```
> coagulation
   coag diet
1    62    A
2    60    A
3    63    A
4    59    A
5    63    B
6    67    B
7    71    B
8    64    B
9    65    B
10   66    B
11   68    C
12   66    C
13   71    C
14   67    C
15   68    C
16   68    C
17   56    D
18   62    D
19   60    D
20   61    D
21   63    D
22   64    D
23   63    D
24   59    D
```

```
> attach(coagulation)
> attributes(diet)
$levels
[1] "A" "B" "C" "D"

$class
[1] "factor"

> group.mean=rep(0,4)
> for(i in 1:4)
+   group.mean[i]=mean(coag[as.numeric(diet)==i])
> group.mean
[1] 61 66 68 61

> g=lm(coag~diet)
> cbind(g$fitted, diet)
      diet
1  61    1
2  61    1
3  61    1
4  61    1
5  66    2
6  66    2
7  66    2
8  66    2
9  66    2
10 66    2
```

```
11 68    3
12 68    3
13 68    3
14 68    3
15 68    3
16 68    3
17 61    4
18 61    4
19 61    4
20 61    4
21 61    4
22 61    4
23 61    4
24 61    4


> anova(g)
Analysis of Variance Table

Response: coag
          Df Sum Sq Mean Sq F value    Pr(>F)
diet       3    228    76.0  13.571 4.658e-05 ***
Residuals 20    112     5.6
```

```
> group.mean
[1] 61 66 68 61

> summary(g)
Estimate Std. Error t value Pr(>|t|)
(Intercept)       61      1.183  51.554    0.000
dietB              5      1.528   3.273    0.004
dietC              7      1.528   4.583    0.000
dietD              0      1.449   0.000    1.000
> contrasts(diet)
  B C D
A 0 0 0
B 1 0 0
C 0 1 0
D 0 0 1

> model.matrix(g)
   (Intercept) dietB dietC dietD
1            1     0     0     0
2            1     0     0     0
3            1     0     0     0
4            1     0     0     0
5            1     1     0     0
6            1     1     0     0
7            1     1     0     0
8            1     1     0     0
```

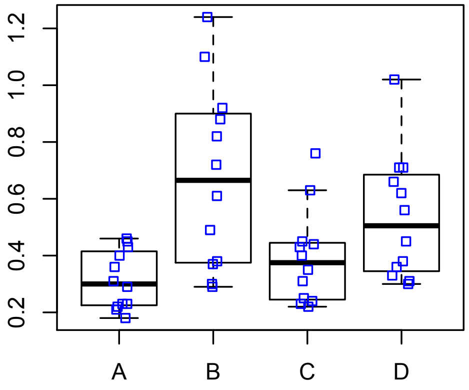
9	1	1	0	0
10	1	1	0	0
11	1	0	1	0
12	1	0	1	0
13	1	0	1	0
14	1	0	1	0
15	1	0	1	0
16	1	0	1	0
17	1	0	0	1
18	1	0	0	1
19	1	0	0	1
20	1	0	0	1
21	1	0	0	1
22	1	0	0	1
23	1	0	0	1
24	1	0	0	1

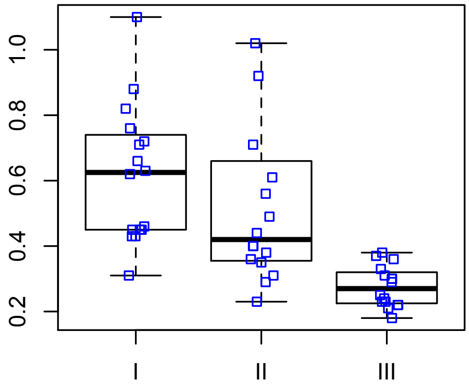
```
# Use different contrasts matrices

> group.mean
[1] 61 66 68 61
> contr.sum(4)
  [,1] [,2] [,3]
1    1    0    0
2    0    1    0
3    0    0    1
4   -1   -1   -1
> contrasts(diet) = contr.sum(4)
> lm(coag~diet)$coef
(Intercept)        diet1        diet2        diet3
         64           -3            2            4

> contr.helmert(4)
  [,1] [,2] [,3]
1   -1   -1   -1
2    1   -1   -1
3    0    2   -1
4    0    0    3
> contrasts(diet) = contr.helmert(4)
> lm(coag~diet)$coef
(Intercept)        diet1        diet2        diet3
       64.0          2.5          1.5         -1.0
```

