1

Multifactor Experiments

Eldho Varghese*[1]*, Seema Jaggi*[2] *and Athulya C.K*[1]*.
[1]ICAR-Central Marine Fisheries Research Institute
[2]Indian Council of Agricultural Research
eldho.varghese@icar.gov.in; seema.jaggi@icar.gov.in;athulyack28@gmail.com

1. Introduction

Multifactor experiments investigate the impact of two or more factors or input parameters on a process' output response. Factorial experiment design, or simply factorial design, is a systematic approach for articulating the procedures required to successfully run a factorial experiment. Estimating the effects of numerous parameters on a process' output with a small number of observations is crucial for process output optimization.

In multifactor experiments, the effects of changing the levels of many factors that contributed to the process outcome are investigated. Each entire trial or replication of the experiment accounted for all possible combinations of these elements' varied amounts in each trial or replication. In order to gather the most information about how input factors affect a process output, effective factorial design ensures that the fewest number of experiment runs are performed.

For example, an experiment on rooting of cuttings involving two factors, each at two levels, such as two hormones at two doses, is referred to as a 2 x 2 or a 2^2 factorial experiment. Its treatments consist of the following four possible combinations of the two levels in each of the two factors.

Treatment number	Treatment Combination	
	Hormone	Dose (ppm)
1	NAA	10
2	NAA	20
3	IBA	10
4	IBA	20

The total number of treatments in a factorial experiment is the product of the number of levels of each factor; in the 2^2 factorial example, the number of treatments is 2 x 2 = 4, in the 2^3 factorial, the number of treatments is 2 x 2 x 2 = 8. With a rise in the number of factors or the levels of each factor, the number of treatments rapidly grows. The total number of treatments in a factorial experiment with 5 clones, 4 spacing, and 3 weed-control strategies would be 5 × 4 x 3 = 60. Because of their large size, complexity, and cost, factorial experiments should not be used indiscriminately. Furthermore, it is not advisable to commit to a large experiment at the outset of a research project when multiple modest preliminary trials may yield promising results. A tree breeder, for example, has brought 30 new clones from a neighbouring country and wants to see how they behave in the local environment. Because the environment is likely to change in terms of soil fertility, moisture levels, and other things, the optimum experiment would be one in which the 30 clones are tested in a factorial experiment with other variables such as fertiliser, moisture level, and population density. When factors other than clones are introduced, however, such an experiment grows exceedingly huge. Even if only one element were introduced, such as nitrogen or fertiliser with three levels, the number of treatments would rise from 30 to 90. Financing, getting a suitable experimental area, regulating soil heterogeneity, and other issues would all be tough with such a vast experiment. As a result, a more feasible method would be to test the 30 clones in a single-factor experiment first, then utilise the results to pick a few clones for more detailed studies. For example, the initial single-factor experiment may reveal that only five clones are worthy of further investigation. These five clones might then be used in a factorial experiment with three levels of nitrogen, yielding a 15-treatment experiment rather than the 90-treatment experiment needed with 30 clones.

The'main effect' of a factor is defined as the amount of change in the process output caused by a change in the 'level' of that factor. Table 1 illustrates a simple factorial experiment with two components, each with two levels. In factorial designs, the two levels of each factor are designated by 'low' and 'high,' which are commonly symbolised by '-' and '+,' respectively.

Table 1. A Simple 2-Factorial Experiment

	A (-)	**A (+)**
B (-)	20	40
B (+)	30	52

The 'average' change in the output response as a component changes from '-' to '+' represents the principal effect of that factor. This is the average of two values in mathematics: 1) the change in output when the factor goes from low to high while the other factor remains low, and 2) the change in output when the factor goes from low to high while the other factor remains high.

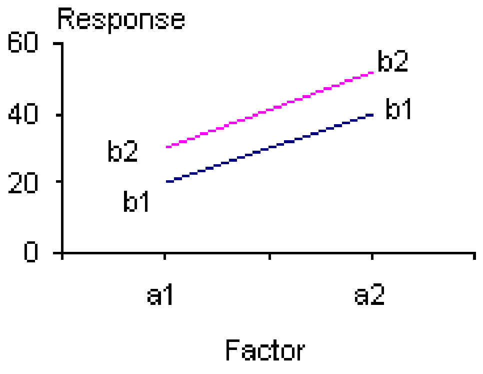
When both A and B are at their '-' level, the process output is simply 20 (lowest output), whereas when both A and B are at their '+' level, the process output is 52 (highest output).

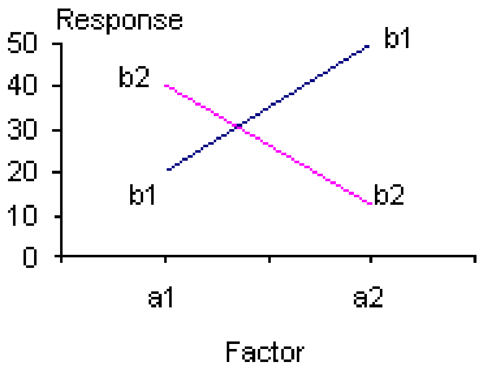
The average of the change in output response when B remains '-' as A goes from '-' to '+', or (40-20) = 20, and the change in output response when B remains '+' as A goes from '-' to '+', or (52-30) = 22 is the main effect of A. As a result, the main effect of A is equal to 21.

Similarly, B's main effect is the average change in output from '-' to'+', i.e. the average of 10 and 12, or 11. As a result, B's main effect on this process is 11. Here, it can be shown that factor A has a bigger impact on the process output, with a major effect of 21 compared to only 11 for factor B. It's worth noting that, in addition to'main effect,' elements can also cause 'interaction effects.' Changes in the process output induced by two or more factors interacting with each other are known as interaction effects. Large interactive effects might overshadow the main effects, making it all the more vital to focus on the interaction of the involved factors than to investigate them individually. In Table 1, as effects of A (B) is not same at all the levels of B (A) hence, A and B are interacting.

Interaction is defined as the failure of differences in response to changes in one factor's levels to maintain the same order and magnitude of performance across all levels of other factors, OR the factors are said to interact if the effect of one factor changes as the levels of other factor(s) change.

Graphical representation of lack of interaction between factors and interaction between factors are shown below. In case of two parallel lines, the factors are non-interacting.





If there are interactions, which is rather typical, we should design our studies so that they can be estimated and tested. It is obvious that we will not be able to accomplish so if we simply change one factor at a time. Multilevel, multifactor studies are required for this.

The basis of factorial experiments is the execution of factorial combinations and the mathematical interpretation of the process' output responses to such combinations. It gives for a better understanding of which aspects have the greatest impact on the process, allowing for improvements (or corrective actions) to be targeted towards these areas.

We may define factorial experiments as experiments in which the effects (main effects and interactions) of more then one factor are studied together. In general if there are 'n' factors, say, F_1, F_2,..., F_n and i^{th} factor has s_i levels, i=1,...,n, then total number of treatment combinations is $\prod_{i=1}^{n} s_i$. Factorial experiments are of two types.

Asymmetrical factorial experiments are those in which all of the factors have the same number of levels, i.e. all s_i's are equal, whereas symmetrical factorial experiments are those in which at least two of the si's are different. Factorial experiments allow researchers to investigate both the individual effects of each element as well as their interactions. They also offer the benefit of conserving experimental resources. When experiments are carried out factor by factor, substantially more resources are needed to achieve the same precision as when they are carried out in factorial studies.

Confounding in Factorial Experiments

When the number of factors and/or their levels increase, the number of treatment combinations increases rapidly, and all of these treatment combinations cannot be accommodated in a single homogenous block. A 2^5 factorial, for example, would contain 32 treatment combinations, and 32 plot blocks are rather large to maintain homogeneity within them. For developing trials with a high number of treatments, a new technique is required. One such device is to use blocks that are smaller than the number of treatments and replicate them multiple times. After that, the treatment combinations are separated into groups equal to the number of blocks in each replication.The different groups of treatments are allocated to the blocks.

There are a number of ways to divide the treatments into as many groups as the number of blocks in each replication. It is well known that in a factorial experiment with two levels of each factor, the treatment combinations are separated into two groups in order to achieve the interaction contrast. Such two groups, each comprising half of the entire number of treatments, can be used to create the contrasts of two blocks, each holding half of the total number of treatments. The interaction contrast and the contrast between the two block totals are both generated by the same function in this situation. As a result, they are mixed and unable to be separated. To put it another way, the interaction has been confounded by the blocks. Because of the reduced block size, the interaction confounded has been lost, but the other interactions and main effects can now be estimated with greater precision. Confounding is a technique for reducing block size by using one or more interaction contrasts that are identical to block contrasts. Only higher order interactions, i.e.

interactions involving three or more components, should be confounded because their loss is insignificant. As an experimenter is generally interested in main effects and two factor interactions, these should not be confounded as far as possible.

When there are two or more replications, complete confounding occurs when the same set of interactions is confounded in all replications, while partial confounding occurs when different sets of interactions are confounded in various replications. Complete confounding eliminates all information about confounded interactions. However, in partial confounding, the confounded interactions can be recovered from non-confounded replications.

Fractional Factorial

When the number of components to be examined in a factorial experiment increases, the total number of factorial treatments may become too high to test simultaneously in a single experiment. An experimental design that permits only a subset of the entire number of treatments to be tested is a logical alternative. The fractional factorial is a design that is particularly well suited to studies with a large number of variables. It allows you to pick and test a subset of the total number of factorial treatment options in a systematic fashion. However, there is a loss of information on some pre-selected effects in exchange. Although this information loss can be significant in studies with only one or two factors, it becomes more manageable when there are many. The number of interaction effects increases rapidly with the number of factors involved, which allows flexibility in the choice of the particular effects to be sacrificed. In fact, in cases where some specific effects are known beforehand to be small or unimportant, use of the fractional factorial results in minimal loss of information.

High order interactions, such as four-factor or five-factor interactions, and even three-factor interactions, are generally sacrificed when using the fractional factorial in practise. Unless the researcher has prior evidence to indicate otherwise, a set of treatments should be chosen to be evaluated in almost all cases so that all main effects and two-factor interactions may be approximated.

The fractional factorial is used in forestry research in exploratory experiments where the main goal is to study the relationships between components. Fractional factorials that sacrifice only interactions involving more than two components are the most suited fractional factorials for such experiments.

With the fractional factorial, the number of effects that can be measured decreases rapidly with the reduction in the number of treatments to be tested. When the number of effects to be measured is large, the number of treatments to be tested may still be too great, even when fractional factorial is used. In such circumstances, limiting the number of replications can help reduce the size of the experiment even further. Although fractional factorial without replication is not commonly used in forestry studies, when it is used in exploratory trials, the number of replications necessary can be decreased to a bare minimum.

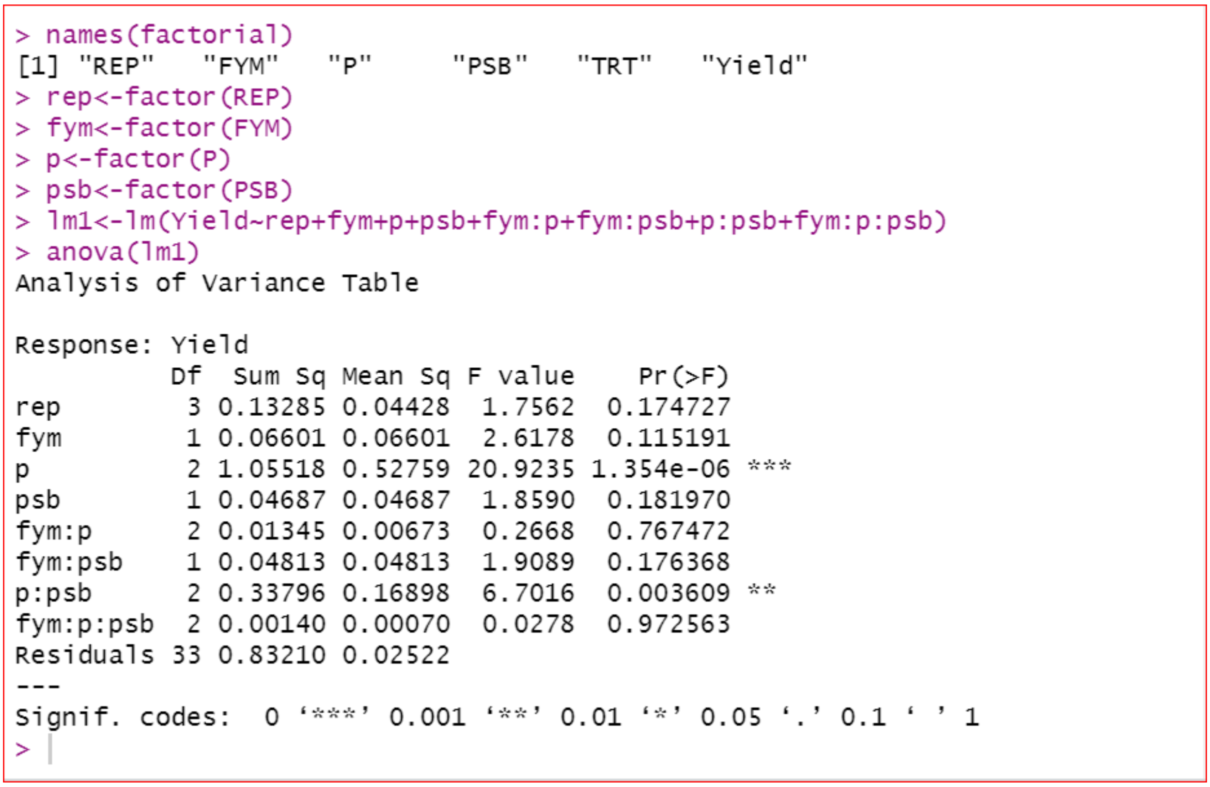
Another advantage of fractional factorial is that it allows for smaller blocks because it does not require each block to contain all of the treatments to be evaluated. The homogeneity of experimental units within a block can be increased in this way. However, in addition to the information already lost due to the reduction in the number of treatments, a reduction in block size results in a loss of information.

Analysis using R

Sample data taken from Design Resource Server https://drs.icar.gov.in/Analysis%20of%20data/Analysis%20of%20Data.html

(steps are similar for partially confounded factorial experiments)

```
attach(factorial)
names(factorial)
rep<-factor(REP)
fym<-factor(FYM)
p<-factor(P)
psb<-factor(PSB)
lm1<-lm(Yield~rep+fym+p+psb+fym:p+fym:psb+p:psb+fym:p:psb)
anova(lm1)
```



```
> names(factorial)
[1] "REP"   "FYM"   "P"     "PSB"   "TRT"   "Yield"
> rep<-factor(REP)
> fym<-factor(FYM)
> p<-factor(P)
> psb<-factor(PSB)
> lm1<-lm(Yield~rep+fym+p+psb+fym:p+fym:psb+p:psb+fym:p:psb)
> anova(lm1)
Analysis of Variance Table

Response: Yield
          Df  Sum Sq Mean Sq F value    Pr(>F)    
rep        3 0.13285 0.04428  1.7562  0.174727    
fym        1 0.06601 0.06601  2.6178  0.115191    
p          2 1.05518 0.52759 20.9235 1.354e-06 ***
psb        1 0.04687 0.04687  1.8590  0.181970    
fym:p      2 0.01345 0.00673  0.2668  0.767472    
fym:psb    1 0.04813 0.04813  1.9089  0.176368    
p:psb      2 0.33796 0.16898  6.7016  0.003609 ** 
fym:p:psb  2 0.00140 0.00070  0.0278  0.972563    
Residuals 33 0.83210 0.02522                      
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> 
```

# To provide pairwise comparison, need to install the package lsmeans or emmeans

install.packages("lsmeans")

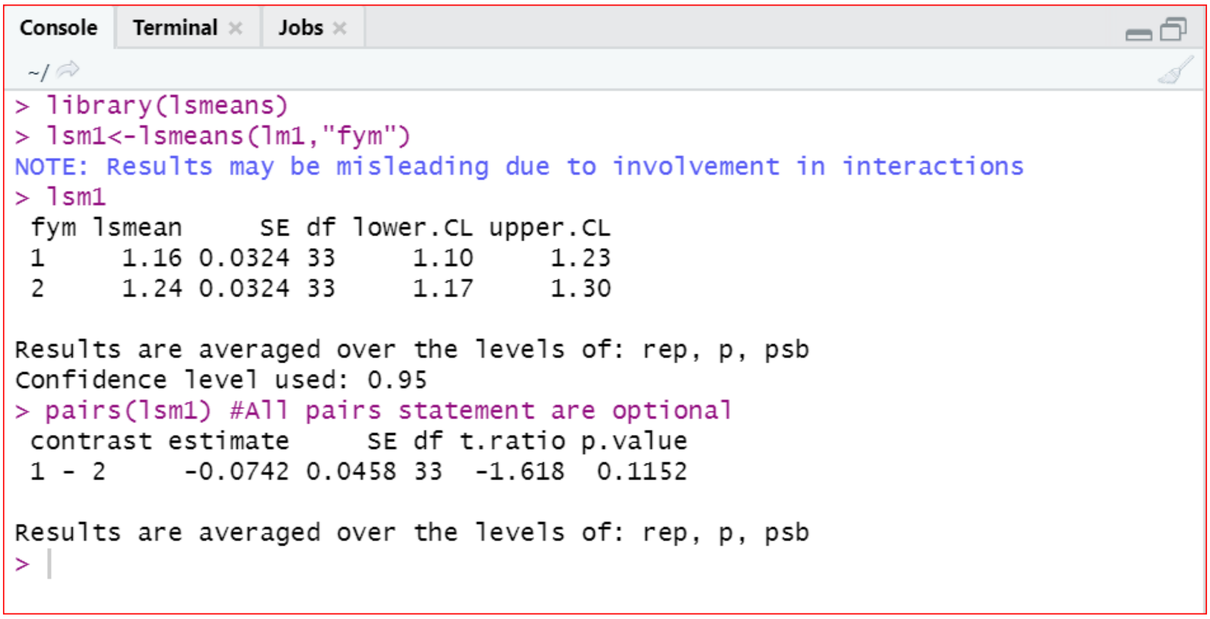
library(lsmeans)

lsm1<-lsmeans(lm1,"fym")

lsm1

#All pairs statement are optional#

pairs(lsm1)

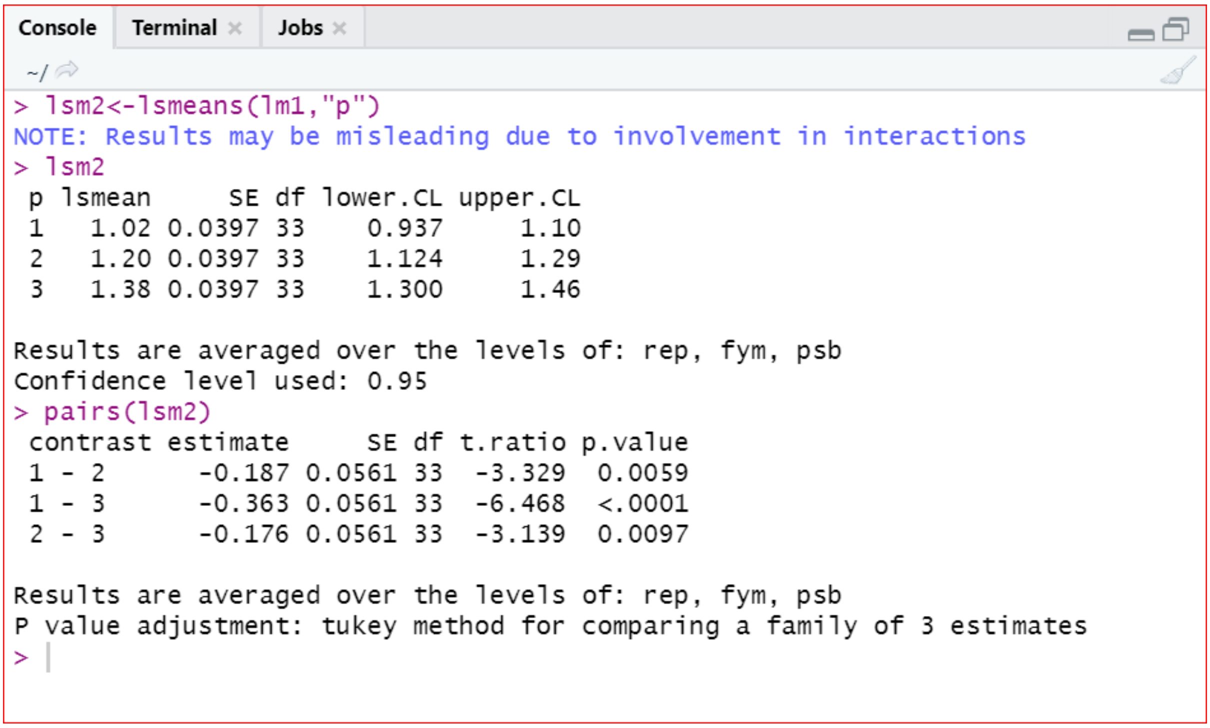


```
> library(lsmeans)
> lsm1<-lsmeans(lm1,"fym")
NOTE: Results may be misleading due to involvement in interactions
> lsm1
 fym lsmean     SE df lower.CL upper.CL
 1     1.16 0.0324 33     1.10     1.23
 2     1.24 0.0324 33     1.17     1.30

Results are averaged over the levels of: rep, p, psb 
Confidence level used: 0.95 
> pairs(lsm1) #All pairs statement are optional
 contrast estimate     SE df t.ratio p.value
 1 - 2     -0.0742 0.0458 33  -1.618  0.1152

Results are averaged over the levels of: rep, p, psb 
> 
```

lsm2<-lsmeans(lm1,"p")

lsm2

pairs(lsm2)

