Statistical analyses for two-factor experimental design with R

Jixiang Wu, South Dakota State University

Introduction

In one-factor experimental designs, only one factor is considered and investigated. In many experiments, treatments consist of two or more than two factors.When two or more than two factors, the principals of experimental designs remain the same. However, the statistical analysis for two-factor experimental designs might be very complicated depending on the combinations of these effects (random or fixed). In this chapter, we will emphasize two-factor cases. The same priciple can be applied to the cases with more than two factors.

Two-factor experimental designs without replications

First we will look at a two-factor experimental designs without relications or blocks. The data set genxenv used for our analysis can be loaded from the R package agrocolae as follows.

```
require(agricolae)

## Loading required package: agricolae

data(genxenv)
dat=genxenv
head(dat)
```

ENV	GEN	YLD
1	1	17.62333
1	2	26.98333
1	3	23.55000
1	4	24.50333
1	5	14.10667
1	6	32.27667

Data analysis for this data structure is very easy, which is similar to analyzing a RCB design data set by treating one factor as a treatment factor and the other one as a block factor.There are several different analyses depending on random or fixed effects. Our first case is to treat both genotype and environment as fixed.

Case 1: Both genotypes and environments are fixed

The codes used for ENV and GEN are integers. Therefore they should be factorized before they can be correctly used with ANOVA analysis.

```
dat=transform(dat,Yield=YLD,Env=factor(ENV),Gen=factor(GEN))
#dat$ENV=factor(dat$ENV)
#dat$GEN=factor(dat$GEN)

lm1=lm(Yield~Env+Gen,data=dat)
summary(lm1)

##
## Call:
## lm(formula = Yield ~ Env + Gen, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -27.7763  -4.3979   0.4386   4.2342  27.7517
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 19.54013    3.86800   5.052 9.99e-07 ***
## Env2        -7.50540    1.66452  -4.509 1.12e-05 ***
## Env3        54.38273    1.66452  32.672  < 2e-16 ***
## Env4        70.29547    1.66452  42.232  < 2e-16 ***
## Env5         8.81320    1.66452   5.295 3.18e-07 ***
## Gen2        -3.42800    5.26368  -0.651 0.515645
## Gen3        -4.99933    5.26368  -0.950 0.343394
## Gen4         4.44133    5.26368   0.844 0.399827
## Gen5        -9.52333    5.26368  -1.809 0.071944 .
## Gen6         9.73067    5.26368   1.849 0.066016 .
## Gen7         6.46933    5.26368   1.229 0.220526
## Gen8         6.01933    5.26368   1.144 0.254201
## Gen9        16.01467    5.26368   3.042 0.002668 **
## Gen10        4.38267    5.26368   0.833 0.406070
## Gen11        6.83067    5.26368   1.298 0.195916
## Gen12        7.78000    5.26368   1.478 0.140999
## Gen13       12.91333    5.26368   2.453 0.015030 *
## Gen14       19.36000    5.26368   3.678 0.000304 ***
## Gen15        7.88333    5.26368   1.498 0.135824
## Gen16        3.85667    5.26368   0.733 0.464621
## Gen17       -1.28333    5.26368  -0.244 0.807634
## Gen18        4.67200    5.26368   0.888 0.375848
## Gen19        7.38067    5.26368   1.402 0.162441
## Gen20       -7.88867    5.26368  -1.499 0.135561
## Gen21        8.75867    5.26368   1.664 0.097714 .
## Gen22        1.43333    5.26368   0.272 0.785673
## Gen23        3.26200    5.26368   0.620 0.536163
## Gen24        6.99067    5.26368   1.328 0.185691
```

```
## Gen25       -0.45200    5.26368  -0.086 0.931656
## Gen26       -0.06933    5.26368  -0.013 0.989504
## Gen27       11.17533    5.26368   2.123 0.035000 *
## Gen28       10.27667    5.26368   1.952 0.052318 .
## Gen29        6.53267    5.26368   1.241 0.216058
## Gen30        8.87933    5.26368   1.687 0.093212 .
## Gen31       -0.52133    5.26368  -0.099 0.921205
## Gen32        5.45133    5.26368   1.036 0.301641
## Gen33        5.34667    5.26368   1.016 0.310993
## Gen34        8.32867    5.26368   1.582 0.115196
## Gen35        3.21933    5.26368   0.612 0.541503
## Gen36        0.14067    5.26368   0.027 0.978707
## Gen37       11.47467    5.26368   2.180 0.030451 *
## Gen38        8.37867    5.26368   1.592 0.113044
## Gen39        5.24667    5.26368   0.997 0.320106
## Gen40        6.27400    5.26368   1.192 0.234725
## Gen41       -1.62333    5.26368  -0.308 0.758103
## Gen42        5.46133    5.26368   1.038 0.300758
## Gen43        7.87600    5.26368   1.496 0.136186
## Gen44        2.09867    5.26368   0.399 0.690543
## Gen45        5.50200    5.26368   1.045 0.297183
## Gen46       -5.74133    5.26368  -1.091 0.276724
## Gen47        5.23200    5.26368   0.994 0.321458
## Gen48        5.44800    5.26368   1.035 0.301936
## Gen49       -1.04400    5.26368  -0.198 0.842984
## Gen50        1.37200    5.26368   0.261 0.794633
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.323 on 196 degrees of freedom
## Multiple R-squared:  0.9486, Adjusted R-squared:  0.9348
## F-statistic:  68.3 on 53 and 196 DF,  p-value: < 2.2e-16
```

We may also run aov for regular F-tests.

```
aov1=aov(Yield~Env+Gen,data=dat)
summary(aov1)
```

```
##              Df Sum Sq Mean Sq F value   Pr(>F)
## Env           4 242922   60731 876.774  < 2e-16 ***
## Gen          49   7830     160   2.307 2.83e-05 ***
## Residuals   196  13576      69
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

When there are no replications available, genotype-by-environment interaction effects are confound with random errors. Therefore, genotype-by-environment interactions and random errors can be separated from each other. The model used for the above analysis is equivalent to that for one-factor with RCB design. We can either treat ENV as block or GEN as block.

After we conduct ANOVA analysis such as F-tests, again it is important to conduct mean comparisons among treatmemts. In the same manner, we can try several multiple comparisons, as you have seen in Chapters 3, 4, and 5.

```
require(agricolae)
mod=aov(Yield~Env+Gen,data=dat)
a=summary(mod)
dfe=mod$df.residual
rn=nrow(a[[1]])
mse=a[[1]][rn,3]
res=LSD.test(dat$Yield, dat$Gen, DFerror=dfe, MSerror=mse)
res
```

```
## $statistics
##       Mean       CV  MSerror      LSD
##   49.24373 16.90085 69.26587 10.38072
##
## $parameters
##    Df ntr  t.value alpha       test   name.t
##   196  50 1.972141  0.05 Fisher-LSD dat$Gen
##
## $means
##    dat$Yield      std r      LCL      UCL       Min       Max
## 1   44.73733 32.55482 5 37.39705 52.07761 17.216667  82.85333
## 10  49.12000 40.48756 5 41.77972 56.46028 13.493333  99.30333
## 11  51.56800 30.84790 5 44.22772 58.90828 19.290000  93.99667
## 12  52.51733 35.94814 5 45.17705 59.85761 15.406667 105.09667
## 13  57.65067 43.27937 5 50.31039 64.99095 17.623333 115.90333
## 14  64.09733 45.74939 5 56.75705 71.43761 21.623333 125.92333
## 15  52.62067 38.08023 5 45.28039 59.96095 18.986667 103.66333
## 16  48.59400 37.48420 5 41.25372 55.93428 12.826667 100.56667
## 17  43.45400 30.58004 5 36.11372 50.79428 15.513333  83.68333
## 18  49.40933 29.50659 5 42.06905 56.74961 17.916667  83.28000
## 19  52.11800 38.79318 5 44.77772 59.45828 18.546667  98.24000
## 2   41.30933 26.25397 5 33.96905 48.64961 13.530000  73.83667
## 20  36.84867 25.66660 5 29.50839 44.18895  9.933333  67.17333
## 21  53.49600 41.25961 5 46.15572 60.83628 17.206667 108.85333
## 22  46.17067 34.89814 5 38.83039 53.51095 15.666667  98.92667
## 23  47.99933 30.50167 5 40.65905 55.33961 15.970000  81.68000
## 24  51.72800 35.76393 5 44.38772 59.06828 18.560000  93.37667
## 25  44.28533 31.68427 5 36.94505 51.62561 12.070000  81.48667
## 26  44.66800 36.89362 5 37.32772 52.00828 10.216667  86.38000
## 27  55.91267 38.47252 5 48.57239 63.25295 18.783333 108.54667
## 28  55.01400 45.72438 5 47.67372 62.35428 14.310000 119.46667
## 29  51.27000 43.61796 5 43.92972 58.61028 14.200000 108.40667
## 3   39.73800 23.44033 5 32.39772 47.07828 17.733333  71.78000
## 30  53.61667 48.21166 5 46.27639 60.95695 13.886667 126.46667
## 31  44.21600 33.28330 5 36.87572 51.55628  9.900000  87.70000
## 32  50.18867 30.00424 5 42.84839 57.52895 17.096667  93.79333
## 33  50.08400 35.97018 5 42.74372 57.42428 18.366667  97.54333
```

