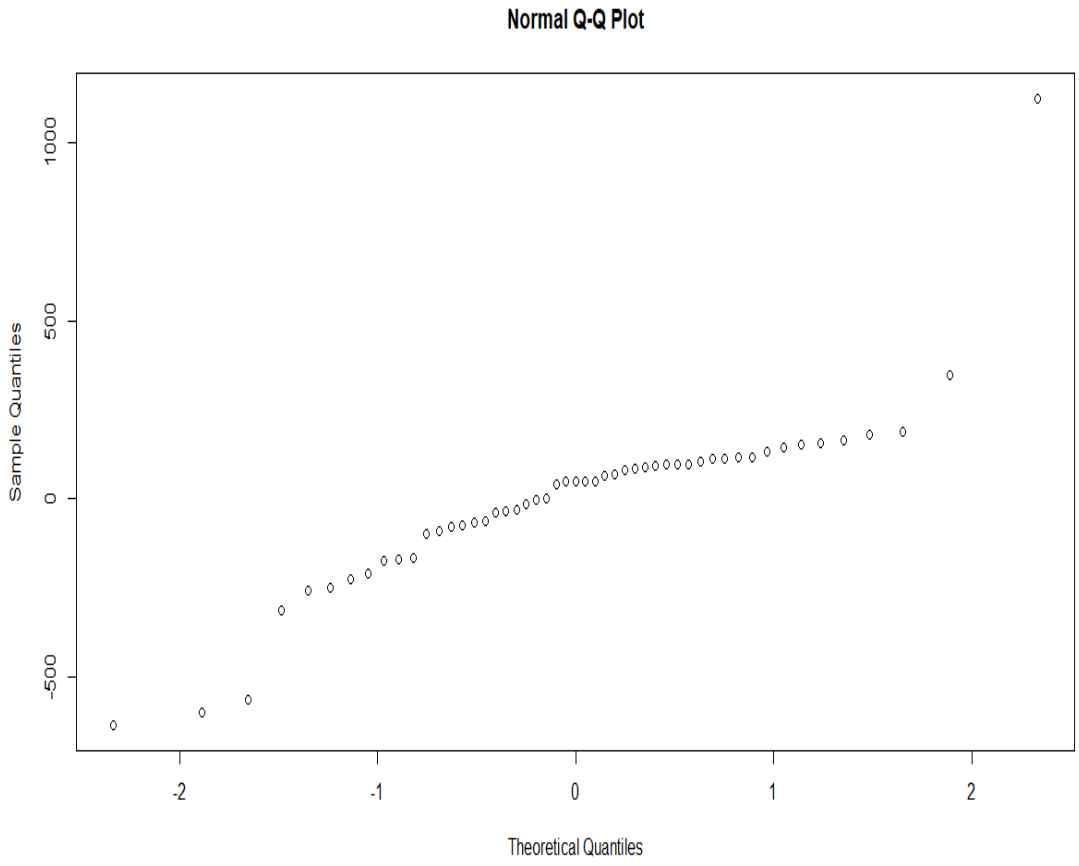

Figure 81: QQ-plot of the residuals

We do the shapiro test to check the result and the code and results are shown as below:

```
> shapiro.test(residuals(employlm))

        Shapiro-Wilk normality test

data:  residuals(employlm)
W = 0.81947, p-value = 2.111e-06
```

Figure 82: Test results

From the QQ-plot we could know that residuals are not taken from a normal population. Therefore, the chosen linear model is inappropriate, there should be some further improvements on this model.