Conservation Biology - Quantitative Methods Module

Yolanda Wiersma

2020-02-20

Contents

Chapter 1

Quantitative Methods Overview

Biology, including ecology and conservation, is becoming increasingly quantitative. Familiarity with a range of statistical tools (frequentist, multivariate, Bayesian) is expected. Many researchers draw on quantitative methods; whether it is process or mathematical modelling, programming, bioinformatics, or bioeconomic analyses. We will only touch on key concepts in quantitative methods; in-depth training would require multiple full-semester courses. Here, we will carry out a few simple exercise that mimics some of the quantitative skills that you need to be a successful ecologist. These are:

1. Use R!
2. Manage data sets
3. Plot data
4. Fit a model to data

In this module there will be a small assignment with each of the above objectives. These are detailed below along with the percentage of your final grade that this is worth (the entire module is worth 10% of your final mark).

##What to hand in (as a single PDF file please!)

1. Written review of the first year guide. How useful was it to help you work through this module?
2. A short document that answers all the questions posed throughout this assignment (they are in italics)
3. Copies of the final version of all plots created (remember you can export these as image files which you can then paste in Word). Convert the

Word document to a PDF file and upload it to the Dropbox folder in Brightspace.

Chapter 2

Using R

We're going to work exclusively with R in this module. Most of you have some R/R studio experience. Dr. Amy Hurford and I, together with Dr. Hurford's graduate student, have prepared a simple "using R" manual that we are planning to introduce to first year Biology students in the near future. This manual is available here (note it is still very much a DRAFT and thus has notes between the authors in the document. Please ignore these).

Your assignment: Provide a written review of the draft manual that gives Dr. Hurford and I some constructive suggestions on how to improve this manual to make it as maximally useful for first year students as possible. You do not need to highlight typos or any residual "notes" we've left each other, rather focus on a "big picture" evaluation of the document. This does not need to be long. It can be as short as a paragraph/half-page of point form suggestions up to a maximum of a full page typed-double spaces (apprx. 250 words).

THIS IS WORTH 10% OF YOUR MARK FOR THIS MODULE.

Chapter 3

Manage a data set

R is really useful for managing data. Here are some commands that come in handy frequently. The reference to "iris" is the built in "iris" data set that comes automatically when you install the baseR package. Try out these commands by typing the code below into the console. *Write down what happens when you enter each command.*

```
data("iris")
```

```
head(iris)
```

```
dim(iris)
```

```
str(iris)
```

```
class(iris)
```

Let's say you want to create a new data frame with only the data for flowers with a petal width greater than 0.3. Here's what the command would look like:

```
lg.petals <- subset(iris, Petal.Width > 0.3)
```

How can you confirm this new data frame is a subset of the larger one?

Your assignmnet Create new data frames from the "iris" data set that contain the following. Provide a copy of the code that you used to generate each new data frame and tell me how many rows are in each new data frame. If you get stuck, start with the first year guide! Then try the "help" files (the guide includes a section on getting help). *For each of these, tell me what the dimensions of each new data frame is.*

1. A data frame with only flowers with Sepal Length less than 4.5
2. A data frame with only flowers with Sepal Width greater than 3.3

3. A data frame with only the species "setosa" (Warning - the "species" field is factor data, which means you need to treat it a little differently! Use the help files, or Google it!)

4. **For graduate students (optional for keen undergraduates**): Create a data frame of all species "setosa" that have Petal.Width great than 0.3. *Tell me how big this data frame is.*

THIS IS WORTH 25% OF YOUR MARK FOR THIS MODULE.

Chapter 4

Plotting Data

The first year help file has a section on how to plot data when you have x and y vectors. Use this to make a plot of the iris data that shows Sepal Length vs. Sepal Width. I've provided code to create the x and y vectors below. Remember to add axis labels!

```
x <- iris$Sepal.Width y <- iris$Sepal.Length
```

Your assignment Modify the graph you created above to show the three species with different symbols. There are different ways you can do this (**HINT** look at the goldfish data example in the first year help manual). Alternatively, you can use the `pch` (for symbol style) and `bg` (for background colour) arguments to specify three colours for the species. Here's a snippet of the code that will assign the colours red, green and blue to the 3 factors in the data set: `pch = 21, bg = c("red", "green", "blue")` **HINT** insert this code into the `plot`function.

