

# 5303hw11

*Jin Yao*

*2019/11/20*

5303 homework 11

## E 15.1

```
library(cfcdae)
library(conf.design)

gen1 <- rbind(c(0,1,1,1,0),c(1,0,1,1,0))
conf.design(gen1,2)
```

##	Blocks	T1	T2	T3	T4	T5
## 1	00	0	0	0	0	0
## 2	00	1	1	1	0	0
## 3	00	1	1	0	1	0
## 4	00	0	0	1	1	0
## 5	00	0	0	0	0	1
## 6	00	1	1	1	0	1
## 7	00	1	1	0	1	1
## 8	00	0	0	1	1	1
## 9	01	1	0	0	0	0
## 10	01	0	1	1	0	0
## 11	01	0	1	0	1	0
## 12	01	1	0	1	1	0
## 13	01	1	0	0	0	1
## 14	01	0	1	1	0	1
## 15	01	0	1	0	1	1
## 16	01	1	0	1	1	1
## 17	10	0	1	0	0	0
## 18	10	1	0	1	0	0
## 19	10	1	0	0	1	0
## 20	10	0	1	1	1	0
## 21	10	0	1	0	0	1
## 22	10	1	0	1	0	1
## 23	10	1	0	0	1	1
## 24	10	0	1	1	1	1
## 25	11	1	1	0	0	0
## 26	11	0	0	1	0	0
## 27	11	0	0	0	1	0
## 28	11	1	1	1	1	0
## 29	11	1	1	0	0	1
## 30	11	0	0	1	0	1
## 31	11	0	0	0	1	1
## 32	11	1	1	1	1	1

```
conf.design
```

```
## function (G, p, block.name = "Blocks", treatment.names = NULL)
## {
##   if (!is.matrix(G))
##     G <- rbind(G)
##   if (is.null(treatment.names))
##     treatment.names <- if (is.null(nam <- dimnames(G)[[2]]))
##       .paste0("T", .zf(1:ncol(G)))
##     else nam
##   stopifnot(is.character(treatment.names), length(treatment.names) ==
##     ncol(G), is.numeric(G), all(G >= 0), all(G%%1 == 0),
##     is.numeric(p), length(p) == 1, p > 0, p%%1 == 0, p %in%
##     primes(p), is.character(block.name), length(block.name) ==
##     1)
##   D <- as.matrix(expand.grid(rep(list(0:(p - 1)), ncol(G))))
##   B <- .listMat((D %*% t(G))%p, format)
##   o <- do.call(order, B)
##   B <- do.call(".paste0", B)
##   D <- cbind(B, format(D))[o, ]
##   dimnames(D) <- list(B[o], c(block.name, treatment.names))
##   data.frame(.listMat(D, factor))
## }
## <bytecode: 0x7fce5d883818>
## <environment: namespace:conf.design>
```

I use a,b,c,d to respectively represent T1,T2,T3,T4,T5

block 1:1,abc,abd,cd,e,ace,abde,cde block 2:a,bc,bd,acd,ae,bce,bde,acde block 3:b,ac,ad,bcd,be,ace,ade,bcde  
block 4:ab,c,d,abcd,abe,ce,de,abcde

## E 15.3

```
gen2 = rbind(c(1,1,1,0,0,1,0,0),c(1,1,0,1,1,0,0,0),c(1,0,1,1,1,0,0,0),c(0,1,1,1,0,0,0,1))
conf.set(gen2,2)
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
## [1,] 1    1    1    0    0    1    0    0
## [2,] 1    1    0    1    1    0    0    0
## [3,] 0    0    1    1    1    1    0    0
## [4,] 1    0    1    1    1    0    0    0
## [5,] 0    1    0    1    1    1    0    0
## [6,] 0    1    1    0    0    0    0    0
## [7,] 1    0    0    0    0    1    0    0
## [8,] 0    1    1    1    0    0    0    1
## [9,] 1    0    0    1    0    1    0    1
## [10,] 1    0    1    0    1    0    0    1
## [11,] 0    1    0    0    1    1    0    1
## [12,] 1    1    0    0    1    0    0    1
## [13,] 0    0    1    0    1    1    0    1
## [14,] 0    0    0    1    0    0    0    1
## [15,] 1    1    1    1    0    1    0    1
```

```
# Use ABCDEFGH respective
```

