Advancing in R

Module 5: ANOVA & ANCOVA

Supplementary exercises

In this exercise, we will evaluate the effectiveness of four experimental food SUPPLEMENT treatments (the control treatment and three experimental supplements called agrimore, supergain, and supersupp) for increasing the weight GAIN of cattle fed three different DIETs (barley, oats, and wheat). This is a full-factorial design, meaning that replicates include assessments at all possible combinations of the two factors' levels. There are consequently twelve groups of subjects. Along the way we'll practice interpreting coefficients (this becomes a slightly hairy business when there are many factor levels, especially if interactions are involved), which will be a good test of your understanding of how GLMs work within the context of linear equations. Finally, you may wish to try building a bar plot to illustrate your findings.

1. Open a new project in a new folder, and start a new script. Save it with a .R suffix to make sure that RStudio interprets commands from it correctly.
2. Read the data from the file "growth.csv" (in your course materials) into a new object.
3. Check that the file has loaded correctly, and examine the data structure using str().
4. Check the data variables for any noticeable errors. If you see any records that look suspicious, delete the entire row in question. Examine the response variable, GAIN, and make a note of any concerns about its distribution.
5. Make exploratory boxplots that illustrate the combined effects of DIET and food SUPPLEMENT on GAIN. You may want to make two alterations to the default settings. First, R will present the treatments in alphabetical order by default, but you might like to change this. Why do you think the alphabetical order might be sub-optimal, especially for the food supplement treatments? To change the "reference" level of a factor, use the ref="" argument within the command relevel(). Consult the help file for relevel() if you are unsure about the syntax. One nice feature of the relevel() command is that it will both change the order of presentation of factors within figures as well as change the factor level assigned the intercept value (as opposed to deviations from the intercept). You might also find that it's hard to read the x-axis labels. Zooming will help, but so will changing the font size of the x-axis text using the argument cex.axis=0.5 within the boxplot command (this changes the font to half the default size).
6. Using this exploratory plot, can you predict the values for the coefficients of your GLM? Remember that you will have a large number of coefficients: you'll need two intercept deviations for the DIET main effect (one deviation for oats relative to barley, and one for wheat relative to barley), and three deviations for the SUPPLEMENT treatment (for the intercept deviations relative to the control of agrimore, supergain and supersupp, respectively). For the interaction you'll need six more coefficients to describe how particular combinations of factor levels deviate from those provided for the main effects. Write down your guesses for the main effects for now – we'll examine the interaction effects more after we have a model.
7. Build a general linear model that assesses the effect of DIET, SUPPLEMENT, and the interaction between the two factors on mass GAIN. Examine the diagnostic plots and make a note of any concerns you have about these.
8. Now examine the model summary. Note the large number of coefficients you have estimated! Are the coefficients for the main effects in line with your predictions from (6)? Try to work out what each of the six interaction coefficients mean. Work through this carefully, as it's important and we'll come back to it later! Use your figure to help you, and consult your classmates or instructors for guidance if necessary.

9. Conduct model simplification by attempting to remove the interaction term first, followed by the two main effects if the interaction F-test is NS. Examine the diagnostics of the simplest adequate model.
10. Repeat the simplification process using comparisons of both the likelihoods and the AIC values for each of your models. Which model has the highest likelihood, and which model fits the best according to the AIC? If this is different, explain why in a couple of sentences.
11. Now look at the parameter estimates of the best fitting model. What do all of the coefficients mean? Make a table of the weight gain predicted by the model for each of the 12 combinations of factor levels (i.e. diet = barley and supplement = control, diet = barley and supplement = agrimore, etc.) using the predict() function. The expand.grid() function might be helpful to generate the twelve combinations automatically.
12. Now check you understand what the coefficients mean and how the predict() function calculates their value by calculating the prediction manually, using the coefficients table only. Your results should exactly match the predictions made by predict().
13. Now examine the parameters for the simple model. Are all the coefficients significant? You will have noted that can't remove either of the main effects without significantly decreasing model fit, but it is nevertheless still possible to simplify the model further. We can do this by collapsing levels of factors that seem to be similar into groups, and testing whether recoding factors in this way results in a significant change in model deviance. As a first step I will create a new vector for the SUPPLEMENT factor so that I don't destroy the information in the original data:

```
SUPP2<-factor(WEIGHTS$SUPPLEMENT)
```

