

Exercise 5

PLSCI 7201

9/11/2020

Summary

The aim of this exercise is to walk you through the analysis of blocked and split plot designs. The first exercise uses an ANOVA to analyze a blocked design. You should already be somewhat familiar with running these analyses, so you can lean on your previous code to get you going. Slides 24 and 25 from the lecture should give you some additional insight, although the example below is slightly more complex. The second and third exercises will show how blocked and split plot designs can be fit using ASREML.

ANOVA approach for a blocked design

The goal of this experiment was to assess the effects of nitrogen (150, 175, and 200 lbs ha⁻¹) on yield for three wheat varieties: Avenger, Bobcat and Chris (A, B, and C, respectively). The experiment was laid out as a randomized complete block design with nine blocks. Each combination of treatment only appeared once in each block. The researcher that provided the data wasn't sure if the data contained any missing observations. The data is in the file "dataSet1.rds".

- **Q1.** Confirm that the data is balanced. This can be done easily using the `table` function. `table` will count the occurrences for the levels of a factor or sets of factors. You need to specify the columns of the data that you want to build a table from.
- **Q2.** Fit an ANOVA model for yield that includes the main effects for nitrogen, line and block, and the interaction effects for nitrogen by line. Make sure to choose the right ANOVA (Type I, II, III). What type did you choose? Why?
- **Q3.** Based on the output from the model above, do you think block effects are contributing to yield?
- **Q4.** What does the test for the main effect of Line tell you? Do you think this factor has an effect on yield? How is the F-test being performed for this factor (i.e. what models are being compared)?
- **Q5.** Analyze this experiment as a completely randomized design using two-way ANOVA. Do these results differ from the blocked design analysis? In this case, how does the choice between a completely randomized design and block design influence the F-tests?

Analyzing a split plot design using ASREML.

An example

I will first demonstrate how to fit the mixed model for a split plot design in `asreml`. This example uses the Yates oat data set. In this data set there are three oat varieties and four nitrogen levels. The varieties were randomly assigned to whole-plots, and nitrogen was applied at random to each subplot. The field was split into six blocks and each of the varieties was assigned once to each block. With this design we would say that the whole plots are nested within blocks. In this analysis we will consider block and the whole-plot effects as random. All other terms (nitrogen level, variety and the interaction between nitrogen level and variety) are fixed.

First, load the data. `Wplots` indicates the whole plot for each observation.

```
library(asreml)
# data(package='asreml')
data("oats")

head(oats)
```

We can look at the design for the whole plot treatments in two ways. Lets first analyze it as a completely randomized design (i.e. no block effect). Fitting the model in asreml is pretty straightforward. The formula is pretty similar to the form used for `lm()`. The difference is that with asreml we can specify a random term along with a variance-covariance structure. Here we assume each whole plot has the same variance and are not correlated (identity variance, `idv`).

```
crdMod <- asreml(fixed = yield ~ Variety * Nitrogen, random = ~idv(Wplots), residual = ~idv(units),
  data = oats)
```

We specify the response variable along with the fixed components of the model in the `fixed` argument. Notice that the formula is specified the same way as `lm()`. We are fitting the main effects of both factors as well as the interaction (`Variety*Nitrogen` expands to `Variety + Nitrogen + Variety:Nitrogen`). We have an argument for the random terms in the model (whole-plot). Again, we assume all whole plots have the same variance and the effects are not correlated. `~idv(Wplots)` tells asreml to use an identity matrix with homogenous variance for the whole plot effect.

Calling `summary()` on the object will return a bunch of useful information.

```
summary(crdMod)
```

- `call` stores the argument formula for the model that was used to generate the output.
- `loglik` is the log-likelihood of observing the data given the parameters. This is pretty important because we can use this data to assess fit and compare between different models.
- `varcomp` are the variance components in the model. Notice that we have one for the whole plot and error (`units`). The first two give the estimate for the variance component and the standard error of the estimate. The next column is the *z* ratio (ratio of the first two columns). The next column tells you the boundary constraints for the variance component. We can set the variance estimates to be positive (P), fixed (F), or unconstrained (U, can be positive or negative). In most cases, we want our estimates to be constrained so that they are positive. The thing to look for in this column is variance components that have a B. This usually means that the estimate is at the boundary. In other words the estimate is so small for this term that the algorithm wants to fit a negative value. Usually it is best to drop these terms and fit a simpler model.
- `aic` and `bic` provide a metric to assess fit and compare models.

Finally, we can test whether any of the fixed terms have an effect on yield using Wald's test.

```
wald(crdMod)
```

Fitting the model with blocking is a little bit more work. The main difference is now we have two random effects (block and whole plot) and the whole-plots nested are within the blocks. To account for this we just add a few more terms to the `random` argument. The `idv(Blocks):id(Wplots)` means that whole plots are nested within blocks.

```
rcbdMod <- asreml(fixed = yield ~ Variety * Nitrogen, random = ~idv(Blocks) + idv(Blocks):id(Wplots),
  residual = ~idv(units), data = oats)
summary(rcbdMod)
```

Notice the new variance component in `varcomp` and the name of the row corresponding to the whole-plot error.

Split plot exercise

You are evaluating the efficacy of a new systemic insecticide. A systemic insecticide is one that is absorbed by the plant, and so it can be applied to individual experimental units relatively easily. To this end, we selected three levels of insect pressure (0, 500 and 1000 insects per m^{-2}) and three dosages of insecticide (0, 0.1 and 0.5% v/v). Since it is difficult to apply pest levels randomly to experimental units, the experiment run using a split plot design. The insect levels were considered the whole plot factor (`wPlot` column lists the whole plot assignment for each observation), and the insecticide dosage was assigned to subplots. The experiment was split into nine blocks and all levels of the whole plot factor appeared once in each block. These data can be found in the file `dataSet2.rds`. As mentioned in the oats example, the randomization at the whole plot level can be viewed as a randomized complete block design (RCBD) or a completely randomized design.

Use the example for the Yates experiment to fit a mixed model that includes fixed main effects for insect pressure (`PestLvl`) and insecticide dosage (`Trt`), and the interaction between the two, as well as random effects for block and whole-plot. Remember that whole-plots are nested within blocks. You may see a warning message. When these messages pop up its best not to ignore them. It can be an indication that there may be an issue with the model. A quick google search will allow you to diagnose things.

- **Q6.** Report the variance estimates for the two random effects. What can you conclude from these estimates? Do you think it is necessary to consider the whole plot design as a RCBD? If not, refit the model and examine the estimates for the variance components.
- **Q7.** Test for the main effects of insect pressure (`PestLvl`), insecticide dosage (`Trt`), and their interaction. What can you conclude from this experiment?

Note: when you compile the Rmd file you may get an error. I think this is because the output from `summary()` with `asreml` has a percent symbol, which is an escape character in latex. You can prevent this by switching `echo = T` to `echo = F` or `eval = T` to `eval = F` in your code block.