counfounded:ABCF,ABDE,CDEF,ACDE,BDEF,BC,AF,BCDH,ADFH,ACEH,BEFH,ABEH,CEFH,DH,ABCDFH

## P 15.1

- (a) RCB I can split the grassland into 50 parts, so I can use RCB.
- (b) PBIBD 16 units, there are 8 treatment combinations, at about 4 blocks(4 runs per day), so  $\lambda = 6/7$
- (c) BIBD 12 units, there are 4 treatments(stints), so 3 units each group,  $\lambda = 6$
- (d) RCB there are 8 treatment combinations, because only 8 per block, so it is RCB
- (e) PBIBD there are 8 treatment combinations(A,B,C each 2), 4 units within each block,  $\lambda = 6/7$

## P 15.2

- (a) youden square, there are two blocks,  $df(\text{doctor})=63, df(\text{drug})=1, df(\text{comb})=1, df(\text{error})=62$
- (b) BIBD there are 6 treatments, 4 units within each block,  $df(\text{brands})=5, df(\text{day})=14, df(\text{error})=40$
- (c) Confounding ab,  $df(\text{protein})=1, df(\text{choline})=1, df(\text{row})=3, df(\text{error})=34$
- (d) latin square, because each subject received drugs and each drug was used equal numbers  $df(\text{subject})=29, df(\text{session})=2, df(\text{drug})=1, df(\text{error})=28$
- (e) BIBD there are 15 combinations of treatment,  $df(\text{days})=2, df(\text{charge})=2, df(\text{additive})=4$ , because there are two treatments,  $df(\text{charge:add})=8, df(\text{error})=13$

## P 15.3

```
data("OdorIntensity")
summary(OdorIntensity)
```

```
##      judge      A      B      C      D      concentration
## 1      :4  high:16  high:16  high:16  high:16  Min.      :    4
## 2      :4  low :16  low :16  low :16  low :16  1st Qu.:   93
## 3      :4                                     Median :  800
## 4      :4                                     Mean   :11832
## 5      :4                                     3rd Qu.:11068
## 6      :4                                     Max.   :90293
## (Other):8
```

```
head(OdorIntensity)
```

