Experiment 4

1.

```
# create design with 7 factors and resolution IV (4)

data<-gen.factorial(levels=c(2,2,2,2,2),nVars=5,varNames=c("A","B","C","D","E"))
data

data <- data %>% mutate(F = A*B*C)
design <- data %>% mutate(G = B*C*D)
design
```

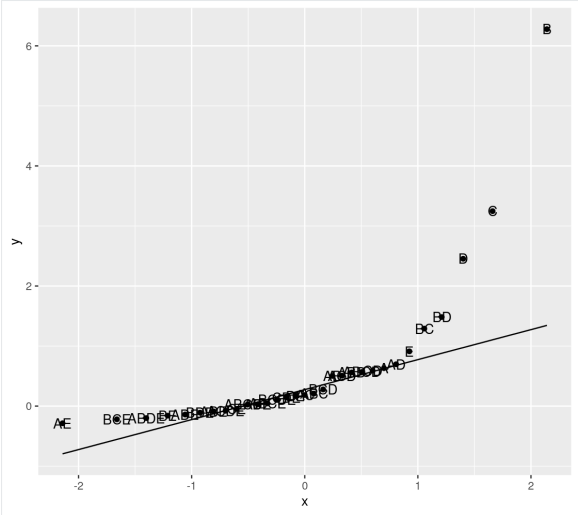
```
> df
    A  B  C  D  E  F  G    y2
1  -1 -1 -1 -1 -1 -1 -1  1.68
2   1 -1 -1 -1 -1  1 -1  1.98
3  -1  1 -1 -1 -1  1  1  4.98
4   1  1 -1 -1 -1 -1  1  5.70
5  -1 -1  1 -1 -1  1  1  3.24
6   1 -1  1 -1 -1 -1  1  3.44
7  -1  1  1 -1 -1 -1 -1  9.97
8   1  1  1 -1 -1  1 -1  9.07
9  -1 -1 -1  1 -1 -1  1  2.07
10  1 -1 -1  1 -1  1  1  2.44
11 -1  1 -1  1 -1  1 -1  7.77
12  1  1 -1  1 -1 -1 -1  9.43
13 -1 -1  1  1 -1  1 -1  4.09
14  1 -1  1  1 -1 -1 -1  4.53
15 -1  1  1  1 -1 -1  1 11.75
16  1  1  1  1 -1  1  1 16.30
17 -1 -1 -1 -1  1 -1 -1  2.69
18  1 -1 -1 -1  1  1 -1  2.30
19 -1  1 -1 -1  1  1  1  6.35
20  1  1 -1 -1  1 -1  1  6.75
21 -1 -1  1 -1  1  1  1  4.34
22  1 -1  1 -1  1 -1  1  4.20
23 -1  1  1 -1  1 -1 -1 10.06
24  1  1  1 -1  1  1 -1  9.35
25 -1 -1 -1  1  1 -1  1  3.22
26  1 -1 -1  1  1  1  1  3.55
27 -1  1 -1  1  1  1 -1  9.33
28  1  1 -1  1  1 -1 -1  9.52
29 -1 -1  1  1  1  1 -1  5.91
30  1 -1  1  1  1 -1 -1  5.83
31 -1  1  1  1  1 -1  1 13.23
32  1  1  1  1  1  1  1 16.40
```

2.

```
library(readr)
group_response <- read_csv("group project/group_response.csv")
y2<- group_response["y2"][complete.cases(group_response["y2"]), ]
y2
```

```
df<- cbind(design, y2)
df

yates<-yates(df$y2)
yates<-as.data.frame(as.matrix(yates))
yates$name <-row.names(yates)
yates

ggplot(yates,aes(sample = V1)) + geom_text(label=yates$name[order(yates$V1)], stat="qq") +
  stat_qq() + stat_qq_line()


m3<-aov(data=df, y2 ~ B+C+D+B*D+B*C+A*E)
summary(m3)

m3<-lm(data=df, y2 ~ B+C+D+E+B*D+B*C)
summary(m3)
```

```
> yates
              V1  name
A       0.631875     A
B       6.278125     B
AB      0.503125    AB
C       3.246875     C
AC      0.184375    AC
BC      1.290625    BC
ABC     0.208125   ABC
D       2.454375     D
AD      0.696875    AD
BD      1.483125    BD
ABD     0.560625   ABD
CD      0.591875    CD
ACD     0.506875   ACD
BCD     0.278125   BCD
ABCD    0.568125  ABCD
E       0.911875     E
AE     -0.285625    AE
BE     -0.159375    BE
ABE    -0.086875   ABE
CE     -0.045625    CE
ACE     0.029375   ACE
BCE    -0.219375   BCE
ABCE    0.045625  ABCE
DE      0.164375    DE
ADE    -0.140625   ADE
BDE    -0.109375   BDE
ABDE   -0.199375  ABDE
CDE     0.144375   CDE
ACDE   -0.078125  ACDE
BCDE    0.103125  BCDE
ABCDE   0.025625 ABCDE
```




```
> summary(m3)
            Df Sum Sq Mean Sq F value   Pr(>F)
B            1 315.32  315.32 384.033 7.54e-16 ***
C            1  84.34   84.34 102.716 5.93e-10 ***
D            1  48.19   48.19  58.694 8.94e-08 ***
A            1   3.19    3.19   3.890 0.060709 .
E            1   6.65    6.65   8.102 0.009140 **
B:D          1  17.60   17.60  21.432 0.000117 ***
B:C          1  13.33   13.33  16.230 0.000524 ***
A:E          1   0.65    0.65   0.795 0.381862
Residuals   23  18.88    0.82
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
> summary(m3)

Call:
lm.default(formula = y2 ~ B + C + D + E + B * D + B * C, data = df)

Residuals:
     Min       1Q   Median       3Q      Max
-1.77906 -0.40125 -0.07594  0.39562  2.77094

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   6.6084     0.1686  39.204  < 2e-16 ***
B             3.1391     0.1686  18.622 3.62e-16 ***
C             1.6234     0.1686   9.631 6.81e-10 ***
D             1.2272     0.1686   7.280 1.25e-07 ***
E             0.4559     0.1686   2.705 0.012122 *
B:D           0.7416     0.1686   4.399 0.000177 ***
B:C           0.6453     0.1686   3.828 0.000769 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.9536 on 25 degrees of freedom
Multiple R-squared:  0.9553,    Adjusted R-squared:  0.9445
F-statistic: 88.98 on 6 and 25 DF,  p-value: 1.221e-15
```