```
## 34  53.06600 44.81368 5 45.72572 60.40628 14.190000 117.64000
## 35  47.95667 31.64728 5 40.61639 55.29695 15.706667  84.62000
## 36  44.87800 35.30855 5 37.53772 52.21828 13.946667  90.73333
## 37  56.21200 32.37567 5 48.87172 63.55228 22.750000  98.92333
## 38  53.11600 38.60790 5 45.77572 60.45628 19.313333 101.00000
## 39  49.98400 36.97341 5 42.64372 57.32428 13.893333 100.31667
## 4   49.17867 32.27384 5 41.83839 56.51895 20.080000  88.26667
## 40  51.01133 35.81602 5 43.67105 58.35161 18.786667 100.99667
## 41  43.11400 30.96953 5 35.77372 50.45428 14.600000  84.17333
## 42  50.19867 36.14306 5 42.85839 57.53895 18.583333  98.85667
## 43  52.61333 40.78926 5 45.27305 59.95361 15.990000 101.75333
## 44  46.83600 34.78686 5 39.49572 54.17628 13.086667  86.79667
## 45  50.23933 35.09432 5 42.89905 57.57961 17.180000  99.40000
## 46  38.99600 27.77105 5 31.65572 46.33628 14.446667  77.47667
## 47  49.96933 36.64249 5 42.62905 57.30961 17.483333 104.58333
## 48  50.18533 32.24445 5 42.84505 57.52561 22.240000  94.51667
## 49  43.69333 26.76515 5 36.35305 51.03361 18.663333  80.75333
## 5   35.21400 25.26154 5 27.87372 42.55428 13.690000  65.74000
## 50  46.10933 32.40048 5 38.76905 53.44961 17.013333  85.18000
## 6   54.46800 38.67896 5 47.12772 61.80828 16.620000 100.51333
## 7   51.20667 32.50882 5 43.86639 58.54695 17.986667  96.80000
## 8   50.75667 33.79318 5 43.41639 58.09695 18.923333  91.75000
## 9   60.75200 49.47199 5 53.41172 68.09228 14.436667 126.08000
##
## $comparison
## NULL
##
## $groups
##    trt    means         M
## 1   14 64.09733         a
## 2    9 60.75200        ab
## 3   13 57.65067       abc
## 4   37 56.21200      abcd
## 5   27 55.91267      abcd
## 6   28 55.01400     abcde
## 7    6 54.46800    abcdef
## 8   30 53.61667    bcdefg
## 9   21 53.49600    bcdefg
## 10  38 53.11600   bcdefgh
## 11  34 53.06600   bcdefgh
## 12  15 52.62067   bcdefgh
## 13  43 52.61333   bcdefgh
## 14  12 52.51733   bcdefgh
## 15  19 52.11800   bcdefgh
## 16  24 51.72800   bcdefgh
## 17  11 51.56800  bcdefghi
## 18  29 51.27000  bcdefghi
## 19   7 51.20667  bcdefghi
## 20  40 51.01133  bcdefghi
## 21   8 50.75667  bcdefghi
```

```
## 22  45 50.23933   cdefghi
## 23  42 50.19867   cdefghi
## 24  32 50.18867   cdefghi
## 25  48 50.18533   cdefghi
## 26  33 50.08400  cdefghij
## 27  39 49.98400  cdefghij
## 28  47 49.96933  cdefghij
## 29  18 49.40933  cdefghij
## 30   4 49.17867 cdefghijk
## 31  10 49.12000 cdefghijk
## 32  16 48.59400 cdefghijk
## 33  23 47.99933 cdefghijk
## 34  35 47.95667 cdefghijk
## 35  44 46.83600 defghijkl
## 36  22 46.17067 defghijkl
## 37  50 46.10933 defghijkl
## 38  36 44.87800 efghijklm
## 39   1 44.73733 efghijklm
## 40  26 44.66800 efghijklm
## 41  25 44.28533  fghijklm
## 42  31 44.21600  fghijklm
## 43  49 43.69333   ghijklm
## 44  17 43.45400   ghijklm
## 45  41 43.11400    hijklm
## 46   2 41.30933     ijklm
## 47   3 39.73800      jklm
## 48  46 38.99600       klm
## 49  20 36.84867        lm
## 50   5 35.21400         m
```

You may also conduct mean comparisons between environments as follows.

```
res=LSD.test(dat$Yield, dat$Env, DFerror=dfe, MSerror=mse)
res

## $statistics
##       Mean       CV  MSerror      LSD
##   49.24373 16.90085 69.26587 3.282673
##
## $parameters
##    Df ntr  t.value alpha       test  name.t
##   196   5 1.972141  0.05 Fisher-LSD dat$Env
##
## $means
##   dat$Yield       std  r      LCL      UCL      Min       Max
## 1  24.04653  7.340494 50 21.72533 26.36773 12.65667  42.47000
## 2  16.54113  3.192049 50 14.21993 18.86233  9.90000  26.13000
## 3  78.42927  9.398327 50 76.10807 80.75047 58.09000 100.79667
## 4  94.34200 16.242465 50 92.02080 96.66320 56.98333 126.46667
## 5  32.85973  4.542469 50 30.53853 35.18093 23.39667  48.00667
```

```
##
## $comparison
## NULL
##
## $groups
##   trt    means M
## 1   4 94.34200 a
## 2   3 78.42927 b
## 3   5 32.85973 c
## 4   1 24.04653 d
## 5   2 16.54113 e
```

You can try this data analysis by using linear mixed model approaches.The following R scripts show how to run an analysis with a linear mixed model approach, MINQUE and with the jackknife resampling procedure.

```
require(minque)

## Loading required package: minque

## Loading required package: klaR

## Loading required package: MASS

## Loading required package: Matrix

res1=lmm.jack(Yield~Env+Gen,data=dat)[[1]]
```

Estimated variance component for random error.

```
data.frame(res1$Var[[1]])
```

	Estimate	SE	PValue	X2.5.LL	X97.5.UL
V(e)	69.26021	2.474358	0	60.52011	78.0003

Estimated fixed effects: population mean, enviromental effects, and genotypic effects

