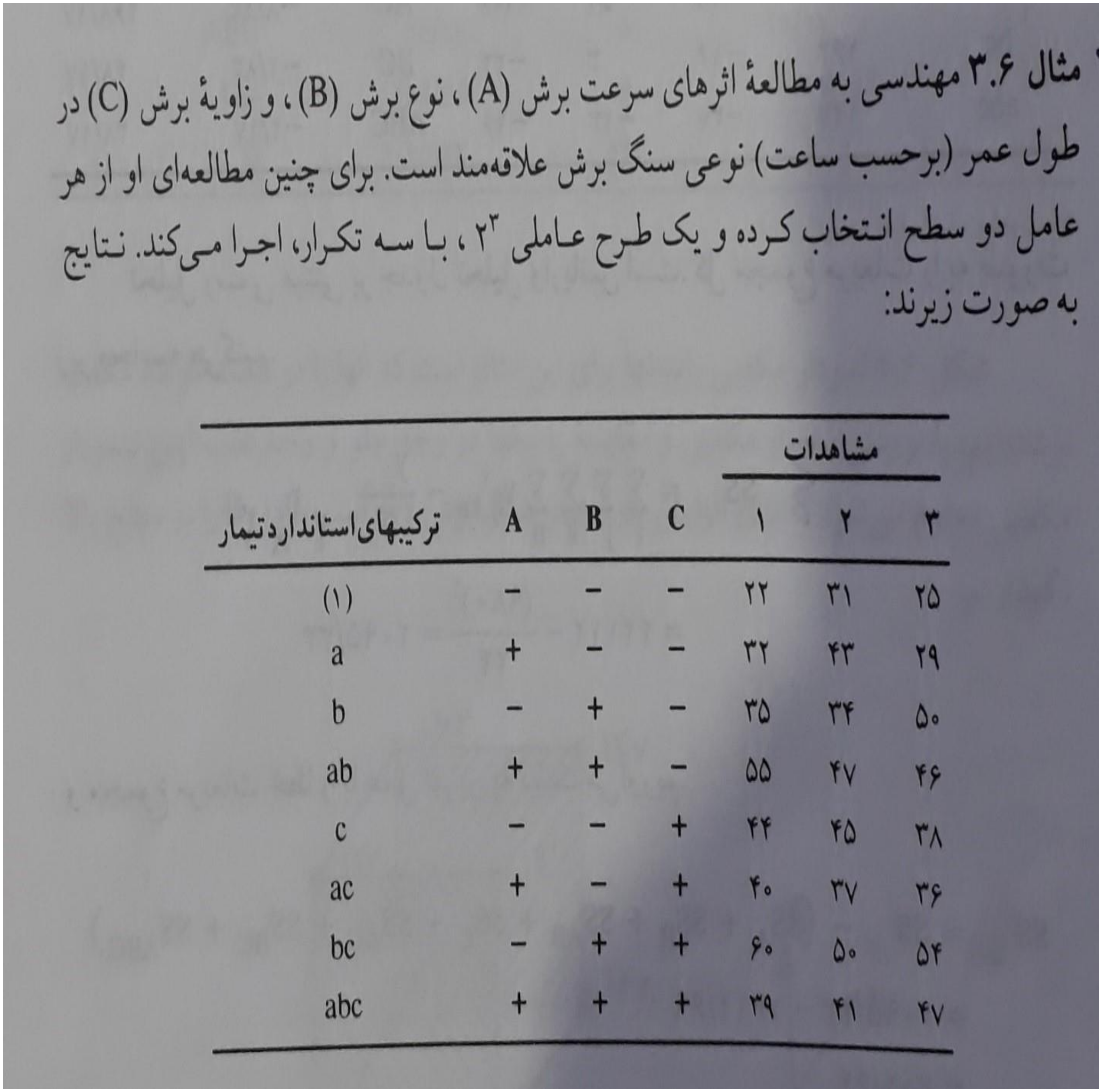
# Project

Mehrab Atighi

1/7/2022

Now we want to solve this question with R:



**مثال ۳.۶** مهندسی به مطالعهٔ اثرهای سرعت برش (A)، نوع برش (B)، و زاویهٔ برش (C) در طول عمر (برحسب ساعت) نوعی سنگ برش علاقه‌مند است. برای چنین مطالعه‌ای او از هر عامل دو سطح انتخاب کرده و یک طرح عاملی $۲^{۳}$، با سه تکرار، اجرا می‌کند. نتایج به صورت زیرند:

