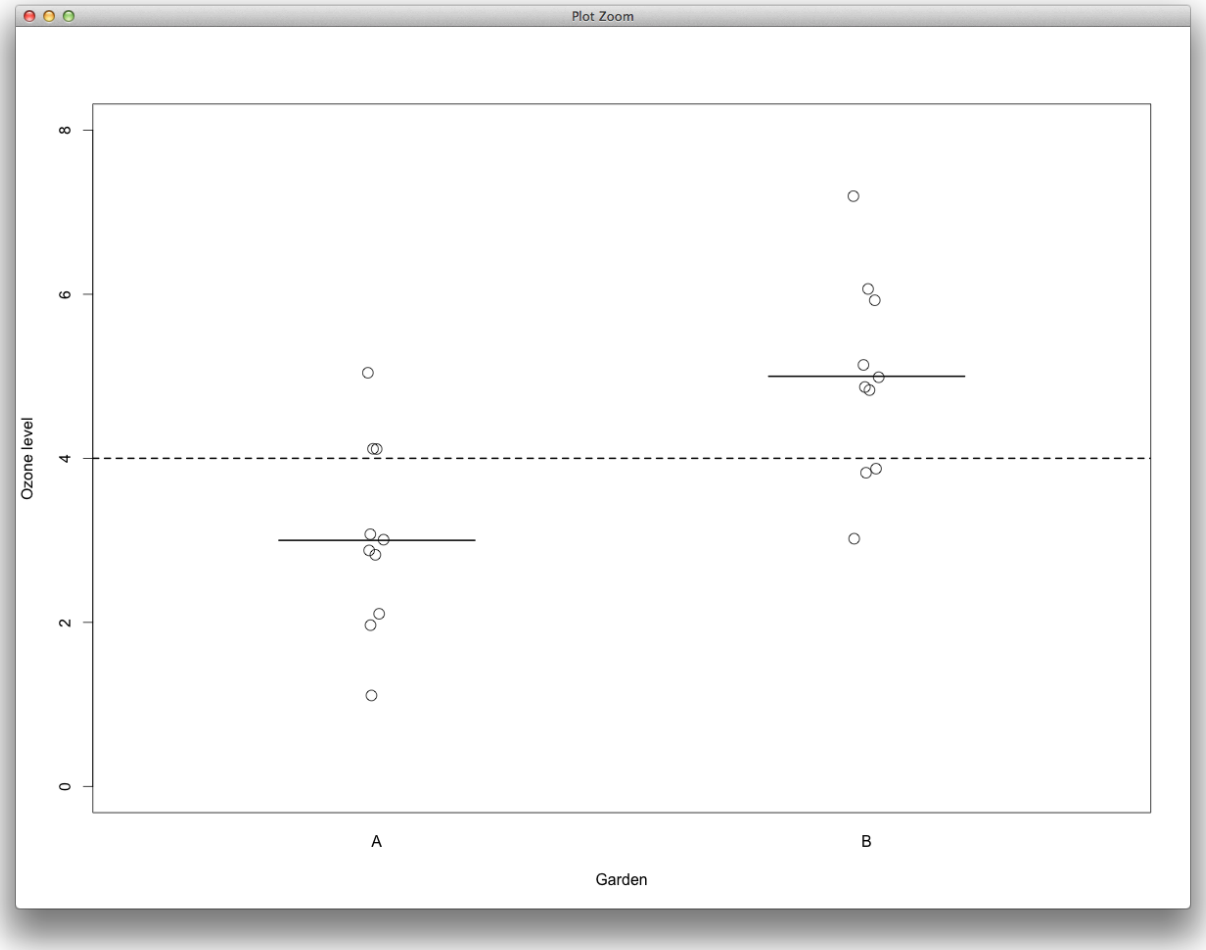
Advancing in R



Module 5:

General linear models and ANCOVA: factorial predictors and model selection

Introduction

In this module, we will broaden the linear model to include categorical predictors. General linear models estimate linear equations by fitting slopes for continuous variables, and separate intercepts for different categories. We will first practice using a simple dataset featuring only a single categorical variable, and note the resemblance between these models and single factor analyses such as t-tests and ANOVA. We will then analyse a dataset with both a factor and a covariate.

