ADS2 Practical 12: R Markdown

MI Stefan

Semester 1, 2019/20

Work through this guide alone or in groups. Facilitators are here to help. The time it takes to complete this practical can vary between individuals - this is OK. Do not worry if you do not finish within the session.

Learning Objective

- Use R Markdown to produce documents containing code and documentation

Installation instructions

- You may have to install the rmarkdown package first. The R command for doing this is

```
install.packages("rmarkdown")
```

- In order to generate pdf documents from R markdown, you need to have LaTeX installed on your computer. If you already have it, there is no need to do anything. If you don't, then install TinyTeX following the instructions here: https://bookdown.org/yihui/rmarkdown/installation.html (If you are not sure, you probably don't have it, so go ahead and install TinyTeX)

Getting started

- In Rstudio, create a new Rmarkdown file by going to File -> New File -> R Markdown You will be asked to provide a name and author for the file, as well as choosing an output format. Please choose PDF
- You can see that Rstudio creates a document for you that already has a lot of content. Save that document.
- When using markdown languages such as R markdown, input and output are separated. You write into an .Rmd file, but you get a .pdf file. The conversion between the two is called "knitting".
- Click on the "Knit" button. This will generate a pdf file. In this tutorial, you will learn to use R markdown by frequently comparing your .Rmd file with your .pdf file in order to see how they are related and how changes in input affect the output.

Document information

- Let's examine the .Rmd file.
- The first few lines contain information about the document, including names and output format. They will look something like this:

```
---
title: "R markdown guide"
author: "Melanie Stefan"
date: "14 March 2018"
output: pdf_document
---
```

This will not appear in the pdf file, but will be saved alongside it as "metadata". If you open the file properties in your pdf viewer, you will see this information.

- The next three lines specify a bit of r code that will set a few options for the knitr package that interprets R markdown. You can ignore or delete them.

Text editing in R markdown

- The text of your R markdown file starts with the following line:

```
## R Markdown
```

- By examining both the .Rmd and the .pdf file, answer the following questions:

1. How do you create a section headline?
2. How do you make text **bold face**?
3. How do you include a clickable link to a website?

- Try the following things and see what the outcome looks like in the pdf after knitting

1. Type the following:

```
I am the *best* student
```

2. Type the following:

```
- Prepare tutorial
- Go to class
- Revise lecture
- Finish ICA report
```

3. Type the following

```
1. Eat breakfast
2. Eat lunch
4. Eat dinner
2. Sleep
```

Do you notice something interesting?

Including code in R markdown

- The most interesting thing about R markdown is that you can include R code. This happens at several places in your .Rmd document. These sections start and end with three back ticks (`) This can be followed by a set of curly brackets specifying what type of code it is and including options. For instance, the following lines include some R code that defines two variables and computes their sum:

````
```{r}
x=2
y=3
```
````

```
x+y
```

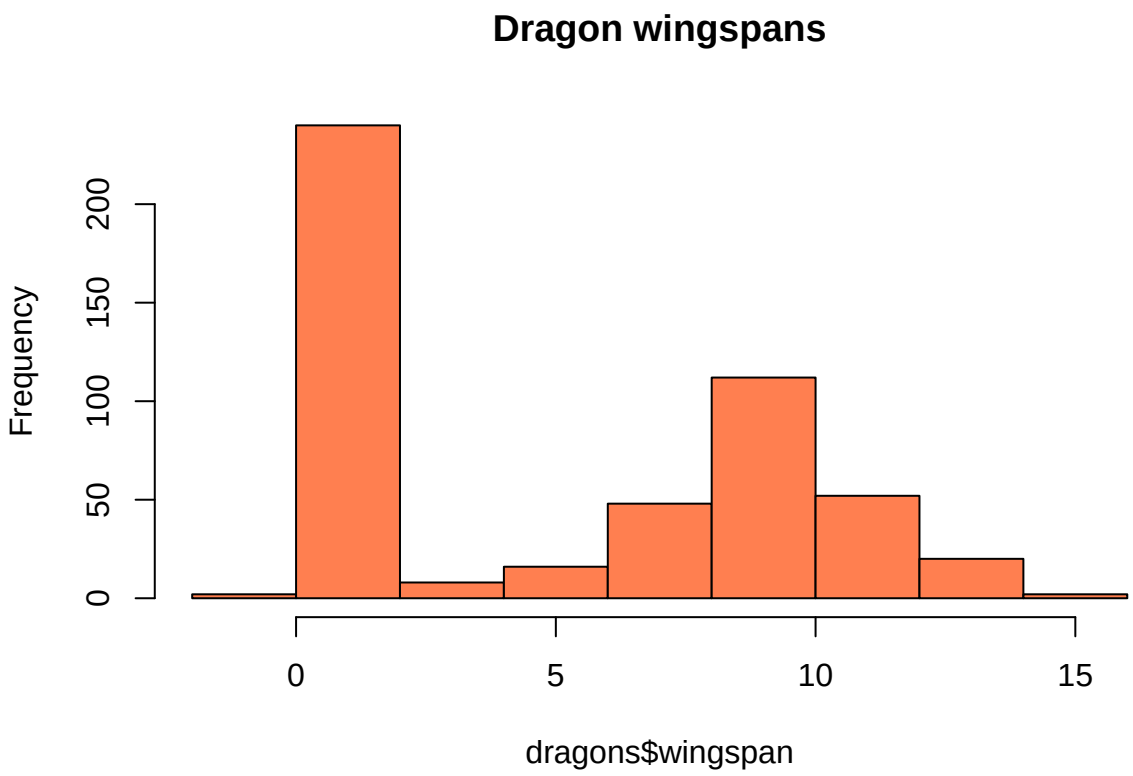
- If you do type that and examine the pdf file, you will see that it contains not only the r command, but also provides the output. How convenient!
- Sometimes, you might not want to see the output. In this case, add the option `eval=FALSE` like this:

```
```{r eval=FALSE}
2+2
```
```