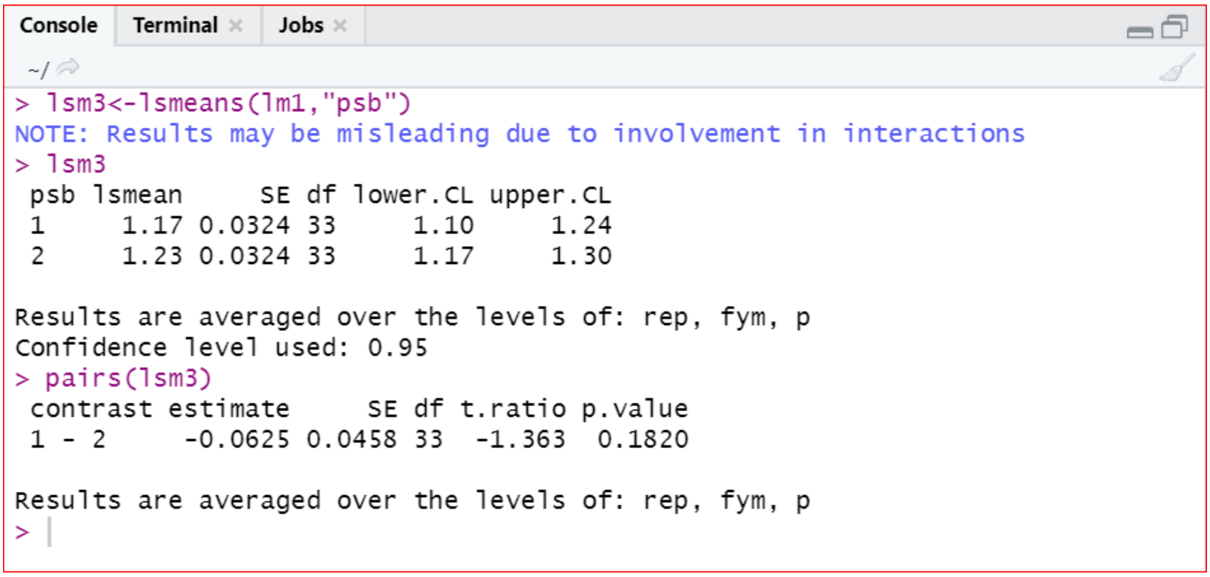


```
> lsm2<-lsmeans(lm1,"p")
NOTE: Results may be misleading due to involvement in interactions
> lsm2
 p lsmean     SE df lower.CL upper.CL
 1   1.02 0.0397 33    0.937     1.10
 2   1.20 0.0397 33    1.124     1.29
 3   1.38 0.0397 33    1.300     1.46

Results are averaged over the levels of: rep, fym, psb 
Confidence level used: 0.95 
> pairs(lsm2)
 contrast estimate     SE df t.ratio p.value
 1 - 2      -0.187 0.0561 33  -3.329  0.0059
 1 - 3      -0.363 0.0561 33  -6.468  <.0001
 2 - 3      -0.176 0.0561 33  -3.139  0.0097

Results are averaged over the levels of: rep, fym, psb 
P value adjustment: tukey method for comparing a family of 3 estimates 
> 
```

lsm3<-lsmeans(lm1,"psb")

lsm3

pairs(lsm3)

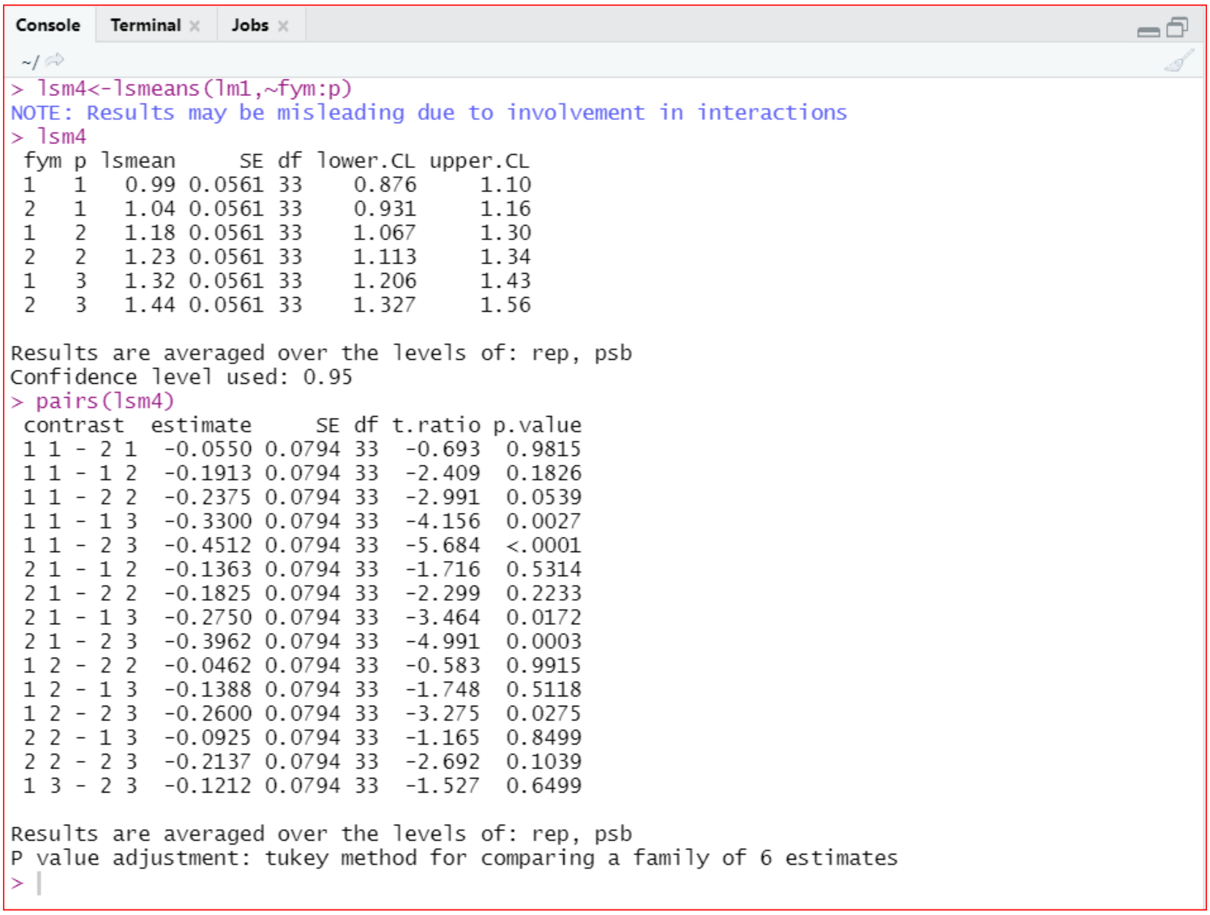


```
> lsm3<-lsmeans(lm1,"psb")
NOTE: Results may be misleading due to involvement in interactions
> lsm3
 psb lsmean     SE df lower.CL upper.CL
 1     1.17 0.0324 33     1.10     1.24
 2     1.23 0.0324 33     1.17     1.30

Results are averaged over the levels of: rep, fym, p 
Confidence level used: 0.95 
> pairs(lsm3)
 contrast estimate     SE df t.ratio p.value
 1 - 2     -0.0625 0.0458 33  -1.363  0.1820

Results are averaged over the levels of: rep, fym, p 
> 
```

lsm4<-lsmeans(lm1,~fym:p)

lsm4

pairs(lsm4)

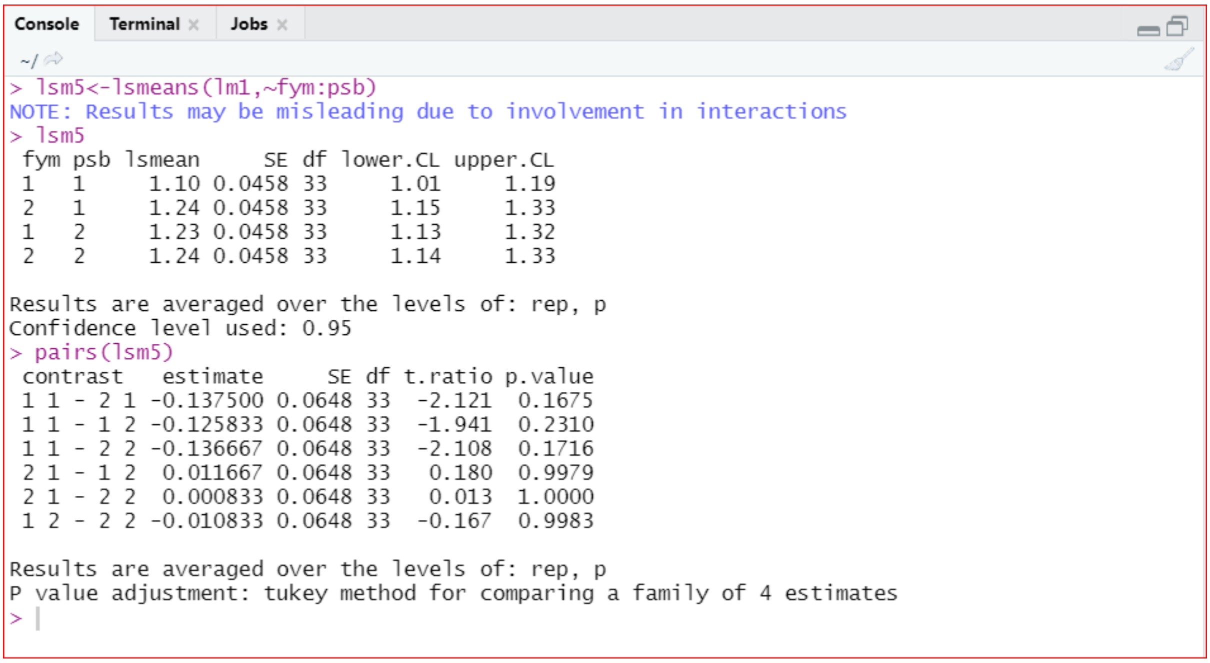


```
> lsm4<-lsmeans(lm1,~fym:p)
NOTE: Results may be misleading due to involvement in interactions
> lsm4
 fym p lsmean     SE df lower.CL upper.CL
 1   1   0.99 0.0561 33    0.876     1.10
 2   1   1.04 0.0561 33    0.931     1.16
 1   2   1.18 0.0561 33    1.067     1.30
 2   2   1.23 0.0561 33    1.113     1.34
 1   3   1.32 0.0561 33    1.206     1.43
 2   3   1.44 0.0561 33    1.327     1.56

Results are averaged over the levels of: rep, psb
Confidence level used: 0.95
> pairs(lsm4)
 contrast  estimate     SE df t.ratio p.value
 1 1 - 2 1  -0.0550 0.0794 33  -0.693  0.9815
 1 1 - 1 2  -0.1913 0.0794 33  -2.409  0.1826
 1 1 - 2 2  -0.2375 0.0794 33  -2.991  0.0539
 1 1 - 1 3  -0.3300 0.0794 33  -4.156  0.0027
 1 1 - 2 3  -0.4512 0.0794 33  -5.684  <.0001
 2 1 - 1 2  -0.1363 0.0794 33  -1.716  0.5314
 2 1 - 2 2  -0.1825 0.0794 33  -2.299  0.2233
 2 1 - 1 3  -0.2750 0.0794 33  -3.464  0.0172
 2 1 - 2 3  -0.3962 0.0794 33  -4.991  0.0003
 1 2 - 2 2  -0.0462 0.0794 33  -0.583  0.9915
 1 2 - 1 3  -0.1388 0.0794 33  -1.748  0.5118
 1 2 - 2 3  -0.2600 0.0794 33  -3.275  0.0275
 2 2 - 1 3  -0.0925 0.0794 33  -1.165  0.8499
 2 2 - 2 3  -0.2137 0.0794 33  -2.692  0.1039
 1 3 - 2 3  -0.1212 0.0794 33  -1.527  0.6499

Results are averaged over the levels of: rep, psb
P value adjustment: tukey method for comparing a family of 6 estimates
> 
```

lsm5<-lsmeans(lm1,~fym:psb)

lsm5

pairs(lsm5)

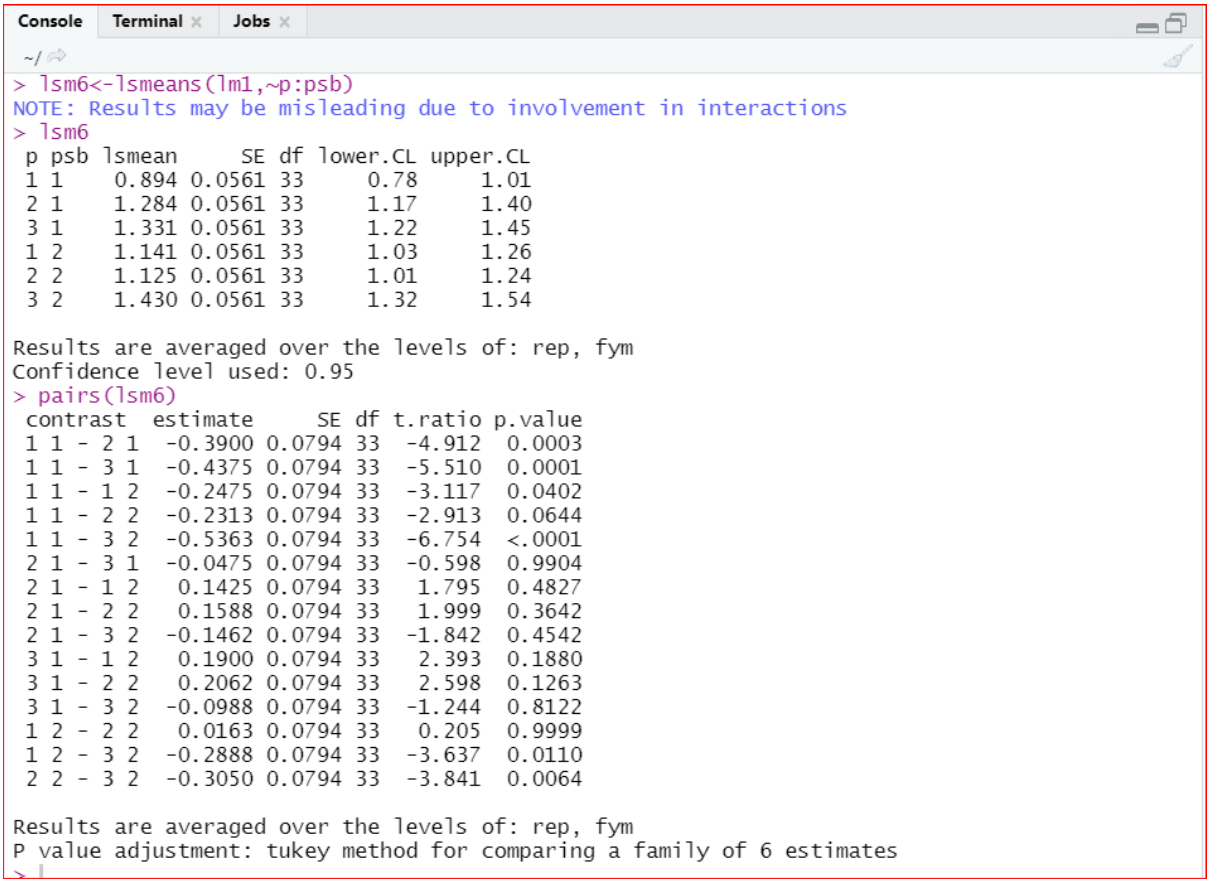


```
> lsm5<-lsmeans(lm1,~fym:psb)
NOTE: Results may be misleading due to involvement in interactions
> lsm5
 fym psb lsmean     SE df lower.CL upper.CL
 1   1     1.10 0.0458 33     1.01     1.19
 2   1     1.24 0.0458 33     1.15     1.33
 1   2     1.23 0.0458 33     1.13     1.32
 2   2     1.24 0.0458 33     1.14     1.33

Results are averaged over the levels of: rep, p
Confidence level used: 0.95
> pairs(lsm5)
 contrast   estimate     SE df t.ratio p.value
 1 1 - 2 1 -0.137500 0.0648 33  -2.121  0.1675
 1 1 - 1 2 -0.125833 0.0648 33  -1.941  0.2310
 1 1 - 2 2 -0.136667 0.0648 33  -2.108  0.1716
 2 1 - 1 2  0.011667 0.0648 33   0.180  0.9979
 2 1 - 2 2  0.000833 0.0648 33   0.013  1.0000
 1 2 - 2 2 -0.010833 0.0648 33  -0.167  0.9983

Results are averaged over the levels of: rep, p
P value adjustment: tukey method for comparing a family of 4 estimates
> 
```

lsm6<-lsmeans(lm1,~p:psb)

lsm6

pairs(lsm6)

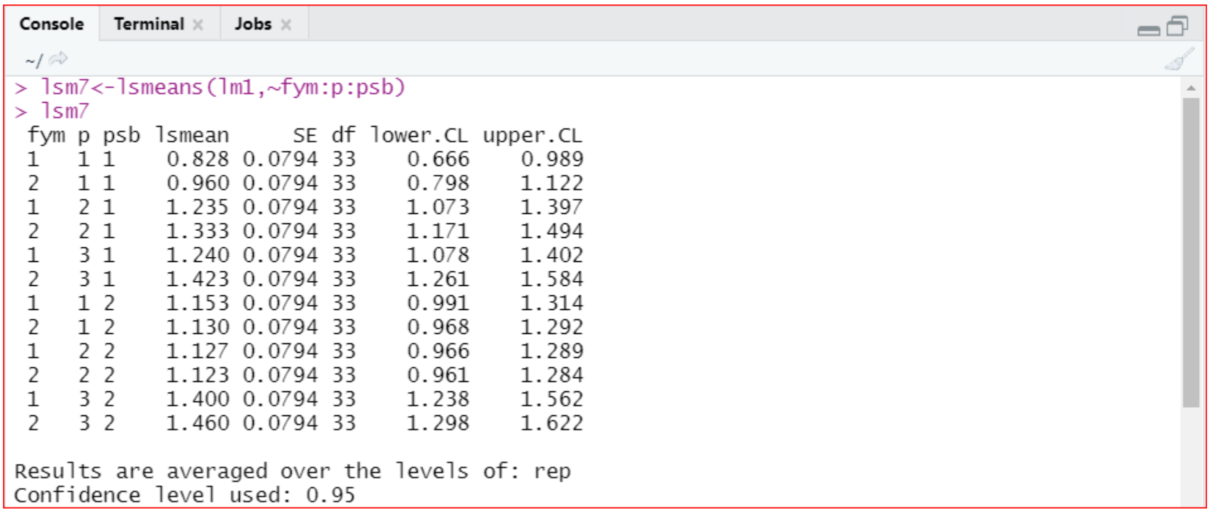


```
> lsm6<-lsmeans(lm1,~p:psb)
NOTE: Results may be misleading due to involvement in interactions
> lsm6
 p psb lsmean     SE df lower.CL upper.CL
 1 1    0.894 0.0561 33     0.78     1.01
 2 1    1.284 0.0561 33     1.17     1.40
 3 1    1.331 0.0561 33     1.22     1.45
 1 2    1.141 0.0561 33     1.03     1.26
 2 2    1.125 0.0561 33     1.01     1.24
 3 2    1.430 0.0561 33     1.32     1.54

Results are averaged over the levels of: rep, fym
Confidence level used: 0.95
> pairs(lsm6)
 contrast  estimate     SE df t.ratio p.value
 1 1 - 2 1  -0.3900 0.0794 33  -4.912  0.0003
 1 1 - 3 1  -0.4375 0.0794 33  -5.510  0.0001
 1 1 - 1 2  -0.2475 0.0794 33  -3.117  0.0402
 1 1 - 2 2  -0.2313 0.0794 33  -2.913  0.0644
 1 1 - 3 2  -0.5363 0.0794 33  -6.754  <.0001
 2 1 - 3 1  -0.0475 0.0794 33  -0.598  0.9904
 2 1 - 1 2   0.1425 0.0794 33   1.795  0.4827
 2 1 - 2 2   0.1588 0.0794 33   1.999  0.3642
 2 1 - 3 2  -0.1462 0.0794 33  -1.842  0.4542
 3 1 - 1 2   0.1900 0.0794 33   2.393  0.1880
 3 1 - 2 2   0.2062 0.0794 33   2.598  0.1263
 3 1 - 3 2  -0.0988 0.0794 33  -1.244  0.8122
 1 2 - 2 2   0.0163 0.0794 33   0.205  0.9999
 1 2 - 3 2  -0.2888 0.0794 33  -3.637  0.0110
 2 2 - 3 2  -0.3050 0.0794 33  -3.841  0.0064

Results are averaged over the levels of: rep, fym
P value adjustment: tukey method for comparing a family of 6 estimates
>
```