```
data.frame(res1$FixedEffect[[1]])
```

	Estimate	SE	PValue	X2.5.LL	X97.5.UL
mu	49.2590913	0.2640429	0.0000000	48.3264210	50.1917615
Env(1)	-25.1747544	0.2095417	0.0000000	-25.9149119	-24.4345968
Env(2)	-32.7207993	0.2473575	0.0000000	-33.5945323	-31.8470662
Env(3)	29.2008644	0.5892729	0.0000000	27.1193941	31.2823347
Env(4)	45.0807278	0.7028253	0.0000000	42.5981601	47.5632955
Env(5)	-16.3860386	0.2998348	0.0000000	-17.4451356	-15.3269415
Gen(1)	-4.5309288	1.4236769	0.0436970	-9.5597382	0.4978806

Gen(2)	-7.8536158	1.5519177	0.0027188	-13.3354061	-2.3718255
Gen(3)	-9.3002987	2.8531906	0.0387075	-19.3785343	0.7779369
Gen(4)	-0.0830955	0.7395458	0.9923865	-2.6953699	2.5291789
Gen(5)	-13.9847565	1.2220158	0.0000046	-18.3012447	-9.6682684
Gen(6)	5.2673986	1.0381384	0.0026702	1.6004149	8.9343822
Gen(7)	1.9940816	0.9240867	0.2138951	-1.2700409	5.2582041
Gen(8)	1.5314615	0.7436966	0.2466384	-1.0954747	4.1583977
Gen(9)	11.4487770	2.4735187	0.0049462	2.7116446	20.1859094
Gen(10)	-0.1385492	1.0673721	0.9916008	-3.9087943	3.6316958
Gen(11)	2.5736475	1.7610943	0.5254407	-3.6470106	8.7943057
Gen(12)	3.0931836	1.9426803	0.4542218	-3.7688852	9.9552525
Gen(13)	8.3814091	1.7894453	0.0045750	2.0606076	14.7022107
Gen(14)	14.8439041	1.4419305	0.0000112	9.7506180	19.9371902
Gen(15)	3.4418517	0.8008335	0.0079600	0.6130927	6.2706107
Gen(16)	-0.6413550	0.8251771	0.8744636	-3.5561023	2.2733923
Gen(17)	-5.7545186	0.6637381	0.0000463	-8.0990200	-3.4100172
Gen(18)	0.2134199	1.7096749	0.9918315	-5.8256110	6.2524509
Gen(19)	2.7892679	0.8832343	0.0454195	-0.3305529	5.9090886
Gen(20)	-12.4070229	3.5514725	0.0268503	-24.9517778	0.1377320
Gen(21)	4.1994374	1.6644378	0.1232190	-1.6798040	10.0786787
Gen(22)	-3.0129388	1.4444001	0.2373983	-8.1149480	2.0890703
Gen(23)	-1.2308369	1.3325172	0.8144641	-5.9376454	3.4759717
Gen(24)	2.2636061	1.2971808	0.3771832	-2.3183850	6.8455972
Gen(25)	-4.9545564	0.9216622	0.0017867	-8.2101150	-1.6989978
Gen(26)	-4.3669026	1.4386790	0.0551179	-9.4487033	0.7148981
Gen(27)	6.6743364	0.9263605	0.0002022	3.4021823	9.9464905
Gen(28)	6.2622299	3.7065553	0.4042665	-6.8303193	19.3547791
Gen(29)	1.9390645	1.3109601	0.5155816	-2.6915989	6.5697279
Gen(30)	4.3250873	2.5233501	0.3921664	-4.5880631	13.2382377
Gen(31)	-5.0216277	0.4291553	0.0000038	-6.5375195	-3.5057358
Gen(32)	0.8289090	1.9338273	0.9636154	-6.0018888	7.6597067
Gen(33)	0.8253136	0.5779725	0.5441723	-1.2162407	2.8668679
Gen(34)	3.8728620	1.9328158	0.2667811	-2.9543627	10.7000868
Gen(35)	-1.3153494	0.8605997	0.4884779	-4.3552189	1.7245201
Gen(36)	-4.3723904	0.7507947	0.0010071	-7.0243990	-1.7203817
Gen(37)	6.9642771	1.2074131	0.0010800	2.6993696	11.2291845

Gen(38)	3.8038382	0.9521284	0.0124657	0.4406650	7.1670114
Gen(39)	0.7055768	0.4324676	0.4336875	-0.8220148	2.2331684
Gen(40)	1.7288006	0.4218011	0.0106788	0.2388857	3.2187155
Gen(41)	-6.1109712	1.0558832	0.0010537	-9.8406343	-2.3813081
Gen(42)	0.9287053	0.5363611	0.3836795	-0.9658663	2.8232770
Gen(43)	3.2521243	1.7526566	0.3267292	-2.9387296	9.4429782
Gen(44)	-2.4939878	1.2346806	0.2607718	-6.8552115	1.8672358
Gen(45)	0.9614119	0.8095109	0.6798057	-1.8979979	3.8208217
Gen(46)	-10.2638297	1.0085576	0.0000124	-13.8263261	-6.7013332
Gen(47)	0.6136088	1.4904853	0.9661629	-4.6511856	5.8784032
Gen(48)	0.9617969	0.6105250	0.4631719	-1.1947415	3.1183353
Gen(49)	-5.5696030	1.6521621	0.0324986	-11.4054833	0.2662773
Gen(50)	-3.2822538	0.8948861	0.0205039	-6.4432318	-0.1212758

Case 2: Both genotypes and environments are assumed random

```
require(minque)
res1=lmm.jack(Yield~1|Env+Gen,data=dat)[[1]]
```

Estimated variance components for environment, genotype, and random error.

```
data.frame(res1$Var[[1]])
```

	Estimate	SE	PValue	X2.5.LL	X97.5.UL
V(Env)	1212.64893	24.055295	0.0000000	1127.679162	1297.61869
V(Gen)	17.94052	2.444239	0.0001749	9.306807	26.57423
V(e)	69.40249	2.371721	0.0000000	61.024929	77.78004

Estimated proportional variance components for environment, genotype, and random error.

```
data.frame(res1$PVar[[1]])
```

	Estimate	SE	PValue	X2.5.LL	X97.5.UL
V(Env)/VP	0.9327985	0.0023149	0.0000000	0.9246217	0.9409753
V(Gen)/VP	0.0137965	0.0017996	0.0001242	0.0074398	0.0201533
V(e)/VP	0.0534050	0.0021149	0.0000000	0.0459347	0.0608752

Estimated population mean

```
data.frame(res1$FixedEffect[[1]])
```

Estimate	SE	PValue	X2.5.LL	X97.5.UL

mu	49.2456	0.2153434	0	48.48495	50.00625

Predicted random effects for enviroments and genotypes

```
data.frame(res1$RandomEffect[[1]])
```

	Pre	SE	PValue	X2.5.LL	X97.5.UL
Env(1)	-25.1378032	0.3430669	0.0000000	-26.3496078	-23.9259987
Env(2)	-32.6980159	0.2436908	0.0000000	-33.5587973	-31.8372345
Env(3)	29.1670771	0.4147866	0.0000000	27.7019394	30.6322147
Env(4)	45.0520522	0.5431687	0.0000000	43.1334344	46.9706700
Env(5)	-16.3833101	0.3770029	0.0000000	-17.7149858	-15.0516344
Gen(1)	-3.3319487	1.0315354	0.0405468	-6.9756090	0.3117117
Gen(2)	-5.7649978	1.1709009	0.0032776	-9.9009345	-1.6290612
Gen(3)	-6.8365667	1.7362477	0.0135930	-12.9694601	-0.7036733
Gen(4)	-0.0616837	0.3737900	0.9898032	-1.3820103	1.2586430
Gen(5)	-10.2440992	1.4798045	0.0002756	-15.4711661	-5.0170322
Gen(6)	3.7476556	0.8668312	0.0076685	0.6857751	6.8095361
Gen(7)	1.4117740	0.8351517	0.4037994	-1.5382062	4.3617542
Gen(8)	1.1320845	1.1695179	0.7939242	-2.9989667	5.2631357
Gen(9)	8.3236254	1.7009342	0.0034157	2.3154690	14.3317818
Gen(10)	-0.0844894	1.0347726	0.9936178	-3.7395842	3.5706053
Gen(11)	1.7071766	0.7724869	0.1981305	-1.0214547	4.4358078
Gen(12)	2.3485000	1.5089329	0.4733264	-2.9814565	7.6784565
Gen(13)	6.1572397	1.4968649	0.0104462	0.8699107	11.4445687
Gen(14)	10.8564834	1.5813414	0.0002938	5.2707608	16.4422059
Gen(15)	2.4213652	0.6861442	0.0254112	-0.0022804	4.8450108
Gen(16)	-0.5185721	0.7082304	0.8902525	-3.0202320	1.9830878
Gen(17)	-4.1354318	0.7312214	0.0012450	-6.7183021	-1.5525614
Gen(18)	0.1397080	1.0869041	0.9916600	-3.6995292	3.9789452
Gen(19)	2.0893570	0.8376008	0.1287851	-0.8692741	5.0479882
Gen(20)	-9.0667129	1.9171643	0.0042921	-15.8386521	-2.2947736
Gen(21)	3.1508874	1.0861026	0.0681341	-0.6855189	6.9872937
Gen(22)	-2.2034248	0.9145467	0.1467808	-5.4338493	1.0269997
Gen(23)	-0.8909641	0.8638394	0.7628597	-3.9422769	2.1603487
Gen(24)	1.8262658	0.5563643	0.0373406	-0.1389622	3.7914939
Gen(25)	-3.6173103	0.7332188	0.0032333	-6.2072362	-1.0273844
Gen(26)	-3.2976049	0.7475427	0.0067500	-5.9381267	-0.6570832

Gen(27)	4.8897263	0.5793789	0.0000576	2.8432044	6.9362483
Gen(28)	4.3026640	1.9988661	0.2155706	-2.7578683	11.3631962
Gen(29)	1.5555623	1.5496697	0.7766064	-3.9182876	7.0294121
Gen(30)	3.0272979	1.4863483	0.2546587	-2.2228835	8.2774793
Gen(31)	-3.6516865	0.3018812	0.0000029	-4.7180120	-2.5853609
Gen(32)	0.7612435	0.8518470	0.8278014	-2.2477088	3.7701958
Gen(33)	0.6307116	0.3100217	0.2554980	-0.4643682	1.7257914
Gen(34)	2.8579938	1.6742954	0.3956195	-3.0560672	8.7720548
Gen(35)	-0.9621686	0.5712078	0.4067868	-2.9798280	1.0554908
Gen(36)	-3.1489557	0.5371542	0.0009598	-5.0463289	-1.2515826
Gen(37)	5.0230507	0.8837385	0.0012009	1.9014490	8.1446524
Gen(38)	2.8271299	0.6771273	0.0095326	0.4353342	5.2189255
Gen(39)	0.5049574	0.4570045	0.7247435	-1.1093054	2.1192201
Gen(40)	1.2647299	0.4595658	0.0861764	-0.3585798	2.8880397
Gen(41)	-4.4030800	0.9284663	0.0042142	-7.6826725	-1.1234874
Gen(42)	0.7004611	0.5266875	0.5996696	-1.1599408	2.5608630
Gen(43)	2.4052288	0.9888693	0.1417053	-1.0877231	5.8981808
Gen(44)	-1.7881783	0.7496707	0.1522606	-4.4362167	0.8598601
Gen(45)	0.7105797	0.4837730	0.5212976	-0.9982363	2.4193957
Gen(46)	-7.4844375	0.6511136	0.0000044	-9.7843455	-5.1845294
Gen(47)	0.4453371	1.0868037	0.9664420	-3.3935457	4.2842198
Gen(48)	0.6486539	0.7041206	0.8155778	-1.8384892	3.1357970
Gen(49)	-4.0891417	1.0568601	0.0150746	-7.8222554	-0.3560279
Gen(50)	-2.2859961	0.4410402	0.0023052	-3.8438685	-0.7281236

Case 3: genotypes are random while environments are assumed fixed

```
require(minque)
res1=lmm.jack(Yield~Env|Gen,data=dat)[[1]]
```

Estimated variance components for genotype and random error.