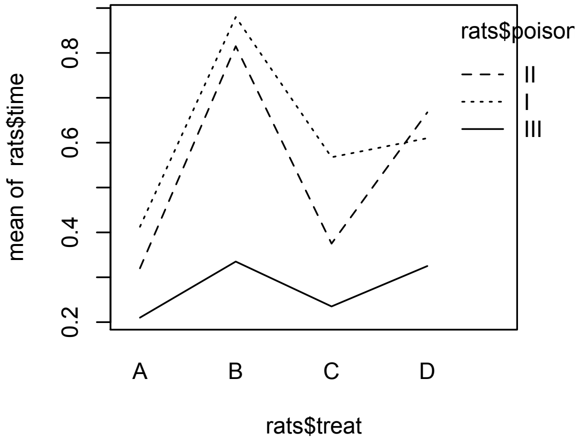
```
# Note that the intercept from contr.sum and contr.helmert
# isn't the sample mean but the mean of the group means.
> coag=coag[-20]
> diet=diet[-20]
> group.mean=rep(0,4)
> for(i in 1:4){
+   group.mean[i]=mean(coag[as.numeric(diet)==i])
+ }
>
> group.mean
[1] 61 66 68 61
> mean(coag)
[1] 64.13043

> contrasts(diet) = contr.sum(4)
> lm(coag~diet)$coef
(Intercept)       diet1       diet2       diet3
         64          -3           2           4
> contrasts(diet) = contr.helmert(4)
> lm(coag~diet)$coef
(Intercept)       diet1       diet2       diet3
       64.0         2.5         1.5        -1.0
```

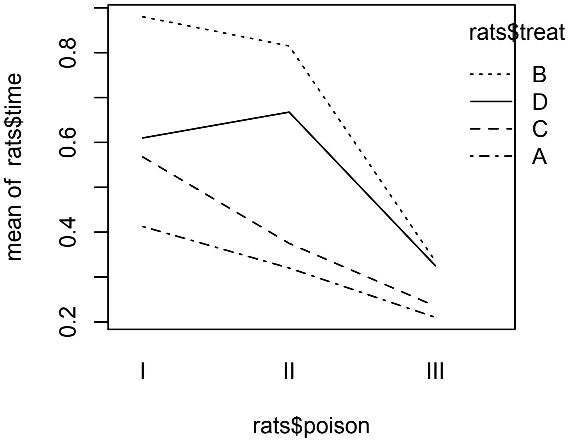
```
> library(faraway)
> ### Rats Example (Balanced Design)

> rats  # display the data
   time poison treat
1  0.31      I     A
2  0.82      I     B
3  0.43      I     C
4  0.45      I     D
5  0.45      I     A
6  1.10      I     B
7  0.45      I     C
8  0.71      I     D
9  0.46      I     A
10 0.88      I     B
11 0.63      I     C
12 0.66      I     D
13 0.43      I     A
14 0.72      I     B
15 0.76      I     C
16 0.62      I     D
17 0.36     II     A
18 0.92     II     B
19 0.44     II     C
20 0.56     II     D
21 0.29     II     A
22 0.61     II     B
23 0.35     II     C
24 1.02     II     D
25 0.40     II     A
26 0.49     II     B
27 0.31     II     C
28 0.71     II     D
29 0.23     II     A
30 1.24     II     B
31 0.40     II     C
32 0.38     II     D
33 0.22    III     A
34 0.30    III     B
35 0.23    III     C
36 0.30    III     D
37 0.21    III     A
38 0.37    III     B
39 0.25    III     C
40 0.36    III     D
41 0.18    III     A
42 0.38    III     B
```

```
43 0.24    III     C
44 0.31    III     D
45 0.23    III     A
46 0.29    III     B
47 0.22    III     C
48 0.33    III     D

# graphical displays: check the main effects
# and the interactions
> par(mfrow=c(2,2))

> boxplot(time~treat, data=rats, outline=FALSE)

> stripchart(time~treat, data=rats, method="jitter",
col="blue", vertical=TRUE, add=TRUE)

> boxplot(time~poison, data=rats, outline=FALSE)

> stripchart(time~poison, data=rats, method="jitter",
col="blue", vertical=TRUE, add=TRUE)

> interaction.plot(rats$treat, rats$poison, rats$time)

> interaction.plot(rats$poison, rats$treat, rats$time)
```