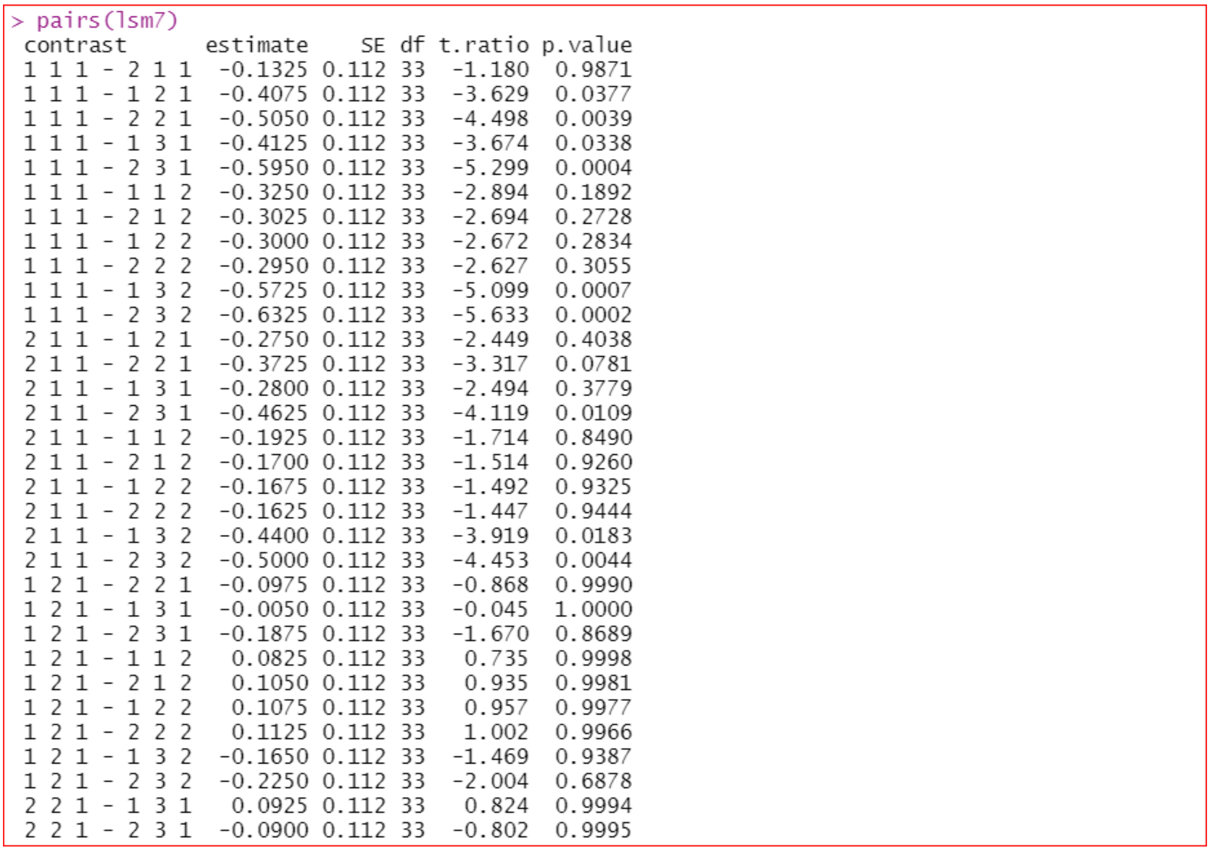
lsm7<-lsmeans(lm1,~fym:p:psb)

lsm7

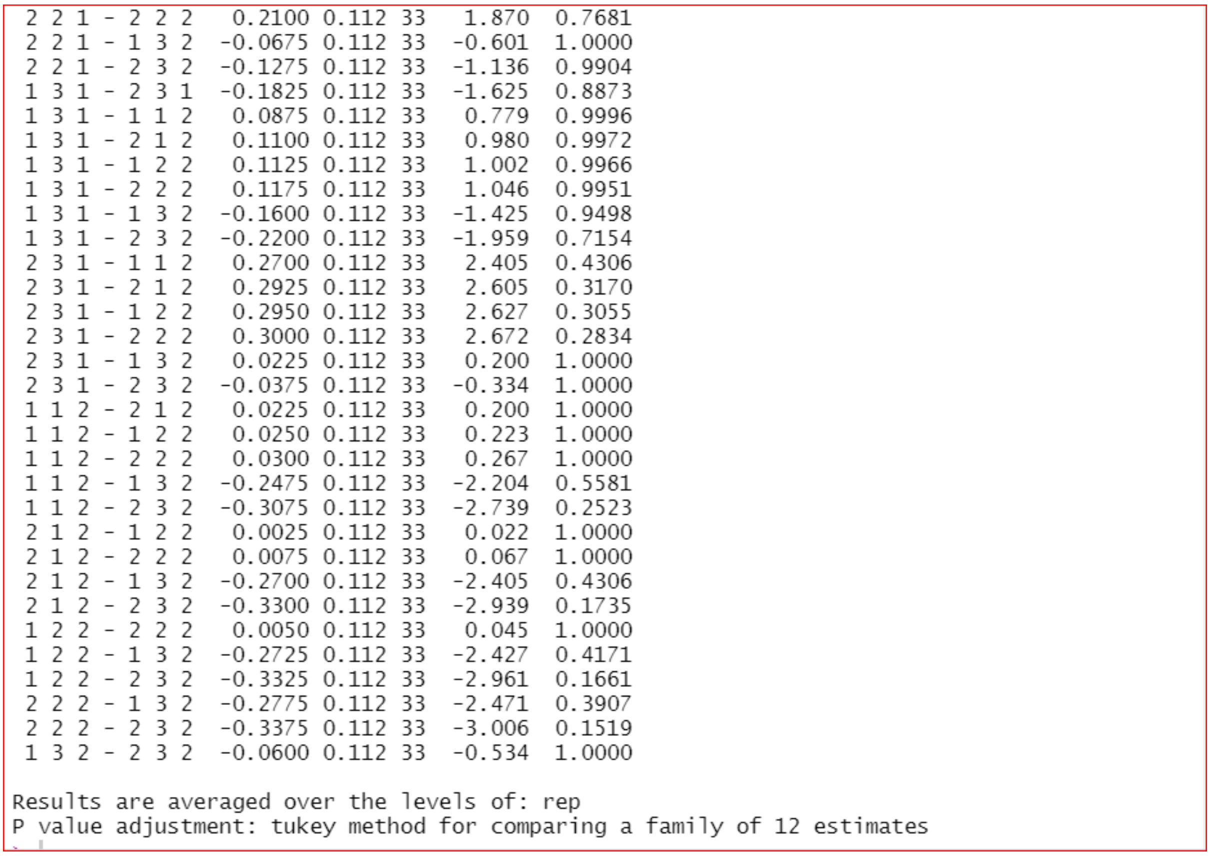
pairs(lsm7)



```
> lsm7<-lsmeans(lm1,~fym:p:psb)
> lsm7
 fym p psb lsmean     SE df lower.CL upper.CL
 1   1 1    0.828 0.0794 33    0.666    0.989
 2   1 1    0.960 0.0794 33    0.798    1.122
 1   2 1    1.235 0.0794 33    1.073    1.397
 2   2 1    1.333 0.0794 33    1.171    1.494
 1   3 1    1.240 0.0794 33    1.078    1.402
 2   3 1    1.423 0.0794 33    1.261    1.584
 1   1 2    1.153 0.0794 33    0.991    1.314
 2   1 2    1.130 0.0794 33    0.968    1.292
 1   2 2    1.127 0.0794 33    0.966    1.289
 2   2 2    1.123 0.0794 33    0.961    1.284
 1   3 2    1.400 0.0794 33    1.238    1.562
 2   3 2    1.460 0.0794 33    1.298    1.622

Results are averaged over the levels of: rep
Confidence level used: 0.95
```



```
> pairs(lsm7)
 contrast      estimate    SE df t.ratio p.value
 1 1 1 - 2 1 1  -0.1325 0.112 33  -1.180  0.9871
 1 1 1 - 1 2 1  -0.4075 0.112 33  -3.629  0.0377
 1 1 1 - 2 2 1  -0.5050 0.112 33  -4.498  0.0039
 1 1 1 - 1 3 1  -0.4125 0.112 33  -3.674  0.0338
 1 1 1 - 2 3 1  -0.5950 0.112 33  -5.299  0.0004
 1 1 1 - 1 1 2  -0.3250 0.112 33  -2.894  0.1892
 1 1 1 - 2 1 2  -0.3025 0.112 33  -2.694  0.2728
 1 1 1 - 1 2 2  -0.3000 0.112 33  -2.672  0.2834
 1 1 1 - 2 2 2  -0.2950 0.112 33  -2.627  0.3055
 1 1 1 - 1 3 2  -0.5725 0.112 33  -5.099  0.0007
 1 1 1 - 2 3 2  -0.6325 0.112 33  -5.633  0.0002
 2 1 1 - 1 2 1  -0.2750 0.112 33  -2.449  0.4038
 2 1 1 - 2 2 1  -0.3725 0.112 33  -3.317  0.0781
 2 1 1 - 1 3 1  -0.2800 0.112 33  -2.494  0.3779
 2 1 1 - 2 3 1  -0.4625 0.112 33  -4.119  0.0109
 2 1 1 - 1 1 2  -0.1925 0.112 33  -1.714  0.8490
 2 1 1 - 2 1 2  -0.1700 0.112 33  -1.514  0.9260
 2 1 1 - 1 2 2  -0.1675 0.112 33  -1.492  0.9325
 2 1 1 - 2 2 2  -0.1625 0.112 33  -1.447  0.9444
 2 1 1 - 1 3 2  -0.4400 0.112 33  -3.919  0.0183
 2 1 1 - 2 3 2  -0.5000 0.112 33  -4.453  0.0044
 1 2 1 - 2 2 1  -0.0975 0.112 33  -0.868  0.9990
 1 2 1 - 1 3 1  -0.0050 0.112 33  -0.045  1.0000
 1 2 1 - 2 3 1  -0.1875 0.112 33  -1.670  0.8689
 1 2 1 - 1 1 2   0.0825 0.112 33   0.735  0.9998
 1 2 1 - 2 1 2   0.1050 0.112 33   0.935  0.9981
 1 2 1 - 1 2 2   0.1075 0.112 33   0.957  0.9977
 1 2 1 - 2 2 2   0.1125 0.112 33   1.002  0.9966
 1 2 1 - 1 3 2  -0.1650 0.112 33  -1.469  0.9387
 1 2 1 - 2 3 2  -0.2250 0.112 33  -2.004  0.6878
 2 2 1 - 1 3 1   0.0925 0.112 33   0.824  0.9994
 2 2 1 - 2 3 1  -0.0900 0.112 33  -0.802  0.9995
```

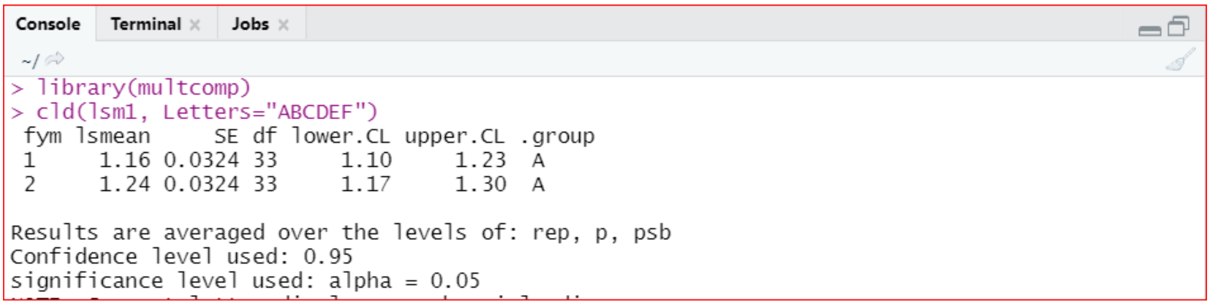


```
 2 2 1 - 2 2 2   0.2100 0.112 33   1.870  0.7681
 2 2 1 - 1 3 2  -0.0675 0.112 33  -0.601  1.0000
 2 2 1 - 2 3 2  -0.1275 0.112 33  -1.136  0.9904
 1 3 1 - 2 3 1  -0.1825 0.112 33  -1.625  0.8873
 1 3 1 - 1 1 2   0.0875 0.112 33   0.779  0.9996
 1 3 1 - 2 1 2   0.1100 0.112 33   0.980  0.9972
 1 3 1 - 1 2 2   0.1125 0.112 33   1.002  0.9966
 1 3 1 - 2 2 2   0.1175 0.112 33   1.046  0.9951
 1 3 1 - 1 3 2  -0.1600 0.112 33  -1.425  0.9498
 1 3 1 - 2 3 2  -0.2200 0.112 33  -1.959  0.7154
 2 3 1 - 1 1 2   0.2700 0.112 33   2.405  0.4306
 2 3 1 - 2 1 2   0.2925 0.112 33   2.605  0.3170
 2 3 1 - 1 2 2   0.2950 0.112 33   2.627  0.3055
 2 3 1 - 2 2 2   0.3000 0.112 33   2.672  0.2834
 2 3 1 - 1 3 2   0.0225 0.112 33   0.200  1.0000
 2 3 1 - 2 3 2  -0.0375 0.112 33  -0.334  1.0000
 1 1 2 - 2 1 2   0.0225 0.112 33   0.200  1.0000
 1 1 2 - 1 2 2   0.0250 0.112 33   0.223  1.0000
 1 1 2 - 2 2 2   0.0300 0.112 33   0.267  1.0000
 1 1 2 - 1 3 2  -0.2475 0.112 33  -2.204  0.5581
 1 1 2 - 2 3 2  -0.3075 0.112 33  -2.739  0.2523
 2 1 2 - 1 2 2   0.0025 0.112 33   0.022  1.0000
 2 1 2 - 2 2 2   0.0075 0.112 33   0.067  1.0000
 2 1 2 - 1 3 2  -0.2700 0.112 33  -2.405  0.4306
 2 1 2 - 2 3 2  -0.3300 0.112 33  -2.939  0.1735
 1 2 2 - 2 2 2   0.0050 0.112 33   0.045  1.0000
 1 2 2 - 1 3 2  -0.2725 0.112 33  -2.427  0.4171
 1 2 2 - 2 3 2  -0.3325 0.112 33  -2.961  0.1661
 2 2 2 - 1 3 2  -0.2775 0.112 33  -2.471  0.3907
 2 2 2 - 2 3 2  -0.3375 0.112 33  -3.006  0.1519
 1 3 2 - 2 3 2  -0.0600 0.112 33  -0.534  1.0000

Results are averaged over the levels of: rep
P value adjustment: tukey method for comparing a family of 12 estimates
```

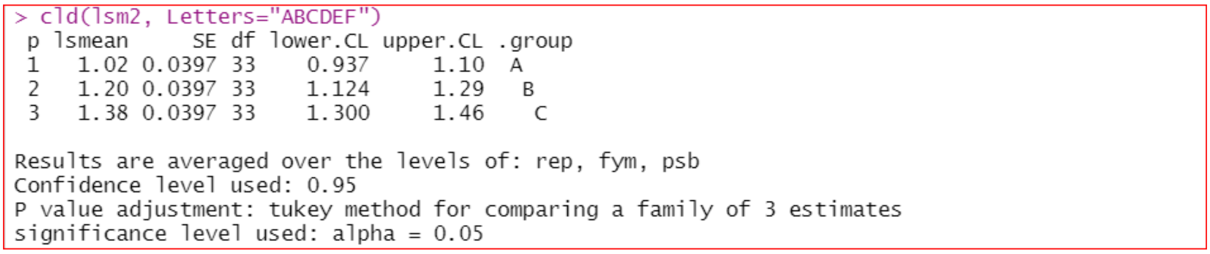
# To provide letter grouping, need to install the package multcomp#

install.packages("multcomp")

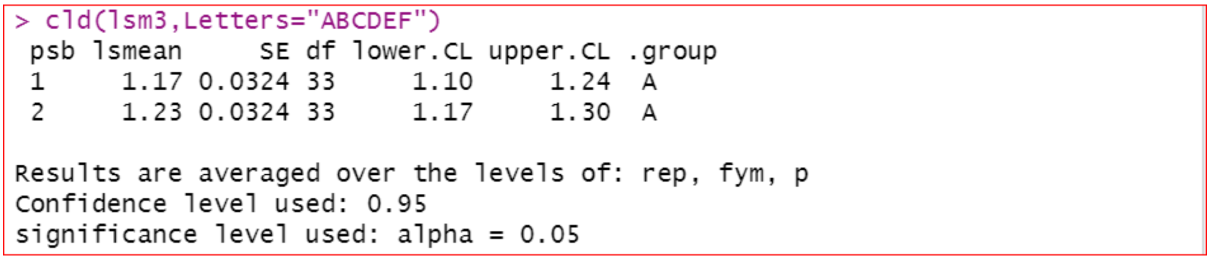
```
library(multcomp)

cld(lsm1, Letters="ABCDEF")

cld(lsm2, Letters="ABCDEF")

cld(lsm3,Letters="ABCDEF")

cld(lsm4,Letters="ABCDEFGHI")

cld(lsm5,Letters="ABCDEFGHI")

cld(lsm6,Letters="ABCDEFGHI")

cld(lsm7,Letters="ABCDEFGHIJKLM")

detach(factorial)
```



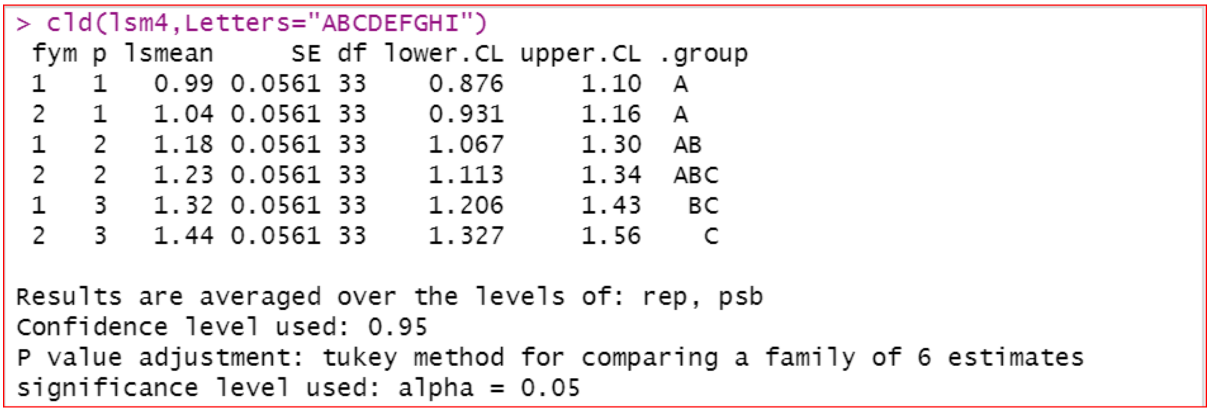
```
> library(multcomp)
> cld(lsm1, Letters="ABCDEF")
 fym lsmean     SE df lower.CL upper.CL .group
 1     1.16 0.0324 33     1.10     1.23  A    
 2     1.24 0.0324 33     1.17     1.30  A    

Results are averaged over the levels of: rep, p, psb 
Confidence level used: 0.95 
significance level used: alpha = 0.05 
```



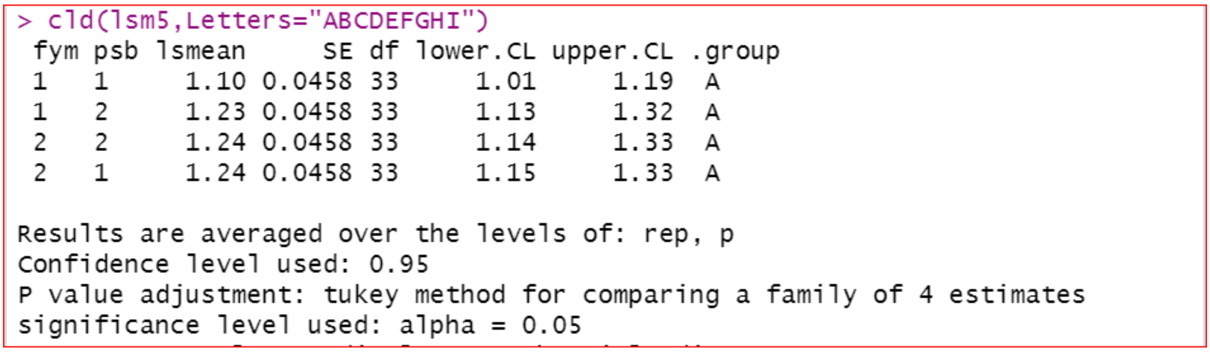
```
> cld(lsm2, Letters="ABCDEF")
 p lsmean     SE df lower.CL upper.CL .group
 1   1.02 0.0397 33    0.937     1.10  A    
 2   1.20 0.0397 33    1.124     1.29   B   
 3   1.38 0.0397 33    1.300     1.46    C  

Results are averaged over the levels of: rep, fym, psb 
Confidence level used: 0.95 
P value adjustment: tukey method for comparing a family of 3 estimates 
significance level used: alpha = 0.05 
```



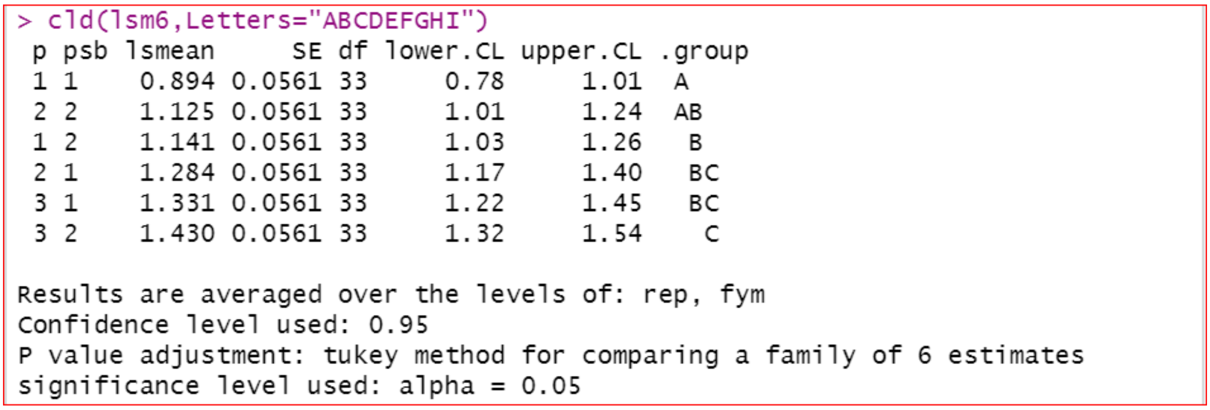
```
> cld(lsm3,Letters="ABCDEF")
 psb lsmean     SE df lower.CL upper.CL .group
 1     1.17 0.0324 33     1.10     1.24  A    
 2     1.23 0.0324 33     1.17     1.30  A    

Results are averaged over the levels of: rep, fym, p 
Confidence level used: 0.95 
significance level used: alpha = 0.05 
```