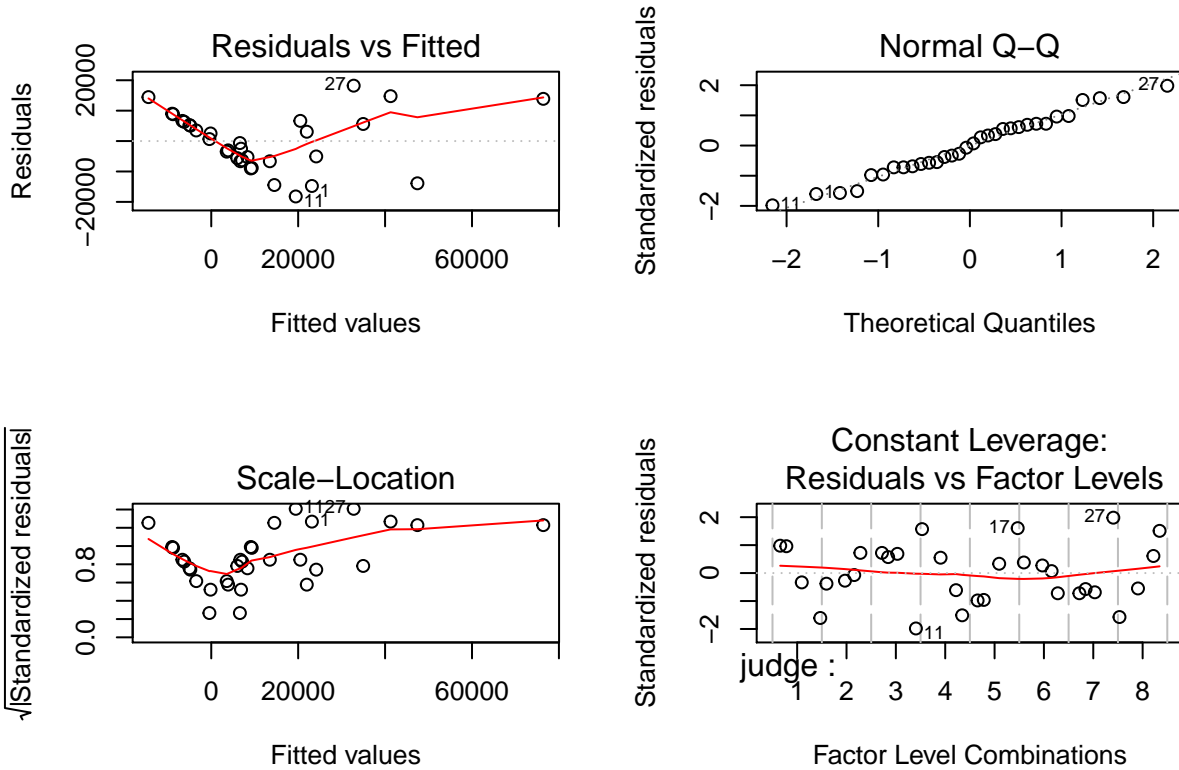
```
##  judge      A      B      C      D      concentration
## 1      1  low  low  low  low          8389
## 2      1  low  high  high  low          816
## 3      1  high  high  low  high          4
## 4      1  high  low  high  high          46
## 5      2  high  low  low  low         4351
## 6      2  high  high  high  low          78
```

```
OdorIntensity$A =as.factor(OdorIntensity$A)
OdorIntensity$B =as.factor(OdorIntensity$B)
OdorIntensity$C =as.factor(OdorIntensity$C)
```

```

OdorIntensity$D =as.factor(OdorIntensity$D)
mod.15.3 <- lm(concentration~judge+A*B*C*D, data = OdorIntensity)
par(mfrow=c(2,2))
plot(mod.15.3)

