# Missing data

- Mixed and random models (EMS tables by hand + R output = Victory)
- Dealing with missing data

## Example 1 *Random effects model [Lab8ex1.R]*

In this example, we're using the data set from Lab 7 (Example 2), a three-way factorial CRD with one replication per three-way combination of factors. As before, there are not enough degrees of freedom to include the three-way interaction in the model. Not as before, *we are going to treat all three factors as random effects*. The table of EMS's, generated using the "recipe" outlined in the supplementary materials on the course website, is shown below:

| **Fixed or Random<br>Number of levels<br>Factor** | R<br>a<br>i | R<br>b<br>j | R<br>c<br>k | **Expected Mean Squares** | **F** |
|---|---|---|---|---|---|
| $A_i$ | 1 | b | c | $\sigma^2_\varepsilon + b\cdot\sigma^2_{AB} + c\cdot\sigma^2_{AC} + bc\cdot\sigma^2_A$ | $(MS_A + MS_\varepsilon) / (MS_{AB} + MS_{AC})$ |
| $B_j$ | a | 1 | c | $\sigma^2_\varepsilon + a\cdot\sigma^2_{BC} + c\cdot\sigma^2_{AB} + ac\cdot\sigma^2_B$ | $(MS_B + MS_\varepsilon) / (MS_{AB} + MS_{BC})$ |
| $C_k$ | a | b | 1 | $\sigma^2_\varepsilon + a\cdot\sigma^2_{AC} + b\cdot\sigma^2_{BC} + ab\cdot\sigma^2_C$ | $(MS_C + MS_\varepsilon) / (MS_{AC} + MS_{BC})$ |
| $(AB)_{ij}$ | 1 | 1 | c | $\sigma^2_\varepsilon + c\cdot\sigma^2_{AB}$ | $MS_{AB} / MS_\varepsilon$ |
| $(AC)_{ik}$ | 1 | b | 1 | $\sigma^2_\varepsilon + b\cdot\sigma^2_{AC}$ | $MS_{AC} / MS_\varepsilon$ |
| $(BC)_{jk}$ | a | 1 | 1 | $\sigma^2_\varepsilon + a\cdot\sigma^2_{BC}$ | $MS_{BC} / MS_\varepsilon$ |
| $\varepsilon_{(ijk)}$ | 1 | 1 | 1 | $\sigma^2_\varepsilon$ | |

*With a = 3, b = 5, and c = 2.*

**The R code, as before:**

```
fact_mod<-lm(Y ~ (A + B + C)^2, fact_dat)
anova(fact_mod)
```

**The output:**

```
          Df Sum Sq Mean Sq F value    Pr(>F)
A          2 3599.3  1799.6 ~~620.56 1.682e-09~~ ***  NO!
B          4 6423.1  1605.8 ~~553.72 8.364e-10~~ ***  NO!!
C          1 5333.3  5333.3 ~~1839.08 9.639e-11~~ ***  NO!!!
A:B        8 9675.1  1209.4  417.03 1.140e-09 ***
A:C        2 5692.5  2846.2  981.46 2.714e-10 ***
B:C        4 7987.0  1996.8  688.53 3.510e-10 ***
Residuals  8   23.2     2.9
```

***NOTE: The above table incorrectly uses the MSE (2.9) as the denominator for all F tests!***

If you'd like to reduce the amount of typing (and thus chances for error) when carrying out the appropriate custom, synthetic F tests, you can first reach into the anova() object and grab the degrees of freedom and mean squares from the above table:

```
d <- anova(fact_mod)$Df
M <- anova(fact_mod)$'Mean Sq'
```

With those in hand, the necessary test is a little less painful. For Factor A:

```
F_A_num <- M[1]+M[7]
F_A_den <- M[4]+M[5]
df_A_num <- F_A_num^2 / ((M[1]^2/d[1]) + (M[7]^2/d[7]))
df_A_den <- F_A_den^2 / ((M[4]^2/d[4]) + (M[5]^2/d[5]))
p_A <- pf(F_A_num/F_A_den, df_A_num, df_A_den, lower.tail = FALSE)
```