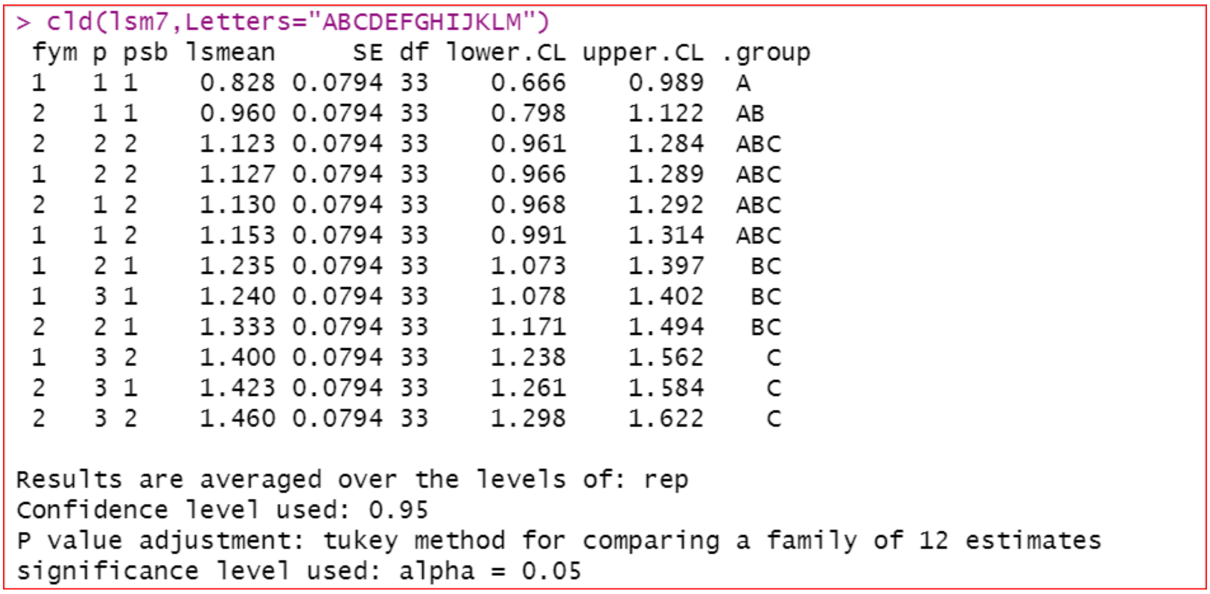
```
> cld(lsm4,Letters="ABCDEFGHI")
 fym p lsmean     SE df lower.CL upper.CL .group
 1   1   0.99 0.0561 33    0.876     1.10  A    
 2   1   1.04 0.0561 33    0.931     1.16  A    
 1   2   1.18 0.0561 33    1.067     1.30  AB   
 2   2   1.23 0.0561 33    1.113     1.34  ABC  
 1   3   1.32 0.0561 33    1.206     1.43   BC  
 2   3   1.44 0.0561 33    1.327     1.56    C  

Results are averaged over the levels of: rep, psb 
Confidence level used: 0.95 
P value adjustment: tukey method for comparing a family of 6 estimates 
significance level used: alpha = 0.05 
```



```
> cld(lsm5,Letters="ABCDEFGHI")
 fym psb lsmean     SE df lower.CL upper.CL .group
 1   1     1.10 0.0458 33     1.01     1.19  A    
 1   2     1.23 0.0458 33     1.13     1.32  A    
 2   2     1.24 0.0458 33     1.14     1.33  A    
 2   1     1.24 0.0458 33     1.15     1.33  A    

Results are averaged over the levels of: rep, p 
Confidence level used: 0.95 
P value adjustment: tukey method for comparing a family of 4 estimates 
significance level used: alpha = 0.05 
```



```
> cld(lsm6,Letters="ABCDEFGHI")
 p psb lsmean     SE df lower.CL upper.CL .group
 1 1    0.894 0.0561 33     0.78     1.01  A    
 2 2    1.125 0.0561 33     1.01     1.24  AB   
 1 2    1.141 0.0561 33     1.03     1.26   B   
 2 1    1.284 0.0561 33     1.17     1.40   BC  
 3 1    1.331 0.0561 33     1.22     1.45   BC  
 3 2    1.430 0.0561 33     1.32     1.54    C  

Results are averaged over the levels of: rep, fym 
Confidence level used: 0.95 
P value adjustment: tukey method for comparing a family of 6 estimates 
significance level used: alpha = 0.05 
```



```
> cld(lsm7,Letters="ABCDEFGHIJKLM")
 fym p psb lsmean     SE df lower.CL upper.CL .group
 1   1 1    0.828 0.0794 33    0.666    0.989  A    
 2   1 1    0.960 0.0794 33    0.798    1.122  AB   
 2   2 2    1.123 0.0794 33    0.961    1.284  ABC  
 1   2 2    1.127 0.0794 33    0.966    1.289  ABC  
 2   1 2    1.130 0.0794 33    0.968    1.292  ABC  
 1   1 2    1.153 0.0794 33    0.991    1.314  ABC  
 1   2 1    1.235 0.0794 33    1.073    1.397   BC  
 1   3 1    1.240 0.0794 33    1.078    1.402   BC  
 2   2 1    1.333 0.0794 33    1.171    1.494   BC  
 1   3 2    1.400 0.0794 33    1.238    1.562    C  
 2   3 1    1.423 0.0794 33    1.261    1.584    C  
 2   3 2    1.460 0.0794 33    1.298    1.622    C  

Results are averaged over the levels of: rep 
Confidence level used: 0.95 
P value adjustment: tukey method for comparing a family of 12 estimates 
significance level used: alpha = 0.05 
```

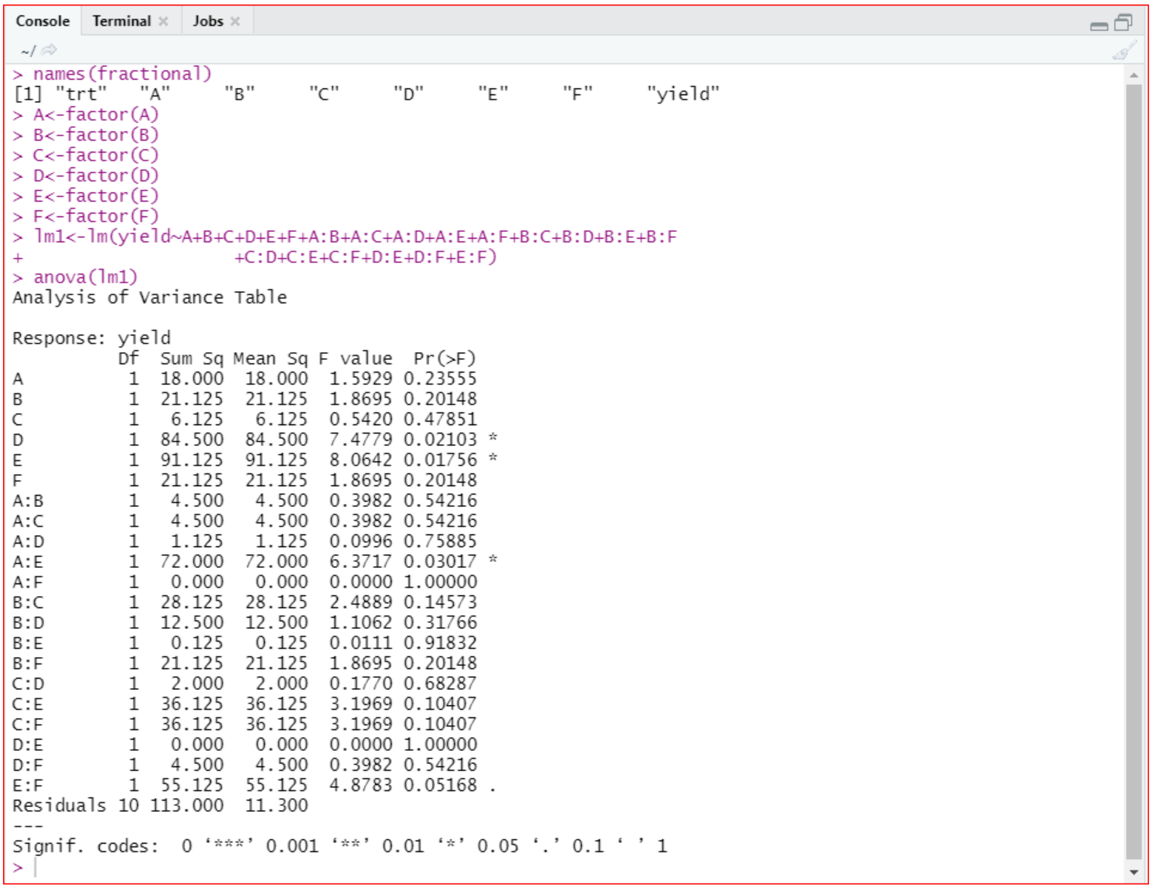
Fractional Factorial Experiment

Datafile: fractional.csv

trt	A	B	C	D	E	F	yield
1	0	0	0	0	0	0	15
ab	1	1	0	0	0	0	4
ac	1	0	1	0	0	0	7
ad	1	0	0	1	0	0	12
ae	1	0	0	0	1	0	13
af	1	0	0	0	0	1	15
bc	0	1	1	0	0	0	17
bd	0	1	0	1	0	0	5
be	0	1	0	0	1	0	6
bf	0	1	0	0	0	1	14
cd	0	0	1	1	0	0	9
ce	0	0	1	0	1	0	12
cf	0	0	1	0	0	1	11
de	0	0	0	1	1	0	5
df	0	0	0	1	0	1	13
ef	0	0	0	0	1	1	6
abcd	1	1	1	1	0	0	3
abce	1	1	1	0	1	0	11
abcf	1	1	1	0	0	1	6
abde	1	1	0	1	1	0	4
abdf	1	1	0	1	0	1	10
abef	1	1	0	0	1	1	4
acde	1	0	1	1	1	0	9
acdf	1	0	1	1	0	1	1
acef	1	0	1	0	1	1	8
adef	1	0	0	1	1	1	1
bcde	0	1	1	1	1	0	6
bcdf	0	1	1	1	0	1	10
bdef	0	1	0	1	1	1	5
bcef	0	1	1	0	1	1	2
cdef	0	0	1	1	1	1	1
abcdef	1	1	1	1	1	1	5

```
attach(fractional)

names(fractional)

A<-factor(A)

B<-factor(B)

C<-factor(C)
```

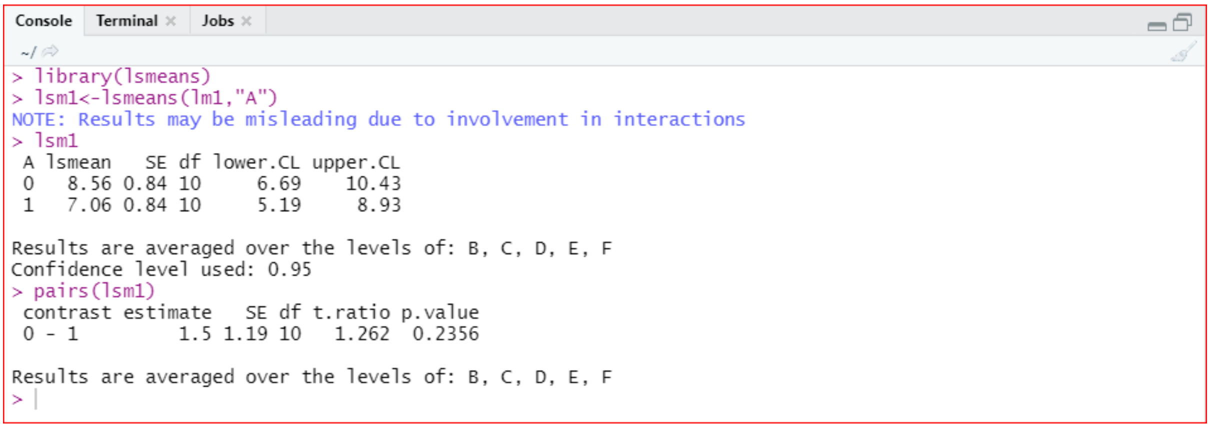
```
D<-factor(D)

E<-factor(E)

F<-factor(F)

lm1<-lm(yield~A+B+C+D+E+F+A:B+A:C+A:D+A:E+A:F+B:C+B:D+B:E+B:F

              +C:D+C:E+C:F+D:E+D:F+E:F)

anova(lm1)
```



```
> names(fractional)
[1] "trt"   "A"     "B"     "C"     "D"     "E"     "F"     "yield"
> A<-factor(A)
> B<-factor(B)
> C<-factor(C)
> D<-factor(D)
> E<-factor(E)
> F<-factor(F)
> lm1<-lm(yield~A+B+C+D+E+F+A:B+A:C+A:D+A:E+A:F+B:C+B:D+B:E+B:F
+                    +C:D+C:E+C:F+D:E+D:F+E:F)
> anova(lm1)
Analysis of Variance Table

Response: yield
          Df  Sum Sq Mean Sq F value  Pr(>F)
A          1  18.000  18.000  1.5929 0.23555
B          1  21.125  21.125  1.8695 0.20148
C          1   6.125   6.125  0.5420 0.47851
D          1  84.500  84.500  7.4779 0.02103 *
E          1  91.125  91.125  8.0642 0.01756 *
F          1  21.125  21.125  1.8695 0.20148
A:B        1   4.500   4.500  0.3982 0.54216
A:C        1   4.500   4.500  0.3982 0.54216
A:D        1   1.125   1.125  0.0996 0.75885
A:E        1  72.000  72.000  6.3717 0.03017 *
A:F        1   0.000   0.000  0.0000 1.00000
B:C        1  28.125  28.125  2.4889 0.14573
B:D        1  12.500  12.500  1.1062 0.31766
B:E        1   0.125   0.125  0.0111 0.91832
B:F        1  21.125  21.125  1.8695 0.20148
C:D        1   2.000   2.000  0.1770 0.68287
C:E        1  36.125  36.125  3.1969 0.10407
C:F        1  36.125  36.125  3.1969 0.10407
D:E        1   0.000   0.000  0.0000 1.00000
D:F        1   4.500   4.500  0.3982 0.54216
E:F        1  55.125  55.125  4.8783 0.05168 .
Residuals 10 113.000  11.300
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> 
```

# To provide pairwise comparison, need to install the package lsmeans or emmeans#

```
install.packages("lsmeans")

library(lsmeans)

lsm1<-lsmeans(lm1,"A")

lsm1

pairs(lsm1)
```

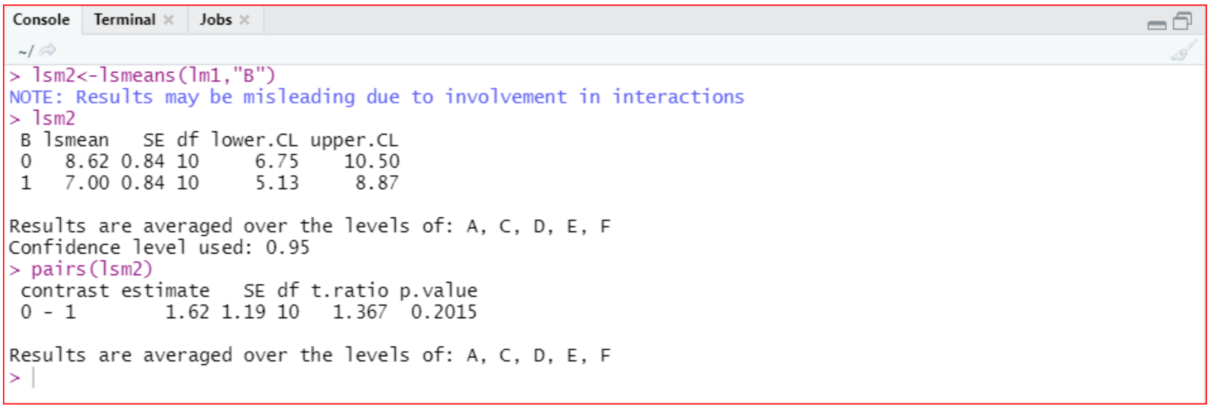


```
> library(lsmeans)
> lsm1<-lsmeans(lm1,"A")
NOTE: Results may be misleading due to involvement in interactions
> lsm1
 A lsmean   SE df lower.CL upper.CL
 0   8.56 0.84 10     6.69    10.43
 1   7.06 0.84 10     5.19     8.93

Results are averaged over the levels of: B, C, D, E, F
Confidence level used: 0.95
> pairs(lsm1)
 contrast estimate   SE df t.ratio p.value
 0 - 1         1.5 1.19 10   1.262  0.2356

Results are averaged over the levels of: B, C, D, E, F
>
```

lsm2<-lsmeans(lm1,"B")

lsm2

pairs(lsm2)

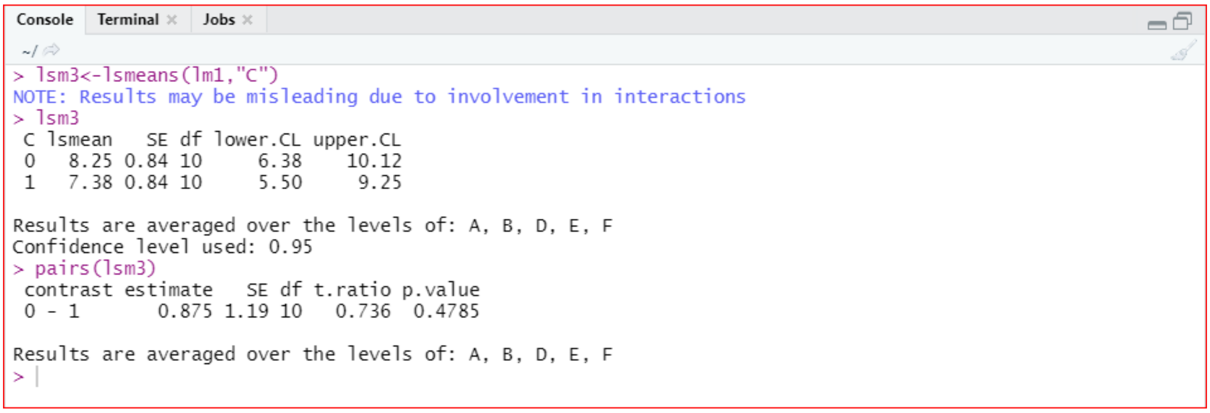


```
> lsm2<-lsmeans(lm1,"B")
NOTE: Results may be misleading due to involvement in interactions
> lsm2
 B lsmean   SE df lower.CL upper.CL
 0   8.62 0.84 10     6.75    10.50
 1   7.00 0.84 10     5.13     8.87

Results are averaged over the levels of: A, C, D, E, F
Confidence level used: 0.95
> pairs(lsm2)
 contrast estimate   SE df t.ratio p.value
 0 - 1        1.62 1.19 10   1.367  0.2015

Results are averaged over the levels of: A, C, D, E, F
>
```

lsm3<-lsmeans(lm1,"C")

lsm3

pairs(lsm3)

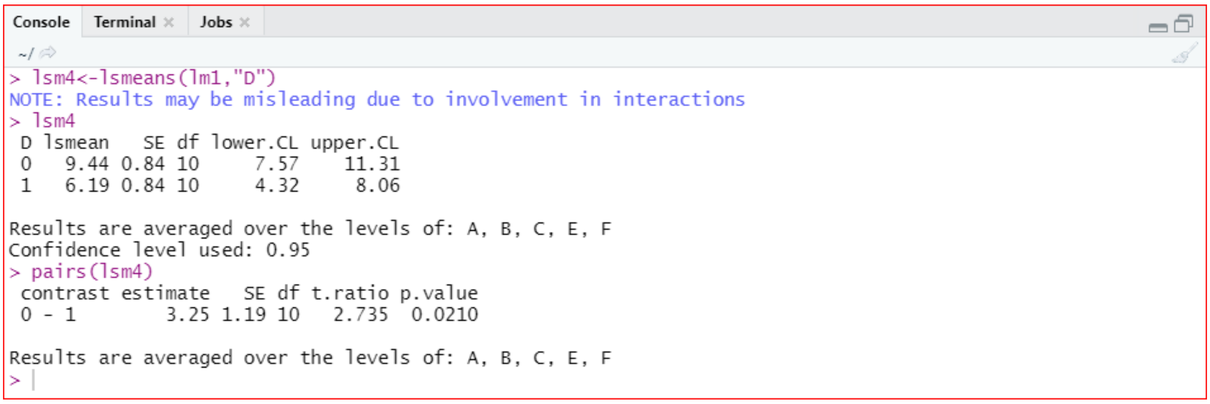


```
> lsm3<-lsmeans(lm1,"C")
NOTE: Results may be misleading due to involvement in interactions
> lsm3
 C lsmean   SE df lower.CL upper.CL
 0   8.25 0.84 10     6.38    10.12
 1   7.38 0.84 10     5.50     9.25

Results are averaged over the levels of: A, B, D, E, F
Confidence level used: 0.95
> pairs(lsm3)
 contrast estimate   SE df t.ratio p.value
 0 - 1       0.875 1.19 10   0.736  0.4785

Results are averaged over the levels of: A, B, D, E, F
>
```

lsm4<-lsmeans(lm1,"D")

lsm4

pairs(lsm4)

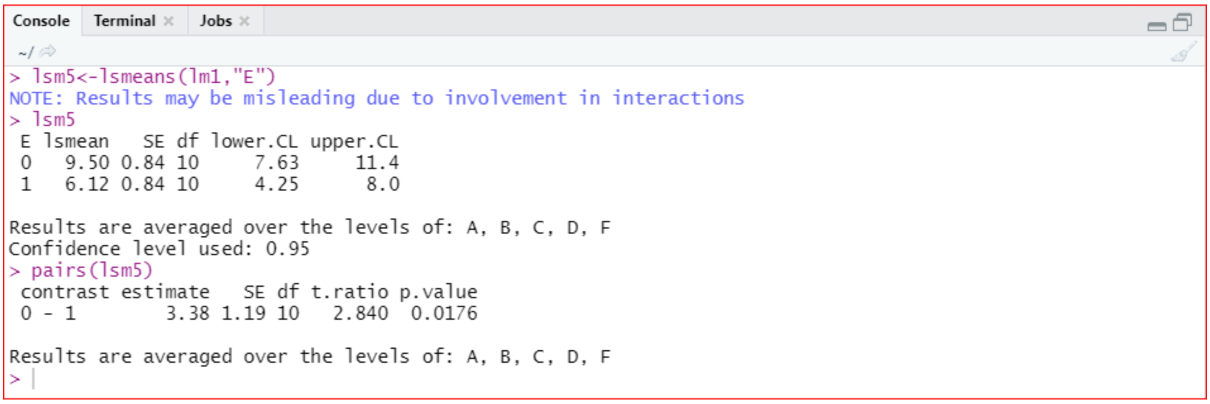


```
> lsm4<-lsmeans(lm1,"D")
NOTE: Results may be misleading due to involvement in interactions
> lsm4
 D lsmean   SE df lower.CL upper.CL
 0   9.44 0.84 10     7.57    11.31
 1   6.19 0.84 10     4.32     8.06

Results are averaged over the levels of: A, B, C, E, F
Confidence level used: 0.95
> pairs(lsm4)
 contrast estimate   SE df t.ratio p.value
 0 - 1        3.25 1.19 10   2.735  0.0210

Results are averaged over the levels of: A, B, C, E, F
> 
```

lsm5<-lsmeans(lm1,"E")

lsm5

pairs(lsm5)

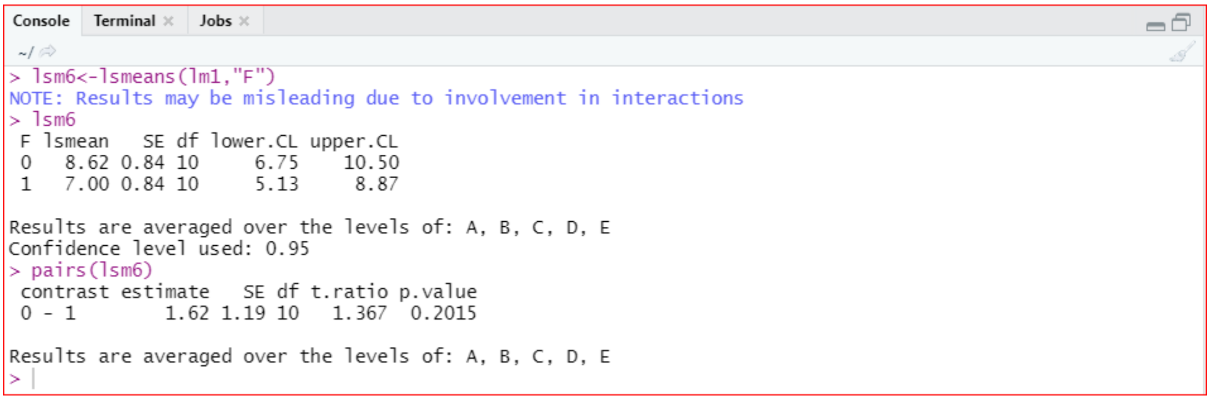


```
> lsm5<-lsmeans(lm1,"E")
NOTE: Results may be misleading due to involvement in interactions
> lsm5
 E lsmean   SE df lower.CL upper.CL
 0   9.50 0.84 10     7.63     11.4
 1   6.12 0.84 10     4.25      8.0

Results are averaged over the levels of: A, B, C, D, F
Confidence level used: 0.95
> pairs(lsm5)
 contrast estimate   SE df t.ratio p.value
 0 - 1        3.38 1.19 10   2.840  0.0176

Results are averaged over the levels of: A, B, C, D, F
> 
```

lsm6<-lsmeans(lm1,"F")

lsm6

pairs(lsm6)