```



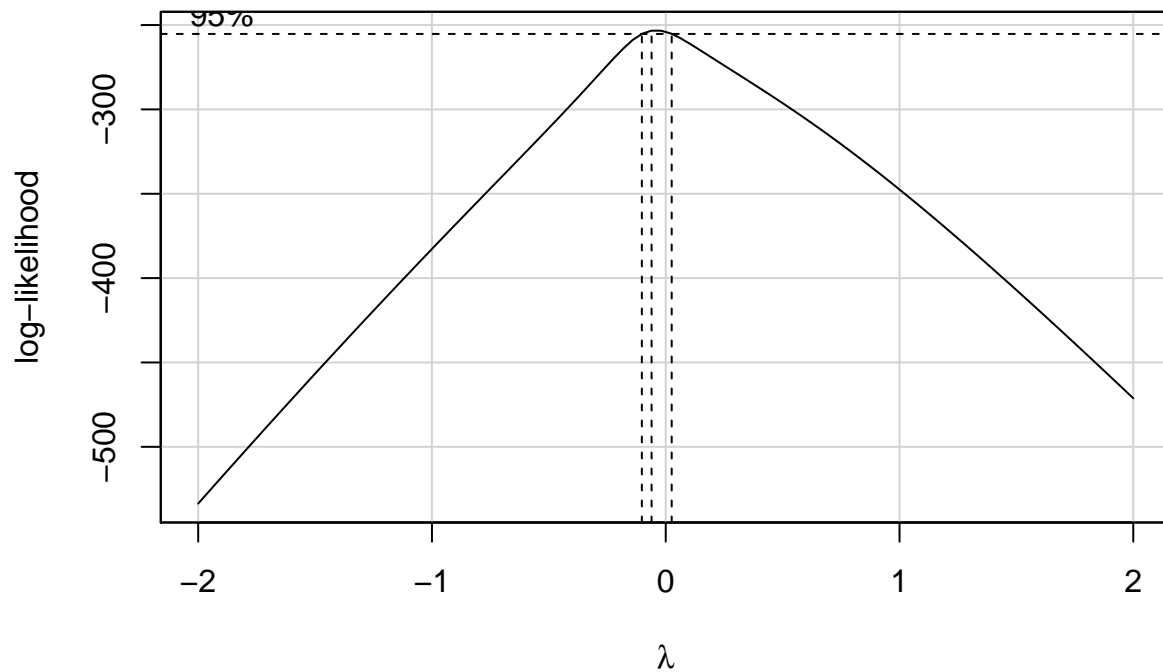
```

# Needs transformation
par(mfrow=c(1,1))
library(car)

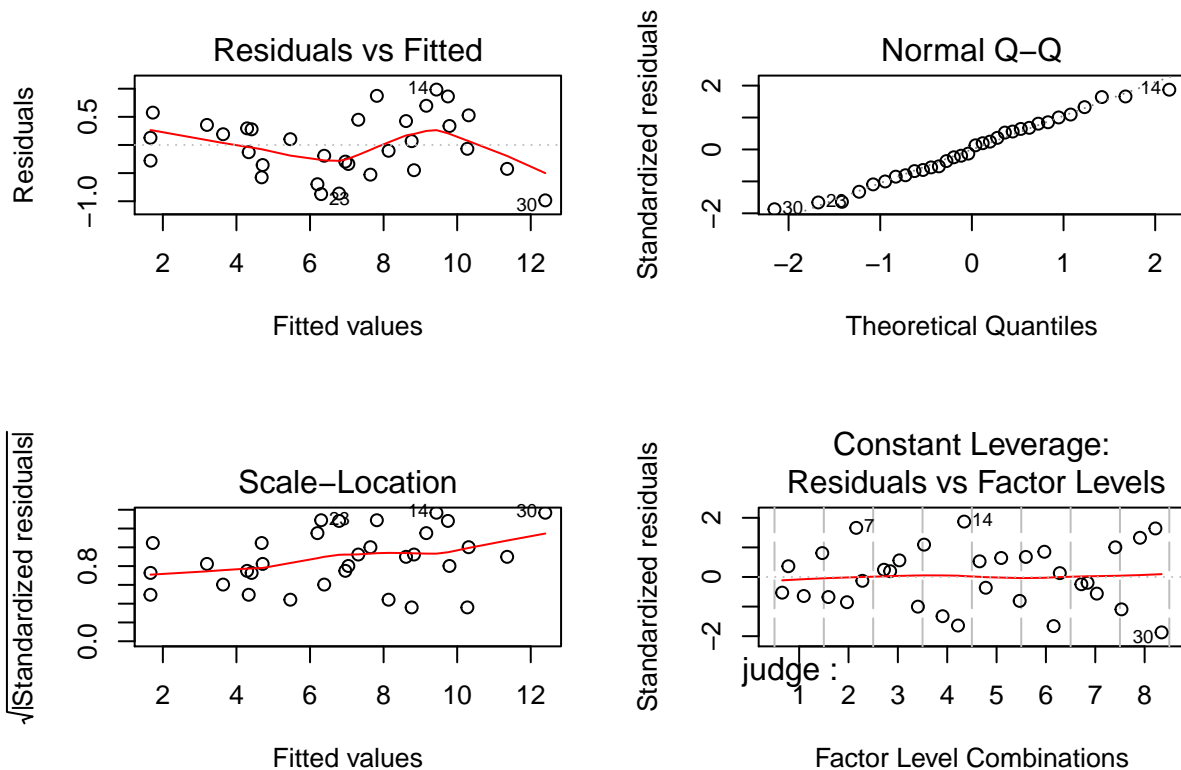
```

```
## Loading required package: carData
```

```
boxCox(mod.15.3) # log
```



```
mod.15.3.log = lm(log(concentration)~judge+A*B*C*D, data = OdorIntensity)
par(mfrow=c(2,2))
plot(mod.15.3.log)
```



```
anova(mod.15.3.log)
```