| | | | | مشاهدات | | |
|---|---|---|---|---|---|---|
| ترکیبهای استاندارد تیمار | A | B | C | ۱ | ۲ | ۳ |
| (۱) | − | − | − | ۲۲ | ۳۱ | ۲۵ |
| a | + | − | − | ۳۲ | ۴۳ | ۲۹ |
| b | − | + | − | ۳۵ | ۳۴ | ۵۰ |
| ab | + | + | − | ۵۵ | ۴۷ | ۴۶ |
| c | − | − | + | ۴۴ | ۴۵ | ۳۸ |
| ac | + | − | + | ۴۰ | ۳۷ | ۳۶ |
| bc | − | + | + | ۶۰ | ۵۰ | ۵۴ |
| abc | + | + | + | ۳۹ | ۴۱ | ۴۷ |

```
# the first question for lesson project :
response<-c(22,31,25,
            32,43,29,
            35,34,50,
            55,47,46,
            44,45,38,
            40,37,36,
            60,50,54,
            39,41,47)
A<-factor(c(rep(c(-1,1),each=3,4)))
B<-factor(c(rep(c(-1,1),each=6 ,2)))
C<-factor(c(rep(c(-1,1),each=12)))
data=data.frame(response,A,B,C)
data
```

```
##    response  A  B  C
## 1        22 -1 -1 -1
## 2        31 -1 -1 -1
## 3        25 -1 -1 -1
## 4        32  1 -1 -1
## 5        43  1 -1 -1
## 6        29  1 -1 -1
## 7        35 -1  1 -1
## 8        34 -1  1 -1
## 9        50 -1  1 -1
## 10       55  1  1 -1
## 11       47  1  1 -1
## 12       46  1  1 -1
## 13       44 -1 -1  1
## 14       45 -1 -1  1
## 15       38 -1 -1  1
## 16       40  1 -1  1
## 17       37  1 -1  1
## 18       36  1 -1  1
## 19       60 -1  1  1
## 20       50 -1  1  1
## 21       54 -1  1  1
## 22       39  1  1  1
## 23       41  1  1  1
## 24       47  1  1  1
```

```
fit1<-lm(response~A*B*C , data=data)
```

```
#anova table**********************************

anova(fit1)

## Analysis of Variance Table
##
## Response: response
##           Df Sum Sq Mean Sq F value    Pr(>F)    
## A          1   0.67    0.67  0.0221 0.8836803    
## B          1 770.67  770.67 25.5470 0.0001173 ***
## C          1 280.17  280.17  9.2873 0.0076787 ** 
## A:B        1  16.67   16.67  0.5525 0.4680784    
## A:C        1 468.17  468.17 15.5193 0.0011722 ** 
## B:C        1  48.17   48.17  1.5967 0.2244753    
## A:B:C      1  28.17   28.17  0.9337 0.3482825    
## Residuals 16 482.67   30.17                      
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

#test significant model**********************

(Fisher=summary(fit1)$fstatistic[[1]])

## [1] 7.636938

(df1=summary(fit1)$fstatistic[[2]])

## [1] 7

(df2=summary(fit1)$fstatistic[[3]])

## [1] 16

alpha=0.05

(cp=qf(1-alpha,df1,df2))

## [1] 2.657197

if(Fisher>cp)print("model is significant") else print("model is  not signific
ant")

## [1] "model is significant"

#test significant model with p-value
(pvalue=1-pf(Fisher,df1,df2))

## [1] 0.0003976576

if(pvalue<alpha)print("model is significant") else print("model is  not signi
ficant")

## [1] "model is significant"
```

```
#***************************************************************

(R2=summary(fit1)$r.squared)

## [1] 0.7696468

(SSR=sum(anova(fit1)$S[1:7]))

## [1] 1612.667

(SSE=(anova(fit1)$S[8]))

## [1] 482.6667

(SST=SSR+SSE)

## [1] 2095.333

(SSR/SST)

## [1] 0.7696468

(R2adj=summary(fit1)$adj.r.squared)

## [1] 0.6688673

#mean data for each treatment*************************


aggregate(response ~ A + B +C ,FUN = mean, data = data)

##    A  B  C response
## 1 -1 -1 -1 26.00000
## 2  1 -1 -1 34.66667
## 3 -1  1 -1 39.66667
## 4  1  1 -1 49.33333
## 5 -1 -1  1 42.33333
## 6  1 -1  1 37.66667
## 7 -1  1  1 54.66667
## 8  1  1  1 42.33333
```

```
#Simplify the model by excluding nonsignificant effects
fit2<-lm(response~B+C+(A:C),data=data)
anova(fit2)
```

```
## Analysis of Variance Table
##
## Response: response
##           Df Sum Sq Mean Sq F value    Pr(>F)
## B          1 770.67  770.67  25.436 7.216e-05 ***
## C          1 280.17  280.17   9.247  0.006724 **
## C:A        2 468.83  234.42   7.737  0.003483 **
## Residuals 19 575.67   30.30
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#fitted.values=yhat*************************
(yhat=fit2$ fitted.values)
```

```
##        1        2        3        4        5        6        7        8
## 27.16667 27.16667 27.16667 36.33333 36.33333 36.33333 38.50000 38.50000
##        9       10       11       12       13       14       15       16
## 38.50000 47.66667 47.66667 47.66667 42.83333 42.83333 42.83333 34.33333
##       17       18       19       20       21       22       23       24
## 34.33333 34.33333 54.16667 54.16667 54.16667 45.66667 45.66667 45.66667
```

```
(yhat=predict(fit2))
```

```
##        1        2        3        4        5        6        7        8
## 27.16667 27.16667 27.16667 36.33333 36.33333 36.33333 38.50000 38.50000
##        9       10       11       12       13       14       15       16
## 38.50000 47.66667 47.66667 47.66667 42.83333 42.83333 42.83333 34.33333
##       17       18       19       20       21       22       23       24
## 34.33333 34.33333 54.16667 54.16667 54.16667 45.66667 45.66667 45.66667
```

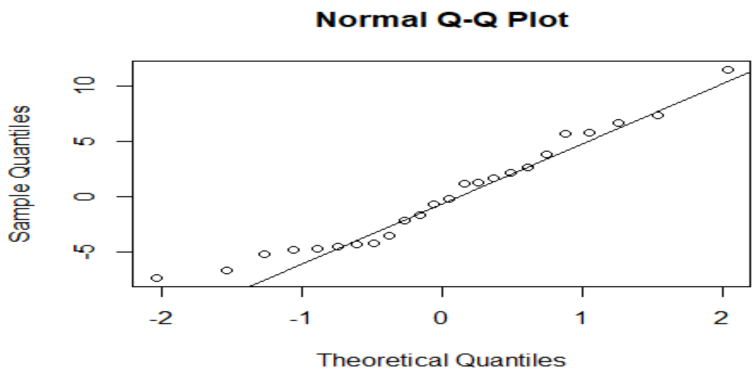
```
#residuals=e********************************
(residual=fit2$ residuals)
```

```
##          1          2          3          4          5          6
7
## -5.1666667  3.8333333 -2.1666667 -4.3333333  6.6666667 -7.3333333 -3.50000
00
##          8          9         10         11         12         13
14
## -4.5000000 11.5000000  7.3333333 -0.6666667 -1.6666667  1.1666667  2.16666
67
##         15         16         17         18         19         20
21
## -4.8333333  5.6666667  2.6666667  1.6666667  5.8333333 -4.1666667 -0.16666
67
##         22         23         24
## -6.6666667 -4.6666667  1.3333333
```