The inconvenience here is the requirement of calculating effective degrees of freedom for this test. Recall from the Satterthwaite method that the test statistic:

$$F = MS' / MS''$$

is distributed approximately as $F_{p,q}$ where $p$ and $q$ are the ***effective degrees of freedom***:

$$p = \frac{(MS_A + \ldots + MS_X)^2}{\frac{MS_A^2}{df_A} + \ldots + \frac{MS_X^2}{df_X}} \quad \text{and} \quad q = \frac{(MS_1 + \ldots + MS_n)^2}{\frac{MS_1^2}{df_1} + \ldots + \frac{MS_n^2}{df_n}}$$

Carrying out similar calculations for Factors B and C then leads to the following modified ANOVA table:

| | Df | Sum Sq | Mean Sq | F value | Pr(>F) | | |
|---|---|---|---|---|---|---|---|
| A | 2 | 3599.3 | 1799.6 | ~~620.56~~ | ~~1.682e-09~~ | *** | NO! |
| Error | 3.89 | | | 0.44 | 0.67 | NS | |
| B | 4 | 6423.1 | 1605.8 | ~~553.72~~ | ~~8.364e-10~~ | *** | NO!! |
| Error | 8.71 | | | 0.50 | 0.74 | NS | |
| C | 1 | 5333.3 | 5333.3 | ~~1839.08~~ | ~~9.639e-11~~ | *** | NO!!! |
| Error | 4.64 | | | 1.10 | 0.35 | NS | |
| A:B | 8 | 9675.1 | 1209.4 | 417.03 | 1.140e-09 | *** | |
| A:C | 2 | 5692.5 | 2846.2 | 981.46 | 2.714e-10 | *** | |
| B:C | 4 | 7987.0 | 1996.8 | 688.53 | 3.510e-10 | *** | |
| Error | 8 | 23.2 | 2.9 | | | | |

*The non-integer error df's are approximated using Satterthwaite's method.*

## Dealing with missing data

The following example demonstrates the effect of missing data (loss of balance) on the computation of means and sums of squares:

**Example 2** *[Lab8ex2.R]*

This experiment was carried out as CRD with with a 2x3 factorial treatment structure. The number of experimental units per A-B combination are not uniform, as shown in the the data below:

| A | B | Y |
|---|---|---|
| 1 | 1 | 5 |
| 1 | 1 | 6 |
| 1 | 2 | 2 |
| 1 | 2 | 3 |
| 1 | 2 | 5 |
| 1 | 2 | 6 |
| 1 | 2 | 7 |
| 1 | 3 | 3 |
| 2 | 1 | 2 |
| 2 | 1 | 3 |
| 2 | 2 | 8 |
| 2 | 2 | 8 |
| 2 | 2 | 9 |
| 2 | 3 | 4 |
| 2 | 3 | 4 |
| 2 | 3 | 6 |
| 2 | 3 | 6 |
| 2 | 3 | 7 |

To begin, let's look at the results of the following two analyses:

```
miss1_mod<-lm(Y ~ A + B + A:B, miss_dat)
anova(miss1_mod)

miss2_mod<-lm(Y ~ B + A + A:B, miss_dat)
anova(miss2_mod)
```

The results:

```
Response: Y
           Df  Sum Sq Mean Sq F value   Pr(>F)
A           1  5.1361  5.1361  2.3645 0.150068
B           2 15.6829  7.8414  3.6099 0.059238 .
A:B         2 30.2255 15.1127  6.9573 0.009859 **
Residuals  12 26.0667  2.1722


Response: Y
           Df  Sum Sq Mean Sq F value   Pr(>F)
B           2 11.1111  5.5556  2.5575 0.118799
A           1  9.7079  9.7079  4.4691 0.056129 .
B:A         2 30.2255 15.1127  6.9573 0.009859 **
Residuals  12 26.0667  2.1722
```

Different results, and all because we swapped the order of the factors in the linear model?! Surely proper analysis should not rely on something as arbitrary as that. Let's compare the above results to those we get using the Anova() function ("car" package) where we specify the calculation of partial (or Type 2) sums of squares:

```
#The ANOVA, with partial (or Type II) SS
#library(car)
Anova(miss1_mod, type=2)
Anova(miss2_mod, type=2)
```