This practical assumes you are familiar with the general syntax of coding in R, with making basic plots, with the theory and practice of linear and multiple regression, and that you have reviewed material covered in the first four practical exercises. If you need to, review this material before attending the practical session.

Learning outcomes

Know how to graphically explore differences between factors, with and without covariates

Know how to change factor reference levels (supplementary exercises)

Know how to interpret diagnostic plots involving factorial predictors

Know how to assess slope homogeneity in ANCOVA

Know how to simplify ANCOVA models

Know how to interpret GLM coefficients and reconstruct linear equations for each treatment group

Know how to produce publication quality plots illustrating GLM findings

Libraries

In addition to the basic libraries automatically loaded with R, you may wish to use the following additional packages. If necessary, make sure you either have internet access on your machine of that you install these packages prior to the practical.

{ggplot2}

Some useful commands

Below are some of the useful commands you may need to run at some stage of today's practical.

abline()	rep()
anova()	seq()
boxplot()	split()
expand.grid()	summary()
ggplot()	t.test()
head()	update()
levels()	AIC()
legend()	logLik()

lm()	coef()
matlines()	predict()
plot()	qplot()
points()	relevel()

Data quality control and exploration

Begin by loading the RStudio package. Open a new project and a new Rscript, and save each of these using meaningful and descriptive filenames. Include the usual annotation at the start of your script that identifies the purpose of the script and its author, as well as commands that clear the workspace.

Find the data file called "oneway.csv" in the folder for this session, and read it into a well-named data frame. Verify that it has loaded properly, and examine the structure of the object using the str() command. This dataset is from a study of ozone levels measured in two different gardens. The analytic question concerns whether the gardens differ in ozone levels. Use plotting commands to exercise data quality control. Check for outliers. If necessary, clean the data and recode any variables that need to be read differently than they currently are. Make sure you check distributions again after cleaning the data to ensure that you have fixed the problems.

Examine the distributions of the response variable. Do you have any concerns about it? Note that the predictor variable is not numeric, so we cannot examine its distribution. Instead, use the levels() and summary() commands to find out the number of levels that the predictor has, and the sample size for each of these. Now explore graphically the possible difference in ozone across gardens. What kind of plot would best illustrate this? Examine your plot carefully, taking note of the mean values for each level of the predictor. These numbers will be useful in assessing your model later on.

Fitting GLMs, diagnostics, and coefficients

Now build a general linear model that predicts OZONE as a function of GARDEN. The command for a general linear model is lm(), the overloaded function with which you are now very familiar. Note that glm() is a different function that calls generalized linear models – we'll learn more about that shortly.

Examine the diagnostic plots for your model. You'll probably find that this plot looks a bit funny. This is partly because the response variable only takes integer values (an unusual situation), but you will also see that there are only two possible fitted values in the first plot (on the x-axis). Why is this true?

Now examine the model summary. You will note two parameters: what are they? Note the suffix affixed to the term for Garden – what does this mean? Compare the values of the coefficients to your plot, and your estimates of group means. Can you reconcile the values one with another?

By default, when fitting factors R will estimate the model intercept as the intercept for the alphabetically first factor (in this case it is GARDEN A). It will then estimate the deviation from this intercept for remaining factors. This means that to get the intercept for GARDEN B, you need to

add its coefficient to the model intercept. Do the values make sense now? Why doesn't the model just estimate the intercept for each group as a deviation from the global mean?

You may already be familiar with alternative methods for testing differences between groups, such as Student's t-test. R is of course ready to oblige! Use the help function for the command t.test(), and run it comparing the ozone across gardens. Compare the t-value and p-value to that for the GLM. Can you explain the similarity and any difference? To wrap up this short analysis, construct a publication quality plot illustrating your findings, and write a sentence of Results text that summarizes them.

ANCOVA

In this exercise we will add a continuous covariate to a model with a categorical predictor. Such an analysis is often called ANCOVA, short for "Analysis of Covariance". Find the data file called "compensationo.csv" in the folder for this practical, and read it into a well-named data frame. Verify that it has loaded properly, and examine the structure of the object using the str() command.

This dataset is from a study of seed production (FRUIT) for plants growing in two pastures. In one of them, cattle are periodically allowed to graze (GRAZING == "Grazed"), while the other pasture is fenced and free of vertebrate herbivores (GRAZING == "Ungrazed"). In addition, the pastures differ in several abiotic variables. In an effort to account for some of these differences, the scientists have measured the root mass (ROOT) of plants early in the season. ROOT is expected to covary with the most important abiotic predictors of seed production.