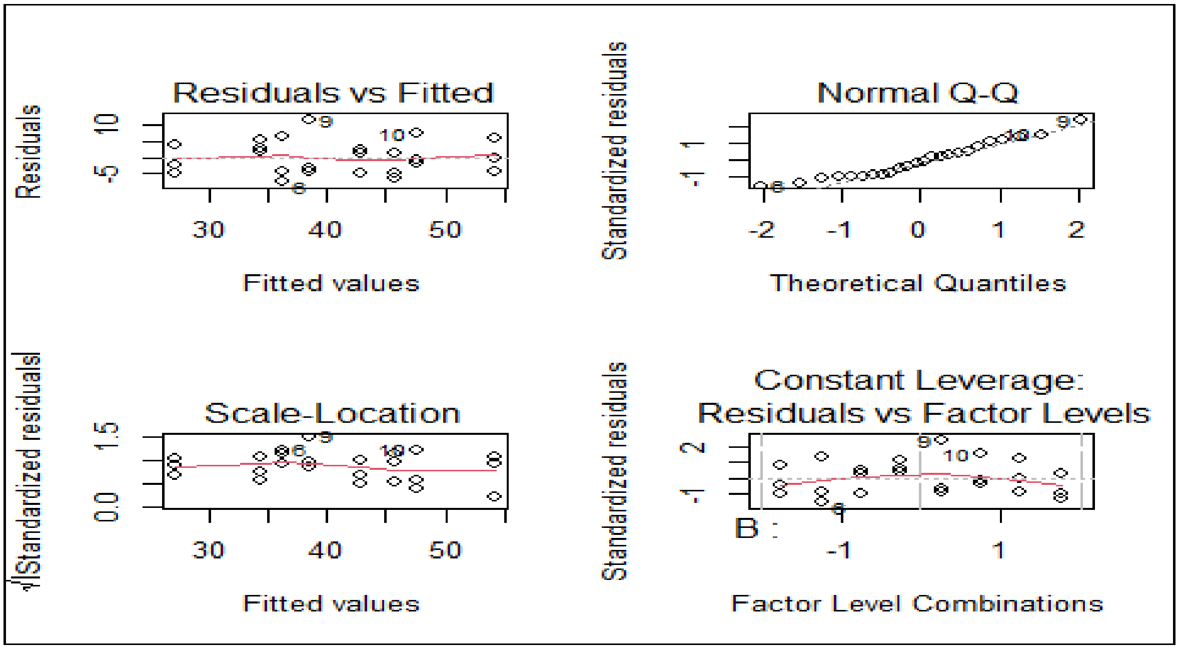
```
cbind(response - yhat,residual)
```

```
##                  residual
## 1  -5.1666667 -5.1666667
## 2   3.8333333  3.8333333
## 3  -2.1666667 -2.1666667
## 4  -4.3333333 -4.3333333
## 5   6.6666667  6.6666667
## 6  -7.3333333 -7.3333333
## 7  -3.5000000 -3.5000000
## 8  -4.5000000 -4.5000000
## 9  11.5000000 11.5000000
## 10  7.3333333  7.3333333
## 11 -0.6666667 -0.6666667
## 12 -1.6666667 -1.6666667
## 13  1.1666667  1.1666667
## 14  2.1666667  2.1666667
## 15 -4.8333333 -4.8333333
## 16  5.6666667  5.6666667
## 17  2.6666667  2.6666667
## 18  1.6666667  1.6666667
## 19  5.8333333  5.8333333
## 20 -4.1666667 -4.1666667
## 21 -0.1666667 -0.1666667
## 22 -6.6666667 -6.6666667
## 23 -4.6666667 -4.6666667
## 24  1.3333333  1.3333333
```

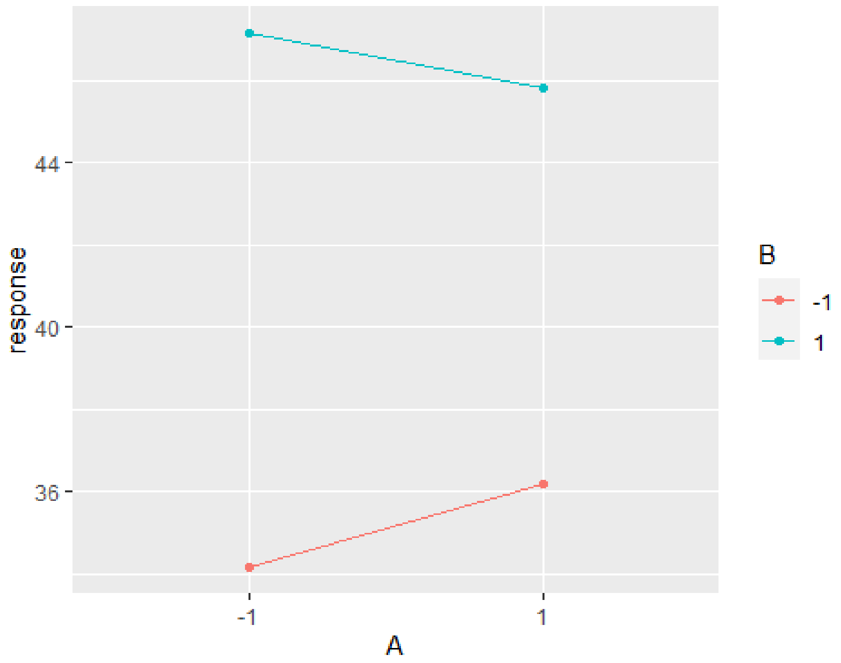
```
#normality of residuals**********************

#qunatile-quantile plot


qqnorm(residual)


qqline(residual)
```



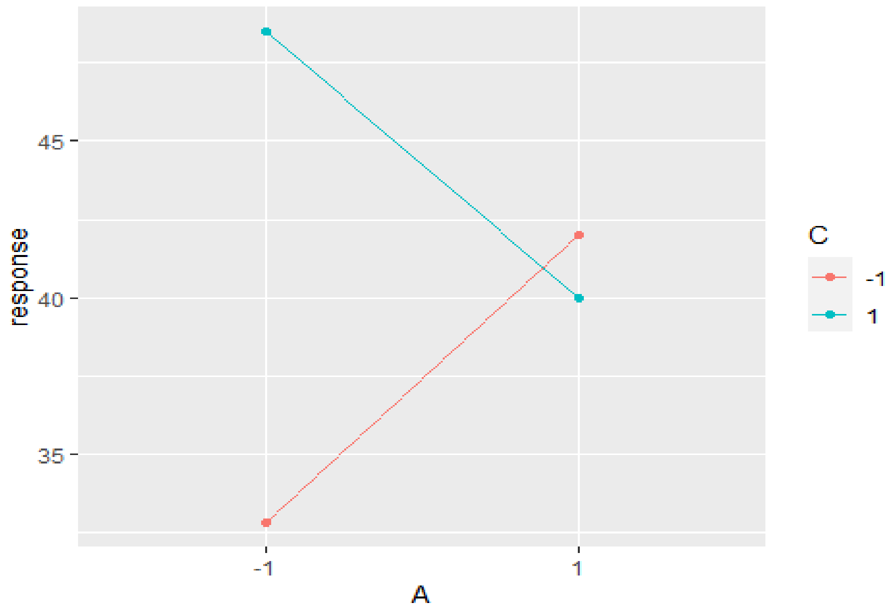
```
#Residuals plots standard chart
par(mfrow=c(2,2))
plot(fit2)
box("outer")
#cheking normality of residuals with shapiro test:
shapiro.test(residual)
##  Shapiro-Wilk normality test
##
## data:  residual
## W = 0.953, p-value = 0.3143

str(shapiro.test(residual))

## List of 4
##  $ statistic: Named num 0.953
##   ..- attr(*, "names")= chr "W"
##  $ p.value  : num 0.314
##  $ method   : chr "Shapiro-Wilk normality test"
##  $ data.name: chr "residual"
##  - attr(*, "class")= chr "htest"

(pvlaue=shapiro.test(residual)$p.value)

## [1] 0.3142911

if(pvalue<alpha)print("residuals have normal distribution") else print("resid
uals have normal distribution")

## [1] "residuals have normal distribution"

#effects interaction of AB*********************************
library(ggplot2)
```