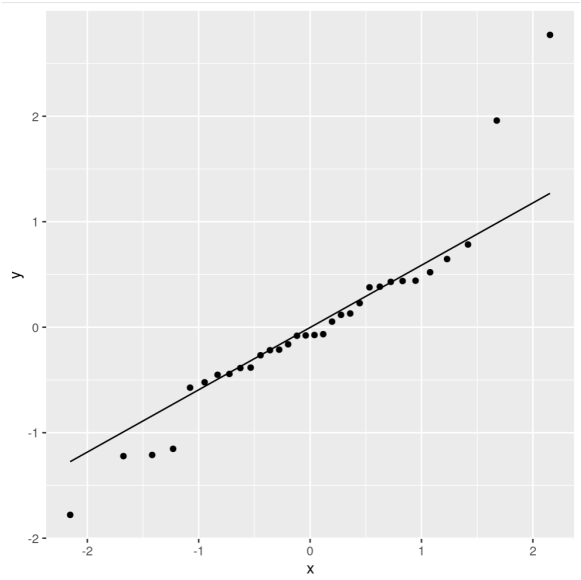
3.

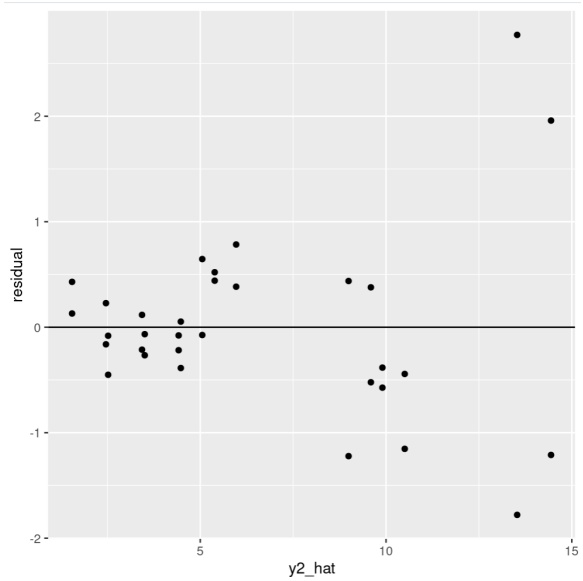
	I=ABCF	I=BCDG	I=ADFG
A=	BCF	ABCDG	DFG
B=	ACF	CDG	ABDFG
C=	ABF	BDG	ACDFG
D=	ABCDF	BCG	AFG
E=	ABCEF	BCDEG	ADEFG
F=	ABC	BCDFG	ADG
G=	ABCFG	BCD	ADF
AB=	CF	ACDG	BDFG
BC=	AF	DG	ABCDFG
AC=	BF	ABDG	CDFG
AD=	BCDF	ABCG	FG
BD=	ACDF	CG	ABFG
CD=	ABDF	BG	ACFG
AE=	BCEF	ABCDEG	DEFG
BE=	ACEF	CDEG	ABDEFG
CE=	ABEF	BDEG	ACDEFG
DE=	ABCDEF	BCEG	AEFG

the shortest word defining relation is 4 so this experiment has resolution IV.

4.