```
data.frame(res1$Var[[1]])
```

	Estimate	SE	PValue	X2.5.LL	X97.5.UL
V(Gen)	18.11014	3.136984	0.001073	7.029472	29.19081
V(e)	69.28976	3.301752	0.000000	57.627085	80.95243

Estimated proportional variance components for genotype and random error.

```
data.frame(res1$PVar[[1]])
```

	Estimate	SE	PValue	X2.5.LL	X97.5.UL
V(Gen)/VP	0.2064461	0.0283396	0.0001856	0.1063430	0.3065492
V(e)/VP	0.7935539	0.0283396	0.0000000	0.6934508	0.8936570

Estimated population mean and environmental effects

```
data.frame(res1$FixedEffect[[1]])
```

	Estimate	SE	PValue	X2.5.LL	X97.5.UL
mu	49.23282	0.2042358	0	48.51140	49.95423
Env(1)	-25.20991	0.3859720	0	-26.57326	-23.84655
Env(2)	-32.69488	0.1827025	0	-33.34024	-32.04953
Env(3)	29.18368	0.2596341	0	28.26659	30.10078
Env(4)	45.10533	0.5366053	0	43.20989	47.00076
Env(5)	-16.38422	0.2958276	0	-17.42916	-15.33928

Predicted genotypic effects

```
data.frame(res1$RandomEffect[[1]])
```

	Pre	SE	PValue	X2.5.LL	X97.5.UL
Gen(1)	-3.2925771	0.9926546	0.0353760	-6.7988999	0.2137458
Gen(2)	-5.7770202	1.8288139	0.0453571	-12.2368822	0.6828417
Gen(3)	-6.8949310	2.2999956	0.0584855	-15.0191334	1.2292713
Gen(4)	-0.0468833	0.5189631	0.9932881	-1.8800003	1.7862336
Gen(5)	-10.2720965	1.7283577	0.0008682	-16.3771203	-4.1670727
Gen(6)	3.8218110	0.6958110	0.0015344	1.3640195	6.2796024
Gen(7)	1.4536942	0.6816512	0.2219242	-0.9540809	3.8614693
Gen(8)	1.1027357	1.1782769	0.8089117	-3.0592550	5.2647264
Gen(9)	8.4501443	2.0100927	0.0091375	1.3499568	15.5503318
Gen(10)	-0.0991480	0.6900110	0.9909270	-2.5364521	2.3381560
Gen(11)	1.7397800	0.7956522	0.2050763	-1.0706774	4.5502374
Gen(12)	2.3380711	1.4015476	0.4149416	-2.6125716	7.2887139
Gen(13)	6.0842817	1.4747554	0.0102587	0.8750494	11.2935141
Gen(14)	10.8589922	1.6950316	0.0004972	4.8716852	16.8462993
Gen(15)	2.5237572	0.7419930	0.0309987	-0.0971615	5.1446758
Gen(16)	-0.4568661	0.8131703	0.9382594	-3.3292019	2.4154697
Gen(17)	-4.0389763	0.8622762	0.0045731	-7.0847674	-0.9931852
Gen(18)	-0.0558708	1.6194795	0.9951745	-5.7763073	5.6645658

Gen(19)	2.0727429	0.8456284	0.1376706	-0.9142437	5.0597295
Gen(20)	-9.1376610	2.0118016	0.0055936	-16.2438846	-2.0314375
Gen(21)	3.1106338	0.8309552	0.0182564	0.1754770	6.0457906
Gen(22)	-2.2593555	0.9459809	0.1515458	-5.6008141	1.0821031
Gen(23)	-0.9197800	1.2720704	0.8933025	-5.4130743	3.5735144
Gen(24)	1.8306636	0.4360385	0.0092113	0.2904586	3.3708685
Gen(25)	-3.6052863	0.6828933	0.0020281	-6.0174490	-1.1931237
Gen(26)	-3.3154719	0.6534731	0.0026712	-5.6237143	-1.0072295
Gen(27)	4.8330676	0.7328805	0.0003995	2.2443366	7.4217985
Gen(28)	4.2528983	1.6354905	0.1092700	-1.5240934	10.0298901
Gen(29)	1.4575684	0.7281532	0.2675331	-1.1144643	4.0296010
Gen(30)	3.2316814	1.9006334	0.3989035	-3.4818665	9.9452294
Gen(31)	-3.6525254	0.3347551	0.0000069	-4.8349702	-2.4700806
Gen(32)	0.7119810	1.1569453	0.9251438	-3.3746606	4.7986226
Gen(33)	0.5735561	0.4319352	0.6008408	-0.9521552	2.0992674
Gen(34)	2.7469074	1.0533399	0.1080125	-0.9737722	6.4675871
Gen(35)	-0.9435256	0.6258015	0.4998010	-3.1540247	1.2669735
Gen(36)	-3.1873631	0.3879233	0.0000715	-4.5576123	-1.8171138
Gen(37)	5.0457848	0.8081741	0.0006031	2.1910966	7.9004730
Gen(38)	2.8182471	0.4068681	0.0002744	1.3810797	4.2554144
Gen(39)	0.5482511	0.2816709	0.2888990	-0.4466863	1.5431885
Gen(40)	1.2189884	0.8044903	0.4956755	-1.6226873	4.0606642
Gen(41)	-4.4865850	0.8642215	0.0022798	-7.5392475	-1.4339225
Gen(42)	0.7078623	0.6182534	0.7032340	-1.4759748	2.8916995
Gen(43)	2.4798292	0.9911789	0.1273131	-1.0212809	5.9809392
Gen(44)	-1.7763655	0.5092224	0.0270640	-3.5750758	0.0223449
Gen(45)	0.7357102	0.6849131	0.7409343	-1.6835870	3.1550074
Gen(46)	-7.4813781	0.9930884	0.0001426	-10.9892332	-3.9735230
Gen(47)	0.5676648	0.2265718	0.1266162	-0.2326478	1.3679775
Gen(48)	0.7028183	0.3925184	0.3558292	-0.6836621	2.0892986
Gen(49)	-4.0686056	1.1444534	0.0244025	-8.1111224	-0.0260888
Gen(50)	-2.2518518	0.4009762	0.0013094	-3.6682075	-0.8354960

Case 4: Genotypes are fixed while environments are assumed random

```
require(minque)
res1=lmm.jack(Yield~Gen|Env,data=dat)[[1]]
```

Estimated variance components for environment and random error.

```r
data.frame(res1$Var[[1]])
```

	Estimate	SE	PValue	X2.5.LL	X97.5.UL
V(Env)	1211.14486	15.999964	0e+00	1154.62869	1267.66103
V(e)	69.25773	4.239908	2e-07	54.28124	84.23423

Estimated proportional variance components for environment and random error.

