Repeated measures in SAS: explanation of code in asparagus.sas

**Computing and analyzing summary statistics**:

One of the advantages of separating the data step (to create or manipulate data sets) and the many proc steps (to do an analysis) is that the proc steps don’t need to care about “where” the data set comes from. We can take advantage of this by computing summary statistics and saving them in a new data set. That new data set is then used in a proc glm to analyze those summary statistics.

Many proc steps can produce new data sets. We will use proc means; to compute means for each plot. These are averages of the 3 years of data for a combination of block and treatment. We use the by statement to compute averages for each subgroup. That requires that the data by sorted so all observations in a subgroup are together.

Sorting the data:

**proc** **sort** data=asparagus;

by block trt;

**run**;

The by statement specifies the variables defining each group of observations.

Computing means for each group:

**proc** **means** noprint;

by block trt;

var yield;

output out=means mean = meanYield;

**run**;

The noprint on the proc means statement suppresses all printed output; the goal here is to produce a new data set, not to see the individual plot means.

The var statement names the variable to be used. You can specify multiple variables if you want.

The output statement provides the information about where and what to store in the new data set.

Out= specifies the name of the new data set. The rest is a set of **keyword = variable** pairs. The keyword defines what, mean= specifies the average. The variable provides the name to use for the results. If you have more than one variable in the var statement, you need more than one variable name, one for each input variable.

After that, all you need to do is write the code to analyze the data. After averaging away the repeated measures (years), you have an RCBD model.

The **lsmestimate** command:

The estimate command is wonderful when you want specific comparisons of means in a 1 way ANOVA or marginal means in a factorial ANOVA. It’s less convenient when you want specific comparisons of cell means. If you try the obvious, using something like *estimate ‘a1 – a2 for b=1’ a\*b 1 -1 0 0;* you get a ‘non-est’ error. That’s because that specific comparison depends on the parameterization of the model. If you write out the model equation for the two cell means, then their difference you get:

What you asked for, , is only part of what you need. You can get the estimate you want by: *estimate ‘a1 – a2 for b=1’ a -1 1 a\*b 1 -1 0 0;* but that’s not very intuitive and not at all convenient.

The lsmestimate command is a relatively recent addition to most SAS modeling procedures. It allows you to specify an estimate in terms of the cell means, not in terms of the effects parameters. The only gotcha is that the order of the bits is not the same as that for the estimate statement. The model term goes first, then the label, then the coefficients. Maybe that’s because you’re only allowed to have one model term in the lsmestimate statement. For the 2 x 2 factorial illustrated above,

lsmestimatea\*b ‘a1 – a2 for b=1’ 1 -1 0 0; gives you the estimate you want.

There are 12 cell means for the asparagus data (4 treatments x 3 years), so you should specify 12 coefficients. There is a short cut: if you specify fewer than 12 coefficients, SAS replaces the missing “trailing” coefficients with 0’s. If you wanted trt 1 – trt 2 in year 2, the SAS statement is:

lsmestimate year\*trt 'trt 1 - trt 2 in year 2'

**0** **0** **0** **0** **1** -**1** **0** **0** **0** **0** **0** **0** ;

Which you could shorten to:

lsmestimate year\*trt 'trt 1 - trt 2 in year 2'

**0** **0** **0** **0** **1** -**1** **0** **0;**

But you can’t get rid of the “leading” zeros. If you did, you would get the difference in year 1, not year 2.

**Specifying models for correlated observations:**

We’re already seen how to use proc mixed to fit mixed models. Remember that a mixed model is specified as one or more random effects (e.g. main plot errors) and conditionally independent residual errors (e.g. split plot errors). Conditionally independent means that the given the main plot error, so within a main plot, the individual observations have independent errors. Repeated measures models generalize that to user-specified correlation matrices. Proc mixed will fit all sorts of correlation models and combinations of random effects and correlations.

Correlation models are specified using a **repeated** statement. The simplest form looks like:

repeated /type = ar(1) subject = block\*trt;

The type= argument specifies the correlation model, using SAS’s specific name for each model. The subject = specifies which observations are correlated and which are independent. Remember that only observations from the same individual are correlated. Observations from different subjects are independent. It’s important to get the subject correct! In the asparagus data set, the “subjects” are plots. These are uniquely identified by the combination of block and treatment.

If you omit the subject= argument, all observations are assumed to be correlated. That would be appropriate if you had one time series (e.g. 31 observations of something measured annually in 1990, 1991, … 2020).

**Useful options for the repeated statement:**

1) repeated year /type = ar(1) subject = block\*trt;

By default, SAS uses the order of observations to relate rows of data to the correlation matrix. This is correct when observations are in the sequence year 1, then year 2, then year 3. It is not correct when observations are jumbled (year 3, year 1, year 2) or some years are missing (e.g., only year 1 and year 3 on a subject). Adding the “ordering” variable lets SAS work out the correct relationship.

I always use an ordering variable. If it’s not necessary, there is no harm. If it is necessary, it really helps.

2) repeated year /type = ar(1) subject = block\*trt r rcorr;

It is often helpful to see the fitted covariance or correlation matrix. SAS prints out the estimated parameters for the correlation matrix, but it can be useful to see the entire matrix. The r option prints the covariance matrix. The rcorr option prints the correlation matrix. Beware: if you have lots of observations per subject, these will be large matrices.

So the full code to fit an ar(1) model to the asparagus data is:

**proc** **mixed** data=asparagus;

class block year trt;

model yield = block trt year year\*trt /ddfm = kr;

repeated year /type = ar(**1**) subject = block\*trt r rcorr;

run;

This would be followed up with any of the “after the ANOVA” commands: lsmeans, estimate, …

Note that the model statement includes specifying KR degrees of freedom. Correlated data models are one situation where KR and Satterthwaite are not equivalent. KR works better.

**Fitting the ar(1) + RE model:**

You can use both random and repeated statements in the same proc mixed. The ar(1) + RE model has ar(1) errors with subject = block\*trt and a random effect for block\*trt. The code is:

**proc** **mixed** data=asparagus;

class block year trt;

model yield = block trt year year\*trt /ddfm = kr;

random block\*trt;

repeated year /type = ar(**1**) subject = block\*trt r rcorr;

run;

**Fitting unequally spaced time intervals:**

The spatial exponential model generalizes ar(1) to arbitrary time intervals. This is type=sp(Exp)(time variable). You need to specify the variable containing time information. That’s year for the asparagus data, so this model would be specified as type=sp(Exp)(year). The difference in time between two observations is treated as a distance. This is computed for you.

The parameter is called the range and is related to the correlation. The estimate for the asparagus data is 2.252. The correlation between two observations 1 time unit apart is exp(-1/range). That is 0.64 here, matching the ar(1) estimate. And if you look at log Likelihood or AIC statistics, you see that the ar(1) and sp(Exp) models have the same values. Again, expected here because time intervals are evenly spaced. They will not be the same if time intervals are irregular.

**Other correlation models:**

|  |  |
| --- | --- |
| Model | Specified as: |
| ar(1) with heterogeneous variances | type=arh(1) |
| antedependence | type=ante(1) |
| ar(1) with unequal times | type=sp(Exp)(year) |
| unstructured | type=un |

These are the most commonly used and useful models. SAS provides many more for all sorts of specialized situations. See the help file for the repeated statement, type= option.