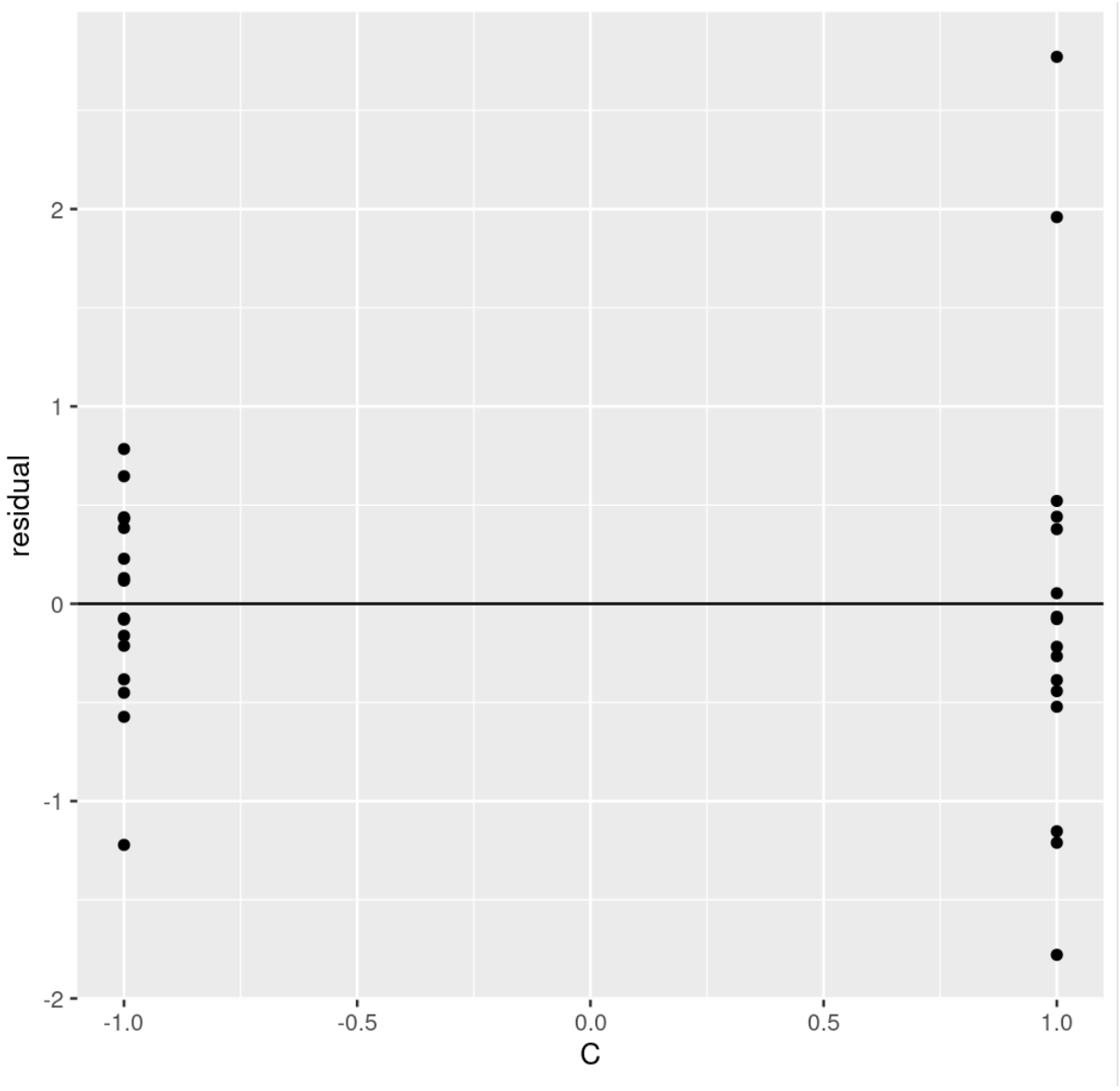
```
points<-get_regression_points(m3)
mold2<- df %>% select(A, F, G)
points<-cbind(mold2, points)
points

ggplot(data=points, aes(sample=residual)) + stat_qq() + stat_qq_line()
ggplot(data=points, aes(x=y2_hat, y=residual)) + geom_point() + geom_hline(yintercept=0)
```





there is a cone shape in residual against predicted values plot, so we should use dispersion effect.

```
points <- points %>% mutate(AB=A*B, AC=A*C, AD=A*D, AE=A*E, AF=A*F, BD=B*D, BF=B*F, ABD=A*B*D, ACD=A*C*D)
varlist<-c("A", "B", "C", "D", "E", "F", "AB", "AC", "AD", "AE", "AF", "BD", "BF", "ABD", "ACD")
disp<-rep(0,15)

dispersion <- function (data, class, var) {
  summ<-as.matrix(data %>% group_by(.data[[class]]) %>% summarise(varres=var(.data[[var]])))
  log(summ[2,2]/summ[1,2])
}

for(i in 1:15){
  disp[i]<-dispersion(points, class=varlist[i], var="residual")
}

disp

dispeffect <- data.frame(varlist, disp)
dispeffect

ggplot(dispeffect,aes(sample = disp)) + geom_text(label=dispeffect$varlist[order(dispeffect$disp)], stat="qq") +
  stat_qq() + stat_qq_line()

ggplot(data=points, aes(x=C, y=residual)) + geom_point() + geom_hline(yintercept=0)
```


residual
2
1
0
-1
-2
-1.0
-0.5
0.0
0.5
1.0
C