Use plotting commands to exercise data quality control. Check for outliers. If necessary, clean the data and recode any variables that need to be read differently than they currently are. How many levels of GRAZING are there, and what is the sample size for each of these?

Explore graphically the possible effects of ROOT and GRAZING on FRUIT. Create what you think is the best plot to illustrate possible bivariate relationships, but also create a plot that illustrates the effect of both predictors in the same panel. You may need to subset your data, e.g., using square brackets, to do this effectively.

Checking for heterogeneity of slopes

Examine the plot illustrating the effect of both ROOT and GRAZING. Do the slopes look like they are equal or nearly so? If so, that's good news: one of the key assumptions when hypothesis testing with ANCOVA is that there is homogeneity of slopes. If the slopes of a continuous covariate differ across treatments (indicated by plotting or the observation of a significant covariate by factor interaction term), then you will want to interpret tests of main effects very carefully. (You may even prefer a different approach altogether – see Logan Ch. 15 for details. The exact approach will depend on which of the predictors is of primary interest and the degree to which the covariate range is similar across treatments.)

In our case there is no evidence of a difference in slopes. Try to guess the slope of the continuous relationship as well as the effect of each treatment. Can you now anticipate what each of these effects will look like in coefficient form? Write down your guesses.

Now build a model that fits an effect for GRAZING, for ROOT, and for the interaction between them. Recall that the operator used for plotting all possible interactions between terms is the * sign.

Examine the model diagnostic plots. Why don't these plots have the narrow range of fitted values that were seen in the first model predicting OZONE? Make a note of any concerning aspects of the diagnostic plots if necessary.

Examine the model summary, but don't get carried away with interpreting coefficients yet! Our first job is to check whether the assumption of homogeneity of slopes has been satisfied, and if so, to remove the interaction term from the model. Examine the interaction term. Should it be removed?

Simplify the model using the update() command, and check that the new model is superior using anova(). Then examine the new, simpler model. Can it be further simplified? Check using further updates and anovas as appropriate.

In addition, calculate the log-Likelihoods for each of the models (including or excluding different factors) using the logLik() function. Which model has the highest likelihood, and what does this mean? Now that you have found the likelihoods for each of the models, also compare them using the AIC (you can calculate this using the AIC() function with each model fit). Which of the models is "best" according to AIC comparisons? Is your conclusion any different to that derived from the F-tests above, or to model likelihoods? If yes, explain why in a couple of sentences.

Once you have found the minimal adequate model, examine its diagnostics once again to verify that they are still OK. Then examine the coefficients. What does the intercept signify? What about the other terms? Write down the model equation, depending on whether or not you are predicting a value for grazed or ungrazed pasture. Use this equation produce a prediction for seed production in both grazed or ungrazed pasture over a range of, for example, 10 root mass values. Hint 1: because GRAZING is factorial you need to use dummy variables to indicate whether the estimated parameter should (1) or should not (0) be used in the calculation of the prediction. Hint 2: you can easily obtain the coefficients from the whole summary table by using the coef() extractor function.

When you have done so, check whether you have done this correctly by predicting the same values but this time using the built-in predict() function and its 'newdata' argument. Hint: the newdata argument takes a dataframe with variables with the same name as in your original data. If you don't remember how to use predict() or the newdata argument, check the previous practical session notes or use the help function.

Visualizing results

Preparing a publication quality plot when there are multiple predictors usually involves using either {ggplot2} to specify the aesthetic values that will encode each variable, or subsetting or splitting your data, then adding points and predictions to an existing plot using low level graphing functions. Use whichever approach you think is best to create a single panel high quality plot illustrating your findings. If you decide to use low-level functions and the predict() command, one function that might speed things up is the command expand.grid(), which creates a dataframe for all possible

combinations of several factors. This becomes an especially useful function when the models you are trying to illustrate have many factors.

Compose one or two sentences of Results text in which you explain the significant predictors of seed production for this study, citing statistics, tables of coefficients, or figures as you see fit. Remember to save your project and R script, and if you want, your figure as well.

~ End of Practical ~