```r
data.frame(res1$PVar[[1]])
```

	Estimate	SE	PValue	X2.5.LL	X97.5.UL
V(Env)/VP	0.9459062	0.0032927	0e+00	0.9342754	0.9575370
V(e)/VP	0.0540938	0.0032927	2e-07	0.0424630	0.0657246

Estimated population mean and genotypic effects

```r
data.frame(res1$FixedEffect[[1]])
```

	Estimate	SE	PValue	X2.5.LL	X97.5.UL
mu	49.2621762	0.2422217	0.0000000	48.4065842	50.1177682
Gen(1)	-4.4419535	1.1164102	0.0127748	-8.3854142	-0.4984928
Gen(2)	-8.0333369	2.2720353	0.0251445	-16.0587761	-0.0078977
Gen(3)	-9.3561555	2.3242956	0.0119124	-17.5661919	-1.1461191
Gen(4)	-0.0246507	0.7468628	0.9952179	-2.6627708	2.6134694
Gen(5)	-14.0161025	2.0266120	0.0002776	-21.1746405	-6.8575646
Gen(6)	5.1352878	1.2565226	0.0108666	0.6969126	9.5736630
Gen(7)	1.8515393	1.1248527	0.4262206	-2.1217425	5.8248211
Gen(8)	1.3606368	1.9057280	0.8962772	-5.3709066	8.0921803
Gen(9)	11.5094010	2.4129144	0.0040547	2.9863393	20.0324628
Gen(10)	-0.1906237	1.8509540	0.9927822	-6.7286905	6.3474432
Gen(11)	2.2232857	1.0294307	0.2133239	-1.4129400	5.8595114
Gen(12)	3.2385037	1.3620135	0.1540214	-1.5724937	8.0495012
Gen(13)	8.6214106	2.7285680	0.0453003	-1.0166246	18.2594458
Gen(14)	14.8490519	2.2615827	0.0004130	6.8605344	22.8375695
Gen(15)	3.3960276	0.8319175	0.0109435	0.4574714	6.3345838
Gen(16)	-0.6326246	1.1042992	0.9356999	-4.5333061	3.2680569
Gen(17)	-5.8220388	0.7756090	0.0001467	-8.5616981	-3.0823794
Gen(18)	0.2238535	1.8331675	0.9919553	-6.2513865	6.6990934
Gen(19)	2.7709309	1.1562376	0.1496781	-1.3132111	6.8550728
Gen(20)	-12.2380535	2.4622307	0.0030742	-20.9353139	-3.5407932

Gen(21)	4.2303059	1.0607021	0.0125951	0.4836213	7.9769905
Gen(22)	-3.0534433	1.1515458	0.1008723	-7.1210123	1.0141257
Gen(23)	-1.2328320	1.8032828	0.9058174	-7.6025112	5.1368471
Gen(24)	2.5488802	0.9797892	0.1090868	-0.9119984	6.0097588
Gen(25)	-4.9496824	0.8833727	0.0013309	-8.0699921	-1.8293727
Gen(26)	-4.5709774	0.9284018	0.0032779	-7.8503419	-1.2916128
Gen(27)	6.5895872	0.8502511	0.0001140	3.5862718	9.5929026
Gen(28)	5.7450999	2.1559116	0.0988093	-1.8701590	13.3603589
Gen(29)	1.9706337	1.1148844	0.3665064	-1.9674376	5.9087050
Gen(30)	4.3506056	2.4768157	0.3717082	-4.3981727	13.0993840
Gen(31)	-5.0184787	0.5269929	0.0000215	-6.8799594	-3.1569980
Gen(32)	0.8426795	1.6462595	0.9489354	-4.9723513	6.6577103
Gen(33)	0.8472141	0.5867912	0.5352688	-1.2254901	2.9199183
Gen(34)	3.9762313	2.3004043	0.3851293	-4.1494146	12.1018771
Gen(35)	-1.3583651	1.0047642	0.5871786	-4.9074620	2.1907319
Gen(36)	-4.3681259	0.3922154	0.0000058	-5.7535360	-2.9827158
Gen(37)	6.9864957	1.1808967	0.0008977	2.8152513	11.1577401
Gen(38)	3.8771367	0.8774751	0.0066795	0.7776590	6.9766143
Gen(39)	0.7900101	0.8657335	0.8194750	-2.2679932	3.8480134
Gen(40)	1.8259523	0.9390837	0.2897056	-1.4911437	5.1430482
Gen(41)	-6.1352759	1.1432373	0.0018081	-10.1734974	-2.0970545
Gen(42)	1.0119151	0.8294054	0.6617735	-1.9177678	3.9415979
Gen(43)	3.2963714	1.2339842	0.0978072	-1.0623921	7.6551350
Gen(44)	-2.4451630	0.5215229	0.0045448	-4.2873221	-0.6030038
Gen(45)	0.9749167	0.8322979	0.6887926	-1.9649831	3.9148164
Gen(46)	-10.2974907	1.3465924	0.0001267	-15.0540168	-5.5409645
Gen(47)	0.7548810	1.1819586	0.9188884	-3.4201143	4.9298763
Gen(48)	0.9663280	0.8233692	0.6875431	-1.9420330	3.8746891
Gen(49)	-5.5158062	1.5317299	0.0227270	-10.9262879	-0.1053245
Gen(50)	-3.0639928	0.6099809	0.0028607	-5.2186093	-0.9093763

Predicted environmental effects

```
data.frame(res1$RandomEffect[[1]])
```

	Pre	SE	PValue	X2.5.LL	X97.5.UL
Env(1)	-25.15561	0.5923304	0	-27.24788	-23.06334
Env(2)	-32.64591	0.3074455	0	-33.73189	-31.55993

Env(3)	29.15014	0.3949698	0	27.75501	30.54528
Env(4)	45.02583	0.4481011	0	43.44301	46.60864
Env(5)	-16.37446	0.3629318	0	-17.65643	-15.09248

Some notes: For a two-factor design without replication, its analysis is very simple. Either ANOVA or LMM approached should work well given the data structure being balanced. If a data structure contains missing data or is unbalanced, LMM approach is more preferred. On the other hand, LMM approaches deal with fixed or random effects directly and the data analysis seems more simplified. It is important

Two-factor factorial experimental designs with replications

Now we will come to one of two-factor experimental designs, which is a factorial design.The data to be used in our demonstrations are built-in in the package minque. The data set is from a factorial mating design. To load the data set, please use the following R codes.

```
library(minque)
data(ncii)

head(ncii)
```

Female	Male	Rep	Yld
1	6	1	74.40
1	7	1	91.82
1	8	1	48.08
2	6	1	59.06
2	7	1	84.16
2	8	1	96.92

```
str(ncii)

## 'data.frame':    60 obs. of  4 variables:
##  $ Female: int  1 1 1 2 2 2 3 3 3 4 ...
##  $ Male  : int  6 7 8 6 7 8 6 7 8 6 ...
##  $ Rep   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Yld   : num  74.4 91.8 48.1 59.1 84.2 ...
```

Now we will start different factorial design analyses.

Case 1: both female and male are fixed but assume no block effects

Since the columns Female, Male, and Rep are not factorized, these three factors (variables) should be factorized before they can be run by an ANOVA model. Please use the following R code to convert these three variables. The within function is similar to the transform function.

```
require(minque)
data(ncii)
str(ncii)

## 'data.frame':    60 obs. of  4 variables:
##  $ Female: int  1 1 1 2 2 2 3 3 3 4 ...
##  $ Male  : int  6 7 8 6 7 8 6 7 8 6 ...
##  $ Rep   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Yld   : num  74.4 91.8 48.1 59.1 84.2 ...

dat=ncii
#dat$Male
#dat$Female

dat <- transform(dat,Female =factor(Female),Male = factor(Male),Rep =factor(R
ep))
dat$Male

##  [1] 6 7 8 6 7 8 6 7 8 6 7 8 6 7 8 6 7 8 6 7 8 6 7 8 6 7 8 6 7 8 6 7 8 6 7
## [36] 8 6 7 8 6 7 8 6 7 8 6 7 8 6 7 8 6 7 8 6 7 8 6 7 8
## Levels: 6 7 8

dat$Female

##  [1] 1 1 1 2 2 2 3 3 3 4 4 4 5 5 5 1 1 1 2 2 2 3 3 3 4 4 4 5 5 5 1 1 1 2 2
## [36] 2 3 3 3 4 4 4 5 5 5 1 1 1 2 2 2 3 3 3 4 4 4 5 5 5
## Levels: 1 2 3 4 5

str(dat)

## 'data.frame':    60 obs. of  4 variables:
##  $ Female: Factor w/ 5 levels "1","2","3","4",..: 1 1 1 2 2 2 3 3 3 4 ...
##  $ Male  : Factor w/ 3 levels "6","7","8": 1 2 3 1 2 3 1 2 3 1 ...
##  $ Rep   : Factor w/ 4 levels "1","2","3","4": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Yld   : num  74.4 91.8 48.1 59.1 84.2 ...
```

ANOVA analysis

You can try the following ANOVA analysis, which is the easiest way for this data set.

