1 Stat 5303 Spring 2005 — Fitting Split Plots in JMP

JMP can directly fit split plot designs only if the whole plot design is CR[1] or RB[1]. If the whole plot design has more than one factor, you need to build the analysis of variance table yourself (or use a different program).

1.1 Whole plot design is CR[1]

Suppose we have a whole plot factor A with a levels, and each level is repeated m times for a total of am whole plots. In the subplot, we can have any design we like, so lets say we have two crossed factors D with d levels and F with f levels. In your data, you will need a variable Y for the response, variables for the levels of the three factors, and another variable W that identifies the whole plot.

In the whole plot, each whole plot is assigned to one of the levels of A, or whole plots are *nested* in levels of A. In JMP notation, the correct model is

$$Y \sim A + W[A]\&\text{Random} + D + F + DF + AD + AF + ADF$$

JMP will then give the correct tests, using the whole plot error to test whole plots and the subplot error to test everything else. The whole plot error consists of the variation between whole plots with the same treatment level, and it will have $am - m = m(a-1)$df.

1.2 Whole plot design is RB

We again have a whole plot factor A with a levels, but these are arranged in m randomized blocks. The subplot design is as given in Section 1.1. In your data, you will need a variable Y for the response, variables for the levels of the three factors, and another variable B that identifies blocks.

In the whole plot, block contains all levels of A, so A and B are crossed. In JMP notation, the correct model is

$$Y \sim A + B + AB\&\text{Random} + D + F + DF + AD + AF + ADF$$

JMP will then give the correct tests, using the whole plot error to test whole plots and the subplot error to test everything else. The whole plot error is now the same as the block by A interaction, and so it has $(a-1)(m-1)$df.

1.3 Whole plot design is RB[2]

We have whole plot factors A with a levels and C with c, arranged in m randomized blocks (so each block has ac whole plots). The subplot design is as given in Section 1.2. In your data, you will need a variable Y for the response, variables for the levels of the three factors, and another variable B that identifies blocks.

In the whole plot, each block contains all levels of A and C, so A, C and B are crossed. The whole plot error term is the obtained by *pooling together all the interactions with block*, in this case the AB, CB and ACB interactions. There doesn't seem to be any way to get JMP to pool terms together, so *you will need to do this by hand.*

Fit the JMP model

$$Y \sim A + C + AC + B + AB + CB + ACB + D + F + DF + AD + AF + ADF \tag{1}$$

If your data are unbalanced, use the "Sequential anova" option. If your data are balanced, then you do not need to use sequential anova.

1. The tests given for subplot effects will be correct (meaning you start at the bottom of the table, and test for interactions first. You may need to refit leaving out insignificant interactions to test for lower-order interactions and main effects).

2. To get the correct whole plot residual sum of squares, add the SS for AB plus CB plus ACB. To get the df for whole plots, add the degrees of freedom for these three terms. To get the whole plot error, divide the whole plot residual SS by the whole plot df.

3. Use this as the error term for testing whole plot effects.

While this method will give the correct F tests for whole plots and for sub-plots, any other computations that the program does, such as comparisons of means (we haven't learned how to do this yet) or computing standard errors, will be incorrect.

1.4 Whole plot design is CR[2]

Suppose we have a whole plot factors A with a levels and C with c, and each combination is repeated m times for a total of acm whole plots. In the subplot, we can have any design we like, so lets say we have two crossed factors D with d levels and F with f levels. In your data, you will need a variable Y for the response, variables for the levels of the three factors, and another variable R that identifies the replication number. This latter variable plays no role in the design, but without I don't see how to trick JMP to give you the right answers.

As in Section 1.1, the whole plot error is computed from the sum of squares between whole plots that get the same combination of factor levels, and so it has $abm - ab = ab(m-1)$df.

To trick JMP, fit the model

$$Y \sim A + C + AC + R + AR + CR + ACR + D + F + DF + AD + AF + ADF \tag{2}$$

which is the same as (1), but with the replication number R replacing the block number B.

1. The tests given for subplot effects will be correct, subject to the same comments given in Section 1.3.
2. To get the correct whole plot residual sum of squares, add the SS for R plus AR plus CR plus ACR. To get the df for whole plots, add the degrees of freedom for these four terms. To get the whole plot error, divide the whole plot residual SS by the whole plot df.
3. Use this as the error term for testing whole plot effects.

F-tests will then be correct, but computations with means will be incorrect.

2 SAS

SAS is a little better (R, S-Plus and SPSS are even better), because it will correctly pool sums of squares together, but it will still not do the tests for you automatically (at least using the GLM procedure).

Here is a SAS program that fits the acacia gum split plot design, and gets the right tests:

```
proc glm data=work.acaciagum;
   class P D pH Batch;
   model y=Batch P D P*D P*D*Batch ph ph*P ph*D ph*p*D;
   random P*D*Batch/test;
run;
```

Points to notice:

1. SAS calls a factor a "class" variable. These need to be specified explicitly every time you fit a model.
2. SAS indicates an interaction using a `*`.
3. You still need to put terms into a model explicitly.
4. The terms `P*Batch` and `D*Batch` are not in the model, so they automatically get pooled into the next larger term, which is `P*D*Batch`. SAS acts correctly here; JMP does not act correctly.
5. The `random` statement tells about error terms.
6. This won't work with type II sums of squares as discussed in this class.