

# **Exercises: R Notebooks**

## Licence

This manual is © 2020, Simon Andrews.

This manual is distributed under the creative commons Attribution-Non-Commercial-Share Alike 2.0 licence. This means that you are free:

- to copy, distribute, display, and perform the work
- to make derivative works

Under the following conditions:

- Attribution. You must give the original author credit.
- Non-Commercial. You may not use this work for commercial purposes.
- Share Alike. If you alter, transform, or build upon this work, you may distribute the resulting work only under a licence identical to this one.

Please note that:

- For any reuse or distribution, you must make clear to others the licence terms of this work.
- Any of these conditions can be waived if you get permission from the copyright holder.
- Nothing in this license impairs or restricts the author's moral rights.

Full details of this licence can be found at

<http://creativecommons.org/licenses/by-nc-sa/2.0/uk/legalcode>

## Exercise 1: Setting up a notebook

- Start a new RStudio project in the `Notebook_data` folder
- Create a new R Notebook and save it under the name "`Plant_uptake.Rmd`"
- Edit the default template so that you have
  - A title (set in the header) of "Plant CO2 uptake"
  - A small introductory piece of text saying what you're going to do
  - A code block which loads the `tidyverse` package
  - A code block which reads in the `plant_co2.csv` file, saves it to a variable and then displays it
  - A piece of text saying you're going to modify the `Treatment` variable
  - A block where you use `mutate` and `factor` to change the `Treatment` column to be a factor where "`nonchilled`" comes before "`chilled`"
  - A piece of text saying you're going to draw a graph
  - A scatterplot of `conc` vs `uptake` coloured by `Treatment`
- Knit the document to an HTML file and see how it looks.

### If you have time

- In the graph modify the point shape by `Type`
- Add a `geom_line` layer and add aesthetic mapping for `group=Plant` so that the lines join points from the same plant
- Modify the colours to use the colourbrewer `Set 1` palette using `scale_colour_brewer`

## Exercise 2: Using Markdown

- Add Level 1 titles saying "**Introduction**", "**Data Preparation**" and "**Visualisation**" at appropriate points in your document
- In the introduction say that this experiment was performed on grass plants of the *Echinochloa crus-galli* species. Make sure the latin name is written in italics.
- In the introduction add some text which describes the experimental factors in the design, these should be presented in a bulleted list. The name of the factor should be in bold. The factors are:
  - **Plant**: The physical plant from which measurements were made
  - **Type**: The species of the plant
  - **Treatment**: Whether the plant was chilled or not
- Also add a separate list for the measures. These are:
  - **conc**: The CO2 concentration
  - **uptake**: The amount of carbon taken up
- Recompile the document to see the changes in the final version

- Add a new graph. This should take only the data where `conc` is 250 or more and plot a strip chart of the `uptake` for nonchilled and chilled. You should use a facet to separate the data for the two plant Types (Mississippi and Quebec)
- Write some text saying what you conclude from the data presented in the stripchart

### If you have time

Try adding the details of the experimental factors as a table instead of a list.

Use `stat_summary` to add a mean line to the stripchart

### Exercise 3: Code Blocks

- Split up any existing code blocks which generate more than one piece of output
- Give all of your code blocks a name and check that you can see both the code block name and the document titles in the navigation section at the bottom of the document
- When printing your tibble use the head function to only show the first 10 lines so that your output file doesn't get too big.
- Remove any unwanted messages (eg `tidyverse` loading or `read_csv` structure information)
- Change the first graph to be printed in SVG (`dev="svg"`) format and have a height of 4 and a width of 9
- Keep the second graph in PNG format, but center it in the document (remember this won't affect the interactive notebook, only the compiled document)
- Add a legend to the figures
- Recompile the document to see the effect of the changes

### If you have time

Use `summarise` to calculate the `mean` and `sd` of the `uptake` for the different types and treatments for concentrations over 250. Print these results both as a table and a barchart. Suppress any unwanted warnings.

### Exercise 4: Document appearance

- Add a table of contents to the document and recompile to see how it works
- Change the theme used for both the main text and the syntax highlighting
- After recompiling the document find the HTML file created and open it directly in a browser

- Try compiling to other formats (Word or PDF) and see how they look. [Note: PDF creation currently fails for me if graphs use SVG format. This looks like a bug so it might be fixed when you try it, but if not then just change the graph type back to PNG and try again]

### **If you have time**

Try changing the appearance of the tibbles within the document and see what effect this has.

Try creating a parameter for the day the document was rendered and add that to the output.

Make a parameter for the file to process and then change this at runtime to automatically render the document.