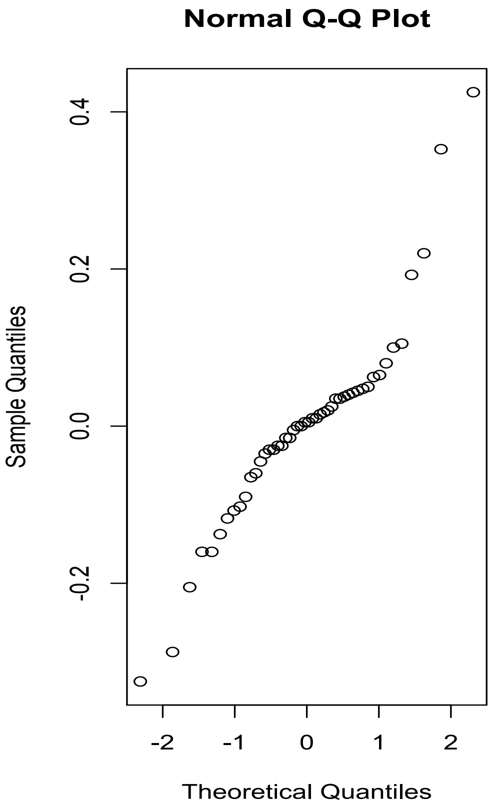

0.2
0.4
0.6
0.8
1.0
1.2
A
B
C
D

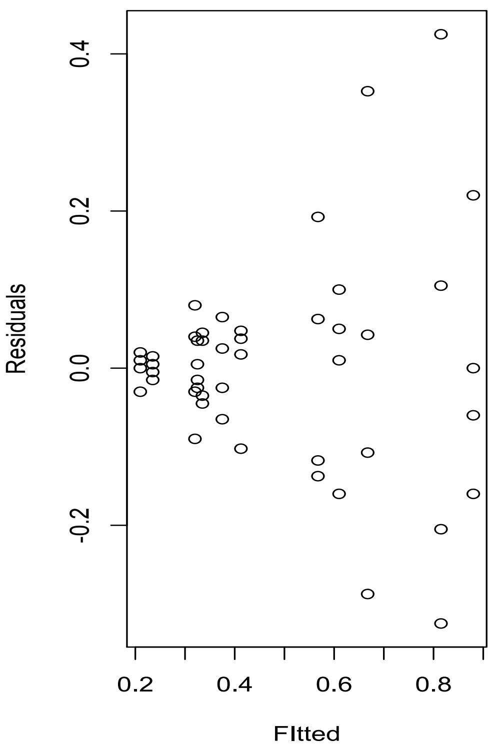

0.2
0.4
0.6
0.8
1.0
I
II
III

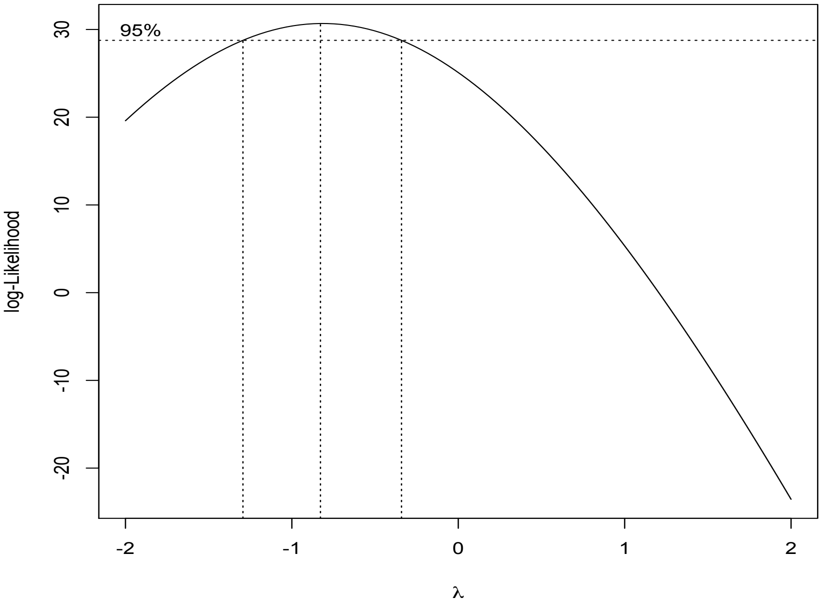

mean of rats$time
0.2
0.4
0.6
0.8
rats$poisor
II
I
III
A
B
C
D
rats$treat

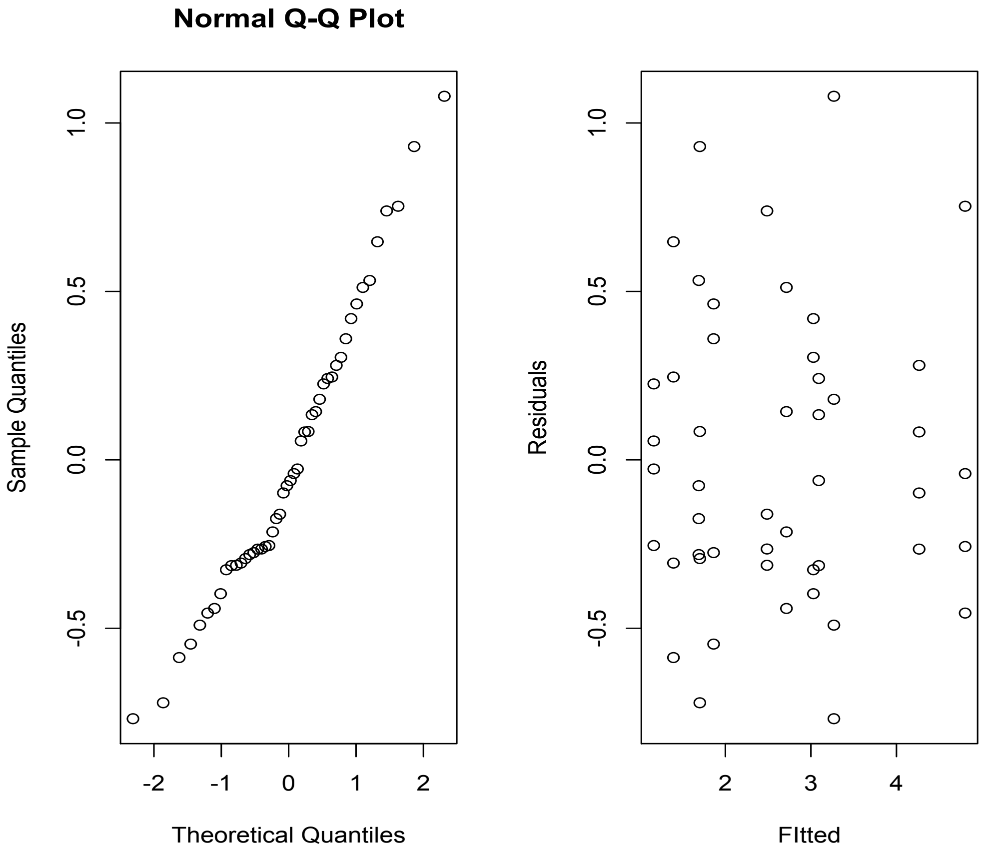

mean of rats$time
0.2
0.4
0.6
0.8
rats$treat
B
D
C
A
I
II
III
rats$poison

```
# F-tests for main effects and the interaction term

> g=lm(time ~ poison*treat, rats)

> anova(g)
Analysis of Variance Table

Response: time
             Df Sum Sq Mean Sq F value Pr(>F)
poison        2  1.033   0.517  23.222 <2e-16 ***
treat         3  0.921   0.307  13.806 <2e-16 ***
poison:treat  6  0.250   0.042   1.874  0.112
Residuals    36  0.801   0.022

# Diagnostic plots
> par(mfrow=c(1,2))

> qqnorm(g$res)

> plot(g$fitted, g$res, xlab="FItted", ylab="Residuals")
# the trumpet pattern

> library(MASS)

> boxcox(g)
```