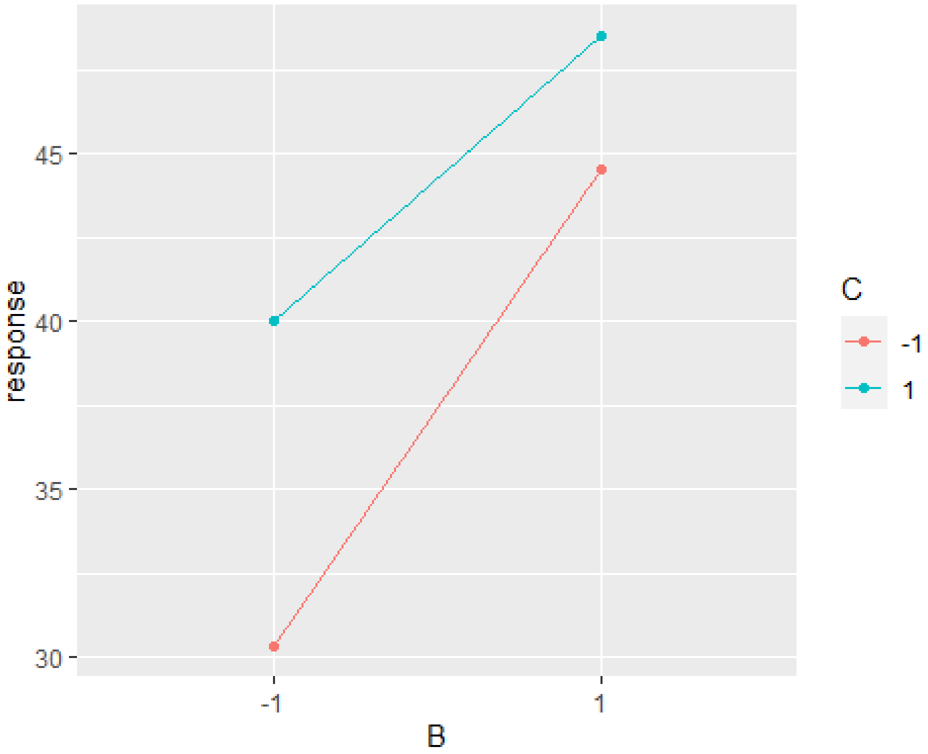
```
intEf <- aggregate(response ~  A +B,FUN = mean, data = data)
effects_interaction <- ggplot(intEf, aes(x = A, y = response, color = B)) +
    geom_point() + geom_line(aes(group = B))
effects_interaction
```



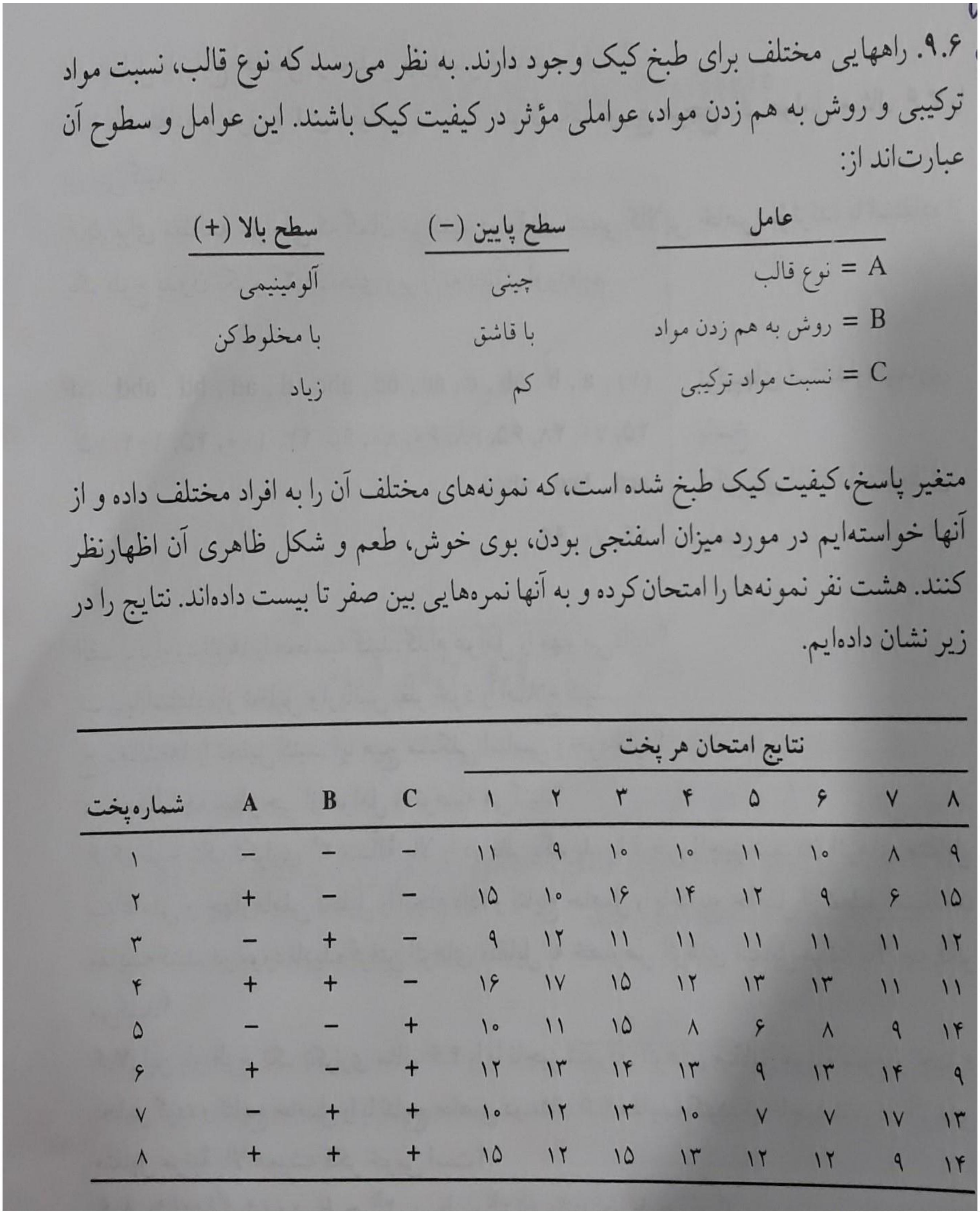
```
#***effects interaction of AC************************************************
******
intEf <- aggregate(response ~ A + C,FUN = mean, data = data)
effects_interaction <- ggplot(intEf, aes(x = A, y = response, color = C)) +
   geom_point() + geom_line(aes(group = C))
effects_interaction
```



```
#***effects interaction of BC************************************************
******


intEf <- aggregate(response ~ B + C,FUN = mean, data = data)


effects_interaction <- ggplot(intEf, aes(x = B, y = response, color = C)) +

  geom_point() + geom_line(aes(group = C))


effects_interaction
```