```
mod=aov(Yld~Female*Male,data=dat)
a=summary(mod)
data.frame(a[[1]]) ##Obtain an ANOVA Table
```

	Df	Sum.Sq	Mean.Sq	F.value	Pr..F.
Female	4	10344.330	2586.08252	28.850986	0.0000000
Male	2	1718.764	859.38216	9.587483	0.0003401
Female:Male	8	14433.452	1804.18144	20.127901	0.0000000
Residuals	45	4033.613	89.63585	NA	NA

```
mod=lm(Yld~Female*Male,data=dat)
a=summary(mod)
a
```

```
##
## Call:
## lm(formula = Yld ~ Female * Male, data = dat)
##
## Residuals:
##      Min       1Q  Median      3Q     Max
## -15.820  -7.614   0.215   5.924  14.310
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)     68.550      4.734  14.481  < 2e-16 ***
## Female2          2.215      6.695   0.331  0.74228
## Female3         32.335      6.695   4.830 1.62e-05 ***
## Female4         37.245      6.695   5.563 1.38e-06 ***
## Female5         15.600      6.695   2.330  0.02434 *
## Male7           39.090      6.695   5.839 5.41e-07 ***
## Male8          -15.910      6.695  -2.377  0.02179 *
## Female2:Male7  -12.215      9.468  -1.290  0.20358
## Female3:Male7  -28.435      9.468  -3.003  0.00435 **
## Female4:Male7  -80.435      9.468  -8.496 6.62e-11 ***
## Female5:Male7  -41.305      9.468  -4.363 7.41e-05 ***
## Female2:Male8   30.795      9.468   3.253  0.00217 **
## Female3:Male8   32.760      9.468   3.460  0.00119 **
## Female4:Male8  -43.075      9.468  -4.550 4.05e-05 ***
## Female5:Male8   26.580      9.468   2.807  0.00736 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.468 on 45 degrees of freedom
## Multiple R-squared:  0.8679, Adjusted R-squared:  0.8268
## F-statistic: 21.11 on 14 and 45 DF,  p-value: 3.136e-15
```

Given that both female and male effects are fixed, the F-tests for all these effects are appropriate.

Linear mixed model analysis

```
res=lmm(Yld~Female*Male,data=dat)[[1]]
data.frame(res$Var[[1]])
```

	Est	SE	Chi_sq	P_value
V(e)	89.63585	18.8969	22.5	1.1e-06

```
data.frame(res$FixedEffect[[1]])
```

	Est	SE	z_value	P_value

mu	86.0670000	1.2843233	67.0135005	0.0000000
Female(1)	-6.9108235	1.7377392	-3.9769049	0.0000698
Female(2)	1.7277059	0.4756402	3.6323799	0.0002808
Female(3)	1.7277059	0.4756402	3.6323799	0.0002808
Female(4)	1.7277059	0.4756402	3.6323799	0.0002808
Female(5)	1.7277059	0.4756402	3.6323799	0.0002808
Male(6)	-3.8633333	1.9611063	-1.9699765	0.0488411
Male(7)	15.6816667	1.9611063	7.9963368	0.0000000
Male(8)	-11.8183333	1.9611063	-6.0263603	0.0000000
Female:Male(1:6)	-6.7428431	2.0620724	-3.2699352	0.0010757
Female:Male(1:7)	12.8021569	2.0620724	6.2083936	0.0000000
Female:Male(1:8)	-14.6978431	2.0620724	-7.1277048	0.0000000
Female:Male(2:6)	-13.1663725	4.9310075	-2.6701181	0.0075825
Female:Male(2:7)	-5.8363725	4.9310075	-1.1836065	0.2365689
Female:Male(2:8)	9.6736275	4.9310075	1.9617953	0.0497863
Female:Male(3:6)	16.9536275	4.9310075	3.4381670	0.0005857
Female:Male(3:7)	8.0636275	4.9310075	1.6352900	0.1019883
Female:Male(3:8)	41.7586275	4.9310075	8.4685792	0.0000000
Female:Male(4:6)	21.8636275	4.9310075	4.4339068	0.0000093
Female:Male(4:7)	-39.0263725	4.9310075	-7.9144825	0.0000000
Female:Male(4:8)	-29.1663725	4.9310075	-5.9148912	0.0000000
Female:Male(5:6)	0.2186275	4.9310075	0.0443373	0.9646356
Female:Male(5:7)	-21.5413725	4.9310075	-4.3685540	0.0000125
Female:Male(5:8)	18.8436275	4.9310075	3.8214559	0.0001327

Please carefully compare the estimated effects by the ANOVA method and by MINQUE method.

Case 2: Both female and male are random effects but no block effects

By ANOVA analysis

We may use ANOVA method to generate an ANOVA table, which contains all DF, SS, MS, and F-tests for these effects.

```
mod1=aov(Yld~Female*Male,data=dat)
a=summary(mod1)

data.frame(a[[1]]) ##Obtain an ANOVA Table
```

	Df	Sum.Sq	Mean.Sq	F.value	Pr..F.
Female	4	10344.330	2586.08252	28.850986	0.0000000
Male	2	1718.764	859.38216	9.587483	0.0003401
Female:Male	8	14433.452	1804.18144	20.127901	0.0000000
Residuals	45	4033.613	89.63585	NA	NA

All F-tests and the corresponding P values are provided; however, not all these F-tests are correctly given when both male and female effects are random. By doing so, we may use the following R codes to run the right F-tests for female and male effects. The following R codes use female x male interaction as error term to run F-test for main effects: male and female.

```
mod2=aov(Yld~Female+Male+Error(Female:Male),data=dat)

## Warning in aov(Yld ~ Female + Male + Error(Female:Male), data = dat):
## Error() model is singular

a=summary(mod2)
a

##
## Error: Female:Male
##              Df Sum Sq Mean Sq F value Pr(>F)
## Female        4  10344  2586.1   1.433  0.307
## Male          2   1719   859.4   0.476  0.638
## Residuals     8  14433  1804.2
##
## Error: Within
##              Df Sum Sq Mean Sq F value Pr(>F)
## Residuals 45   4034   89.64

#data.frame(a[[1]]) ##Obtain an ANOVA Table
```

However, the above ANOVA table does not provide a F test for male-by-female interaction effect. Thus, we need combine these two ANOVA tables for all appropriate F-tests.

By a linear mixed model approach

Using linear mixed model approaches could be a little convient, though the process may not be computationally easy.

```
res1=lmm.jack(Yld~1|Female*Male,data=dat)[[1]]
```

Estimated proportional variance components for male, female, male*female, and random error.

```
data.frame(res1$Var[[1]])
```

	Estimate	SE	PValue	X2.5.LL	X97.5.UL

V(Female)	65.04606	12.934725	0.0028386	19.35714	110.7350
V(Male)	0.00000	0.000000	1.0000000	0.00000	0.0000
V(Female:Male)	428.78048	26.524525	0.0000002	335.08874	522.4722
V(e)	90.47962	3.886761	0.0000000	76.75053	104.2087

Estimated proportional variance components for male, female, male*female, and random error.

```
data.frame(res1$PVar[[1]])
```

	Estimate	SE	PValue	X2.5.LL	X97.5.UL
V(Female)/VP	0.1116956	0.0239390	0.0046922	0.0271368	0.1962544
V(Male)/VP	0.0000000	0.0000000	1.0000000	0.0000000	0.0000000
V(Female:Male)/VP	0.7333347	0.0250923	0.0000000	0.6447019	0.8219674
V(e)/VP	0.1549697	0.0076015	0.0000000	0.1281192	0.1818203

Estimated population mean

```
data.frame(res1$FixedEffect[[1]])
```

	Estimate	SE	PValue	X2.5.LL	X97.5.UL
mu	86.07161	0.5170937	0	84.24509	87.89812

Predicted male, female, male*female effects

```
data.frame(res1$RandomEffect[[1]])
```

	Pre	SE	PValue	X2.5.LL	X97.5.UL
Female(1)	-5.3455069	0.5544895	0.0000194	-7.304113	-3.386901
Female(2)	-0.7599826	0.5772982	0.6073327	-2.799155	1.279190
Female(3)	13.1040043	1.3274841	0.0000159	8.414974	17.793035
Female(4)	-7.4900510	0.8482288	0.0000399	-10.486223	-4.493879
Female(5)	0.4915363	0.4277055	0.7008947	-1.019235	2.002307
Male(6)	0.0000000	0.0000000	1.0000000	0.000000	0.000000
Male(7)	0.0000000	0.0000000	1.0000000	0.000000	0.000000
Male(8)	0.0000000	0.0000000	1.0000000	0.000000	0.000000
Female:Male(1:6)	-13.2062872	1.3363513	0.0000158	-17.926639	-8.485936
Female:Male(1:7)	28.8080628	1.6567200	0.0000001	22.956083	34.660043
Female:Male(1:8)	-26.5661387	1.6952013	0.0000003	-32.554045	-20.578232
Female:Male(2:6)	-17.7917498	1.5657675	0.0000049	-23.322461	-12.261038
Female:Male(2:7)	9.1134979	1.5038231	0.0007523	3.801591	14.425405

Female:Male(2:8)	7.1090618	1.3412161	0.0019725	2.371526	11.846597
Female:Male(3:6)	-2.4541043	1.2691588	0.2940136	-6.937114	2.028905
Female:Male(3:7)	4.2971871	1.5351432	0.0800084	-1.125351	9.719725
Female:Male(3:8)	24.9956848	0.8968362	0.0000000	21.827819	28.163551
Female:Male(4:6)	36.5492650	2.3025464	0.0000003	28.416052	44.682478
Female:Male(4:7)	-21.4464594	1.6398253	0.0000015	-27.238763	-15.654156
Female:Male(4:8)	-30.4321968	1.6135770	0.0000001	-36.131784	-24.732609
Female:Male(5:6)	-3.1259330	1.3499550	0.1692242	-7.894337	1.642471
Female:Male(5:7)	-12.4342899	1.1683859	0.0000085	-16.561343	-8.307237
Female:Male(5:8)	16.5843997	1.7878151	0.0000266	10.269356	22.899443

It is important to know how to make appropriate statistical conclusions based on P values and confidence intervals (such as significant difference from zero or each other). The same principle is applicable for many other cases in this course or your own research data analyses.

Case 3: Female is random but male is fixed and assume no block effects

By ANOVA method

