collard.r: 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 a block design

We will use functions in the emmeans library to calculate type III (partial) F tests and do all the 'after the ANOVA' analysis. emmeans is the successor to the lsmeans library and comes with an extensive set of vignettes.

Vignettes are optional information provided by some packages. Vignettes illustrate the use of functions and interpretations of their results. If a package has one or vignettes, I start with those to learn how to use the package.

Displaying available vignettes: `vignette(package='emmeans')`
lists all the vignettes in the named library

Displaying a vignette: `vignette('confidence-intervals')`
opens the named vignette in a browser window

We use the collard data set in collard.txt. This is the lecture example. Two factors: clarify and size, each with two levels. The response is the measured concentration of riboflavin in collard leaves.

Fitting a 2 way ANOVA model:
`collard.lm <- lm(conc ~ clarify.f + size.f + clarify.f:size.f, data=collard)`

We've seen the lm() formula specification of main effects and interactions when we discussed multiple regression. The application to factorial designs is identical. The only difference is that the interaction is always included (usual US practice) except when it can not be included (e.g., in most block designs).

clarify.f and size.f are the factor versions of clarify and size. The third term in the model,
`clarify.f:size.f`, with the colon indicates the interaction between clarify and size. A shortcut specification if you don't care about the order of terms is:
`collard.lm <- lm(conc ~ clarify.f*size.f, data=collard)`.
Connecting two factor names with an asterisk is equivalent to each component and their interaction. This can especially useful with three factors because `A*B*C` generates
`A + B + C + A:B + A:C + B:C + A:B:C`.

The default anova() function provides sequential (type I) tests. US practice is to prefer partial (type III) tests. Functions in the emmeans library make these easy to obtain. There are other ways to get type III tests. If you search the web, you will see illustrations using drop1() and the Anova() function in the car library. Both of these require resetting the default indicator variables, to something called an orthogonal set of indicators. This is easy to forget to do and there is no warning when you forget. You just get the wrong results. Actually, you get the right results for the wrong questions. Ask me in lab or office hours if you will be using a lot of factorial ANOVA and want to know why the default answers the wrong question.

Using emmeans: `emmeans(collard.lm, c('clarify.f', 'size.f'))`
Before you can do anything using emmeans, you need to create a reference grid. This is information about the factors you are interested in. To get the Type III F tests, provide the result from lm (collard.lm) and a vector with the factors. You do not include the interaction. When you specify two or more factors, the interaction is included if it is in the lm model. Later, we will see how to get marginal means by specifying only one factor to emmeans().

Type III F tests: `joint_tests(collard.emm)`
To get the Type III tests, pass the result from emmeans into `joint_tests()`. The result is an F test for each factor named in the emmeans() call and their interaction.

After the ANOVA: `emmeans(collard.lm, 'clarify.f')` and subsequent code
The emmeans library provides lots of useful results about marginal means (also called lsmeans). To get them for a factor of interest, call emmeans() with only the name of that factor. The results using collard.clarify will all be about the clarify marginal means (i.e., averaged over size levels). If you want results about size, save and use the results from `emmeans(collard.lm, 'size.f')`.

Print the marginal means and related info: just print the result from emmeans
You get the estimate, its se, the error df, and a 95% CI.

Pairwise differences: `pairs()`
Just like pairwise differences in 1-way ANOVA, but now based on the main effect emmeans object. For each pair of levels, you get the estimated difference, the se, error df and a T test of difference = 0. When more than one difference, these are Tukey adjusted, but you can change that with the adjust= argument. See `?contrast` for the names of available adjustments.

a-priori constrasts: `constrast()`
Just like contrasts in a 1-way ANOVA, but now based on the main effect emmeans object. The order of the marginal means is the order they are printed (going from top down).

"compact letter displays": `CLD()`
This uses letters or numbers to present information about pairwise differences. Two groups are NOT significantly different if they share the same letter or number. Uses Tukey adjustment by default.

You can also plot the results. The plot shows the mean and 95% confidence intervals for each group.

The default plot has the response on the X axis and the groups along the Y axis (a horizontal plot). Adding horizontal=F (as in the last plot statement) gives you a vertical dot plot. You can also add xlab="label" and ylab="label" to change the X axis and Y axis labels.

You can plot main effects or cell (treatment) means. The last two statements, a plot and a CLD, use the emmeans object with both clarify and size, so they show information about cell means.

Analyzing data from a randomized complete block design:
No code, because this is identical to the steps above. Just leave the interaction out of the lm model, because it can not be estimated without replication within a block.