Now we want to solve this question with R:



۹.۶. راههایی مختلف برای طبخ کیک وجود دارند. به نظر می‌رسد که نوع قالب، نسبت مواد ترکیبی و روش به هم زدن مواد، عواملی مؤثر در کیفیت کیک باشند. این عوامل و سطوح آن عبارت‌اند از:

| عامل | سطح پایین (−) | سطح بالا (+) |
|---|---|---|
| A = نوع قالب | چینی | آلومینیمی |
| B = روش به هم زدن مواد | با قاشق | با مخلوط کن |
| C = نسبت مواد ترکیبی | کم | زیاد |

متغیر پاسخ، کیفیت کیک طبخ شده است، که نمونه‌های مختلف آن را به افراد مختلف داده و از آنها خواسته‌ایم در مورد میزان اسفنجی بودن، بوی خوش، طعم و شکل ظاهری آن اظهارنظر کنند. هشت نفر نمونه‌ها را امتحان کرده و به آنها نمره‌هایی بین صفر تا بیست داده‌اند. نتایج را در زیر نشان داده‌ایم.

| | | | | نتایج امتحان هر پخت | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| شماره پخت | A | B | C | ۱ | ۲ | ۳ | ۴ | ۵ | ۶ | ۷ | ۸ |
| ۱ | − | − | − | ۱۱ | ۹ | ۱۰ | ۱۰ | ۱۱ | ۱۰ | ۸ | ۹ |
| ۲ | + | − | − | ۱۵ | ۱۰ | ۱۶ | ۱۴ | ۱۲ | ۹ | ۶ | ۱۵ |
| ۳ | − | + | − | ۹ | ۱۲ | ۱۱ | ۱۱ | ۱۱ | ۱۱ | ۱۱ | ۱۲ |
| ۴ | + | + | − | ۱۶ | ۱۷ | ۱۵ | ۱۲ | ۱۳ | ۱۳ | ۱۱ | ۱۱ |
| ۵ | − | − | + | ۱۰ | ۱۱ | ۱۵ | ۸ | ۶ | ۸ | ۹ | ۱۴ |
| ۶ | + | − | + | ۱۲ | ۱۳ | ۱۴ | ۱۳ | ۹ | ۱۳ | ۱۴ | ۹ |
| ۷ | − | + | + | ۱۰ | ۱۲ | ۱۳ | ۱۰ | ۷ | ۷ | ۱۷ | ۱۳ |
| ۸ | + | + | + | ۱۵ | ۱۲ | ۱۵ | ۱۳ | ۱۲ | ۱۲ | ۹ | ۱۴ |

```
#the second question for lesson project:
response <- c(11,9,10,10,11,10,8,9,
              15,10,16,14,12,9,6,15,
              9,12,11,11,11,11,11,12,
              16,17,15,12,13,13,11,11,
              10,11,15,8,6,8,9,14,
              12,13,14,13,9,13,14,9,
              10,12,13,10,7,7,17,13,
              15,12,15,13,12,12,9,14)
A = factor(rep(rep(c(-1,+1) , each = 8),4))
B = factor(rep(rep(c(-1,-1,+1,+1), each = 8) ,2))
C = factor(rep(c(-1,+1),each = 32))

data2 = data.frame(Response = response , A = A , B = B , C = C)

fit3 = lm(response ~ A*B*C ,data = data2)

#anova table**********************************

anova(fit3)

## Analysis of Variance Table
##
## Response: response
##           Df Sum Sq Mean Sq F value   Pr(>F)
## A          1  72.25  72.250 11.9527 0.001049 **
## B          1  18.06  18.062  2.9882 0.089385 .
## C          1   0.06   0.062  0.0103 0.919370
## A:B        1   0.06   0.063  0.0103 0.919370
## A:C        1   1.56   1.563  0.2585 0.613154
## B:C        1   1.00   1.000  0.1654 0.685751
## A:B:C      1   0.25   0.250  0.0414 0.839584
## Residuals 56 338.50   6.045
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

#test significant model**********************


(Fisher=summary(fit3)$fstatistic[[1]])

## [1] 2.20384

(df1=summary(fit3)$fstatistic[[2]])

## [1] 7

(df2=summary(fit3)$fstatistic[[3]])

## [1] 56
```

```
alpha=0.05


(cp=qf(1-alpha,df1,df2))

## [1] 2.178156

if(Fisher>cp)print("model is significant") else print("model is  not signific
ant")

## [1] "model is significant"

#test significant model with p-value


(pvalue=1-pf(Fisher,df1,df2))

## [1] 0.04749937

if(pvalue<alpha)print("model is significant") else print("model is  not signi
ficant")

## [1] "model is significant"

#***********************************************************

(R2=summary(fit3)$r.squared)

## [1] 0.2159815

(SSR=sum(anova(fit3)$S[1:7]))

## [1] 93.25

(SSE=(anova(fit3)$S[8]))

## [1] 338.5

(SST=SSR+SSE)

## [1] 431.75

(SSR/SST)

## [1] 0.2159815

(R2adj=summary(fit3)$adj.r.squared)

## [1] 0.1179792
```

