Reproducilble Research

Malay (mbasu@kumc.edu)

November 9, 2023

Contents

1 What is reproducible research

In science replicability is of prime importance. There is a difference between “replication” and “reproduction”. Replication of an experiments is to repeat the same experiment to see whether the result holds up. There is a assumption that the results will differ next time. The results of experimental replication are of two types: biological and technical replicates. Unlike replication, reproduction of and experment is to repeat is exactly to check whether another groups’ results holds up under scrutiny. This is of vital importance.

In bioinformatics, reproducible research is a concept where data and code are packaged together into one live document. It provides the facility to reproduce the data analysis step exactly on another computer.

2 Text processing vs word processing

In word processing (like in Microsoft Word), the environment is WYSIWYG (What You See Is What You Get). You exactly see the formatting of the document live. In text processing, you think about the overall structure of the document and `markup` the document is special syntax then use a software to convert the document in more presentation format, like PDF. The prime software for text processing in scientific computing is LaTeX.

3 LaTeX

LaTeX has been a mainstay of scientific publishing. A very simple document is LaTeX is:

```
    \documentclass{article}
    \title{My first document}
    \author{Malay}
    \begin{document}
    \maketitle
    \section{This is the first section}
    Hello World!
    \end{document}
```

3.1 Exercise

Type the above code block in a text file. Name the file as `hello.tex`. You can compile the document in LaTeX using

```
xelatex hello.tex
```

Note: You might have to install xelatex and the assoicated packages.

4 `Pandoc`

LaTeX is a type-setting software and no one should use it directly. `Pandoc` (http://johnmacfarlane.net/pandoc/) is a highly useful text conversion software that can convert any form of text to another. Additionally, it supports a variant of a very popular and simple text markup language called `markdown`. When you convert a `markdown` formatted file to PDF using `pandoc`, it uses LaTeX behind the scene. This makes our job is crating LaTeX a lot easier.

Let's create our first `pandoc` file.

```
---
title: My first document
author: Malay
---
# This is the first section
Hello World!
```

Type the above code in a file called `hello1.md`. Then type the following:

```
pandoc --number-sections -o hello1.pdf hello1.md
```

Interesting thing is that the same source can be now used to create an HTML document:

```
pandoc -s --number-sections -o hello1.html hello1.md
```

Also a Microsof Word file:

```
pandoc -o hello1.docx hello1.md
```

5 Literate programming

1. Originally created by Don Knuth.
2. Document that describes the data analysis and the code are in the same text file.
3. You **weave** a document to create the human readable text.
4. You **tangle** a document ot create the machine readable code.

6 knitr

`knitr` is a package in `R` to mix `markdown` formatted document with `R` code called, `Rmarkdown`(http://rmarkdown.rstudio.com/). The `R` code is live and executed on the fly to generate the final document with the result of the `R` already embedded in. The overall steps are:

```
.Rmd -> (use knitr)-> .md -> (use pandoc)-> HTML or PDF
```

Within `RStudio` all these steps are put together under one button click.

6.1 Exercise

Type the following code into `RStudio`:

~~~
---
title: My first knitr document
author: Me
---

This is some text (text chunk).

Here is some code (code chunk).

```{r}
set.seed(1)
x<-rnorm(100)
mean(x)
```
~~~

6.2 Chunk options

For a list of chunk options see http://yihui.name/knitr/options#chunk_options. There are several options you can use to control what should be included in the final document. For example, `echo=FALSE` with not include the code in the final document.

~~~
```{r echo=FALSE}
set.seed(1)
x<-rnorm(100)
mean(x)
```
~~~

`results="hide"` will hide the output but the code will still be shown.

~~~
```{r results="hide"}
set.seed(1)
x<-rnorm(100)
mean(x)
```
~~~

`include=FALSE` will suppress both the output and the code, but the code will will be executed.

~~~
```{r include=FALSE}
set.seed(1)
x<-rnorm(100)
mean(x)
```
~~~

```
head(x)
```

```
## [1] -0.6264538  0.1836433 -0.8356286  1.5952808  0.3295078 -0.8204684
```

To prevent the execution of the code, use `eval=FALSE`.

6.3 Global chunk option

You can set a chunk option globally for every chunk.

~~~
```{r include=FALSE}
knitr::opts_chunk$set(message=FALSE)
```
~~~

One important option for chunk is caching R calculations.

~~~
```{r include=FALSE}
knitr::opts_chunk$set(cache=TRUE)
```
~~~

6.4 In-line R

You can include a piece of data from the calculation inside a text chunk.

```
We have `r length(x)` rows in our data.
```

6.5 Nice tables

You can use R package `xtable` for creating nice looking table.

~~~
```{r fitmodel}
    library(datasets)
    data(airquality)
    fit <- lm(Ozone ~ Wind, data=airquality)
```
~~~

We can now print a nice looking table of regression coefficients.

```
library(xtable)
options(xtable.comment=FALSE)
xt <- xtable( summary(fit) )
print.xtable(xt, type="latex")
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	96.8729	7.2387	13.38	0.0000
Wind	-5.5509	0.6904	-8.04	0.0000

You can also use the `knitr` function `kable` to create a table.

```
knitr::kable(summary(airquality))
```

	Ozone	Solar.R	Wind	Temp	Month	Day
	Min. : 1.00	Min. : 7.0	Min. : 1.700	Min. :56.00	Min. :5.000	Min. : 1.0
	1st Qu.: 18.00	1st Qu.:115.8	1st Qu.: 7.400	1st Qu.:72.00	1st Qu.:6.000	1st Qu.: 8.0
	Median : 31.50	Median :205.0	Median : 9.700	Median :79.00	Median :7.000	Median :16.0
	Mean : 42.13	Mean :185.9	Mean : 9.958	Mean :77.88	Mean :6.993	Mean :15.8
	3rd Qu.: 63.25	3rd Qu.:258.8	3rd Qu.:11.500	3rd Qu.:85.00	3rd Qu.:8.000	3rd Qu.:23.0
	Max. :168.00	Max. :334.0	Max. :20.700	Max. :97.00	Max. :9.000	Max. :31.0
	NA's :37	NA's :7	NA	NA	NA	NA

7 Git and github

`Git` is a version control system, created by Linus Tovalds. Github is website for storing and sharing `git` repositories.

7.1 Demo how to create github account and create a repository

Check the Jupyter notebook.