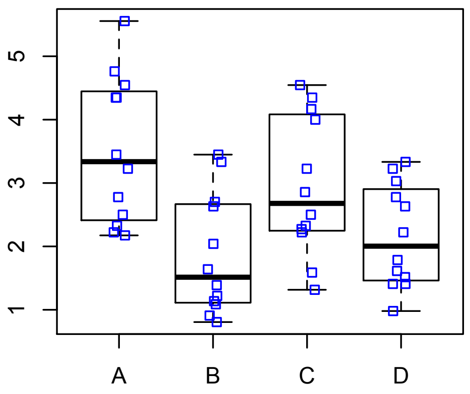

Normal Q-Q Plot
Sample Quantiles
Theoretical Quantiles
0.4
0.2
0.0
-0.2
-2
-1
0
1
2

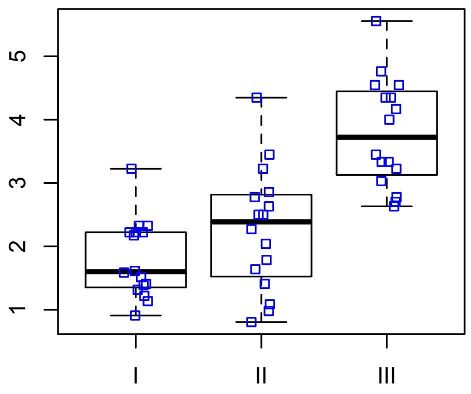

Residuals
FItted
0.4
0.2
0.0
-0.2
0.2
0.4
0.6
0.8

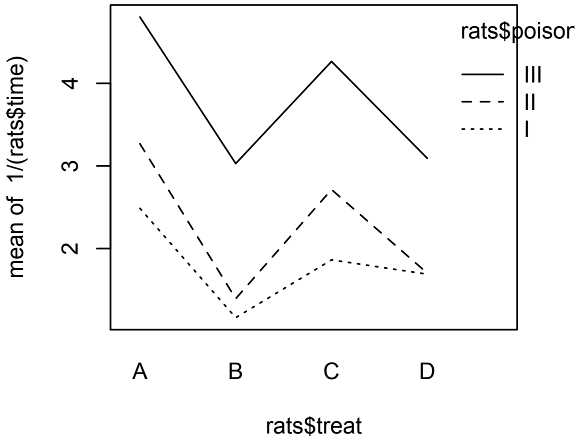

95%
log-Likelihood
λ
30
20
10
0
-10
-20
-2
-1
0
1
2

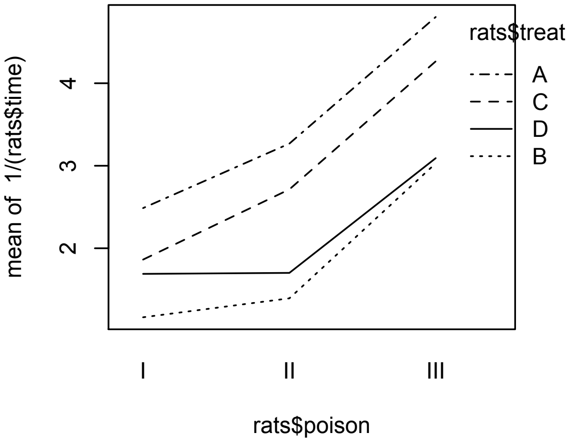


```
# try the reciprocal transformation

> g=lm(1/time ~ poison*treat, rats)

> par(mfrow=c(1,2))

> qqnorm(g$res)

> plot(g$fitted, g$res, xlab="FItted", ylab="Residuals")
```