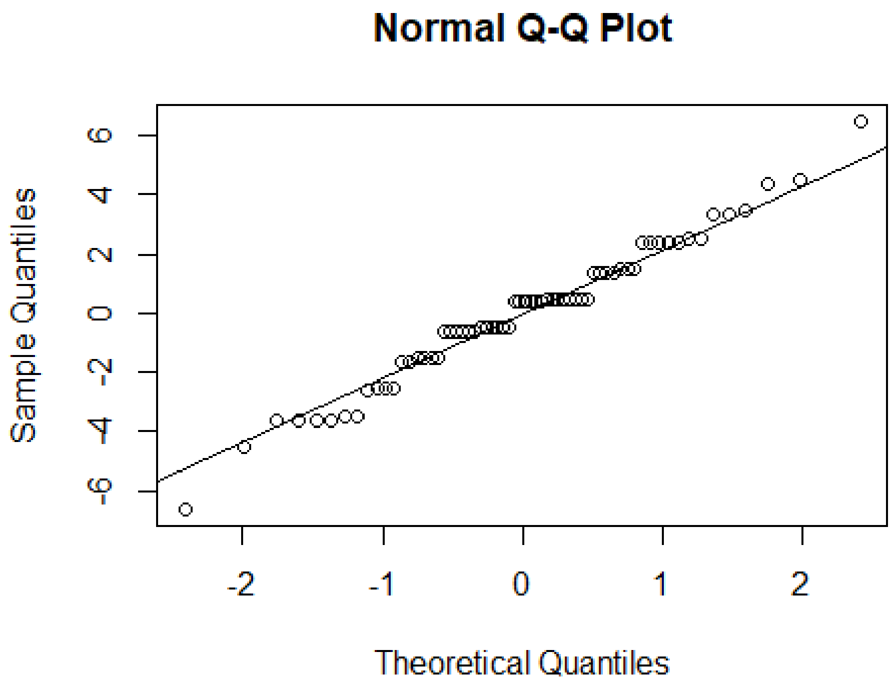
```
#mean data for each treatment*************************
aggregate(response ~ A + B +C ,FUN = mean, data = data2)

##    A  B  C response
## 1 -1 -1 -1    9.750
## 2  1 -1 -1   12.125
## 3 -1  1 -1   11.000
## 4  1  1 -1   13.500
## 5 -1 -1  1   10.125
## 6  1 -1  1   12.125
## 7 -1  1  1   11.125
## 8  1  1  1   12.750

#Simplify the model by excluding nonsignificant effects
fit4<-lm(response~A,data=data2)
anova(fit4)

## Analysis of Variance Table
##
## Response: response
##           Df Sum Sq Mean Sq F value    Pr(>F)
## A          1  72.25  72.250   12.46 0.0007901 ***
## Residuals 62 359.50   5.798
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

#fitted.values=yhat**************************
(yhat=fit4$ fitted.values)

##      1      2      3      4      5      6      7      8      9     10
11
## 10.500 10.500 10.500 10.500 10.500 10.500 10.500 10.500 12.625 12.625 12.6
25
##     12     13     14     15     16     17     18     19     20     21
22
## 12.625 12.625 12.625 12.625 12.625 10.500 10.500 10.500 10.500 10.500 10.5
00
##     23     24     25     26     27     28     29     30     31     32
33
## 10.500 10.500 12.625 12.625 12.625 12.625 12.625 12.625 12.625 12.625 10.5
00
##     34     35     36     37     38     39     40     41     42     43
44
## 10.500 10.500 10.500 10.500 10.500 10.500 10.500 12.625 12.625 12.625 12.6
25
##     45     46     47     48     49     50     51     52     53     54
55
## 12.625 12.625 12.625 12.625 10.500 10.500 10.500 10.500 10.500 10.500 10.5
00
##     56     57     58     59     60     61     62     63     64
## 10.500 12.625 12.625 12.625 12.625 12.625 12.625 12.625 12.625
```

```
(yhat=predict(fit4))

##      1      2      3      4      5      6      7      8      9     10     
11 
## 10.500 10.500 10.500 10.500 10.500 10.500 10.500 10.500 12.625 12.625 12.6
25 
##     12     13     14     15     16     17     18     19     20     21     
22 
## 12.625 12.625 12.625 12.625 12.625 10.500 10.500 10.500 10.500 10.500 10.5
00 
##     23     24     25     26     27     28     29     30     31     32     
33 
## 10.500 10.500 12.625 12.625 12.625 12.625 12.625 12.625 12.625 12.625 10.5
00 
##     34     35     36     37     38     39     40     41     42     43     
44 
## 10.500 10.500 10.500 10.500 10.500 10.500 10.500 12.625 12.625 12.625 12.6
25 
##     45     46     47     48     49     50     51     52     53     54     
55 
## 12.625 12.625 12.625 12.625 10.500 10.500 10.500 10.500 10.500 10.500 10.5
00 
##     56     57     58     59     60     61     62     63     64 
## 10.500 12.625 12.625 12.625 12.625 12.625 12.625 12.625 12.625

#residualse*********************************
(residual=fit4$ residuals)

##      1      2      3      4      5      6      7      8      9     10     
11 
##  0.500 -1.500 -0.500 -0.500  0.500 -0.500 -2.500 -1.500  2.375 -2.625  3.3
75 
##     12     13     14     15     16     17     18     19     20     21     
22 
##  1.375 -0.625 -3.625 -6.625  2.375 -1.500  1.500  0.500  0.500  0.500  0.5
00 
##     23     24     25     26     27     28     29     30     31     32     
33 
##  0.500  1.500  3.375  4.375  2.375 -0.625  0.375  0.375 -1.625 -1.625 -0.5
00 
##     34     35     36     37     38     39     40     41     42     43     
44 
##  0.500  4.500 -2.500 -4.500 -2.500 -1.500  3.500 -0.625  0.375  1.375  0.3
75 
##     45     46     47     48     49     50     51     52     53     54     
55 
## -3.625  0.375  1.375 -3.625 -0.500  1.500  2.500 -0.500 -3.500 -3.500  6.5
00 
##     56     57     58     59     60     61     62     63     64 
##  2.500  2.375 -0.625  2.375  0.375 -0.625 -0.625 -3.625  1.375
```