```
mod2=aov(Yld~Male+Error(Female:Male)+Female,data=dat)

## Warning in aov(Yld ~ Male + Error(Female:Male) + Female, data = dat):
## Error() model is singular

a=summary(mod2)
a

##
## Error: Female:Male
##           Df Sum Sq Mean Sq F value Pr(>F)
## Male       2   1719   859.4   0.476  0.638
## Female     4  10344  2586.1   1.433  0.307
## Residuals  8  14433  1804.2
##
## Error: Within
##           Df Sum Sq Mean Sq F value Pr(>F)
## Residuals 45   4034   89.64
```

Only the F-test for male effect is right. However, for other F-tests, please refer to the regular ANOVA table. We still can use the following R code to a model with fixed male effects and random female effects.

By LMM method

```
res=lmm.jack(Yld~Male|Female*Male,data=dat)[[1]]
```

Estimated variance components for female and female-by-male interaction effects.

```r
data.frame(res$Var[[1]])
```

	Estimate	SE	PValue	X2.5.LL	X97.5.UL
V(Female)	64.30107	27.037771	0.1539132	-31.20359	159.8057
V(Male:Female)	430.59441	29.130218	0.0000005	327.69865	533.4902
V(e)	88.93307	6.112707	0.0000006	67.34135	110.5248
Estimated proport	ional varian	ce component	s for female	and female-	by-male interaction effects.

```r
data.frame(res$PVar[[1]])
```

	Estimate	SE	PValue	X2.5.LL	X97.5.UL
V(Female)/VP	0.1097951	0.0446434	0.1359357	-0.0478975	0.2674877
V(Male:Female)/VP	0.7377426	0.0439838	0.0000002	0.5823798	0.8931053
V(e)/VP	0.1524623	0.0109480	0.0000009	0.1137911	0.1911334

Estimated population mean and male effects.

```r
data.frame(res$FixedEffect[[1]])
```

	Estimate	SE	PValue	X2.5.LL	X97.5.UL
mu	86.0932930	0.6117023	0.0000000	83.932596	88.253990
Male(6)	0.0105872	0.7025796	0.9957130	-2.471113	2.492287
Male(7)	6.6043310	0.3911125	0.0000002	5.222817	7.985845
Male(8)	-6.6149182	0.7022897	0.0000235	-9.095594	-4.134243

Predicted female and female-by-male interaction effects.

```r
data.frame(res$RandomEffect[[1]])
```

	Pre	SE	PValue	X2.5.LL	X97.5.UL
Female(1)	-5.2499544	1.2786052	0.0105621	-9.7663314	-0.7335774
Female(2)	-0.7986280	0.5130833	0.4732570	-2.6109760	1.0137200
Female(3)	12.8212212	2.6584740	0.0037671	3.4307768	22.2116656
Female(4)	-7.2955791	1.7549778	0.0097913	-13.4946322	-1.0965260
Female(5)	0.5229402	0.5586220	0.8088030	-1.4502627	2.4961431
Male:Female(6:1)	-13.2633903	1.5655907	0.0000559	-18.7934772	-7.7333033
Male:Female(7:1)	27.2144385	2.8047918	0.0000184	17.3071604	37.1217167
Male:Female(8:1)	-24.9379313	2.2130596	0.0000053	-32.7550523	-17.1208103

Male:Female(6:2)	-17.8268975	2.0425716	0.0000439	-25.0418093	-10.6119857
Male:Female(7:2)	7.4637566	2.4696345	0.0562739	-1.2596560	16.1871691
Male:Female(8:2)	8.6994573	2.7235471	0.0429052	-0.9208427	18.3197572
Male:Female(6:3)	-2.3168595	1.8663740	0.6498147	-8.9093940	4.2756749
Male:Female(7:3)	2.4149486	1.6931625	0.5451011	-3.5657564	8.3956536
Male:Female(8:3)	26.7891004	1.2093873	0.0000000	22.5172196	31.0609813
Male:Female(6:4)	36.6275386	2.0642773	0.0000001	29.3359567	43.9191206
Male:Female(7:4)	-23.0239745	2.8700728	0.0000866	-33.1618426	-12.8861064
Male:Female(8:4)	-28.8791076	1.6622421	0.0000001	-34.7505932	-23.0076219
Male:Female(6:5)	-3.2203913	2.1086859	0.4891356	-10.6688366	4.2280539
Male:Female(7:5)	-14.0691692	2.0256162	0.0002687	-21.2241898	-6.9141487
Male:Female(8:5)	18.3284811	2.0633634	0.0000380	11.0401271	25.6168352

Case 4: Both female and male are random with block effects

By ANOVA method

```
mod2=aov(Yld~Male+Female+Error(Female:Male)+Rep,data=dat)

## Warning in aov(Yld ~ Male + Female + Error(Female:Male) + Rep, data = dat)
:
## Error() model is singular

a=summary(mod2)
a

##
## Error: Female:Male
##           Df Sum Sq Mean Sq F value Pr(>F)
## Male       2   1719   859.4   0.476  0.638
## Female     4  10344  2586.1   1.433  0.307
## Residuals  8  14433  1804.2
##
## Error: Within
##           Df Sum Sq Mean Sq F value Pr(>F)
## Rep        3    114   38.04   0.408  0.748
## Residuals 42   3919   93.32
```

The F-test for male*female interaction effect is not available in above ANOVA tables; however, it is available in the regular ANOVA table.

By LMM method

```
res=lmm.jack(Yld~1|Female*Male+Rep,data=dat)[[1]]
data.frame(res$Var[[1]])
```

	Estimate	SE	PValue	X2.5.LL	X97.5.UL

V(Female)	66.10546	19.455217	0.0311684	-2.615587	134.8265
V(Male)	0.00000	0.000000	1.0000000	0.000000	0.0000
V(Rep)	0.00000	0.000000	1.0000000	0.000000	0.0000
V(Female:Male)	426.89429	26.914452	0.0000003	331.825217	521.9634
V(e)	93.97127	5.168524	0.0000001	75.714658	112.2279

```
data.frame(res$PVar[[1]])
```

	Estimate	SE	PValue	X2.5.LL	X97.5.UL
V(Female)/VP	0.1127684	0.0334704	0.0325955	-0.0054580	0.2309948
V(Male)/VP	0.0000000	0.0000000	1.0000000	0.0000000	0.0000000
V(Rep)/VP	0.0000000	0.0000000	1.0000000	0.0000000	0.0000000
V(Female:Male)/VP	0.7269299	0.0306026	0.0000000	0.6188335	0.8350264
V(e)/VP	0.1603017	0.0113371	0.0000008	0.1202561	0.2003473

```
data.frame(res$FixedEffect[[1]])
```

	Estimate	SE	PValue	X2.5.LL	X97.5.UL
mu	86.07004	0.4645859	0	84.429	87.71108

```
data.frame(res$RandomEffect[[1]])
```

	Pre	SE	PValue	X2.5.LL	X97.5.UL
Female(1)	-5.3428068	0.9905220	0.0017445	-8.841596	-1.844017
Female(2)	-0.7012139	0.5712231	0.6575648	-2.718927	1.316500
Female(3)	13.0973453	2.0034624	0.0004268	6.020578	20.174113
Female(4)	-7.5417828	1.3807388	0.0015962	-12.418923	-2.664642
Female(5)	0.4884582	0.4967474	0.7865899	-1.266187	2.243103
Male(6)	0.0000000	0.0000000	1.0000000	0.000000	0.000000
Male(7)	0.0000000	0.0000000	1.0000000	0.000000	0.000000
Male(8)	0.0000000	0.0000000	1.0000000	0.000000	0.000000
Rep(1)	0.0000000	0.0000000	1.0000000	0.000000	0.000000
Rep(2)	0.0000000	0.0000000	1.0000000	0.000000	0.000000
Rep(3)	0.0000000	0.0000000	1.0000000	0.000000	0.000000
Rep(4)	0.0000000	0.0000000	1.0000000	0.000000	0.000000
Female:Male(1:6)	-13.2115162	1.1392329	0.0000041	-17.235593	-9.187439
Female:Male(1:7)	28.8098596	2.3371866	0.0000024	20.554289	37.065430
Female:Male(1:8)	-26.5735890	1.5516502	0.0000001	-32.054434	-21.092744
Female:Male(2:6)	-17.7548879	1.7647682	0.0000136	-23.988523	-11.521252
Female:Male(2:7)	9.0700465	1.4762050	0.0006795	3.855694	14.284399

Female:Male(2:8)	7.2015347	1.7642143	0.0109462	0.969856	13.433214
Female:Male(3:6)	-2.3543528	1.4165316	0.4180542	-7.357923	2.649217
Female:Male(3:7)	4.3477750	1.6108508	0.0936482	-1.342183	10.037733
Female:Male(3:8)	24.8094327	1.1126289	0.0000000	20.879329	28.739537
Female:Male(4:6)	36.3505027	2.5156289	0.0000006	27.464626	45.236380
Female:Male(4:7)	-21.4034440	2.0049161	0.0000083	-28.485346	-14.321542
Female:Male(4:8)	-30.3337906	1.3616009	0.0000000	-35.143331	-25.524251
Female:Male(5:6)	-3.0826623	1.6723699	0.3320848	-8.989922	2.824598
Female:Male(5:7)	-12.4592519	1.0011472	0.0000023	-15.995573	-8.922931
Female:Male(5:8)	16.5843436	1.3498070	0.0000025	11.816463	21.352225

Case 5: Female is random but male is fixed with block effects

By LMM method