Next, I'll lump together the two best and the two worst diet supplements (I make this call by inspecting the exploratory plots, which suggest that control and supergain are similar, as are agrimore and supersupp).

```
levels(SUPP2)[c(2,4)]<-"best"
levels(SUPP2)[c(1,3)]<-"worst"
```

Now inspect SUPP2 and see what's happened. Then use it instead of SUPPLEMENT in a new model, and use anova() to test whether the new model is any worse than the one with four levels. If the new model is acceptable, examine the diagnostics again to see if they have improved.
14. Examine the table of coefficients for your minimal adequate model, and re-evaluate your predictions generated in question (6). How much of the variation in mass is explained by diet and supplement?
15. Now prepare a publication quality plot. If you want, you can make a boxplot using whichever package you prefer, but try to adjust the labels so that the figure looks nice, and include a legend and colour or grayscale fills to help distinguish the two factor levels. Or if you prefer, you can try to make a barplot. To do this, we will exploit another handy function called tapply(), which stands for table apply. This command computes summaries of variables, and arranges the output in a tabular form. We'll use it to extract the means and standard deviations of each combination of DIET and SUPPLEMENT. tapply() needs as input the variable to summarize, then a list containing the factor variables, then the function to apply to the records belonging in each factor group:

```
MN.GAIN<-tapply(WEIGHTS$GAIN, list(WEIGHTS$DIET,
WEIGHTS$SUPPLEMENT), mean)
```

Examine the object created by this command. This is a matrix rather than a dataframe, and sometimes needs to be treated slightly differently as a consequence, but it has nice properties that are useful when making a barplot. Create two more matrices like the one created above for the standard deviations (substitute the function sd for mean in the code) and sample sizes (use the function length) of all combinations of factors as well. To begin the barplot, call the matrix of means in a barplot() command. Note that below I not only plot the graph, I also store an object called MIDS, which is a matrix of the x-axis position for each of the bars in the plot. By storing this object now, I can use it later to add error bars and sample sizes to the plot. In the command below, beside=TRUE forces the barplot to plot bars from the same column of the matrix beside instead of on top of one another.

```
MIDS<-barplot(MN.GAIN,beside=TRUE,col=c("dark gray","light
gray","white"), ylim=c(0,40), ylab="Mass gain (kg)")
```

Have a look at the new object called MIDS in the workspace -- you should see that it contains only the x-axis values for the midpoints of the bars in our bar plot.
To add error bars indicating SDs, we'll use the command arrow(), which draws an arrow between two coordinates on a plot. We'll start the arrow at the mean value - 1 SD (we need to provide two coordinates for this point: MIDS gives the x-axis position, and MN.GAIN-SD.GAIN the y-axis position), and then stretch it up to the mean value + 1 SD.

```
arrows(MIDS,MN.GAIN-SD.GAIN,MIDS, MN.GAIN+SD.GAIN,
angle=90,length=0.05,code=3)
```

The argument angle=90 tells R to make the arrowhead straight across, and the argument code=3 makes a straight line at both ends of the arrow.
Let's be fancy and add a sample size designation to each bar. Again we need to give coordinates, so I use MIDS for x, and an arbitrary height of 1 for y.

```
text(MIDS,1,paste("N =",N.GAIN),cex=0.6)
```

Finally, add a legend so anyone can work out which bar is which.

16. Compose several lines of text that would be suitable for a Results section, in which you describe the effects of diet and supplement on weight gain, and include the details in parenthetical statements that cite the appropriate statistical parameters. You might find it easier to refer to a table of parameter estimates rather than cite all of the coefficients individually, but you should provide test statistics and p-values for key null hypothesis tests. Feel free to refer to your figures in this sentence as well.
17. Make sure you have annotated your script well enough for your future self or someone else to make sense of it, and then save it.