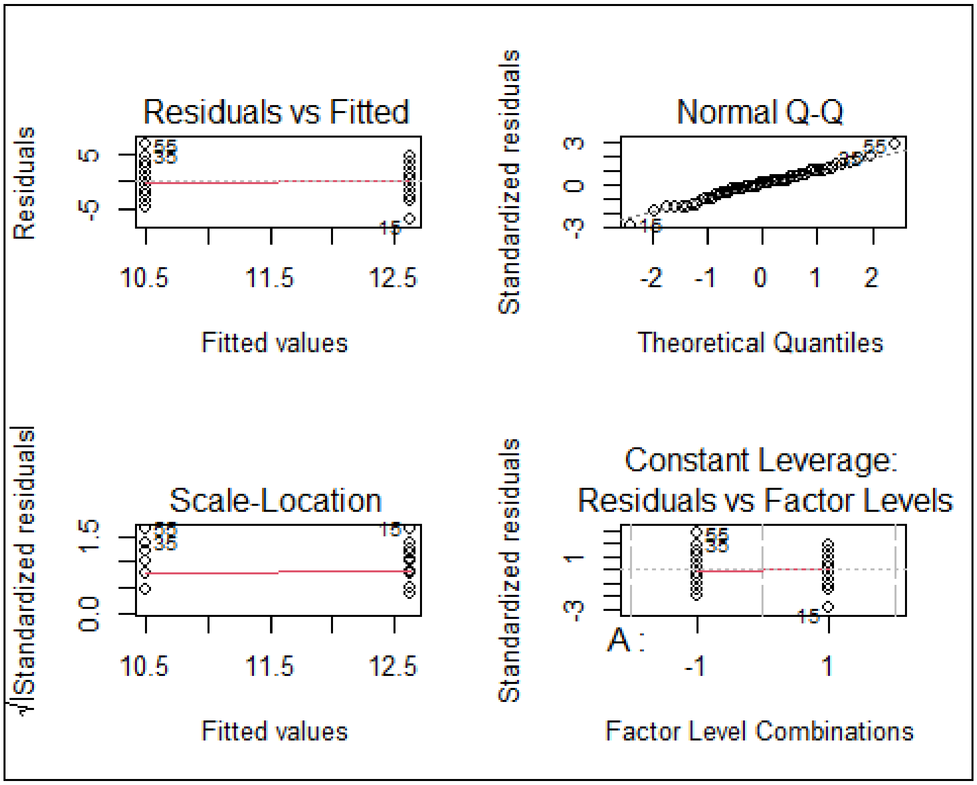
```
cbind(response - yhat,residual)
```

```
##            residual
## 1   0.500    0.500
## 2  -1.500   -1.500
## 3  -0.500   -0.500
## 4  -0.500   -0.500
## 5   0.500    0.500
## 6  -0.500   -0.500
## 7  -2.500   -2.500
## 8  -1.500   -1.500
## 9   2.375    2.375
## 10 -2.625   -2.625
## 11  3.375    3.375
## 12  1.375    1.375
## 13 -0.625   -0.625
## 14 -3.625   -3.625
## 15 -6.625   -6.625
## 16  2.375    2.375
## 17 -1.500   -1.500
## 18  1.500    1.500
## 19  0.500    0.500
## 20  0.500    0.500
## 21  0.500    0.500
## 22  0.500    0.500
## 23  0.500    0.500
## 24  1.500    1.500
## 25  3.375    3.375
## 26  4.375    4.375
## 27  2.375    2.375
## 28 -0.625   -0.625
## 29  0.375    0.375
## 30  0.375    0.375
## 31 -1.625   -1.625
## 32 -1.625   -1.625
## 33 -0.500   -0.500
## 34  0.500    0.500
## 35  4.500    4.500
## 36 -2.500   -2.500
## 37 -4.500   -4.500
## 38 -2.500   -2.500
## 39 -1.500   -1.500
## 40  3.500    3.500
## 41 -0.625   -0.625
## 42  0.375    0.375
## 43  1.375    1.375
## 44  0.375    0.375
## 45 -3.625   -3.625
## 46  0.375    0.375
## 47  1.375    1.375
## 48 -3.625   -3.625
## 49 -0.500   -0.500
## 50  1.500    1.500
## 51  2.500    2.500
## 52 -0.500   -0.500
## 53 -3.500   -3.500
## 54 -3.500   -3.500
## 55  6.500    6.500
## 56  2.500    2.500
## 57  2.375    2.375
## 58 -0.625   -0.625
## 59  2.375    2.375
## 60  0.375    0.375
## 61 -0.625   -0.625
## 62 -0.625   -0.625
## 63 -3.625   -3.625
## 64  1.375    1.375
```

```
#normality of residuals**********************

#qunatile-quantile plot


qqnorm(residual)

qqline(residual)
```

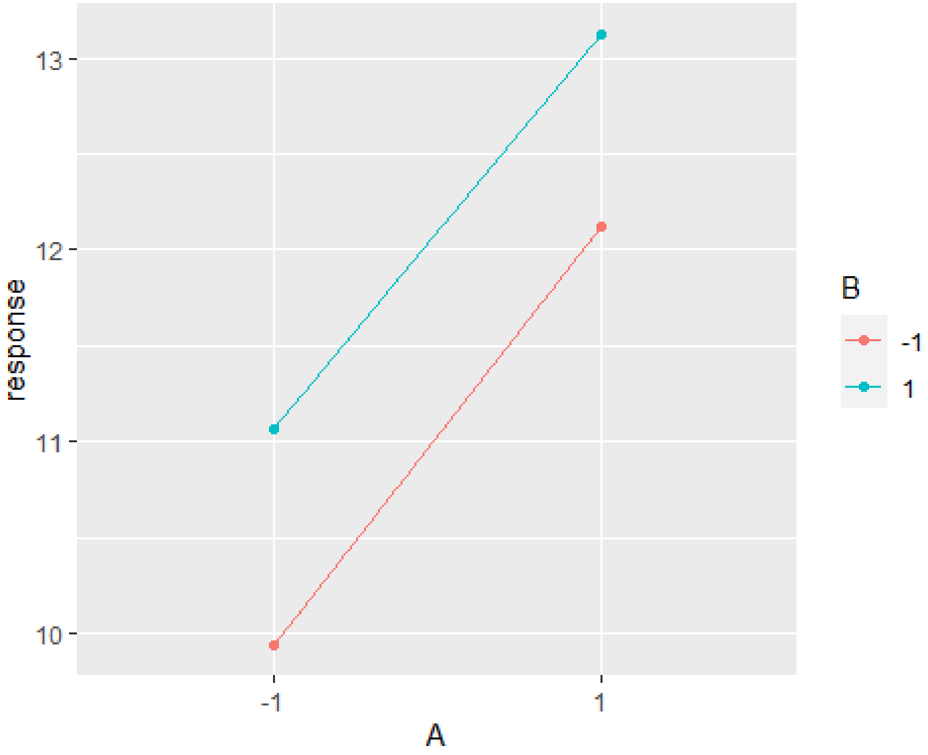


```
#Residuals plots standard chart

par(mfrow=c(2,2))

plot(fit4)

box("outer")
```

