

collard.sas: Explanation of code

Goals of code:

- Fitting a 2-way ANOVA model
- After the ANOVA options for main effects and cell means
- Analyzing data from an RCBD

**Fitting a 2-way ANOVA model:** `model conc = clarify size clarify*size;`

Earlier, we used `proc glm;` to fit ANOVA models. `proc glm;` can also be used to fit more complicated models. Just name the components on the right-hand side of the model statement. The interaction is given by two terms connected with an asterisk.

Remember that anything identifying groups must be named in a class statement before being used in the model statement. You only need the factor names in the class statement; you do not need the interaction there. The interaction only goes in the model statement.

`Proc glm` provides three sets of F tests:

At the top of the output: box with model and error. This is the overall test of the entire model: It compares the equal means (intercept only) model to the specified model. Probably not useful for a two-way ANOVA and certainly not useful when you have blocks in the model.

The error line in this output is where you will find the MSE, the estimate of the pooled error variance.

Type III (partial) F tests: These are the tests that we are using in this class because I believe they answer the most interesting questions. I usually ignore the type I (sequential) F tests.

**After the ANOVA:**

You can do all the “after the ANOVA” analyses using the principles discussed in the one-way ANOVA example. You just name the factor of interest in the `estimate` or `lsmeans` statement. You can name multiple factors, e.g. `lsmeans clarify size;` to get both the size `lsmeans` and the clarify `lsmeans` from one statement.

If you want information or comparisons between individual treatments, put the interaction, e.g. `clarify*size`, in the `lsmeans` or `estimate` statement (example in the `proc glimmix` also works in `proc glm`).

You may notice the code has 4 `estimate` statements but only three results. The log file tells you why. The last estimate is not estimable. The goal was to estimate the difference between large and small in the NO group. That is a simple effect and is a comparison of the first two LSMEANS. The log file tells you that is non-estimable. That means the answer depends on the choice of the model

parameterization (not good). It is possible to write an estimable contrast for the simple effect, but it takes writing out each cell mean in terms of the model parameters. It is much easier to use the `lsestimate` statement in `proc glimmix`.

### **Alternate way to fit models: `proc glimmix`;**

`Proc glm` is one of the very first SAS procedures; it dates from the early 1970's. Over the years, SAS has added options to old procs and added a lot of new procs. Some of the new procedures provide useful output not in the earlier procs.

`proc glimmix` is one of the newest SAS procs. It fits models all the models `proc glm` fits and a huge number more. The `glim` part of the name refers to models with non-normal error distributions. The `mix` part refers to mixed-effect models with more than one random effect. `Glimmix` extends both `proc glm` (which we have used) and `proc mixed` (used heavily in Stat 402).

You should know about `glimmix` because SAS has decided to put all the newest bells and whistles into `proc glimmix` and not include them in the earlier procs (`glm` and `mixed`).

`glimmix` provides improved output and some new features that are not in `proc glm`. Three that are useful now are improved results for differences of means (either treatment means or factor `lsmeans`), a letter display for differences, and the ability to easily write certain sorts of contrasts.

The syntax for `proc glimmix` is exactly the same as for `proc glm`, when you want to fit models with normally distributed errors. Same `class` statement; same model statement.

The `lsmeans /diff` option provides nicer results for pairwise differences between means.

The `lsmeans /lines` option provides a compact display of which groups are significantly different from each other. `adjust=` names the multiple comparisons adjustment; it can be used with either `/diff` or `/lines`.

The `lsestimate` statement provides a way to write estimates in terms of the cell or marginal means, not in terms of model parameters. We don't usually need to care about the two different methods unless we want certain contrasts such as simple effects. The `lsestimate` works here, but the analogous `estimate` statement in `proc glm` did not.

Notice that in an `lsestimate` statement, the label is second and the factor name is first. This is switched from an `estimate` statement.

### **Analyzing data from a randomized complete block design:**

No code, because this is identical to the steps above. Name both the blocking variable and the treatment variables in the `class` statement. Just leave the `block*treatment` interaction out of the model, because it can not be estimated without replication within a block. You can have treatment interactions (and should have them if you have a 2 way factorial in blocks).