The results:

```
Response: Y
            Sum Sq Df F value   Pr(>F)
A           9.7079  1  4.4691 0.056129 .
B          15.6829  2  3.6099 0.059238 .
A:B        30.2255  2  6.9573 0.009859 **
Residuals  26.0667 12

Response: Y
            Sum Sq Df F value   Pr(>F)
B          15.6829  2  3.6099 0.059238 .
A           9.7079  1  4.4691 0.056129 .
B:A        30.2255  2  6.9573 0.009859 **
Residuals  26.0667 12
```

Two things happen: 1) The results no longer depend upon the order of the factors in the model (good), and 2) The results we obtain are identical to the anova() approach, assuming that the main effect of each factor was adjusted for the main effect of the other. The Type 2 approach does a better job of minimizing the overlap due to the broken orthogonality of the treatments by evaluating each main effect (A, B) after first accounting for the other.

**Means**

```
meansA <- aggregate(miss_dat$Y, list(miss_dat$A), mean)
meansB <- aggregate(miss_dat$Y, list(miss_dat$B), mean)
```

```
> meansA
  Group.1     x
1       1 4.625
2       2 5.700
> meansB
  Group.1 x
1       1 4
2       2 6
3       3 5
```

**Least squares adjusted means**

```
> lsmeans(miss1_mod, "A")

NOTE: Results may be misleading due to involvement in interactions
 A   lsmean        SE df lower.CL upper.CL
 1 4.366667 0.6405534 12 2.971021 5.762313
 2 5.411111 0.4994029 12 4.323006 6.499217

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

> lsmeans(miss1_mod, "B")

NOTE: Results may be misleading due to involvement in interactions
 B   lsmean        SE df lower.CL upper.CL
 1 4.000000 0.7369230 12 2.394383 5.605617
 2 6.466667 0.5381725 12 5.294090 7.639244
 3 4.200000 0.8072587 12 2.441134 5.958866

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

**Output of LSMeans and p-values for protected (Tukey) pairwise comparisons**

```
missA_lsm <- lsmeans(miss1_mod, "A")
missB_lsm <- lsmeans(miss1_mod, "B")
```

```
> contrast(missA_lsm, method = "pairwise", adjust = "tukey")

 contrast  estimate        SE df t.ratio p.value
 1 - 2    -1.044444 0.8122265 12  -1.286  0.2227

Results are averaged over the levels of: B
```

```
> contrast(missB_lsm, method = "pairwise", adjust = "tukey")

 contrast  estimate        SE df t.ratio p.value
 1 - 2    -2.466667 0.9125159 12  -2.703  0.0470
 1 - 3    -0.200000 1.0930335 12  -0.183  0.9817
 2 - 3     2.266667 0.9702043 12   2.336  0.0887

Results are averaged over the levels of: A
P value adjustment: tukey method for a family of 3 means
```

Notice that the estimates in the above tables are based on the least squares adjusted means (e.g. for Factor B, Level 1 - Level 3 = 4 - 4.2 = -0.2).

**Output of LSMeans and p-values for Dunnett comparisons**

```
> contrast(missB_lsm, method = "trt.vs.ctrl")

 contrast estimate        SE df t.ratio p.value
 2 - 1    2.466667 0.9125159 12   2.703  0.0380
 3 - 1    0.200000 1.0930335 12   0.183  0.9798

Results are averaged over the levels of: A
P value adjustment: sidak method for 2 tests
```

**Output of LSMeans and p-values for group comparisons (contrasts)**

```
> contrast(missB_lsm, list("1 vs. 23"=c(2,-1,-1), "2 vs. 3"=c(0,1,-1)))

 contrast   estimate        SE df t.ratio p.value
 X1.vs..23 -2.666667 1.7645165 12  -1.511  0.1566
 X2.vs..3   2.266667 0.9702043 12   2.336  0.0376

Results are averaged over the levels of: A
```

**Output of LSMeans and p-values for trend analysis**

```
> contrast(missB_lsm, method = "poly")

 contrast   estimate        SE df t.ratio p.value
 linear     0.200000 1.093033 12   0.183  0.8579
 quadratic -4.733333 1.534028 12  -3.086  0.0094

Results are averaged over the levels of: A
```

**The Take Home Message**

For unbalanced designs, use LS Means and partial SS.