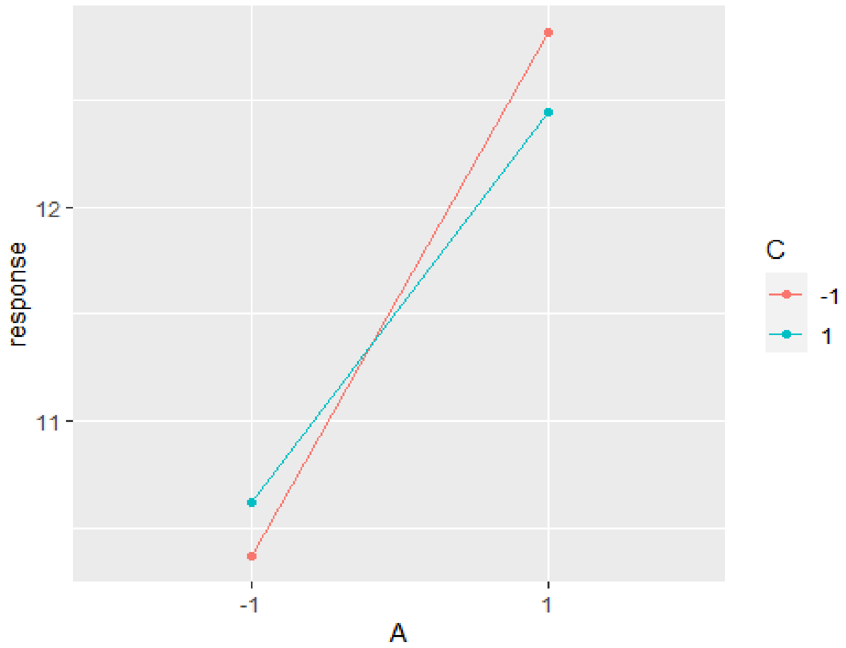


```
#**********************
#cheking normality of residuals with shapiro test:
shapiro.test(residual)
##  Shapiro-Wilk normality test
##
## data:  residual
## W = 0.98244, p-value = 0.4953

str(shapiro.test(residual))

## List of 4
##  $ statistic: Named num 0.982
##   ..- attr(*, "names")= chr "W"
##  $ p.value  : num 0.495
##  $ method   : chr "Shapiro-Wilk normality test"
##  $ data.name: chr "residual"
##  - attr(*, "class")= chr "htest"
```

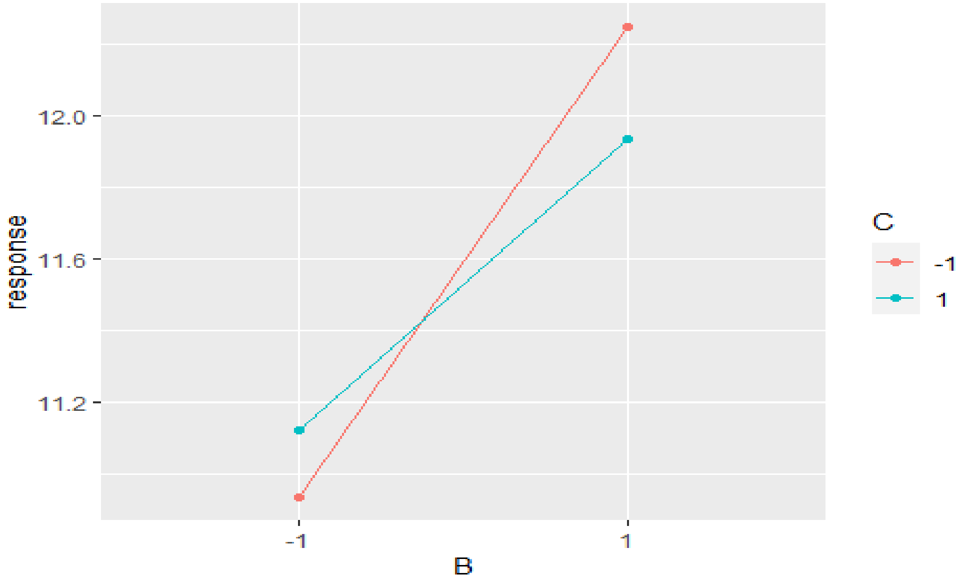
```
(pvlaue=shapiro.test(residual)$p.value)

## [1] 0.4952593

if(pvalue<alpha)print("residuals have normal distribution") else print("residuals have normal distribution")

## [1] "residuals have normal distribution"

#effects interaction of AB*********************************

library(ggplot2)

intEf <- aggregate(response ~  A +B,FUN = mean, data = data2)


effects_interaction <- ggplot(intEf, aes(x = A, y = response, color = B)) +

  geom_point() + geom_line(aes(group = B))


effects_interaction
```



```
#***effects interaction of AC*******************
intEf <- aggregate(response ~ A + C,FUN = mean, data = data2)
effects_interaction <- ggplot(intEf, aes(x = A, y = response, color = C)) +
    geom_point() + geom_line(aes(group = C))
effects_interaction
```



```
#***effects interaction of BC*******************************
intEf <- aggregate(response ~ B + C,FUN = mean, data = data2)
effects_interaction <- ggplot(intEf, aes(x = B, y = response, color = C)) +
  geom_point() + geom_line(aes(group = C))
effects_interaction
```


response
B
C