- What is the difference between `eval=FALSE` and `echo=FALSE`?
- R markdown can be used to show output, not just if the output is a number, but also if it is a plot. Try this one, for instance:

```
```{r}
hist(rnorm(10000), col ="tomato")
```
```

- Those "chunks" of r code are not isolated from each other. You can refer back to them later. For instance, at any point to the document, you can recall the value of a variable you defined earlier. Try this:

```
```{r}
x
```
```

- You can also use small bits of R code in text. For instance, type the following:

We determined that the sum 1 and 2 was `` `r 1+2` `` surprisingly

Just like in a "normal" R document, you can also use R code chunks within an R markdown document to import data and then work with a dataframe. For instance, import data from dragons.csv (the same file we used in Practical 3), and plot a histogram of their wingspans



Tables and images

Tables

You may want some data presented in a table. This can be done using the `kable` function. For instance, if you want to show the first five rows of the dragon dataset, you would do this

```
library(knitr)
kable(dragons[1:5,])
```

x
1.34
0.67
0.80
0.66
1.29

(The `library(knitr)` command is only needed once, to tell R to use this particular library and the functions it provides.)

You can even do something more clever and plot the summary statistics for the dragon dataset in a table:

```
kable(summary(dragons))
```

wingspan
Min. :-0.140
1st Qu.: 1.058

wingspan
Median : 3.310
Mean : 5.075
3rd Qu.: 9.070
Max. :14.460

This is more interesting for datasets with more than one variable. For instance, see the file `mouse_report.csv`, which contains information about ID, meaurement date, age, genotype, and weight for a number of mice in a lab. Let's say you want to table ID, weight, and age for the first 20 mice, you can do this as follows:

```
mice <- read.csv("mouse_report.csv")
kable(mice[1:20,c(2,4,5)])
```

ID	weight	age
qg487	20	23
sw743	20	20
je649	21	21
pr476	16	13
nh236	20	16
nt852	15	11
ot185	18	29
mq231	20	7
pg148	21	15
zt398	16	14
nu634	20	38
lj546	17	25
jt739	20	12
af041	20	15
gk257	18	13
cj459	18	28
lu039	21	22
qn532	21	5
ey679	18	21
mp250	20	30

- How would you create a table that shows the output of the `summary()` function on the mice dataset?

Images

You may also want to include an image in your R markdown file. The syntax for this is quite easy. Start with an exclamation mark, then add the figure legend in square brackets and then a path to the file (with the correct extension) in round brackets:

`![Figure legend](path_to_file.png)`

The path can be to a local file on your computer (make sure you point to the correct directory!) or to an image file online. (Be careful with using links to images online if you are not the maintainer of the online resource - the image may be changed or taken down without you knowing.)

We provide an example image file (`haining_bridge_snow.png`) - see if you can include it in your .Rmd file (or use a picture of your own). You may notice that the location of the image in the .pdf file is not exactly where you put it in your .Rmd file - this is because during knitting, the overall aesthetics of the final document is taken into account and the ideal place for the figure is chosen accordingly.

Figure 1: ZJU International Campus bridge in the snow. By Melanie Stefan, 2018 (CC-BY-SA 4.0)

Little side note: With R Markdown files (as well a with any other work you produce), be careful about licensing and attribution if you use a picture that is not your own. A lot of images are copyrighted, and they cannot be reproduced without permission. Others are published under a *license* which explains the ways in which they can be used. (For instance CC-BY-SA is a Creative Commons license which specifies, essentially, that the piece of work under the license can be re-used, provided that the author is correctly attributed - you can think of this as similar to citing a scientific paper if you use a fact or idea from it.) We are not going deeper into this topic at the moment, but this seems like a good opportunity to make you aware of the issue.

Exploring further

If you want to explore further, read through Prof. Cozma Shalizi's R markdown introduction: http://www.stat.cmu.edu/~cshalizi/rmarkdown/ This document goes into more depth, and discusses some additional advanced topics. Try out some of the commands.