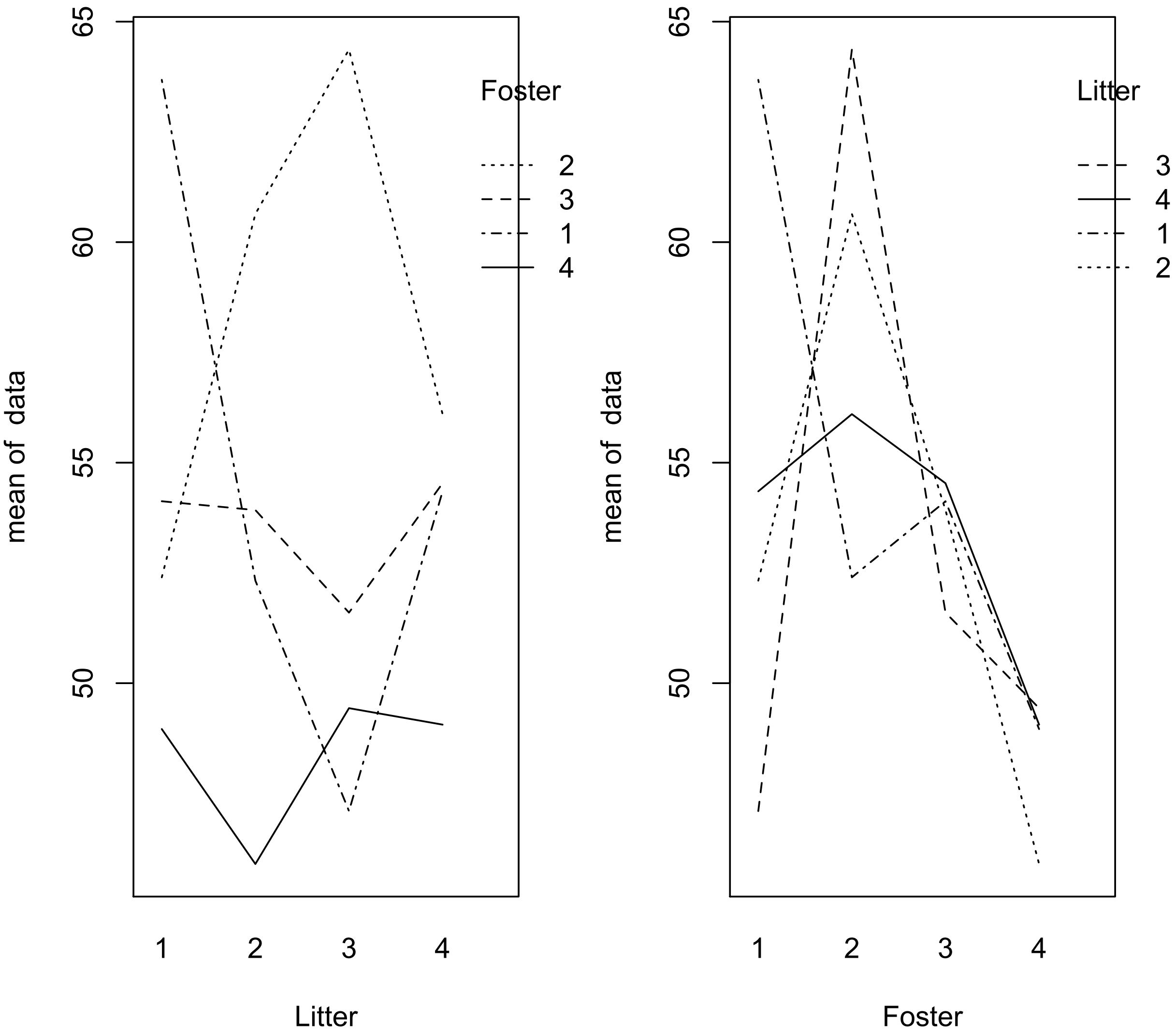


```
> anova(g)
Analysis of Variance Table

Response: 1/time
             Df Sum Sq Mean Sq F value Pr(>F)
poison        2 34.877  17.439  72.635 <2e-16 ***
treat         3 20.414   6.805  28.343 <2e-16 ***
poison:treat  6  1.571   0.262   1.090  0.387
Residuals    36  8.643   0.240
```

```
# Pairwise CIs for "treat"
> TukeyHSD(aov(1/time ~ poison + treat, data=rats),
"treat")
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = 1/time ~ poison + treat, data = rats)

$treat
          diff        lwr         upr     p adj
B-A -1.6574024 -2.1959343 -1.11887050 0.0000000
C-A -0.5721354 -1.1106673 -0.03360355 0.0335202
D-A -1.3583383 -1.8968702 -0.81980640 0.0000002
C-B  1.0852669  0.5467351  1.62379883 0.0000172
D-B  0.2990641 -0.2394678  0.83759598 0.4550931
D-C -0.7862029 -1.3247347 -0.24767096 0.0018399

> treat.CI = TukeyHSD(aov(1/time ~ poison + treat,
data=rats), "treat")$treat
>
> treat.CI[,1]-treat.CI[,2]  # half-width of the CIs
      B-A       C-A       D-A       C-B       D-B       D-C
0.5385319 0.5385319 0.5385319 0.5385319 0.5385319 0.5385319

> qtukey(0.95,4,42)*summary(g)$sigma*sqrt(2/(48/4))/sqrt(2)
[1] 0.5350869

> # Pairwise CIs for "poison"
> poison.CI=TukeyHSD(aov(1/time ~ poison + treat,
data=rats), "poison")$poison

> poison.CI[,1]-poison.CI[,2]  # half-width of the CIs
     II-I     III-I    III-II
0.4235854 0.4235854 0.4235854

> qtukey(0.95,3,42)*summary(g)$sigma*sqrt(2/(48/3))/sqrt(2)
[1] 0.4208757
```

```
### Rats Example (unBalanced Design)
> newrats=rats;

> newrats=newrats[-1,]

> anova(lm(1/time ~ poison*treat, newrats))
Analysis of Variance Table

Response: 1/time
             Df Sum Sq Mean Sq F value    Pr(>F)
poison        2 36.672 18.3358 81.0799 7.276e-14 ***

treat         3 18.567  6.1889 27.3670 2.706e-09 ***

poison:treat  6  1.980  0.3300  1.4592    0.2207

Residuals    35  7.915  0.2261

> anova(lm(1/time ~ treat*poison, newrats))
Analysis of Variance Table

Response: 1/time
             Df Sum Sq Mean Sq F value    Pr(>F)
treat         3 20.136  6.7121 29.6807 9.986e-10 ***