```
res=lmm.jack(Yld~Male|Female*Male+Rep,data=dat)[[1]]
data.frame(res$Var[[1]])
```

	Estimate	SE	PValue	X2.5.LL	X97.5.UL
V(Female)	64.320198	23.0162669	0.0805971	-16.979440	145.619836
V(Rep)	0.309927	0.9800753	0.9780305	-3.151962	3.771816
V(Male:Female)	428.835007	41.5368485	0.0000110	282.115699	575.554315
V(e)	92.550758	6.1471431	0.0000004	70.837397	114.264119

```
data.frame(res$PVar[[1]])
```

	Estimate	SE	PValue	X2.5.LL	X97.5.UL
V(Female)/VP	0.1096545	0.0384433	0.0735788	-0.0261376	0.2454465
V(Rep)/VP	0.0006241	0.0019736	0.9780305	-0.0063472	0.0075954
V(Male:Female)/VP	0.7310809	0.0408639	0.0000001	0.5867386	0.8754232
V(e)/VP	0.1586405	0.0163970	0.0000188	0.1007217	0.2165593

```
data.frame(res$FixedEffect[[1]])
```

	Estimate	SE	PValue	X2.5.LL	X97.5.UL
mu	85.9821090	0.3171805	0.0000000	84.861742	87.102476
Male(6)	-0.0096078	0.8605360	0.9958146	-3.049252	3.030037
Male(7)	6.5463094	0.9991818	0.0004198	3.016931	10.075688
Male(8)	-6.5367016	0.6309662	0.0000107	-8.765444	-4.307959

```
data.frame(res$RandomEffect[[1]])
```

	Pre	SE	PValue	X2.5.LL	X97.5.UL

Female(1)	-5.2659588	1.0579291	0.0030436	-9.0028487	-1.529069
Female(2)	-0.6996356	0.7771033	0.8248833	-3.4445732	2.045302
Female(3)	12.8647108	2.2161852	0.0010311	5.0365492	20.692872
Female(4)	-7.3642731	1.6905976	0.0073128	-13.3359181	-1.392628
Female(5)	0.4651568	0.7943388	0.9326603	-2.3406612	3.270975
Rep(1)	-0.1067438	0.3375537	0.9780305	-1.2990741	1.085586
Rep(2)	-0.1771916	0.5603289	0.9780305	-2.1564240	1.802041
Rep(3)	0.0716920	0.2267101	0.9780305	-0.7291089	0.872493
Rep(4)	0.2122434	0.6711725	0.9780305	-2.1585183	2.583005
Male:Female(6:1)	-13.3010297	1.7239301	0.0001181	-19.3904139	-7.211646
Male:Female(7:1)	27.2810545	2.5741256	0.0000088	18.1885514	36.373558
Male:Female(8:1)	-24.9681082	1.4772254	0.0000002	-30.1860650	-19.750151
Male:Female(6:2)	-17.7388358	2.2150829	0.0000878	-25.5631036	-9.914568
Male:Female(7:2)	7.3368691	2.7296899	0.0953147	-2.3051289	16.978867
Male:Female(8:2)	8.8423692	2.4335299	0.0216141	0.2464877	17.438251
Male:Female(6:3)	-2.2637787	1.7415458	0.6167456	-8.4153863	3.887829
Male:Female(7:3)	2.6551350	1.8643984	0.5463222	-3.9304210	9.240691
Male:Female(8:3)	26.4721105	2.5678862	0.0000111	17.4016464	35.542575
Male:Female(6:4)	36.5474151	2.7040265	0.0000011	26.9960670	46.098763
Male:Female(7:4)	-23.2565771	1.7828050	0.0000015	-29.5539233	-16.959231
Male:Female(8:4)	-28.5728307	3.0298635	0.0000233	-39.2751224	-17.870539
Male:Female(6:5)	-3.2437708	2.6743400	0.6657545	-12.6902583	6.202717
Male:Female(7:5)	-14.0164816	1.0818665	0.0000016	-17.8379247	-10.195038
Male:Female(8:5)	18.2264592	1.9041902	0.0000206	11.5003480	24.952571

Case 6: Both female and male are fixed with block effects

```
res=lmm.jack(Yld~Female*Male|Rep,data=dat)[[1]]
data.frame(res$Var[[1]])
```

	Estimate	SE	PValue	X2.5.LL	X97.5.UL
V(Rep)	0.00000	0.000000	1e+00	0.00000	0.0000
V(e)	93.66863	5.854858	3e-07	72.98769	114.3496

```
data.frame(res$PVar[[1]])
```

	Estimate	SE	PValue	X2.5.LL	X97.5.UL
V(Rep)/VP	0	0	1	0	0
V(e)/VP	1	0	0	1	1

```
data.frame(res$FixedEffect[[1]])
```

	Estimate	SE	PValue	X2.5.LL	X97.5.UL
mu	86.0639556	0.5839629	0.0000000	84.001242	88.126669
Female(1)	-6.8587137	0.5166146	0.0000013	-8.683535	-5.033892
Female(2)	1.7146784	0.1291537	0.0000013	1.258473	2.170884
Female(3)	1.7146784	0.1291537	0.0000013	1.258473	2.170884
Female(4)	1.7146784	0.1291537	0.0000013	1.258473	2.170884
Female(5)	1.7146784	0.1291537	0.0000013	1.258473	2.170884
Male(6)	-3.8307222	0.7517062	0.0025915	-6.485951	-1.175494
Male(7)	15.6844444	0.7816145	0.0000000	12.923572	18.445317
Male(8)	-11.8537222	0.7930883	0.0000005	-14.655123	-9.052321
Female:Male(1:6)	-6.6885196	0.7253962	0.0000280	-9.250814	-4.126225
Female:Male(1:7)	12.8266471	0.7654111	0.0000002	10.123009	15.530285
Female:Male(1:8)	-14.7115196	0.9120988	0.0000002	-17.933298	-11.489741
Female:Male(2:6)	-13.1829118	2.2924715	0.0011039	-21.280537	-5.085287
Female:Male(2:7)	-5.8230784	2.1946308	0.1006005	-13.575104	1.928947
Female:Male(2:8)	9.7250882	2.0595254	0.0043359	2.450291	16.999885
Female:Male(3:6)	16.9370882	1.3443986	0.0000020	12.188311	21.685865
Female:Male(3:7)	7.9835882	1.5273195	0.0021738	2.588685	13.378491
Female:Male(3:8)	41.8100882	1.1668207	0.0000000	37.688564	45.931612
Female:Male(4:6)	21.8470882	2.4109134	0.0000323	13.331094	30.363082
Female:Male(4:7)	-39.0304118	1.8066106	0.0000000	-45.411846	-32.648978
Female:Male(4:8)	-29.1149118	1.1149210	0.0000000	-33.053112	-25.176711
Female:Male(5:6)	0.2020882	2.1877933	0.9932090	-7.525785	7.929962
Female:Male(5:7)	-21.5280784	1.4691089	0.0000006	-26.717366	-16.338791
Female:Male(5:8)	18.7477549	1.6629489	0.0000052	12.873773	24.621737

```
data.frame(res$RandomEffect[[1]])
```

	Pre	SE	PValue	X2.5.LL	X97.5.UL
Rep(1)	0	0	1	0	0
Rep(2)	0	0	1	0	0
Rep(3)	0	0	1	0	0
Rep(4)	0	0	1	0	0

Conclusions

When two or more factors are employed in an experiment, the data can be analyzed in several ways: full or reduced models, fixed, random, or mixed effects models. When using

ANOVA methods, we may need consider to combining different ANOVA results for different appropriate F-tests when one or two factors are random. This, sometimes, will need EMS and could be challenging. On the other hand, LMM methods provide a much easier frame to deal with fixed or random effects models. The key for these experimental analyses is to write an appropriate linear model. It is also very important to determine which effects are random or fixed.