

Reproducible papers in the life sciences using R

Ariel Mundo Ortiz

Université de Montréal



@amundortiz



@aimundo

CANSSI Statistical Software Conference
November 10, 2022

Introduction

- RMarkdown is a powerful tool to create reproducible papers

Introduction

- RMarkdown is a powerful tool to create reproducible papers
- However, R is rarely used in the life sciences as a default method to create papers

Introduction

- RMarkdown is a powerful tool to create reproducible papers
- However, R is rarely used in the life sciences as a default method to create papers
- Why?

Reasons

- “R is just for Stats”

Reasons

- “R is just for Stats”
- “There is a learning curve”

Reasons

- “R is just for Stats”
- “There is a learning curve”
- **“I can’t create figures for publication”**

The Problem

- Papers in the life sciences usually require figures where the following are combined:

The Problem

- Papers in the life sciences usually require figures where the following are combined:
 - Images from cells/tissues

The Problem

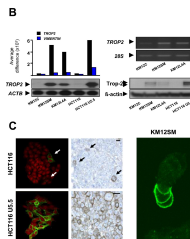
- Papers in the life sciences usually require figures where the following are combined:
 - Images from cells/tissues
 - Figures that summarize data

The Problem

- Papers in the life sciences usually require figures where the following are combined:
 - Images from cells/tissues
 - Figures that summarize data
 - Figures that present statistical analyses (with “p-values”)

The Problem

Figure 1: A typical figure



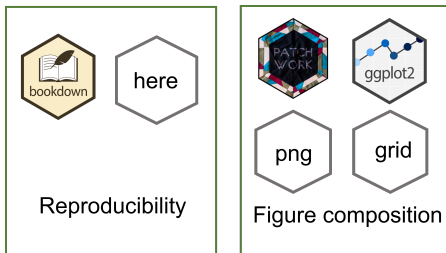
Guerra et al. *Neoplasia*
2021

- Can figures like this be created using 'RMarkdown'?

The Solution

- Yes, we can create figures like this using R!

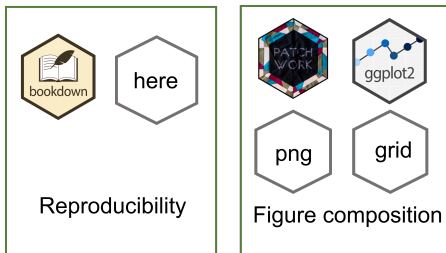
Figure 2: The packages



The Solution

- Yes, we can create figures like this using R!
- We will need to use a combination of packages to achieve this

Figure 2: The packages



The Solution

- **bookdown** allows to create a reproducible paper:

The Solution

- **bookdown** allows to create a reproducible paper:
 - Each section of the paper: (Materials and Methods, Results, etc.) is in a separate Rmd file

The Solution

- **bookdown** allows to create a reproducible paper:
 - Each section of the paper: (Materials and Methods, Results, etc.) is in a separate Rmd file
 - More details can be found in <https://bookdown.org/>

The Solution

- **bookdown** allows to create a reproducible paper:
 - Each section of the paper: (Materials and Methods, Results, etc.) is in a separate `Rmd` file
 - More details can be found in <https://bookdown.org/>
- **here** allows to easily call scripts within the document (we will look at this later)

The Solution

- Let's think about a typical scenario, where you:

The Solution

- Let's think about a typical scenario, where you:
 - Have written your paper sections (Methods, Results, etc) each section is in a `Rmd` file

The Solution

- Let's think about a typical scenario, where you:
 - Have written your paper sections (Methods, Results, etc) each section is in a `Rmd` file
 - Have some images

The Solution

- Let's think about a typical scenario, where you:
 - Have written your paper sections (Methods, Results, etc) each section is in a `Rmd` file
 - Have some images
 - Have some data that needs to be analyzed

The Solution

- Let's think about a typical scenario, where you:
 - Have written your paper sections (Methods, Results, etc) each section is in a `Rmd` file
 - Have some images
 - Have some data that needs to be analyzed
 - Want to create a composite figure of images/data analysis