poison        2 35.102 17.5510 77.6094 1.362e-13 ***

treat:poison  6  1.980  0.3300  1.4592    0.2207

Residuals    35  7.915  0.2261
```

```
### Another unBalanced example
### The experiment had litters of rats born to one mother
### but  raised by another. The row factor is the genotype
### of the litter and the column factor is the genotype
### of the foster mother. There are n=61 litters, and the
### y's are the average weight of the litters in grams at
### 28 days. The design is unbalanced

> data=c(61.5, 68.2, 64,65,59.7, 55, 42, 60.2, 52.5, 61.8,
49.5, 52.7, 42, 54, 61,
+  48.2, 39.6, 60.3, 51.7, 49.3, 48, 50.8, 64.7, 61.7, 64,
62, 56.5, 59,47.2, 53,
+ 51.3, 40.5, 37, 36.3, 68, 56.3, 69.8, 67, 39.7, 46, 61.3,
55.3, 55.7, 50, 43.8,
+ 54.5, 59, 57.4, 54, 47, 59.5, 52.8, 56, 45.2, 57, 61.4,
44.8, 51.5, 53, 42, 54);

> n=matrix(c(5,3,4,5,4,5,4,2,3,3,5,3,4,3,3,5),4,4); n=t(n);
> n
     [,1] [,2] [,3] [,4]
[1,]    5    3    4    5
[2,]    4    5    4    2
[3,]    3    3    5    3
[4,]    4    3    3    5

> Foster=c(); Litter=c();
> for(i in 1:4){
+         for (j in 1:4){
+         Foster=c(Foster, rep(j, n[i,j]));
+         Litter=c(Litter, rep(i, n[i,j]));
+         }
+         }
> Litter=as.factor(Litter);
> Foster=as.factor(Foster);
>
> par(mfrow=c(1,2))
> interaction.plot(Litter, Foster, data)
> interaction.plot(Foster, Litter, data)
```


mean of data
65
60
55
50
Foster
2
3
1
4
1
2
3
4
Litter
mean of data
65
60
55
50
Litter
3
4
1
2
1
2
3
4
Foster

```
> g1=lm(data~Foster*Litter)

> g2=lm(data~Litter*Foster)

> anova(g1)
Analysis of Variance Table

Response: data
              Df  Sum Sq Mean Sq F value   Pr(>F)
Foster         3  771.61 257.202  4.7419 0.005869 **

Litter         3   63.63  21.211  0.3911 0.760004

Foster:Litter  9  824.07  91.564  1.6881 0.120053

Residuals     45 2440.82  54.240

> anova(g2)
Analysis of Variance Table

Response: data
              Df  Sum Sq Mean Sq F value   Pr(>F)
Litter         3   60.16  20.052  0.3697 0.775221

Foster         3  775.08 258.360  4.7632 0.005736 **

Litter:Foster  9  824.07  91.564  1.6881 0.120053

Residuals     45 2440.82  54.240
```