Hands-on excercises - class 2

R Markdown

## R Markdown

This document will introduce you to R Markdown notebooks, and give you some hands-on experience adding to the document.

### Reference material

[R Markdown cheat sheet](https://rstudio.com/wp-content/uploads/2016/03/rmarkdown-cheatsheet-2.0.pdf)

Yihui Xie, J. J. Allaire, Garrett Grolemund, [*R Markdown: The Definitive Guide*](https://bookdown.org/yihui/rmarkdown/)

### Getting started

You can create a new R Markdown document in RStudio by clicking the menu item:

* File / New File > R Notebook

*or*

* File / New File > R Markdown…

For this course, we will be using the Document option…for your project, you may wish to use another format.

### Typing, running, and knitting

When you type in an R Markdown document, you just get what you type.

The magic starts to happen when you run R code chunks in the document…we will get to that soon.

And the real whiz-bang thing is that when you’ve finished, you can “knit” the document. At that point, the code chunks are run, and the markdown formatting is interpreted, and you get a beautiful document.

## 1. Markdown syntax

Markdown is a great way to write a report, slide deck, book, … even hands-on exercises for a class!

Formatting is simple. Check the cheatsheet for some details, or this section of [*R for Data Science*](https://r4ds.had.co.nz/r-markdown.html#text-formatting-with-markdown)

### Your Turn 1.1

Write something that you want bolded:

Now, make a bullet list with the names of three colours:

### Your Turn 1.2

1. Add a hypertext link to the text book’s title: Hadley Wickham and Garrett Grolemund, *R for Data Science*, so that it creates a link to the online version of the book at r4ds.had.co.nz .
2. Add an image of the data science process…the file is “data-science.png”

![The data science process]("data-science.png")

## 2. The first chunk: “setup”

It’s good practice to load all of the packages you’ll require at the top of your script. When a package is loaded, all of the functions and data in the package are available.

Loading an installed package is done with the library() function. The R chunk below, called “setup”, will load the {tidyverse} package (which in turn will load other packages, such as the data manipulation package {dplyr} and the plotting package {ggplot2}).

To run a chunk, press the green arrow at the far right of the chunk, or put your cursor in the chunk and press Ctrl + Shift + Enter (on macOS, Cmd + Shift + Enter)

### Your Turn 2.1

Add the package {cowsay} to the setup chunk, and run the chunk.

library(tidyverse)

## -- Attaching packages ----------------------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.2 v purrr 0.3.4  
## v tibble 3.0.3 v dplyr 1.0.2  
## v tidyr 1.1.2 v stringr 1.4.0  
## v readr 1.3.1 v forcats 0.5.0

## -- Conflicts -------------------------------------------------------- tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

# solution  
library(cowsay)

### Your Turn 2.2

1. Add a new chunk to the R Markdown document.

You can do this with the “Insert” toolbar button, or the keyboard shortcut

* Ctrl + Alt + I
* Cmd + Option + I on macOS

You could also type the three back ticks, the curly braces, the letter R, and three more backticks to end the chunk–but that’s a very WET approach!

1. Name the chunk so that you can navigate to it later.

### Your Turn 2.3

In the new chunk you have just created, type the function say() (which is from the {cowsay} package), inside the parentheses put a short quote (for example, “I like BIDA302”), and then run the chunk…

# solution  
  
say("I like BIDA302")

##   
## --------------   
## I like BIDA302   
## --------------  
## \  
## \  
## \  
## |\\_\_\_/|  
## ==) ^Y^ (==  
## \ ^ /  
## )=\*=(  
## / \  
## | |  
## /| | | |\  
## \| | |\_|/\  
## jgs //\_// \_\_\_/  
## \\_)  
##

### Getting help

Now, type ?say in the console to activate the Help menu for this function. Try some of the other arguments. (Note that an important by = option is not listed: “yoda”.)

Try experimenting with other arguments. For example, instead of your quote (which is the what = argument), you could put what = "time".

Now you have had a chance to experiement with that silliness…onward!

## 3. Chunk contents

Code chunks can create an object, which is stored in the Global Environment…it will be listed under the “Environment” tab in the top right of your RStudio screen.

That object can then be referenced in a future chunk and in inline text. This can be very useful!

### Your Turn 3.1

Use the seq() function to create an object “my\_numbers” that contains the integers from 3 to 7.

If you’re not sure about the syntax of the function, don’t hesitate to use the ?seq command in the console…

Now, use the next chunk to calculate the sum of the numbers in my\_list. Assign the sum to an object my\_total, and print the total.

### Your Turn 3.2

R code chunks can contain anything you want to do in R.

Here’s a chunk that contains code to create a chart from everyone’s favourite data set, mpg

Run the chunk and see what happens…

ggplot(data = mpg) +  
 geom\_point(mapping = aes(x = displ, y = hwy))



## 4. Inline R code

R Markdown allows you to insert a calculated value into the middle of your text. This is accomplished by putting a single backtick followed by the letter “r” (lower case, with no space after the backtick) and then your code, and ended with another backtick.

Here’s an example.

The area of a circle with a radius of 3 mm is 28.2743339 sq. mm.

When this is rendered (as we will later) this code chunk will be evaluated, and the result inserted into the text.

### Your Turn 4.1

Now is your chance to get fancy: write a sentence that uses inline code and the function sqrt to calculate the square root of my\_total, which you calculated before.

## 5. Rendering the HTML file

At this point, we want to knit it all together.

### Your Turn 5.1 - rendering the HTML file

Below is that chunk from above that creates a chart. When we are rendering the final document, we might not want to have the code displayed (it might be a fancy report for your boss’s boss)—we just want the chart.

The solution is to use one of many code chunk options that are available. These options control how the chunk is interpreted…you can leave the code visible but not run at all, or to omit rendering any messages that might clutter your output.

ggplot(data = mpg) +  
 geom\_point(mapping = aes(x = displ, y = hwy))



The solution is to put the option “echo = FALSE” after the “r” in the opening bit of the code. There are other options for the chunks that you may find useful.

## 6. The YAML

When we started our new R markdown file, there is a chunk of text at the top that we’ve ignored so far. This is the YAML (which now stands for [“YAML Ain’t Markup Language”](https://en.wikipedia.org/wiki/YAML#History_and_name).)

The YAML contents gives us the opportunity to control what happens when we knit our document.

For example, there are a variety of themes, which you can find listed at the [Bootstrap site](https://bootswatch.com/3/).

What happens when you replace the output: html\_document with the following?

output:  
 html\_document:  
 theme: united  
 highlight: tango

You can add a table of contents with

output:  
 html\_document:  
 toc: true

One of my favourites is the floating table of contents:

output:  
 html\_document:  
 toc: true  
 toc\_float: true

You can change the type of document being output. The current setting for this file is output: html\_document. But by changing it to output: word\_document, the result of knitting is a Word format document!

## REFERENCE MATERIAL

You can find more details in *R for Data Science*: <https://r4ds.had.co.nz/r-markdown.html>

You can find more code chunk options described here: <https://bookdown.org/yihui/rmarkdown/r-code.html>

There’s also a short listing of them in the R Markdown cheatsheet and reference guide, available online and through the “Help / Cheatsheets >” menu item in RStudio