The Solution

- Let's think about a typical scenario, where you:
 - Have written your paper sections (Methods, Results, etc) each section is in a Rmd file
 - Have some images
 - Have some data that needs to be analyzed
 - Want to create a composite figure of images/data analysis
 - **For the sake of time, I will focus on the image composition/data analysis part**

Project Organization

■ How to organize the files:

```
... {% raw %}
```

```
Project
```

```
├── Data
```

```
│   ├── csv files
```

```
├── Code
```

```
│   ├── R Script(s)
```

```
├── Figures
```

```
│   ├── PNGs (or other image files)
```

```
├── Manuscript
```

```
│   ├── Rmd files (Sections)
```

```
{% endraw %}
```

A Handy Script: Images

- We can read the images (in PNG format), get the data, do the analysis and create the figure in a single script!

A Handy Script: Images

- We can read the images (in PNG format), get the data, do the analysis and create the figure in a single script!
 - Reading the images is achieved by `grid` and `png`

A Handy Script: Images

- We can read the images (in PNG format), get the data, do the analysis and create the figure in a single script!
 - Reading the images is achieved by `grid` and `png`
 - `ggplot2` creates the plot of our analysis

A Handy Script: Images

- We can read the images (in PNG format), get the data, do the analysis and create the figure in a single script!
 - Reading the images is achieved by `grid` and `png`
 - `ggplot2` creates the plot of our analysis
 - `patchwork` allows us to assemble everything

A Handy Script: Images

- How do we read a PNG image?

```
cells<-rasterGrob(readPNG(here("figures",  
                             "cells.png"),  
                    native=TRUE))
```

A Handy Script: Images

- How do we read a PNG image?

```
cells<-rasterGrob(readPNG(here("figures",  
                              "cells.png"),  
                    native=TRUE))
```

- We use **here** to call the PNG file located in the “figures” directory.

A Handy Script: Data Analysis

```
# for regression

formula<-y~x

# create a plot of the data and the regression
a1<-ggplot(data=data,
           aes(x=weight,y=body_fat,fill=Group,color=Group))
  geom_point(show.legend=FALSE,shape=21,colour='black',size=5,
            alpha=0.7)+
  geom_smooth(method="lm",formula=formula, se=T)+
  stat_poly_eq(use_label(c("R2","p.value")),
              formula = formula, size = 3)
```

- Try **ggpmisc**

A Handy Script: Layout

- Provide a layout for the figure <https://patchwork.data-imaginist.com/articles/guides/layout.html>

A Handy Script: Layout

- Provide a layout for the figure <https://patchwork.data-imaginist.com/articles/guides/layout.html>

A Handy Script: Layout

- Provide a layout for the figure <https://patchwork.data-imaginist.com/articles/guides/layout.html>

```
layout<-"  
AAAABBBB  
AAAABBBB  
AAAABBBB  
AAAABBBB  
CCCCDDDD  
CCCCDDDD  
CCCCEEEE  
CCCCEEEE  
"
```

A Handy Script: Assemble!

- Use `wrap_elements` and `patchwork`

```
image_a<-wrap_elements(  
  panel=cells  
) +  
  wrap_elements(  
    panel=molecule  
  ) +  
  wrap_elements(  
    panel=jellyfish  
  ) +  
  ylab("jellyfish") +  
  a1 +  
  a2 +  
  plot_layout(design=layout)
```

Conclusion

- We can use R to create reproducible papers and complex figures

Conclusion

- We can use R to create reproducible papers and complex figures
- There **is** a learning curve, but once you learn you won't go back to W**d!

Acknowledgments

- Nasri Lab @Université de Montréal



Acknowledgments

- Nasri Lab @Université de Montréal
- Centre de recherches mathématiques (CRM)



Acknowledgments

- Nasri Lab @Université de Montréal
- Centre de recherches mathématiques (CRM)
- CANSSI