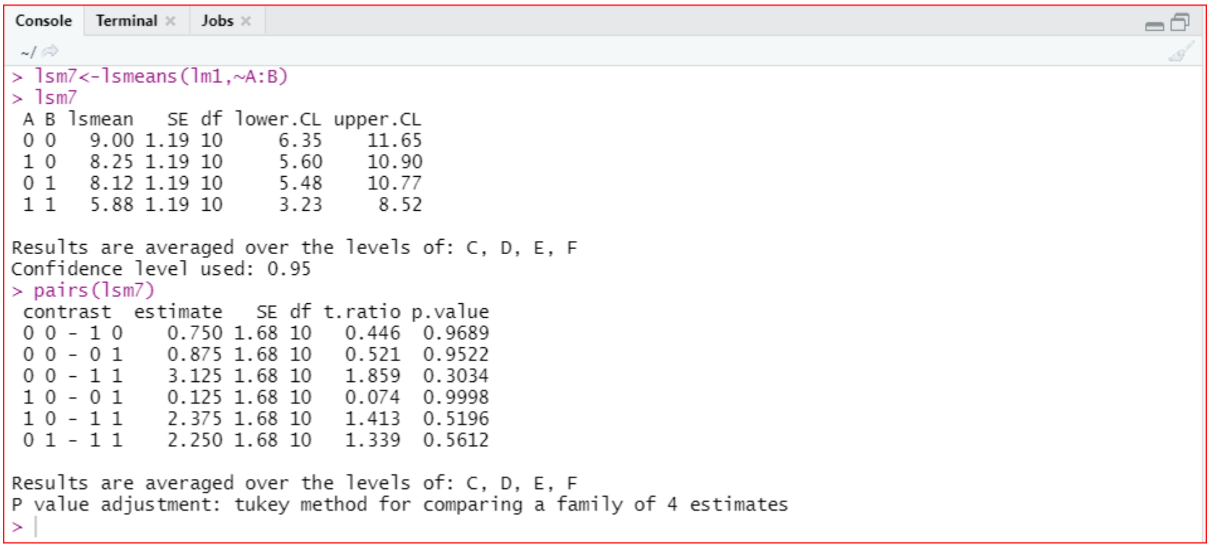


```
> lsm6<-lsmeans(lm1,"F")
NOTE: Results may be misleading due to involvement in interactions
> lsm6
 F lsmean   SE df lower.CL upper.CL
 0   8.62 0.84 10     6.75    10.50
 1   7.00 0.84 10     5.13     8.87

Results are averaged over the levels of: A, B, C, D, E
Confidence level used: 0.95
> pairs(lsm6)
 contrast estimate   SE df t.ratio p.value
 0 - 1        1.62 1.19 10   1.367  0.2015

Results are averaged over the levels of: A, B, C, D, E
> 
```

lsm7<-lsmeans(lm1,~A:B)

lsm7

pairs(lsm7)

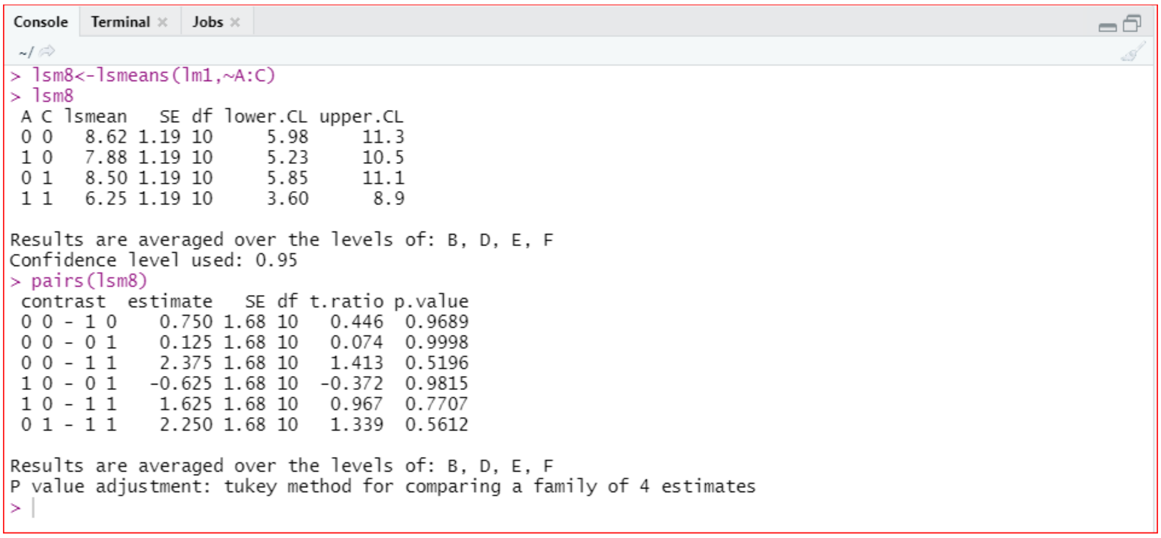


```
> lsm7<-lsmeans(lm1,~A:B)
> lsm7
 A B lsmean   SE df lower.CL upper.CL
 0 0   9.00 1.19 10     6.35    11.65
 1 0   8.25 1.19 10     5.60    10.90
 0 1   8.12 1.19 10     5.48    10.77
 1 1   5.88 1.19 10     3.23     8.52

Results are averaged over the levels of: C, D, E, F
Confidence level used: 0.95
> pairs(lsm7)
 contrast  estimate   SE df t.ratio p.value
 0 0 - 1 0    0.750 1.68 10   0.446  0.9689
 0 0 - 0 1    0.875 1.68 10   0.521  0.9522
 0 0 - 1 1    3.125 1.68 10   1.859  0.3034
 1 0 - 0 1    0.125 1.68 10   0.074  0.9998
 1 0 - 1 1    2.375 1.68 10   1.413  0.5196
 0 1 - 1 1    2.250 1.68 10   1.339  0.5612

Results are averaged over the levels of: C, D, E, F
P value adjustment: tukey method for comparing a family of 4 estimates
> 
```

lsm8<-lsmeans(lm1,~A:C)

lsm8

pairs(lsm8)

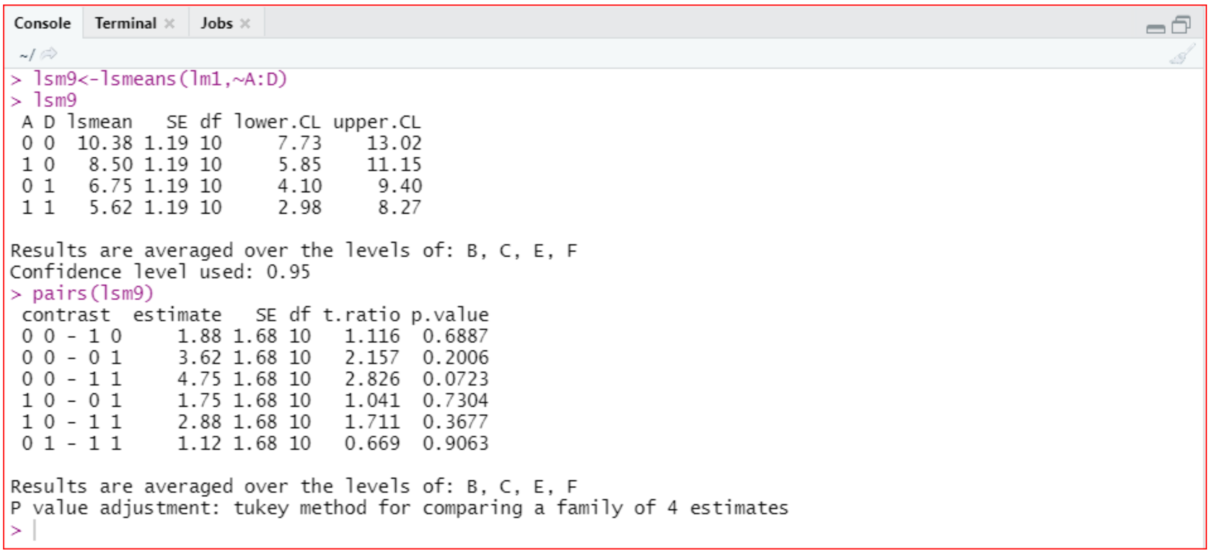


```
> lsm8<-lsmeans(lm1,~A:C)
> lsm8
 A C lsmean   SE df lower.CL upper.CL
 0 0   8.62 1.19 10     5.98     11.3
 1 0   7.88 1.19 10     5.23     10.5
 0 1   8.50 1.19 10     5.85     11.1
 1 1   6.25 1.19 10     3.60      8.9

Results are averaged over the levels of: B, D, E, F
Confidence level used: 0.95
> pairs(lsm8)
 contrast  estimate   SE df t.ratio p.value
 0 0 - 1 0    0.750 1.68 10   0.446  0.9689
 0 0 - 0 1    0.125 1.68 10   0.074  0.9998
 0 0 - 1 1    2.375 1.68 10   1.413  0.5196
 1 0 - 0 1   -0.625 1.68 10  -0.372  0.9815
 1 0 - 1 1    1.625 1.68 10   0.967  0.7707
 0 1 - 1 1    2.250 1.68 10   1.339  0.5612

Results are averaged over the levels of: B, D, E, F
P value adjustment: tukey method for comparing a family of 4 estimates
> 
```

lsm9<-lsmeans(lm1,~A:D)

lsm9

pairs(lsm9)

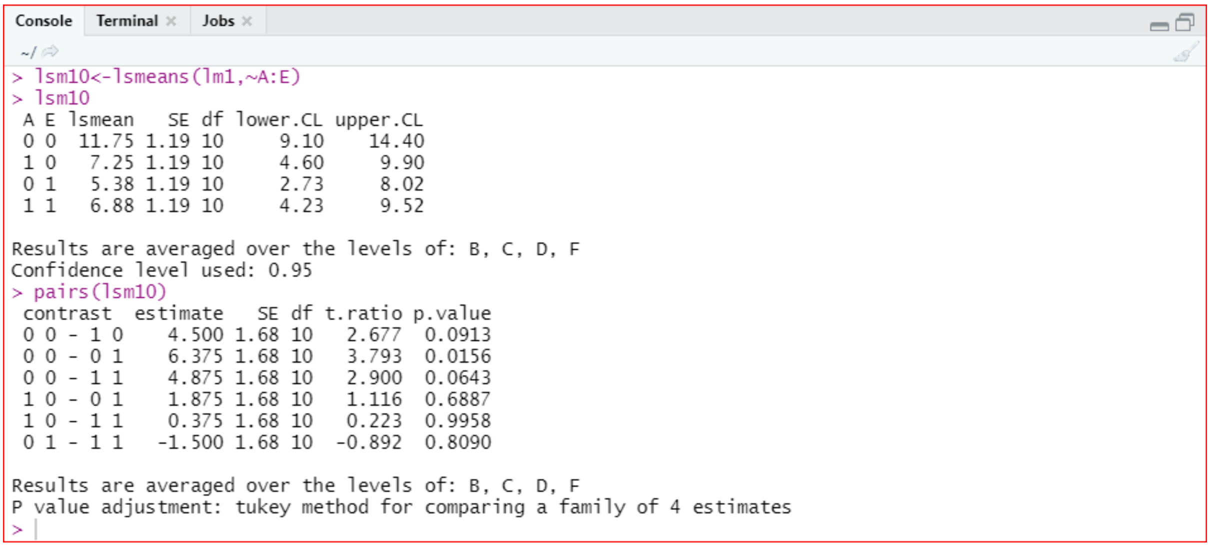


```
> lsm9<-lsmeans(lm1,~A:D)
> lsm9
 A D lsmean   SE df lower.CL upper.CL
 0 0  10.38 1.19 10     7.73    13.02
 1 0   8.50 1.19 10     5.85    11.15
 0 1   6.75 1.19 10     4.10     9.40
 1 1   5.62 1.19 10     2.98     8.27

Results are averaged over the levels of: B, C, E, F
Confidence level used: 0.95
> pairs(lsm9)
 contrast  estimate   SE df t.ratio p.value
 0 0 - 1 0     1.88 1.68 10   1.116  0.6887
 0 0 - 0 1     3.62 1.68 10   2.157  0.2006
 0 0 - 1 1     4.75 1.68 10   2.826  0.0723
 1 0 - 0 1     1.75 1.68 10   1.041  0.7304
 1 0 - 1 1     2.88 1.68 10   1.711  0.3677
 0 1 - 1 1     1.12 1.68 10   0.669  0.9063

Results are averaged over the levels of: B, C, E, F
P value adjustment: tukey method for comparing a family of 4 estimates
> 
```

lsm10<-lsmeans(lm1,~A:E)

lsm10

pairs(lsm10)

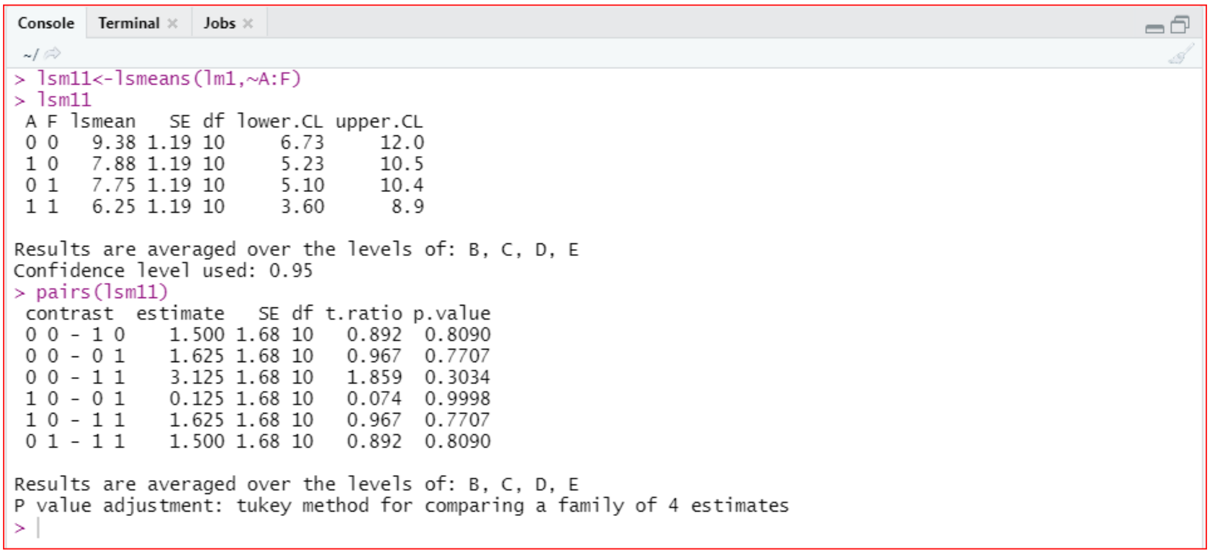


```
> lsm10<-lsmeans(lm1,~A:E)
> lsm10
 A E lsmean   SE df lower.CL upper.CL
 0 0  11.75 1.19 10     9.10    14.40
 1 0   7.25 1.19 10     4.60     9.90
 0 1   5.38 1.19 10     2.73     8.02
 1 1   6.88 1.19 10     4.23     9.52

Results are averaged over the levels of: B, C, D, F
Confidence level used: 0.95
> pairs(lsm10)
 contrast  estimate   SE df t.ratio p.value
 0 0 - 1 0    4.500 1.68 10   2.677  0.0913
 0 0 - 0 1    6.375 1.68 10   3.793  0.0156
 0 0 - 1 1    4.875 1.68 10   2.900  0.0643
 1 0 - 0 1    1.875 1.68 10   1.116  0.6887
 1 0 - 1 1    0.375 1.68 10   0.223  0.9958
 0 1 - 1 1   -1.500 1.68 10  -0.892  0.8090

Results are averaged over the levels of: B, C, D, F
P value adjustment: tukey method for comparing a family of 4 estimates
> 
```

lsm11<-lsmeans(lm1,~A:F)

lsm11

pairs(lsm11)

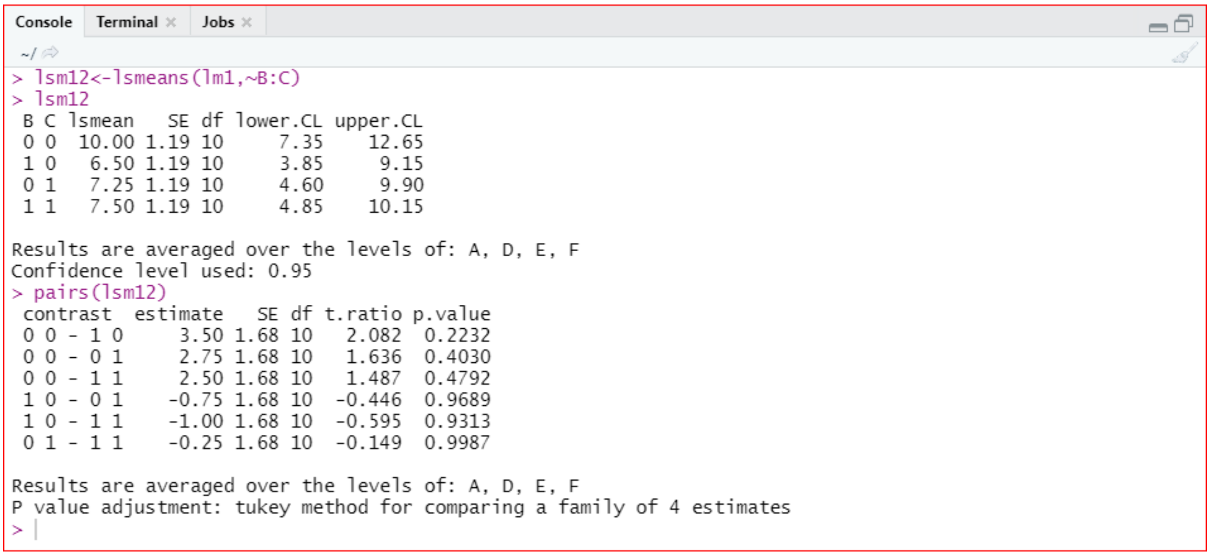


```
> lsm11<-lsmeans(lm1,~A:F)
> lsm11
 A F lsmean   SE df lower.CL upper.CL
 0 0   9.38 1.19 10     6.73     12.0
 1 0   7.88 1.19 10     5.23     10.5
 0 1   7.75 1.19 10     5.10     10.4
 1 1   6.25 1.19 10     3.60      8.9

Results are averaged over the levels of: B, C, D, E
Confidence level used: 0.95
> pairs(lsm11)
 contrast  estimate   SE df t.ratio p.value
 0 0 - 1 0    1.500 1.68 10   0.892  0.8090
 0 0 - 0 1    1.625 1.68 10   0.967  0.7707
 0 0 - 1 1    3.125 1.68 10   1.859  0.3034
 1 0 - 0 1    0.125 1.68 10   0.074  0.9998
 1 0 - 1 1    1.625 1.68 10   0.967  0.7707
 0 1 - 1 1    1.500 1.68 10   0.892  0.8090

Results are averaged over the levels of: B, C, D, E
P value adjustment: tukey method for comparing a family of 4 estimates
> 
```

lsm12<-lsmeans(lm1,~B:C)

lsm12

pairs(lsm12)

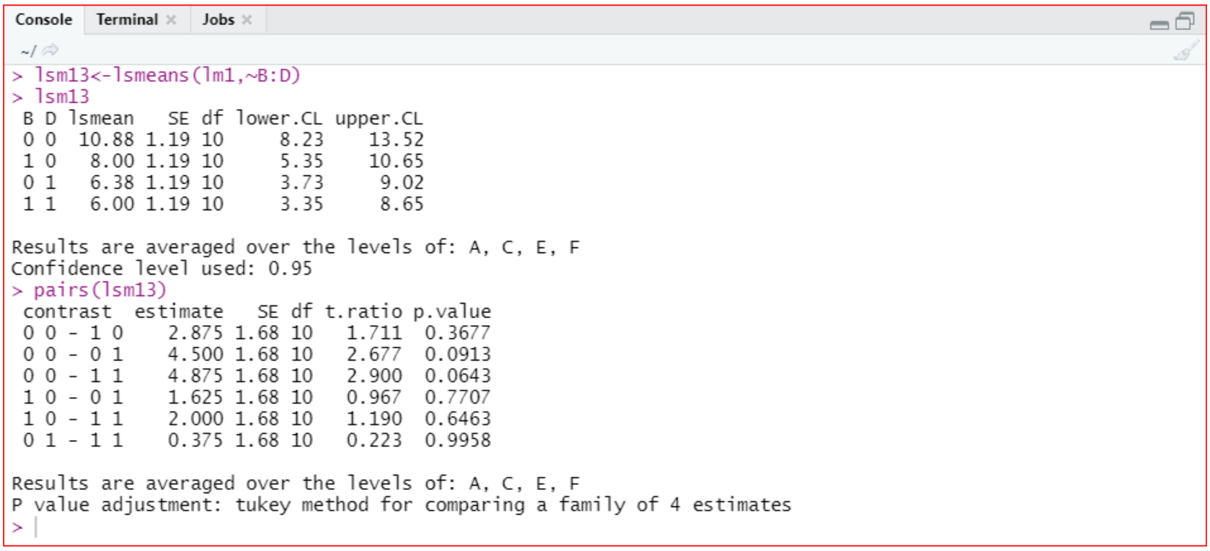


```
> lsm12<-lsmeans(lm1,~B:C)
> lsm12
 B C lsmean   SE df lower.CL upper.CL
 0 0  10.00 1.19 10     7.35    12.65
 1 0   6.50 1.19 10     3.85     9.15
 0 1   7.25 1.19 10     4.60     9.90
 1 1   7.50 1.19 10     4.85    10.15

Results are averaged over the levels of: A, D, E, F
Confidence level used: 0.95
> pairs(lsm12)
 contrast  estimate   SE df t.ratio p.value
 0 0 - 1 0     3.50 1.68 10   2.082  0.2232
 0 0 - 0 1     2.75 1.68 10   1.636  0.4030
 0 0 - 1 1     2.50 1.68 10   1.487  0.4792
 1 0 - 0 1    -0.75 1.68 10  -0.446  0.9689
 1 0 - 1 1    -1.00 1.68 10  -0.595  0.9313
 0 1 - 1 1    -0.25 1.68 10  -0.149  0.9987

Results are averaged over the levels of: A, D, E, F
P value adjustment: tukey method for comparing a family of 4 estimates
> 
```

lsm13<-lsmeans(lm1,~B:D)

lsm13

pairs(lsm13)

