

Applied Data Analysis School

LITERATE PROGRAMMING IN R MARKDOWN

Paulo Guimaraes and Miguel Portela

October 27 and 29, 2020

Setup and references

1 Pandoc

2 Latex

- You can install a lightweight Latex flavor: [TinyTex](#)

```
install.packages('tinytex')  
tinytex::install_tinytex()
```

- or a complete distribution: [MikTex](#) or [MacTex](#)

3 Sumatra PDF or Skim

4 Markdown references

- [Markdown Guide](#)
- [R Markdown - Get Started](#)
- [R Rarkdown: The definitive guide](#)

By the end of the day

you can write a report

- Claxton, G., Rae, M., Long, M., Damico, A., Foster, G. and Whitmore, H., 2017. [2017 Employer Health Benefits Survey](#). *Kaiser Family Foundation and Health Research & Educational Trust*.

a paper

- Hartgerink, C. H. J., Wicherts, J. M., & van Assen, M. A. L. M. (2017). [Too Good to be False: Nonsignificant Results Revisited](#). *Collabra: Psychology*, 3(1), 9.

or edit a book

- Xie, Y., Allaire, J.J. and Grolemund, G., 2018. [R Markdown: The Definitive Guide](#). CRC Press.

- What is **Literate programming**?

Literate programming refers to melding a descriptive narrative and computer code into a single document, from which both human-friendly documentation and computer readable files can be created

- Your work should be transparent, easy to update, easy to maintain, and easy to replicate
- Data analysis requires reproducibility of results
- Useful for teaching

Preliminaries: Markdown

- **Markdown** is a lightweight markup language with plain text formatting syntax that was invented by John Gruber
- There are several “flavors” of Markdown
- Markdown files can be created in any text editor
- and can be easily converted to other formats
- They are text files usually with the extension “.md”
- There are many editors specific for Markdown (eg: **Typora**, **Mou**, **Draft**, etc) but you can use general editors (eg: **Atom**, **Sublime Text** or **RStudio**)
- Used in popular sites such as **GitHub**, **Reddit**, **Stack Exchange**, etc

Markdown components

- 1 Metadata: written between a pair of three dashes ---
- 2 Text: regular text and math elements, plus tables, images and figures
- 3 Code:
 - inline R code: starts with ``r`` and ends with a backtick ```
 - code chunk: starts with three backticks like ````{r}` where `r` indicates the language name and ends with three backticks `````

Markdown Syntax: Quick Introduction

- Creating Headers

```
This is header 1  
=====
```

or

```
Another header  
-----
```

- or headers with different levels

```
# This is Header 1  
...some text...  
## And now Header 2  
...more text...  
### Finally Header 3  
... and a little more text.
```

Markdown Syntax (continued)

- Emphasizing text

this is **Italic** but `_this_` also works

this will write in ****bold**** but you can also `__do this__`

It is also possible to ~~~~Strikeout~~~~ text

- Creating unordered lists (use "*" "+" or "-")

* Point 1

+ Point 2

Markdown Syntax (continued)

- You can create structured lists

- Point 1
 - Point 12
- Point 2
 - Point 21

- insert links to text. Simply write

`[text](url)`

- insert images

`![text](file "Description")`

- You can add tables
- and even references

- You can find a quick guide for syntax [here](#) or more detailed information [here](#)
- To get an idea how Markdown works you can use an online editor such as [Dillinger](#) or [Markdown here](#)
- and if you need to create tables you can use [this](#) great online tool

Markdown Syntax (continued)

- Markdown documents are highly flexible because they can incorporate other languages
- An example is *LaTeX*. To add *LaTeX* code you need to enclose it in “\$” If you write

`$a^2+b^2=c^2$`

you will see $a^2 + b^2 = c^2$

- Markdown also accepts raw HTML code
- You can incorporate code from other languages (eg: Python, Java, R, Stata, etc)
- This is typically done in a “code fence”: lines with three or more backticks or tildes inserted before and after

Preliminaries: Pandoc

- **Pandoc** is a command line tool to convert across different document formats. Try it online [here](#)
- It supports many formats such as Markdown, HTML, docx, pdf, *LaTeX*, etc
- It is simple to use: to convert the file *example1.md* to

- HTML

```
pandoc -s example1.md -o example1.html
```

- PDF

```
pandoc -s example1.md -o example1.pdf
```

- docx

```
pandoc -s example1.md -o example1.pdf
```

- Note: -s stands for source and -o for output

Metadata

- You can add a title, author and date as metadata. Simply start the document as

```
% Fill in title  
% Fill in Name  
% 23 January 2019
```

- Or you can use the **YAML** format. See **Pandoc User's Guide** for more info. To add title, author and date place at the top of the document

```
---  
title: Fill in title  
author: Fill in name  
date: 23 January 2019  
---
```

Adding citations to the document

- it is possible to add citations from a **bibtex** file
- place the “bib” file (say “references.bib”) in the working folder
- modify the YAML by adding a line pointing to the bib file

```
bibliography: references.bib
```

- At the bottom of the document add a line

```
## References
```

- use the syntax [**@key**] to identify the reference

Advanced use: customizing outputs

- The default style for citations is the Chicago Manual of Style author-date format.
- But you can use any style available in Citation Style Language (CSL) in the [Zotero Style Repository](#).
- To change style download the “csl” file and add a reference to it in the YAML block. for example

```
---  
title: Title of my presentation  
author: My name  
date: 23 January 2019  
bibliography: references.bib  
csl: thisstyle.csl  
---
```

Further information

Random text generator

- [Lorem ipsum](#)