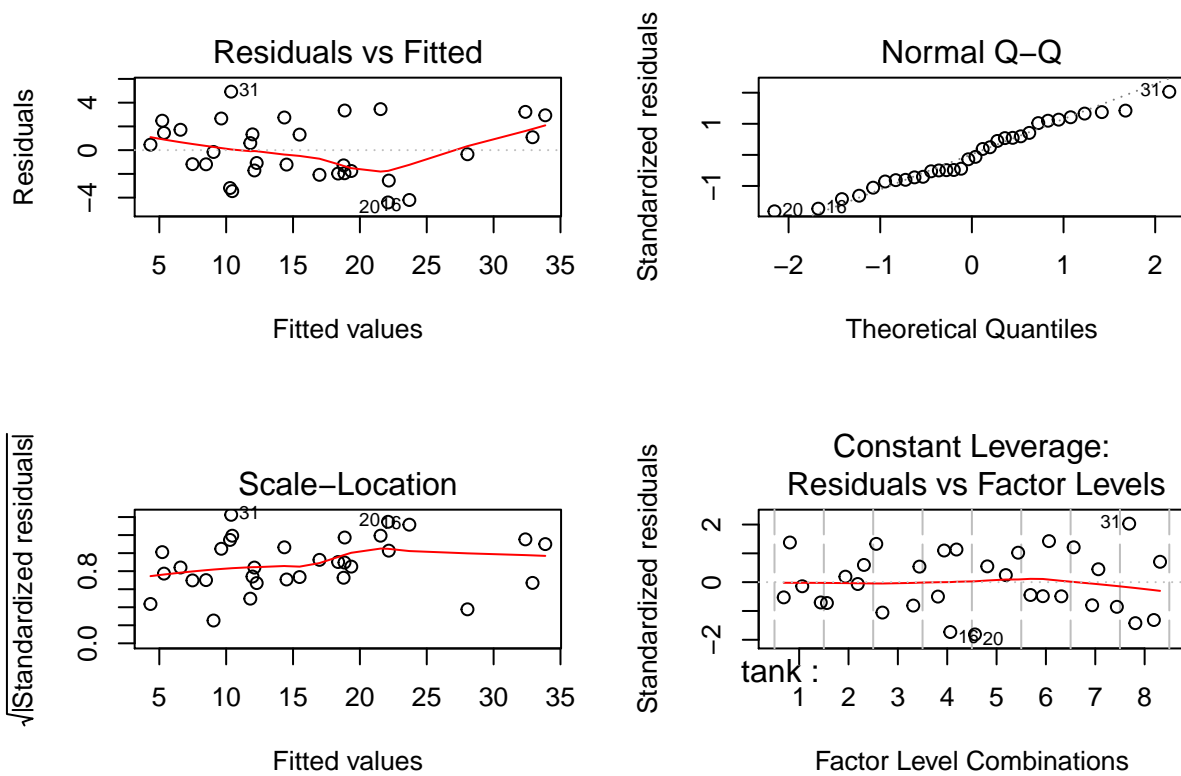
```
## Analysis of Variance Table
##
```

```
## Response: log(concentration)
##           Df Sum Sq Mean Sq  F value    Pr(>F)
## judge      7 60.710   8.673   11.7590 0.0001586 ***
## A          1 98.026  98.026  132.9079 7.548e-08 ***
## B          1 74.362  74.362  100.8229 3.426e-07 ***
## C          1  0.862   0.862    1.1686 0.3009266
## D          1 10.788  10.788   14.6273 0.0024193 **
## A:B         1  2.350   2.350    3.1863 0.0995380 .