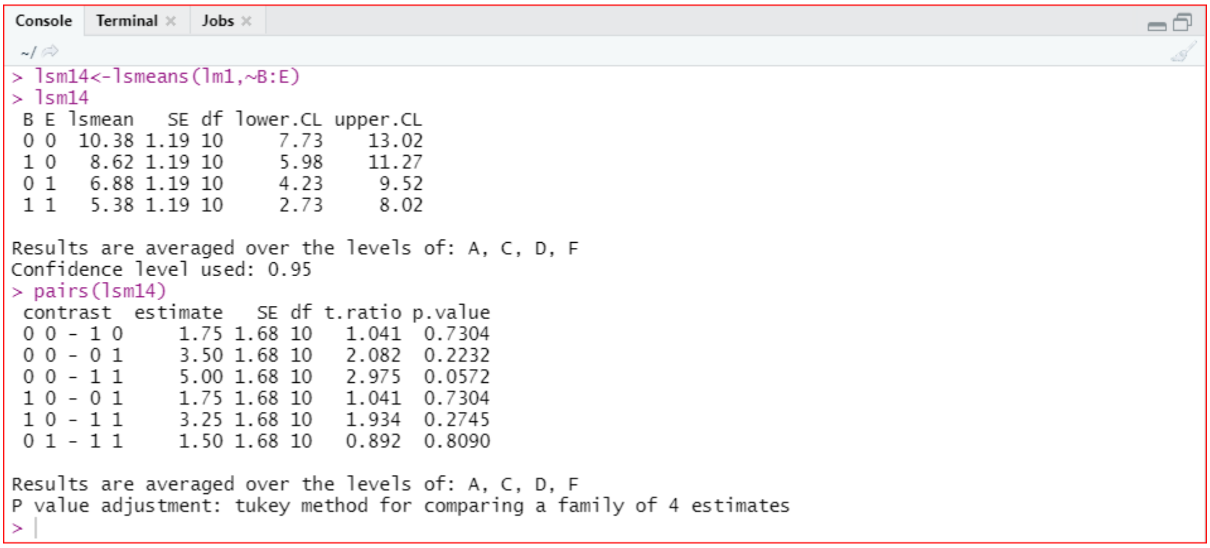


```
> lsm13<-lsmeans(lm1,~B:D)
> lsm13
 B D lsmean   SE df lower.CL upper.CL
 0 0  10.88 1.19 10     8.23    13.52
 1 0   8.00 1.19 10     5.35    10.65
 0 1   6.38 1.19 10     3.73     9.02
 1 1   6.00 1.19 10     3.35     8.65

Results are averaged over the levels of: A, C, E, F 
Confidence level used: 0.95 
> pairs(lsm13)
 contrast  estimate   SE df t.ratio p.value
 0 0 - 1 0    2.875 1.68 10   1.711  0.3677
 0 0 - 0 1    4.500 1.68 10   2.677  0.0913
 0 0 - 1 1    4.875 1.68 10   2.900  0.0643
 1 0 - 0 1    1.625 1.68 10   0.967  0.7707
 1 0 - 1 1    2.000 1.68 10   1.190  0.6463
 0 1 - 1 1    0.375 1.68 10   0.223  0.9958

Results are averaged over the levels of: A, C, E, F 
P value adjustment: tukey method for comparing a family of 4 estimates 
> 
```

lsm14<-lsmeans(lm1,~B:E)

lsm14

pairs(lsm14)

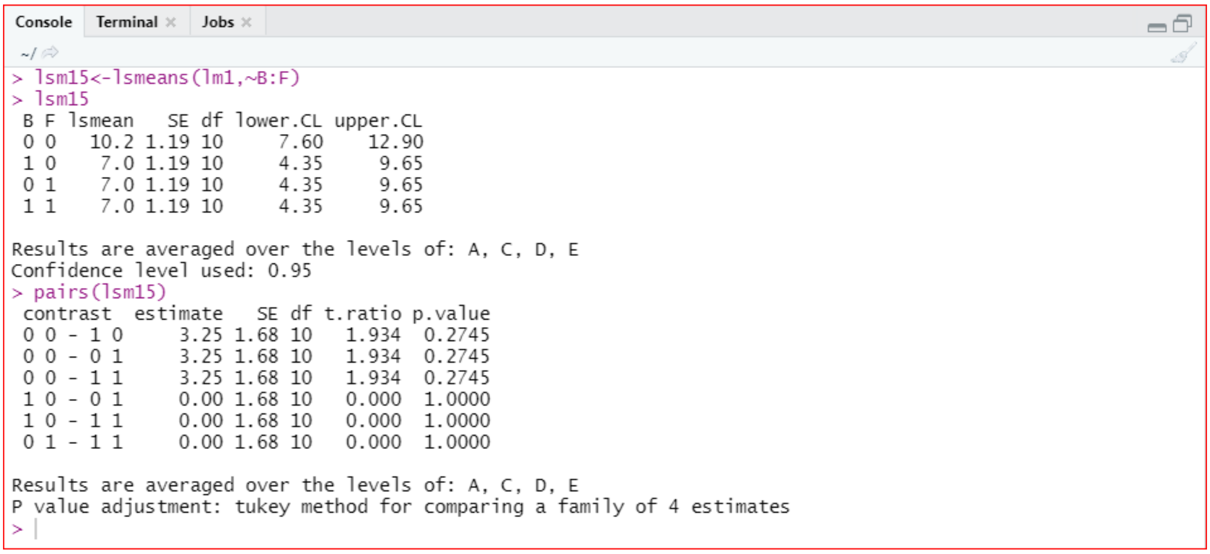


```
> lsm14<-lsmeans(lm1,~B:E)
> lsm14
 B E lsmean   SE df lower.CL upper.CL
 0 0  10.38 1.19 10     7.73    13.02
 1 0   8.62 1.19 10     5.98    11.27
 0 1   6.88 1.19 10     4.23     9.52
 1 1   5.38 1.19 10     2.73     8.02

Results are averaged over the levels of: A, C, D, F 
Confidence level used: 0.95 
> pairs(lsm14)
 contrast  estimate   SE df t.ratio p.value
 0 0 - 1 0     1.75 1.68 10   1.041  0.7304
 0 0 - 0 1     3.50 1.68 10   2.082  0.2232
 0 0 - 1 1     5.00 1.68 10   2.975  0.0572
 1 0 - 0 1     1.75 1.68 10   1.041  0.7304
 1 0 - 1 1     3.25 1.68 10   1.934  0.2745
 0 1 - 1 1     1.50 1.68 10   0.892  0.8090

Results are averaged over the levels of: A, C, D, F 
P value adjustment: tukey method for comparing a family of 4 estimates 
> 
```

lsm15<-lsmeans(lm1,~B:F)

lsm15

pairs(lsm15)

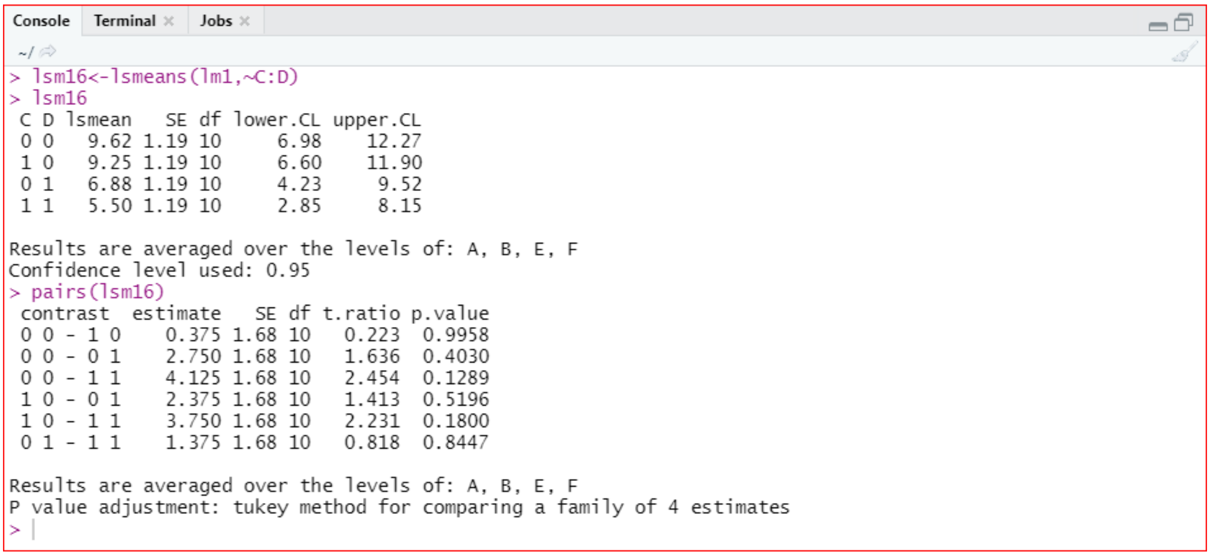


```
> lsm15<-lsmeans(lm1,~B:F)
> lsm15
 B F lsmean   SE df lower.CL upper.CL
 0 0   10.2 1.19 10     7.60    12.90
 1 0    7.0 1.19 10     4.35     9.65
 0 1    7.0 1.19 10     4.35     9.65
 1 1    7.0 1.19 10     4.35     9.65

Results are averaged over the levels of: A, C, D, E
Confidence level used: 0.95
> pairs(lsm15)
 contrast  estimate   SE df t.ratio p.value
 0 0 - 1 0     3.25 1.68 10   1.934  0.2745
 0 0 - 0 1     3.25 1.68 10   1.934  0.2745
 0 0 - 1 1     3.25 1.68 10   1.934  0.2745
 1 0 - 0 1     0.00 1.68 10   0.000  1.0000
 1 0 - 1 1     0.00 1.68 10   0.000  1.0000
 0 1 - 1 1     0.00 1.68 10   0.000  1.0000

Results are averaged over the levels of: A, C, D, E
P value adjustment: tukey method for comparing a family of 4 estimates
> 
```

lsm16<-lsmeans(lm1,~C:D)

lsm16

pairs(lsm16)

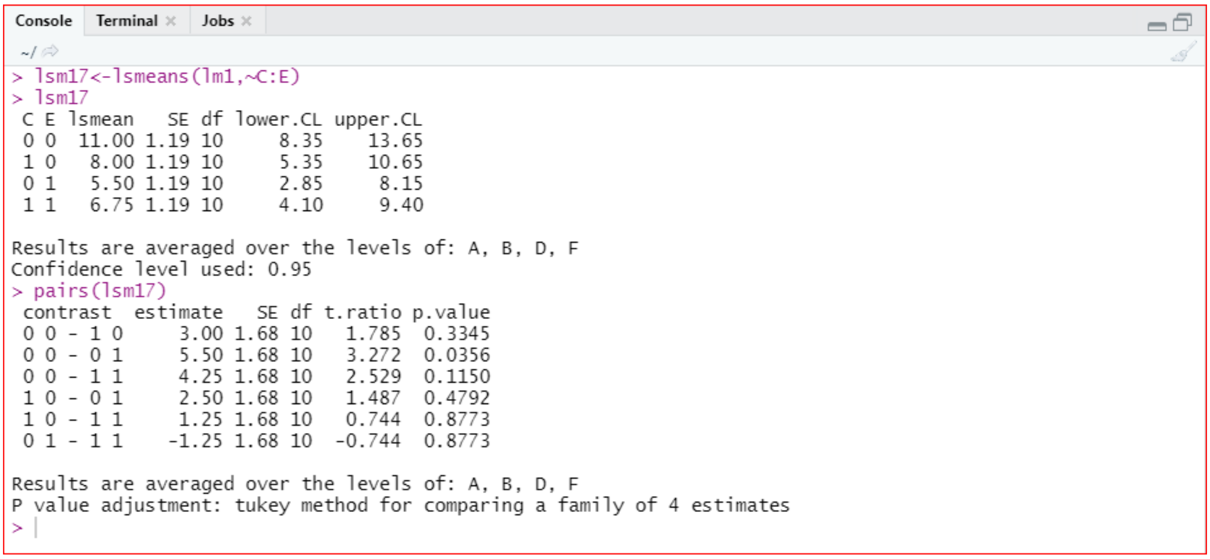


```
> lsm16<-lsmeans(lm1,~C:D)
> lsm16
 C D lsmean   SE df lower.CL upper.CL
 0 0   9.62 1.19 10     6.98    12.27
 1 0   9.25 1.19 10     6.60    11.90
 0 1   6.88 1.19 10     4.23     9.52
 1 1   5.50 1.19 10     2.85     8.15

Results are averaged over the levels of: A, B, E, F
Confidence level used: 0.95
> pairs(lsm16)
 contrast  estimate   SE df t.ratio p.value
 0 0 - 1 0    0.375 1.68 10   0.223  0.9958
 0 0 - 0 1    2.750 1.68 10   1.636  0.4030
 0 0 - 1 1    4.125 1.68 10   2.454  0.1289
 1 0 - 0 1    2.375 1.68 10   1.413  0.5196
 1 0 - 1 1    3.750 1.68 10   2.231  0.1800
 0 1 - 1 1    1.375 1.68 10   0.818  0.8447

Results are averaged over the levels of: A, B, E, F
P value adjustment: tukey method for comparing a family of 4 estimates
> 
```

lsm17<-lsmeans(lm1,~C:E)

lsm17

pairs(lsm17)

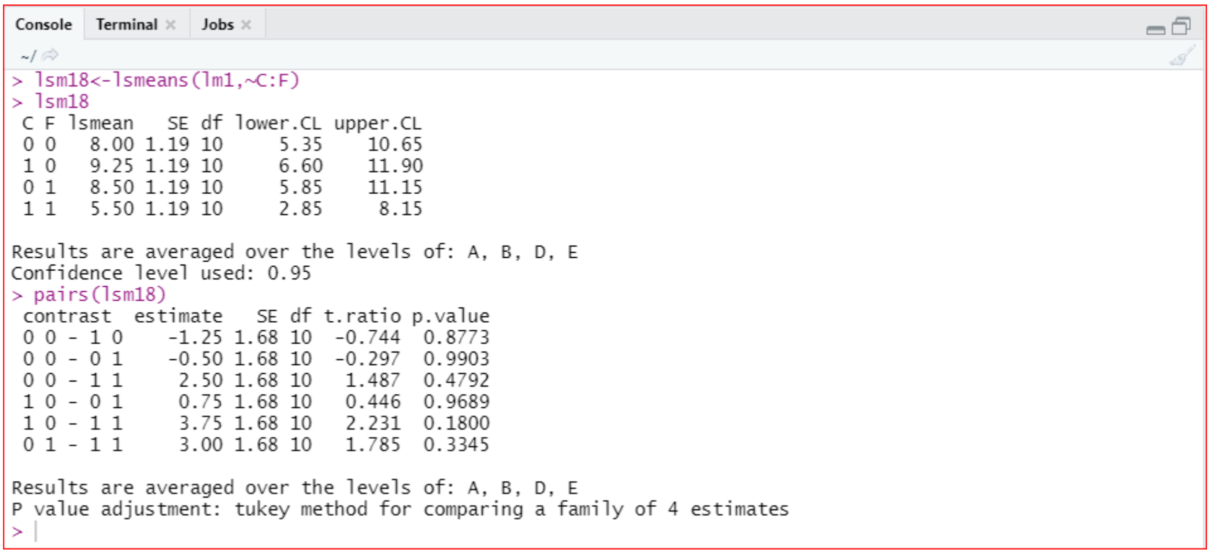


```
> lsm17<-lsmeans(lm1,~C:E)
> lsm17
 C E lsmean   SE df lower.CL upper.CL
 0 0  11.00 1.19 10     8.35    13.65
 1 0   8.00 1.19 10     5.35    10.65
 0 1   5.50 1.19 10     2.85     8.15
 1 1   6.75 1.19 10     4.10     9.40

Results are averaged over the levels of: A, B, D, F
Confidence level used: 0.95
> pairs(lsm17)
 contrast  estimate   SE df t.ratio p.value
 0 0 - 1 0     3.00 1.68 10   1.785  0.3345
 0 0 - 0 1     5.50 1.68 10   3.272  0.0356
 0 0 - 1 1     4.25 1.68 10   2.529  0.1150
 1 0 - 0 1     2.50 1.68 10   1.487  0.4792
 1 0 - 1 1     1.25 1.68 10   0.744  0.8773
 0 1 - 1 1    -1.25 1.68 10  -0.744  0.8773

Results are averaged over the levels of: A, B, D, F
P value adjustment: tukey method for comparing a family of 4 estimates
> 
```

lsm18<-lsmeans(lm1,~C:F)

lsm18

pairs(lsm18)

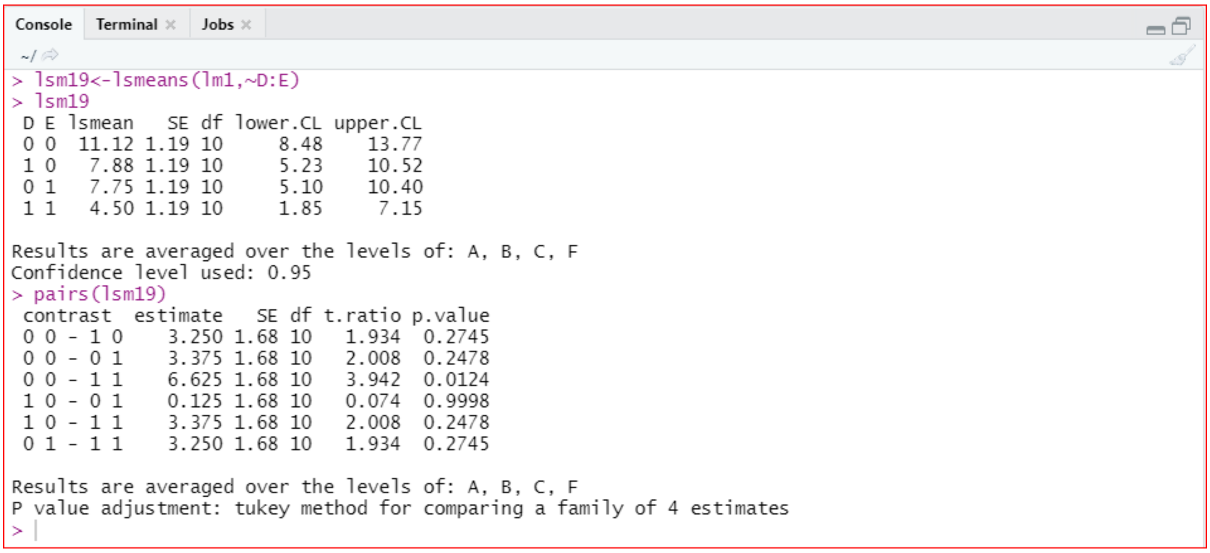


```
> lsm18<-lsmeans(lm1,~C:F)
> lsm18
 C F lsmean   SE df lower.CL upper.CL
 0 0   8.00 1.19 10     5.35    10.65
 1 0   9.25 1.19 10     6.60    11.90
 0 1   8.50 1.19 10     5.85    11.15
 1 1   5.50 1.19 10     2.85     8.15

Results are averaged over the levels of: A, B, D, E
Confidence level used: 0.95
> pairs(lsm18)
 contrast  estimate   SE df t.ratio p.value
 0 0 - 1 0    -1.25 1.68 10  -0.744  0.8773
 0 0 - 0 1    -0.50 1.68 10  -0.297  0.9903
 0 0 - 1 1     2.50 1.68 10   1.487  0.4792
 1 0 - 0 1     0.75 1.68 10   0.446  0.9689
 1 0 - 1 1     3.75 1.68 10   2.231  0.1800
 0 1 - 1 1     3.00 1.68 10   1.785  0.3345

Results are averaged over the levels of: A, B, D, E
P value adjustment: tukey method for comparing a family of 4 estimates
> 
```

lsm19<-lsmeans(lm1,~D:E)

lsm19

pairs(lsm19)

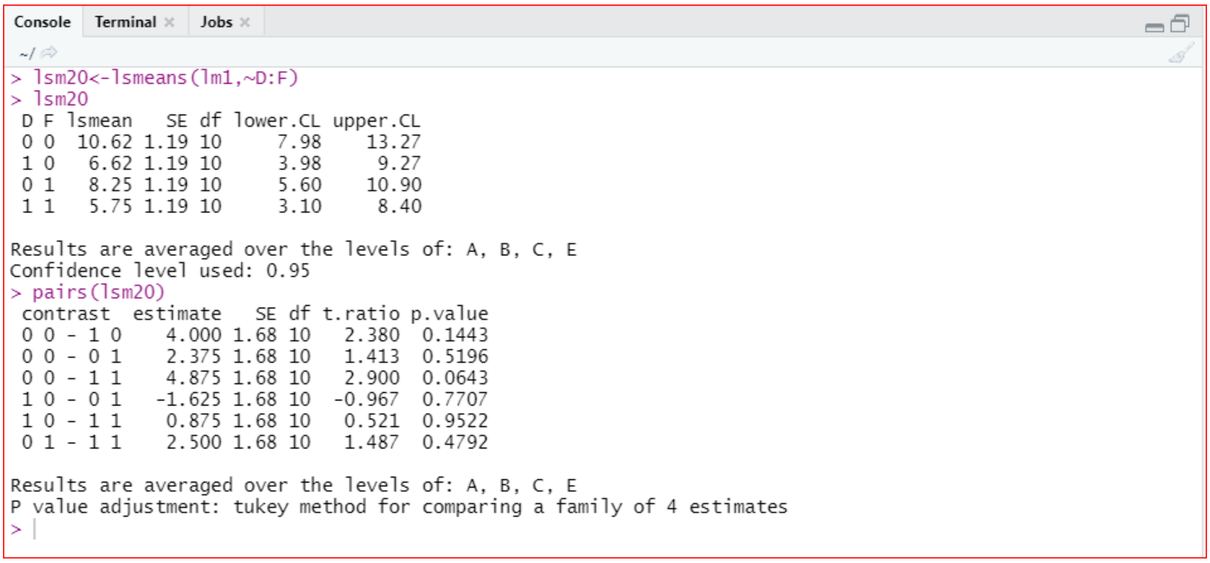


```
> lsm19<-lsmeans(lm1,~D:E)
> lsm19
 D E lsmean   SE df lower.CL upper.CL
 0 0  11.12 1.19 10     8.48    13.77
 1 0   7.88 1.19 10     5.23    10.52
 0 1   7.75 1.19 10     5.10    10.40
 1 1   4.50 1.19 10     1.85     7.15

Results are averaged over the levels of: A, B, C, F
Confidence level used: 0.95
> pairs(lsm19)
 contrast  estimate   SE df t.ratio p.value
 0 0 - 1 0    3.250 1.68 10   1.934  0.2745
 0 0 - 0 1    3.375 1.68 10   2.008  0.2478
 0 0 - 1 1    6.625 1.68 10   3.942  0.0124
 1 0 - 0 1    0.125 1.68 10   0.074  0.9998
 1 0 - 1 1    3.375 1.68 10   2.008  0.2478
 0 1 - 1 1    3.250 1.68 10   1.934  0.2745

Results are averaged over the levels of: A, B, C, F
P value adjustment: tukey method for comparing a family of 4 estimates
> 
```

lsm20<-lsmeans(lm1,~D:F)

lsm20

pairs(lsm20)

