

csci297b Exercise 4

Base R and lattice graphics

Part 1

1. Log on to the RStudio Workbench Server. In the same project as the previous exercise, open a new R script called `exercise_4`. Make sure you include any metadata you feel is appropriate (title, description of task, date of script creation etc). Don't forget to comment out your metadata with a `#` at the beginning of the line.
2. This exercise will deal with the squid dataset, found in the file `data/squid1.txt`. These data were originally collected as part of a study published in Aquatic Living Resources¹ in 2005. The aim of the study was to investigate the seasonal patterns of investment in somatic and reproductive tissues in the long finned squid *Loligo forbesi* caught in Scottish waters. Squid were caught monthly from December 1989 - July 1991 (month and year variables). After capture, each squid was given a unique specimen code, weighed (weight) and the sex determined (sex - only female squid are included here). The size of individuals was also measured as the dorsal mantle length (DML) and the mantle weight measured without internal organs (eviscerate.weight). The gonads were weighed (ovary.weight) along with the accessory reproductive organ (the nidamental gland, nid.weight, nid.length). Each individual was also assigned a categorical measure of maturity (maturity.stage, ranging from 1 to 5 with 1 = immature, 5 = mature). The digestive gland weight (dig.weight) was also recorded to assess nutritional status of the individual. If you're not familiar with squid morphology and are interested in finding out more see https://en.wikipedia.org/wiki/Cephalopod_size.
3. Load the squid data from the same data folder as the last exercise. `squid1.txt` is a tab-delimited file, so the easiest way to load it is

```
1 squid <- read.delim('data/squid1.txt')
```

Enter this in your script and source it to the console.

Use the `str()` function to display the structure of the dataset and the `summary()` function to summarise the dataset. How many observations are in this dataset? How many variables?

4. The year, month and maturity.stage variables were coded as integers in the original dataset. Here we would like to code them as factors. Recode each of them as factors. Month, for example, can be changed as follows:

```
1 squid$month <- factor(squid$month, labels=month.name)
```

The year and maturity.stage don't really need new names, so recode them as factors using the default labels.

Use the `str()` function again to check the coding of these new variables.

5. How many observations are there per month and year combination (hint: remember the `table()` or `xtabs()` functions)? Don't forget to use the factor recoded versions of these variables. Do you have data for each month in each year? Which years have the most observations? Use a combination of the `xtabs()` and `ftable()` functions to create a flattened table of the number of observations for each year, maturity stage and month (aka a contingency table).
6. The humble cleveland dotplot is a great way of identifying if you have potential outliers in continuous variables (See Section 4.2.4. Create dotplots (using the `dotchart()` function) for the following variables; DML, weight, nid.length and ovary.weight. Do these variables contain any unusually large or small observations?
7. It looks like the variable nid.length contains an unusually large value. Actually, this value is biologically implausible and clearly an error. The researchers were asked to go back and check their field notebooks and sure enough they discover a typo. It looks like a tired researcher accidentally inserted a zero by mistake when transcribing these data (mistakes in data are very common and why we always explore, check and validate any data we are working on). We can clearly see this value is over 400 so we can use the `which()` function to identify which observation this is (`which(squid$nid.length > 400)`). It looks like this is the 11th observation of the `squid$nid.length` variable. Use your skill with the square brackets `[]` to first confirm the this is the correct value (you should get 430.2) and then change this value in the dataset to 43.2. Now redo the dotchart to visualise your correction.

Caution: You can only do this because you have confirmed that this is an transcribing error. You should not remove or change values in your data just because you feel like it or they look 'unusual'. This is scientific fraud! Also, the advantage of making this change in your R script rather than in Excel is that you now have a permanent record of the change you made and can state a clear reason for the change.

8. When exploring your data it is often useful to visualise the distribution of continuous variables. Take a look at Section 4.2.2 and then create histograms for the variables; DML, weight, eviscerate.weight and ovary.weight.

One potential problem with histograms is that the distribution of data can look quite different depending on the number of 'breaks' used. The `hist()` function does it's best to create appropriate 'breaks' for your plots (it uses the Sturges algorithm for those that want to know) but experiment with changing the number of breaks for the DML variable to see how the shape of the distribution changes (see Section 4.2.2 of the book for further details of how to change the breaks).

9. Scatterplots are great for visualising relationships between two continuous variables (Section 4.2.1). Plot the relationship between DML on the x axis and weight on the y axis. How would you describe this relationship? Is it linear? One approach to linearising relationships is to apply a transformation on one or both variables. Try transforming the weight variable with either a natural log (`log()`) or square root (`sqrt()`) transformation. I suggest you create new variables in the squid dataframe for your transformed variables and use these variables when creating your new plots. Which transformation best linearises this relationship?

Part 2

10. When visualising differences in a continuous variable between levels of a factor (categorical variable) then a boxplot is your friend (avoid using bar plots - Google 'bar plots are evil' for more info). Create a boxplot to visualise the differences in DML at each maturity stage (don't forget to use the recoded version of this variable you created above) . Include x and y axes labels in your plot. Make sure you understand the anatomy of a boxplot before moving on - please ask if you're not sure (also see Section 4.2.3 of the book).
11. To visualise the relationship between two continuous variables but for different levels of a factor variable you can create a conditional scatterplot. Use the `coplot()` function (Section 4.2.6) to plot the relationship between DML and square root transformed weight (you created this variable above) for each level of maturity stage. Does the relationship between DML and weight look different for each maturity stage (suggesting an interaction)?
12. To explore the relationships between multiple continuous variables it's hard to beat a pairs plot. Create a pairs plot for the variables; DML, weight, eviscerate.weight, ovary.weight, nid.length, and nid.weight (see Section 4.2.5 of the book for more details). If it looks a little cramped in RStudio then click on the 'zoom' button in the plot viewer to see a larger version. One of the great things about the `pairs()` function is that you can customise what goes into each panel. Modify your pairs plot to include a histogram of the variables on the diagonal panel and include a correlation coefficient for each relationship on the upper triangle panels. Also include a smoother (wiggly line) in the lower triangle panels to help visualise these relationships. Take a look at the Introduction to R book to see how to do all this (or `?pairs`).
13. Almost every aspect of the figures you create in R is customisable. Learning how to get your plot looking just right is not only rewarding but also means that you will never have to include a plot in your paper/thesis that you're not completely happy with. When you start learning how to use R it can sometimes seem to take a lot of work to customise your plots. Don't worry, it gets easier with experience (most of the time anyway) and you will always have your code if y'ou want to create a similar plot in the future.

Use the `plot()` function to produce a scatterplot of DML on the x axis and ovary weight on the y axis (you might need to apply a transformation on the variable ovary.weight). Use a different colour to highlight points for each level of maturity stage. Produce a

legend explaining the different colours and place it in a suitable position on the plot. Format the graph further to make it suitable for inclusion into your paper/thesis (i.e. add axes labels, change the axes scales etc). See Section 4.3 for more details about customising plots.

14. Don't to forget to save your R script. Since we put this script in the same project as the last one, it should already be shared with your instructor.

Close your Project by selecting File→Close Project on the main menu.