## A:C         1  1.125   1.125    1.5251 0.2404777
## B:C         1  0.663   0.663    0.8986 0.3618536
## B:D         1  0.002   0.002    0.0029 0.9582659
## C:D         1  2.166   2.166    2.9364 0.1122962
## A:B:D        1  1.175   1.175    1.5936 0.2307968
## A:C:D        1  0.006   0.006    0.0082 0.9293287
## A:B:C:D       1  0.407   0.407    0.5520 0.4717972
## Residuals  12  8.851   0.738
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

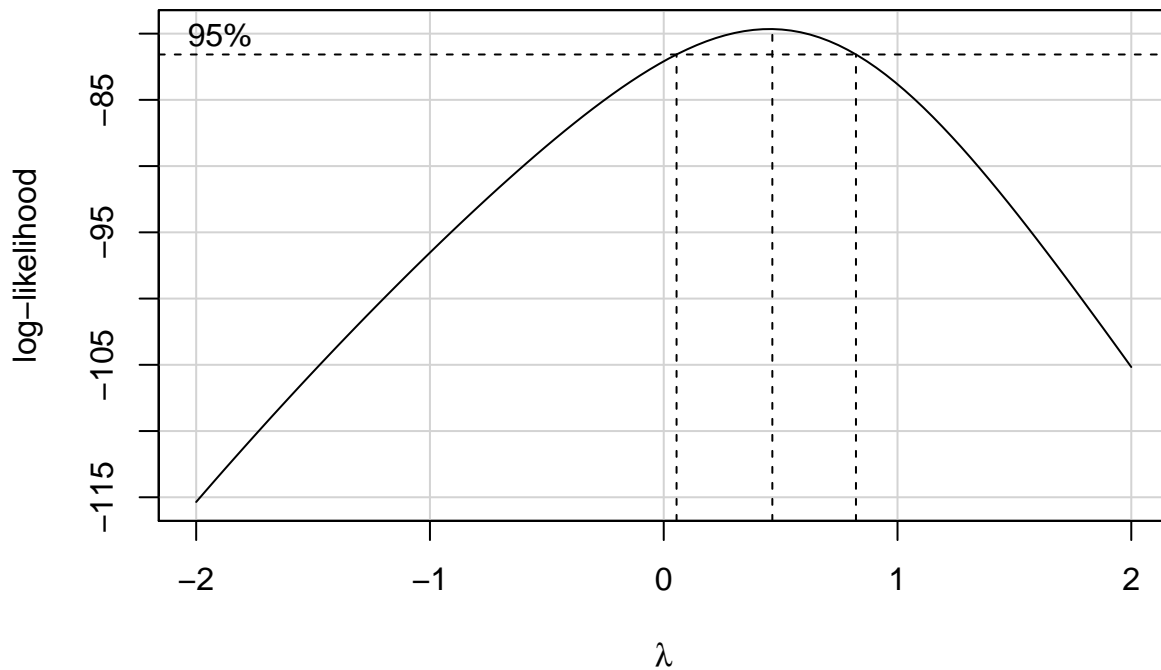
# the judge, A ,B, D are significant, the A:B is marginally significant
# I did not find A:D,A:B:C,B:C:D in the table, so it not very good.
```