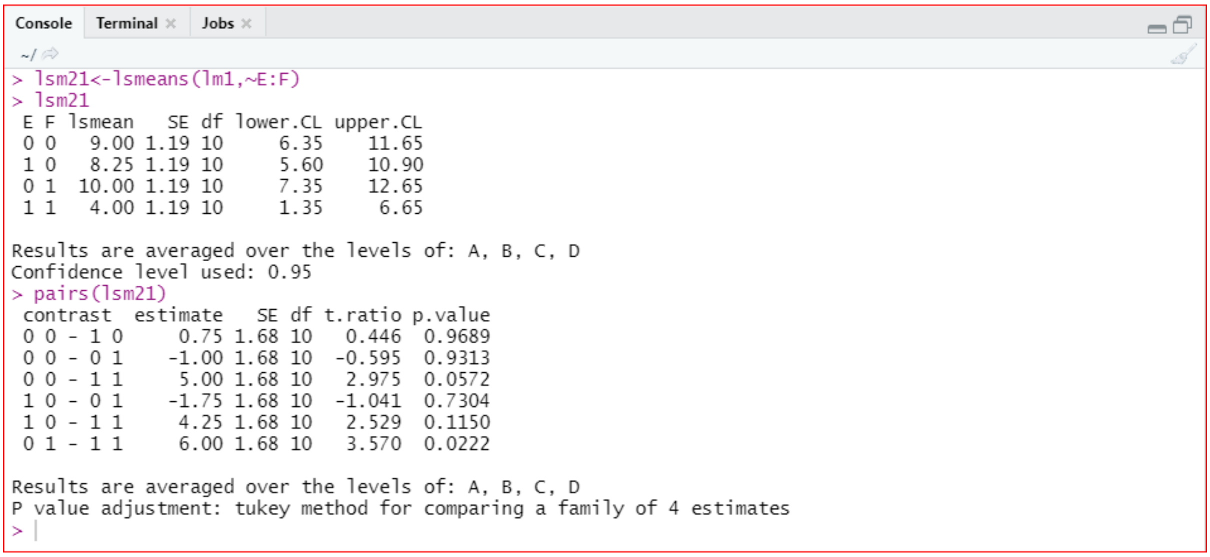


```
> lsm20<-lsmeans(lm1,~D:F)
> lsm20
 D F lsmean   SE df lower.CL upper.CL
 0 0  10.62 1.19 10     7.98    13.27
 1 0   6.62 1.19 10     3.98     9.27
 0 1   8.25 1.19 10     5.60    10.90
 1 1   5.75 1.19 10     3.10     8.40

Results are averaged over the levels of: A, B, C, E
Confidence level used: 0.95
> pairs(lsm20)
 contrast  estimate   SE df t.ratio p.value
 0 0 - 1 0    4.000 1.68 10   2.380  0.1443
 0 0 - 0 1    2.375 1.68 10   1.413  0.5196
 0 0 - 1 1    4.875 1.68 10   2.900  0.0643
 1 0 - 0 1   -1.625 1.68 10  -0.967  0.7707
 1 0 - 1 1    0.875 1.68 10   0.521  0.9522
 0 1 - 1 1    2.500 1.68 10   1.487  0.4792

Results are averaged over the levels of: A, B, C, E
P value adjustment: tukey method for comparing a family of 4 estimates
> 
```

lsm21<-lsmeans(lm1,~E:F)

lsm21

pairs(lsm21)



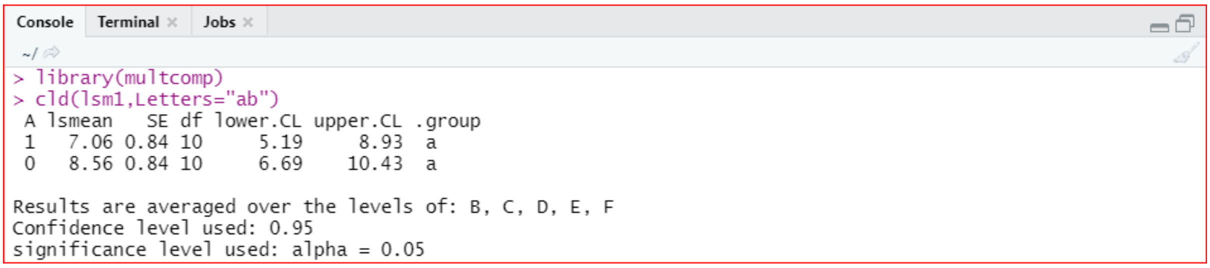
```
> lsm21<-lsmeans(lm1,~E:F)
> lsm21
 E F lsmean   SE df lower.CL upper.CL
 0 0   9.00 1.19 10     6.35    11.65
 1 0   8.25 1.19 10     5.60    10.90
 0 1  10.00 1.19 10     7.35    12.65
 1 1   4.00 1.19 10     1.35     6.65

Results are averaged over the levels of: A, B, C, D
Confidence level used: 0.95
> pairs(lsm21)
 contrast  estimate   SE df t.ratio p.value
 0 0 - 1 0     0.75 1.68 10   0.446  0.9689
 0 0 - 0 1    -1.00 1.68 10  -0.595  0.9313
 0 0 - 1 1     5.00 1.68 10   2.975  0.0572
 1 0 - 0 1    -1.75 1.68 10  -1.041  0.7304
 1 0 - 1 1     4.25 1.68 10   2.529  0.1150
 0 1 - 1 1     6.00 1.68 10   3.570  0.0222

Results are averaged over the levels of: A, B, C, D
P value adjustment: tukey method for comparing a family of 4 estimates
> |
```

# To provide letter grouping, need to install the package multcomp#

install.packages("multcomp")

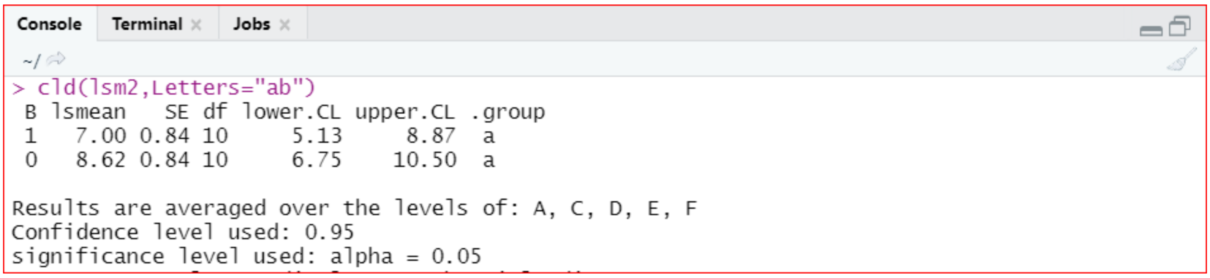
library(multcomp)

cld(lsm1,Letters="ab")



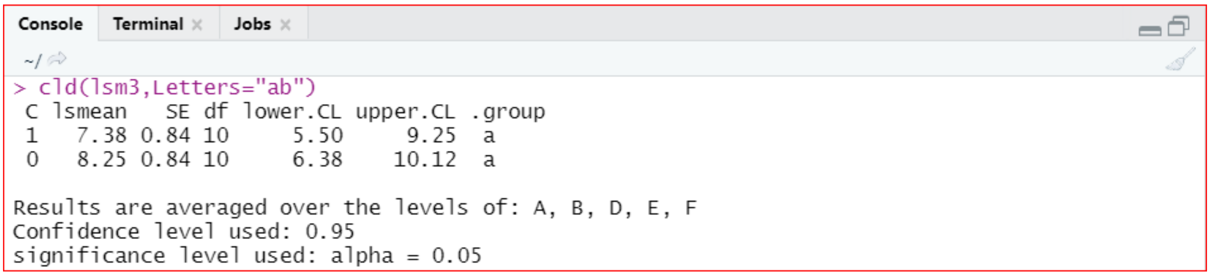
```
> library(multcomp)
> cld(lsm1,Letters="ab")
 A lsmean   SE df lower.CL upper.CL .group
 1   7.06 0.84 10     5.19     8.93  a
 0   8.56 0.84 10     6.69    10.43  a

Results are averaged over the levels of: B, C, D, E, F
Confidence level used: 0.95
significance level used: alpha = 0.05
```

cld(lsm2,Letters="ab")



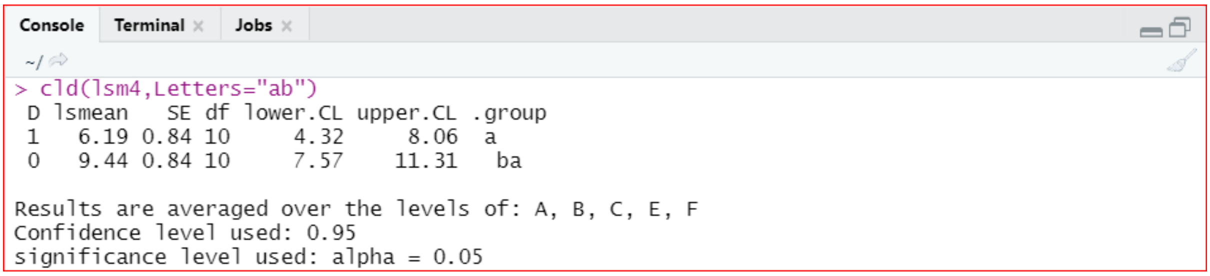
```
> cld(lsm2,Letters="ab")
 B lsmean   SE df lower.CL upper.CL .group
 1   7.00 0.84 10     5.13     8.87  a
 0   8.62 0.84 10     6.75    10.50  a

Results are averaged over the levels of: A, C, D, E, F
Confidence level used: 0.95
significance level used: alpha = 0.05
```

cld(lsm3,Letters="ab")



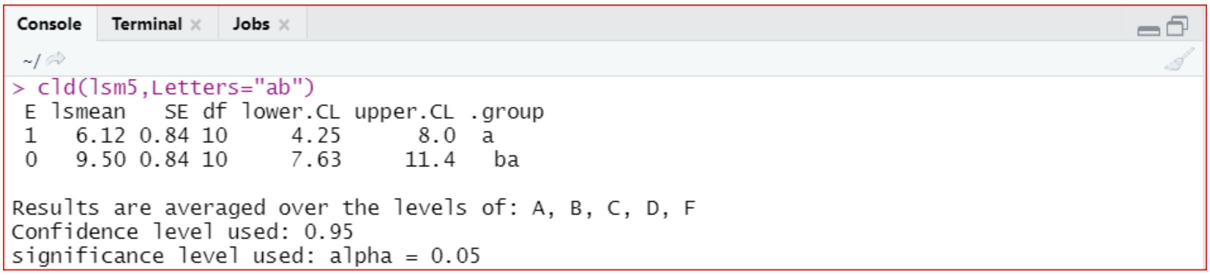
```
> cld(lsm3,Letters="ab")
 C lsmean   SE df lower.CL upper.CL .group
 1   7.38 0.84 10     5.50     9.25  a
 0   8.25 0.84 10     6.38    10.12  a

Results are averaged over the levels of: A, B, D, E, F
Confidence level used: 0.95
significance level used: alpha = 0.05
```

cld(lsm4,Letters="ab")



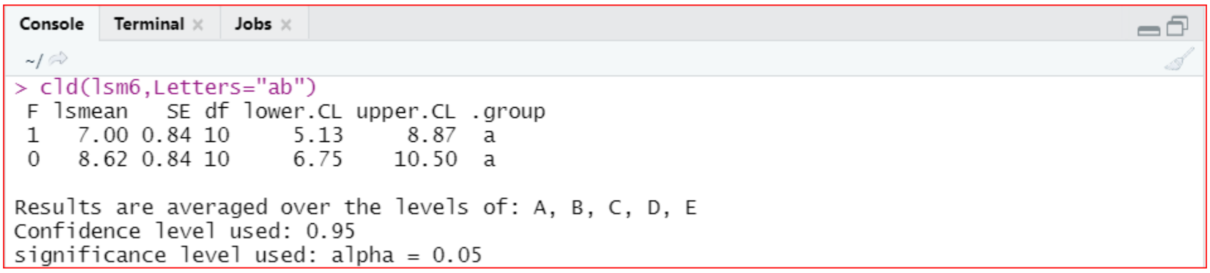
```
> cld(lsm4,Letters="ab")
 D lsmean   SE df lower.CL upper.CL .group
 1   6.19 0.84 10     4.32     8.06  a
 0   9.44 0.84 10     7.57    11.31   ba

Results are averaged over the levels of: A, B, C, E, F
Confidence level used: 0.95
significance level used: alpha = 0.05
```

cld(lsm5,Letters="ab")



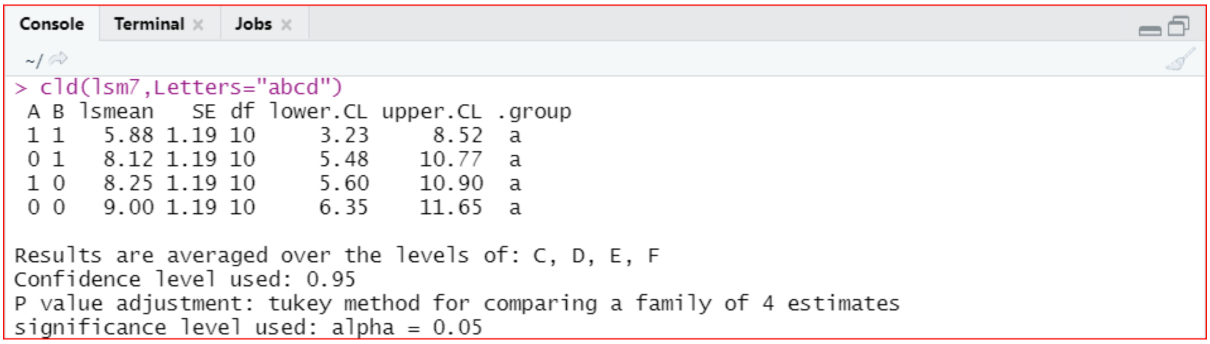
```
> cld(lsm5,Letters="ab")
 E lsmean   SE df lower.CL upper.CL .group
 1   6.12 0.84 10     4.25      8.0  a
 0   9.50 0.84 10     7.63     11.4   ba

Results are averaged over the levels of: A, B, C, D, F
Confidence level used: 0.95
significance level used: alpha = 0.05
```

cld(lsm6,Letters="ab")



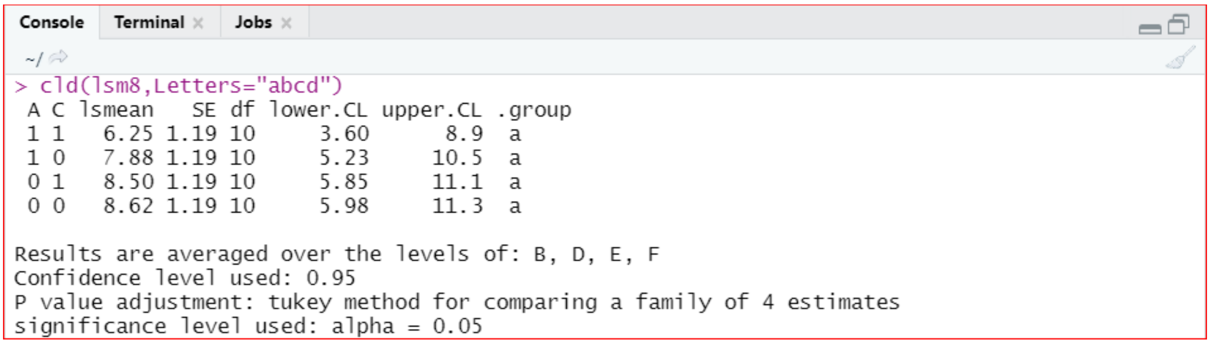
```
> cld(lsm6,Letters="ab")
 F lsmean   SE df lower.CL upper.CL .group
 1   7.00 0.84 10     5.13     8.87  a
 0   8.62 0.84 10     6.75    10.50  a

Results are averaged over the levels of: A, B, C, D, E
Confidence level used: 0.95
significance level used: alpha = 0.05
```

cld(lsm7,Letters="abcd")



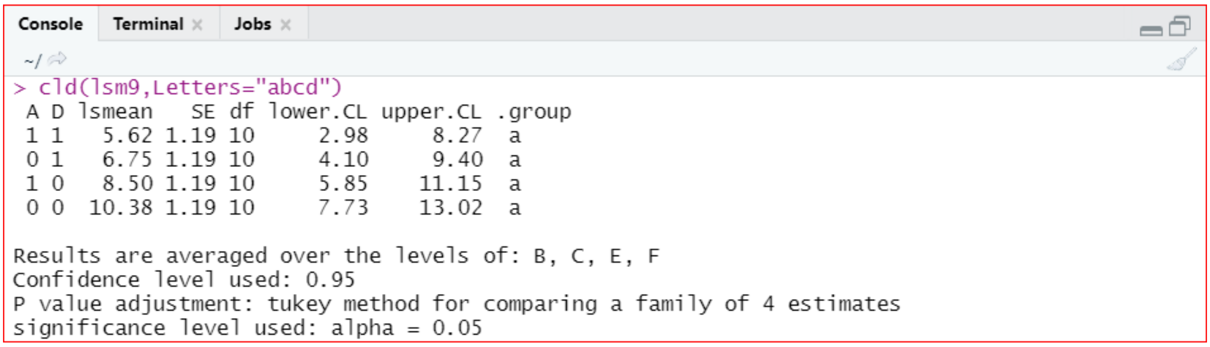
```
> cld(lsm7,Letters="abcd")
 A B lsmean   SE df lower.CL upper.CL .group
 1 1   5.88 1.19 10     3.23     8.52  a
 0 1   8.12 1.19 10     5.48    10.77  a
 1 0   8.25 1.19 10     5.60    10.90  a
 0 0   9.00 1.19 10     6.35    11.65  a

Results are averaged over the levels of: C, D, E, F
Confidence level used: 0.95
P value adjustment: tukey method for comparing a family of 4 estimates
significance level used: alpha = 0.05
```

cld(lsm8,Letters="abcd")



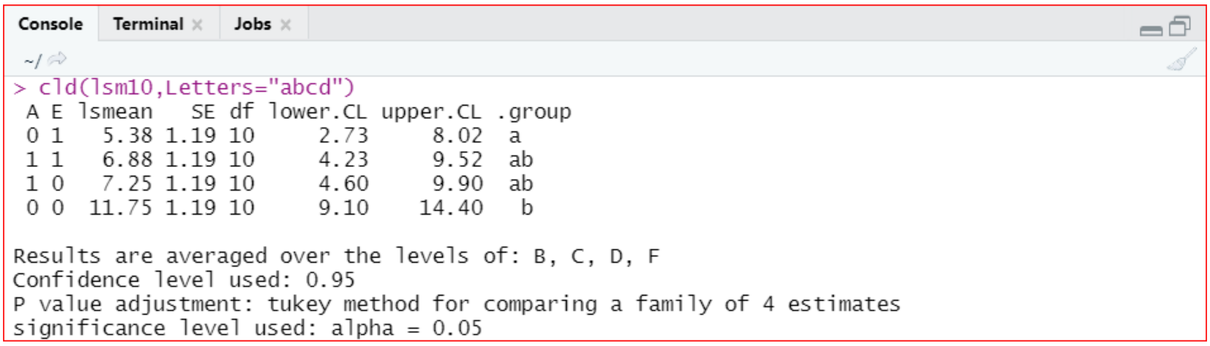
```
> cld(lsm8,Letters="abcd")
 A C lsmean   SE df lower.CL upper.CL .group
 1 1   6.25 1.19 10     3.60      8.9  a
 1 0   7.88 1.19 10     5.23     10.5  a
 0 1   8.50 1.19 10     5.85     11.1  a
 0 0   8.62 1.19 10     5.98     11.3  a

Results are averaged over the levels of: B, D, E, F
Confidence level used: 0.95
P value adjustment: tukey method for comparing a family of 4 estimates
significance level used: alpha = 0.05
```

cld(lsm9,Letters="abcd")



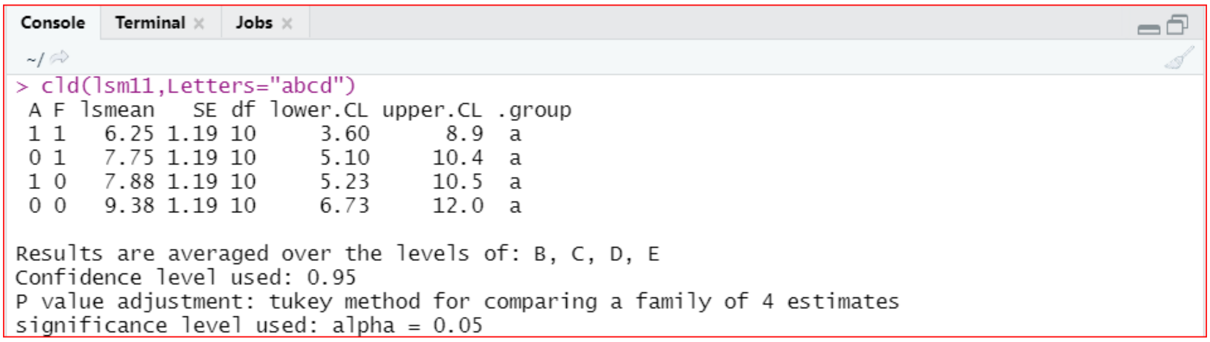
```
> cld(lsm9,Letters="abcd")
 A D lsmean   SE df lower.CL upper.CL .group
 1 1   5.62 1.19 10     2.98     8.27  a
 0 1   6.75 1.19 10     4.10     9.40  a
 1 0   8.50 1.19 10     5.85    11.15  a
 0 0  10.38 1.19 10     7.73    13.02  a

Results are averaged over the levels of: B, C, E, F
Confidence level used: 0.95
P value adjustment: tukey method for comparing a family of 4 estimates
significance level used: alpha = 0.05
```

cld(lsm10,Letters="abcd")