To make a legend use this code: `legend(2, 7.5, pch= 16, leg.text, col = c("red", "green", "blue"))`. Experiment with changing the variables to see what happens!

For graduate students (optional for keen undergraduates): Make a boxplot of sepal lengths by species. **HINT** you need to insert a formula of the form `y~grp` where `y` is a numerical vector of the data you want split into the boxplots and `grp` is the groupings you want to show on the x-axis (i.e., the factor data). Don't forget meaningful axis labels!

THIS IS WORTH 30% OF YOUR MARK FOR THIS MODULE.

Chapter 5

Fitting a model

Let's say you want to test whether there is a statical relationship between sepal length and sepal width in the "Versicolor" Species of iris. We suspect it is a linear fit, so we'll just fit a linear model of the form y = a(x) + b, where "y" is sepal length, x is sepal width, a is the coefficient for the slope of the line, and b is the intercept.

We'll use the glm (generalized linear model) not the ls (least squares) model, since it allows for better post-hoc model evalution. To do that we need to load and install the package "vegan".

Your assignment

1. Install the package "vegan". **HINT* Refer to the instructions on using packages in the first-year R guide.
2. Subset the data frame "iris" to create a new data frame that only has the "Versicolor" species (**HINT** you already did somehting like this, but for a different species in step 3 of the "Managing a data set" part of the assignment).
3. *Inspect new data frame and write down how many rows are in it.*
4. Fit a model by writing code that tells R to create an object called `Sepal.Model` using the function `glm`. Write the code as below, but where I have things inside <> replace with correct parameters from your new data frame. *Give a copy of the code when you hand in the work*

```
Sepal.Model <- glm(<y> ~ <x>, data = <new data frame)
```

If you execute this code and nothing happens, then it has worked. You can see a summary of the model fit with the command `summary(Sepal.Model)`

Is there a significant relationship between Sepal Length and Sepal width? What's the coefficeint?

It's always important to check that your model doesn't violate assmptions of normality. You can assess this by inspecting the ratio of the null deviance to residual deviance in the model summary.

You can also inspect the model by plotting the model fits. Do this with the code `plot(Speal.Model)`. You will see plots of 1) residuals vs. fitted data; 2) a Q-Q plot; 3) Scale-location plot; 4) Residulas vs. Leverage.

Click the links here to read about how to interpret the Residuals vs Fitted plot; the Q-Q plot. If your model does not meet the assumptions of normality, you may need to use a different error strucutre (e.g., Poisson, logNormal). Going into the whys and hows of that is beyond the scope of this course. The fit here is slightly skewed but the null and residual deviances are small enought that we can accept this fit.

Of course, you probably want to plot a line of best fit! To do that, we create a range of values for "x" (Sepal Width) that matches the range of the actual data. It's helpful to plot the data first.

So… make a scatter plot with "Sepal Width" on the x-axis and "Sepal Length" on the y-axis. **HINT** refer back to the "Plotting Data" part of the module.

Note, you could also query the data to see what the range of real x values are by typing `range(iris$Sepal.Width)`.

Then specify a range of values that fall within this for fitting the model. We'll create a sequence called "xvalues" that ranges between 2 and 5 in increments of 0.1 using this code `xvalues <- seq(2,5, 0.1)`. **aside** you've probably been ignoring the other windows in R studio, but notice how the "xvalues" popped up in the "Environment" window and shows you the length of the vector and the first values.

Then we use the `predict` function to create the model for these "xvalues" we just created, based on the "Sepal.model" you built with the glm.

```
yvalues <- predict(Sepal.Model, list(Sepal.Width = xvalues),
type="response")
```

Then simply add the line of fit to your plot with `lines(xvalues, yvalues)`.

THIS IS WORTH 35% OF YOUR MARK FOR THIS MODULE.

Chapter 6

Other sources of info

R "cheat sheets" can be really handy. See this link for an example cheat sheet for baseR. There are other cheat sheet for other R packages. For example, I wrote this document using Rmarkdown which has a cheat sheet here.

If you want more detailed help using R than what's in the first year guide provided as part of this module, check out "The Student's Guide to R" by Nochlas Horton, Randall Pruim and Daniel Kaplan. It's availalbe online here