## P 15.4

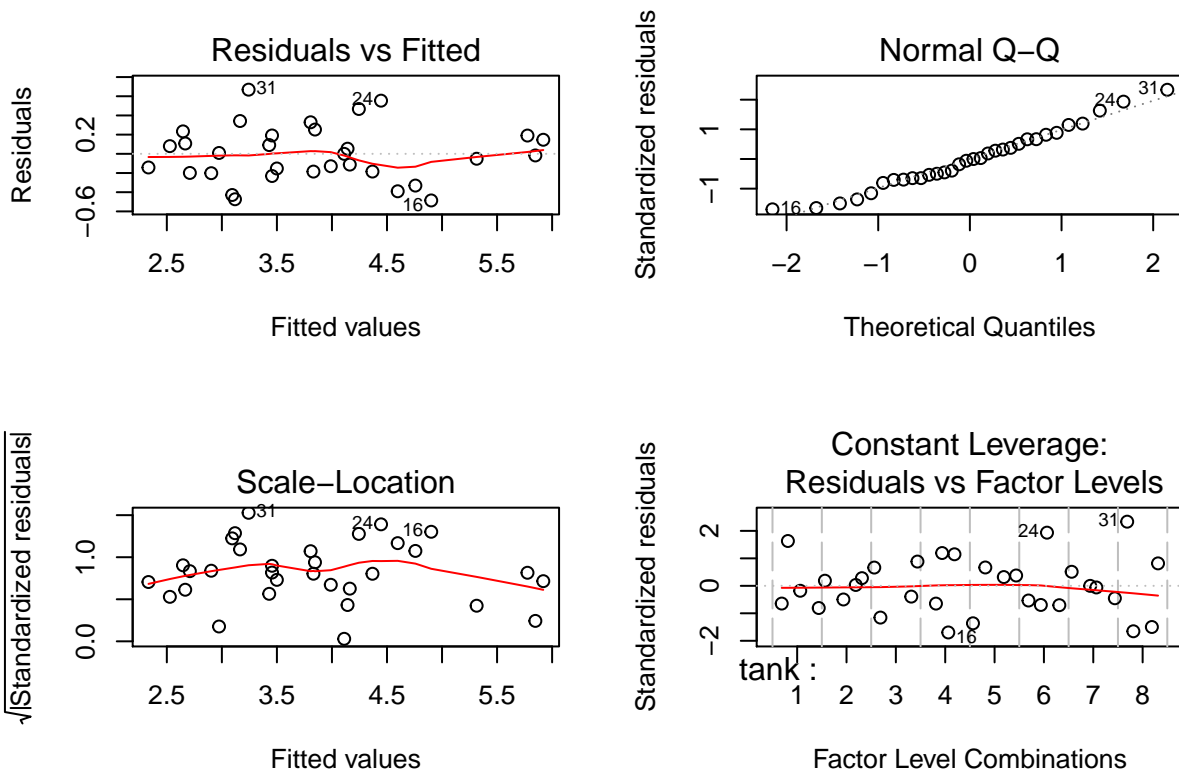
```
data("Milfoil")
Milfoil$A =as.factor(Milfoil$A)
Milfoil$B =as.factor(Milfoil$B)
Milfoil$C =as.factor(Milfoil$C)
mod.15.4 = lm(biomass~tank+A*B*C,data = Milfoil)
par(mfrow=c(2,2))
plot(mod.15.4)
```



```
# not so good, take a look at boxcox
par(mfrow=c(1,1))
boxCox(mod.15.4)
```



```
# about 0.5
mod.15.4.0.5 = lm(biomass^(0.5)~tank+A*B*C,data = Milfoil)
par(mfrow=c(2,2))
plot(mod.15.4.0.5)
```



```
anova(mod.15.4.0.5)
```

```
## Analysis of Variance Table
##
## Response: biomass^(0.5)
##          Df Sum Sq Mean Sq F value    Pr(>F)
## tank      7 15.3510   2.1930 14.2064 5.597e-06 ***
## A          1  0.3323   0.3323  2.1527  0.16057
## B          1  8.9774   8.9774 58.1561 6.961e-07 ***
## C          1  4.7941   4.7941 31.0566 3.367e-05 ***
## A:B        1  0.0522   0.0522  0.3380  0.56863
## A:C        1  0.0280   0.0280  0.1815  0.67543
## B:C        1  0.4796   0.4796  3.1070  0.09593 .
## A:B:C      1  0.0687   0.0687  0.4448  0.51376
## Residuals 17  2.6242   0.1544
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
# B,C are significant, B:C is marginally significant
```