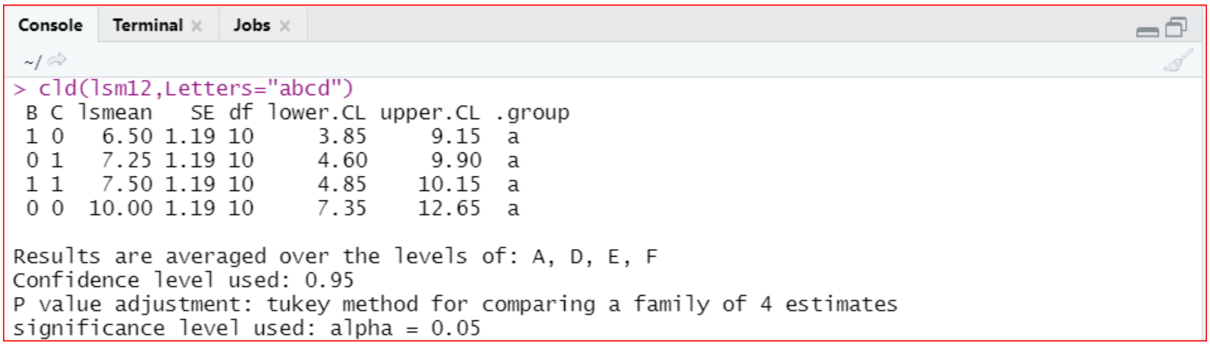
```
> cld(lsm10,Letters="abcd")
 A E lsmean   SE df lower.CL upper.CL .group
 0 1   5.38 1.19 10     2.73     8.02  a
 1 1   6.88 1.19 10     4.23     9.52  ab
 1 0   7.25 1.19 10     4.60     9.90  ab
 0 0  11.75 1.19 10     9.10    14.40   b

Results are averaged over the levels of: B, C, D, F
Confidence level used: 0.95
P value adjustment: tukey method for comparing a family of 4 estimates
significance level used: alpha = 0.05
```

cld(lsm11,Letters="abcd")



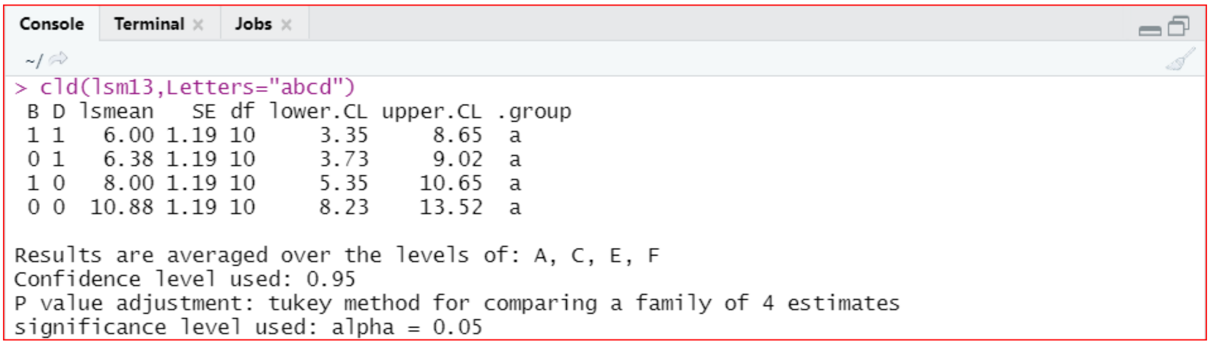
```
> cld(lsm11,Letters="abcd")
 A F lsmean   SE df lower.CL upper.CL .group
 1 1   6.25 1.19 10     3.60      8.9  a
 0 1   7.75 1.19 10     5.10     10.4  a
 1 0   7.88 1.19 10     5.23     10.5  a
 0 0   9.38 1.19 10     6.73     12.0  a

Results are averaged over the levels of: B, C, D, E
Confidence level used: 0.95
P value adjustment: tukey method for comparing a family of 4 estimates
significance level used: alpha = 0.05
```

cld(lsm12,Letters="abcd")



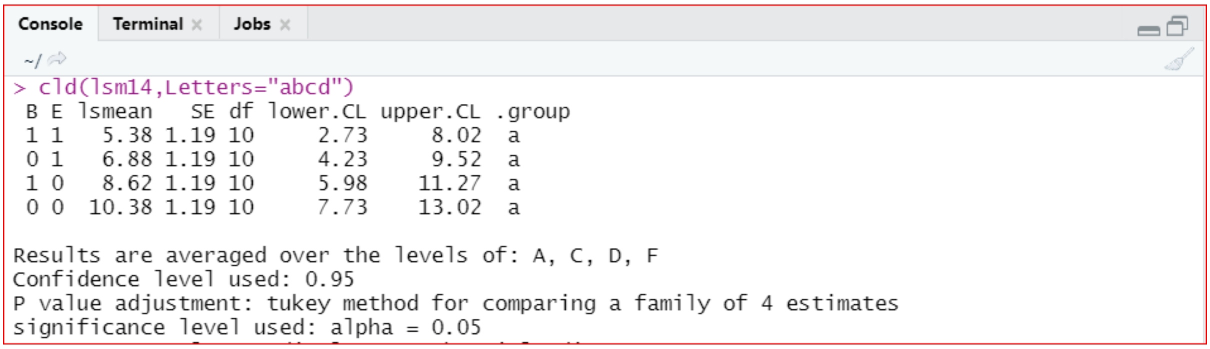
```
> cld(lsm12,Letters="abcd")
 B C lsmean   SE df lower.CL upper.CL .group
 1 0   6.50 1.19 10     3.85     9.15  a
 0 1   7.25 1.19 10     4.60     9.90  a
 1 1   7.50 1.19 10     4.85    10.15  a
 0 0  10.00 1.19 10     7.35    12.65  a

Results are averaged over the levels of: A, D, E, F
Confidence level used: 0.95
P value adjustment: tukey method for comparing a family of 4 estimates
significance level used: alpha = 0.05
```

cld(lsm13,Letters="abcd")



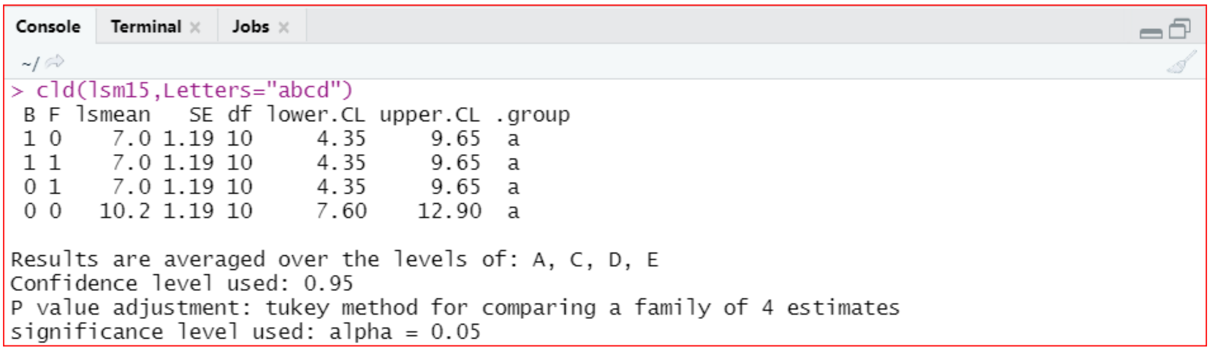
```
> cld(lsm13,Letters="abcd")
 B D lsmean   SE df lower.CL upper.CL .group
 1 1   6.00 1.19 10     3.35     8.65  a    
 0 1   6.38 1.19 10     3.73     9.02  a    
 1 0   8.00 1.19 10     5.35    10.65  a    
 0 0  10.88 1.19 10     8.23    13.52  a    

Results are averaged over the levels of: A, C, E, F 
Confidence level used: 0.95 
P value adjustment: tukey method for comparing a family of 4 estimates 
significance level used: alpha = 0.05 
```

cld(lsm14,Letters="abcd")



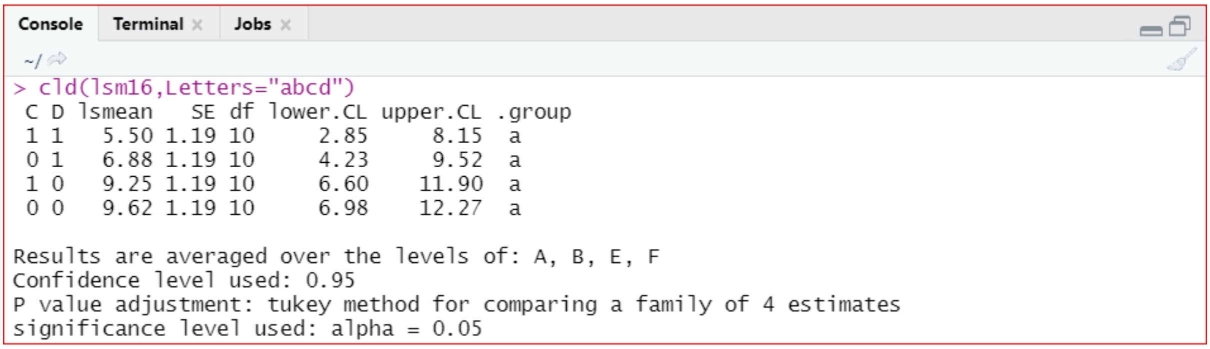
```
> cld(lsm14,Letters="abcd")
 B E lsmean   SE df lower.CL upper.CL .group
 1 1   5.38 1.19 10     2.73     8.02  a    
 0 1   6.88 1.19 10     4.23     9.52  a    
 1 0   8.62 1.19 10     5.98    11.27  a    
 0 0  10.38 1.19 10     7.73    13.02  a    

Results are averaged over the levels of: A, C, D, F 
Confidence level used: 0.95 
P value adjustment: tukey method for comparing a family of 4 estimates 
significance level used: alpha = 0.05 
```

cld(lsm15,Letters="abcd")



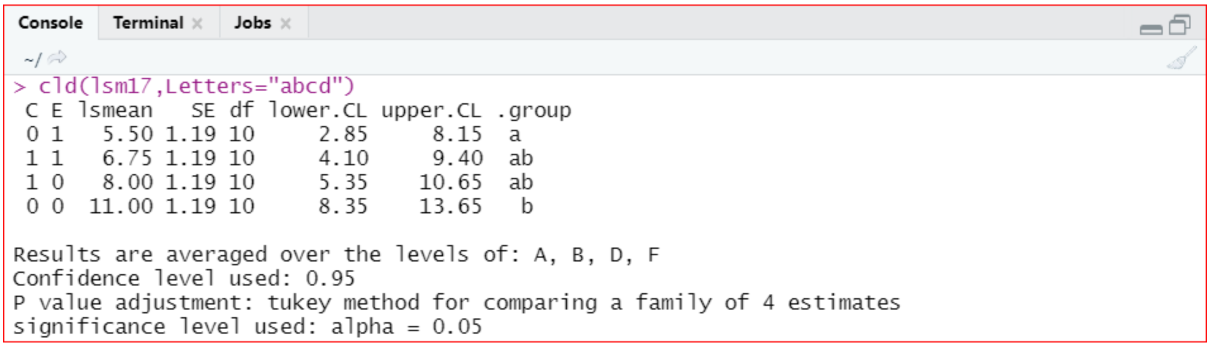
```
> cld(lsm15,Letters="abcd")
 B F lsmean   SE df lower.CL upper.CL .group
 1 0    7.0 1.19 10     4.35     9.65  a    
 1 1    7.0 1.19 10     4.35     9.65  a    
 0 1    7.0 1.19 10     4.35     9.65  a    
 0 0   10.2 1.19 10     7.60    12.90  a    

Results are averaged over the levels of: A, C, D, E 
Confidence level used: 0.95 
P value adjustment: tukey method for comparing a family of 4 estimates 
significance level used: alpha = 0.05 
```

cld(lsm16,Letters="abcd")



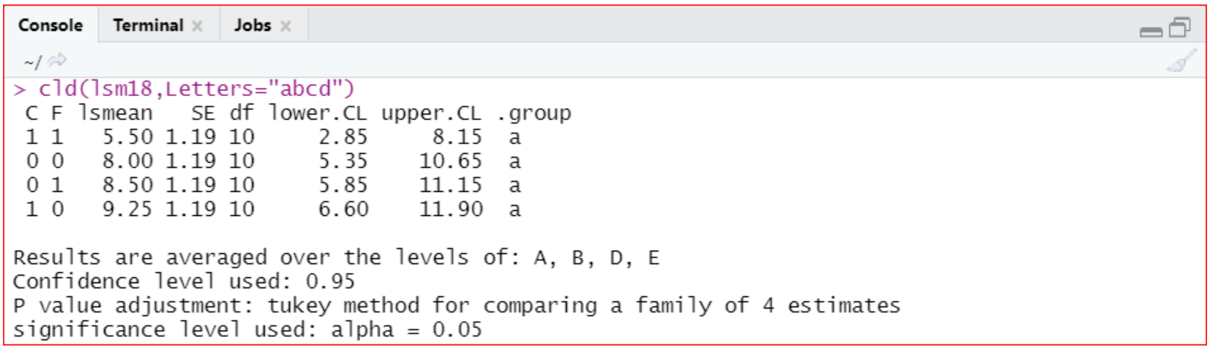
```
> cld(lsm16,Letters="abcd")
 C D lsmean   SE df lower.CL upper.CL .group
 1 1   5.50 1.19 10     2.85     8.15  a    
 0 1   6.88 1.19 10     4.23     9.52  a    
 1 0   9.25 1.19 10     6.60    11.90  a    
 0 0   9.62 1.19 10     6.98    12.27  a    

Results are averaged over the levels of: A, B, E, F 
Confidence level used: 0.95 
P value adjustment: tukey method for comparing a family of 4 estimates 
significance level used: alpha = 0.05 
```

cld(lsm17,Letters="abcd")



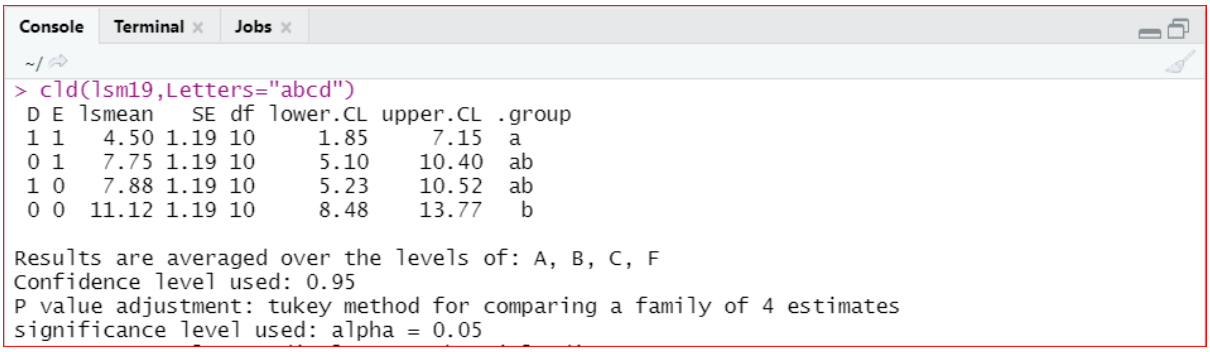
```
> cld(lsm17,Letters="abcd")
 C E lsmean   SE df lower.CL upper.CL .group
 0 1   5.50 1.19 10     2.85     8.15  a
 1 1   6.75 1.19 10     4.10     9.40  ab
 1 0   8.00 1.19 10     5.35    10.65  ab
 0 0  11.00 1.19 10     8.35    13.65   b

Results are averaged over the levels of: A, B, D, F
Confidence level used: 0.95
P value adjustment: tukey method for comparing a family of 4 estimates
significance level used: alpha = 0.05
```

cld(lsm18,Letters="abcd")



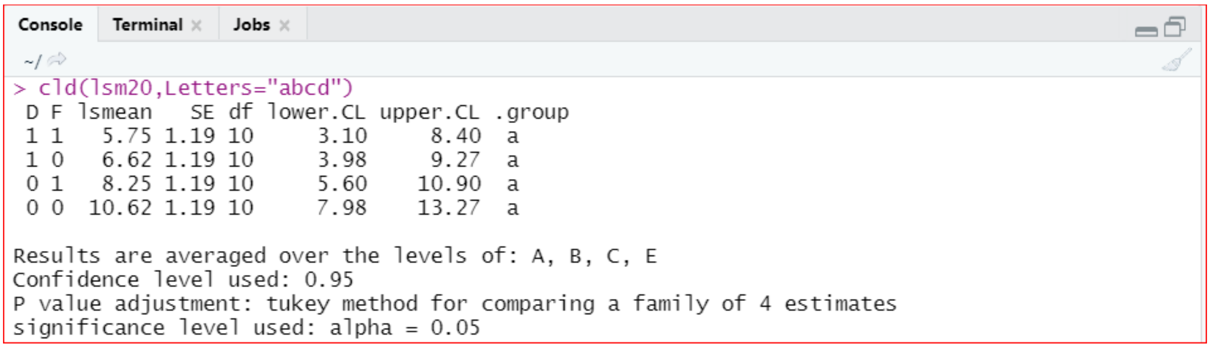
```
> cld(lsm18,Letters="abcd")
 C F lsmean   SE df lower.CL upper.CL .group
 1 1   5.50 1.19 10     2.85     8.15  a
 0 0   8.00 1.19 10     5.35    10.65  a
 0 1   8.50 1.19 10     5.85    11.15  a
 1 0   9.25 1.19 10     6.60    11.90  a

Results are averaged over the levels of: A, B, D, E
Confidence level used: 0.95
P value adjustment: tukey method for comparing a family of 4 estimates
significance level used: alpha = 0.05
```

cld(lsm19,Letters="abcd")



```
> cld(lsm19,Letters="abcd")
 D E lsmean   SE df lower.CL upper.CL .group
 1 1   4.50 1.19 10     1.85     7.15  a
 0 1   7.75 1.19 10     5.10    10.40  ab
 1 0   7.88 1.19 10     5.23    10.52  ab
 0 0  11.12 1.19 10     8.48    13.77   b

Results are averaged over the levels of: A, B, C, F
Confidence level used: 0.95
P value adjustment: tukey method for comparing a family of 4 estimates
significance level used: alpha = 0.05
```

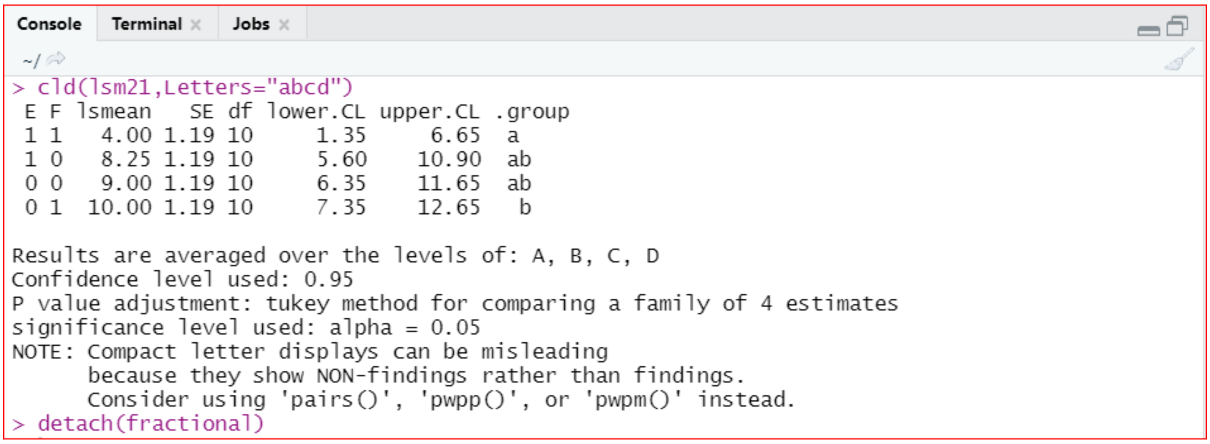
cld(lsm20,Letters="abcd")



```
> cld(lsm20,Letters="abcd")
 D F lsmean   SE df lower.CL upper.CL .group
 1 1   5.75 1.19 10     3.10     8.40  a
 1 0   6.62 1.19 10     3.98     9.27  a
 0 1   8.25 1.19 10     5.60    10.90  a
 0 0  10.62 1.19 10     7.98    13.27  a

Results are averaged over the levels of: A, B, C, E
Confidence level used: 0.95
P value adjustment: tukey method for comparing a family of 4 estimates
significance level used: alpha = 0.05
```

cld(lsm21,Letters="abcd")

detach(fractional)



```
> cld(lsm21,Letters="abcd")
 E F lsmean   SE df lower.CL upper.CL .group
 1 1   4.00 1.19 10     1.35     6.65  a
 1 0   8.25 1.19 10     5.60    10.90  ab
 0 0   9.00 1.19 10     6.35    11.65  ab
 0 1  10.00 1.19 10     7.35    12.65   b

Results are averaged over the levels of: A, B, C, D
Confidence level used: 0.95
P value adjustment: tukey method for comparing a family of 4 estimates
significance level used: alpha = 0.05
NOTE: Compact letter displays can be misleading
      because they show NON-findings rather than findings.
      Consider using 'pairs()', 'pwpp()', or 'pwpm()' instead.
> detach(fractional)
```

References:

- Cochran, W.G. and Cox, G.M. (1957). Experimental designs. John Wiley & Sons, Inc., New York.
- Das, M.N. and Giri, N.C. (1986). Design and Analysis of Experiments. Wiley Eastern Limited, New Delhi.
- Hinkelmann, K. and Kempthorne, O. (2005). Design and Analysis of Experiments. Wiley Interscience Publications, New Jersey.
- Design Resources Server. Indian Agricultural Statistics Research Institute(ICAR), New Delhi-110012, India. www.iasri.res.in/design.