

RMarkdown Basics

Recap

- Reproducibility across all sciences is a **huge** issue!
- We need to be able to reproduce scientific findings
 - This means the data & code from a given paper must be easy to access and **human readable**
- When it comes to programming in **R**, using **RMarkdown** can help us make sure that other humans can understand our analyses



Anatomy of RNotebook

The anatomy of all `.Rmd` files (RNotebook or RMarkdown):

1. Document Information

2. Formatted Text

3. Code Chunks

Anatomy of RNotebook

The anatomy of all `.Rmd` files (RNotebook or RMarkdown):

1. Document Information

2. Formatted Text

3. Code Chunks

Document Information

- Title
- Subtitle
- Date
- Author
- Output type

```
---  
title: "Making Pretty Code"  
subtitle: "with RMarkdown"  
author: "Shelly Cooper"  
output: html_document  
---
```

This is sometimes called a YAML header.

Document Information

- Title
- Subtitle
- Date
- Author
- Output type

```
---
title: "Making Pretty Code"
author: "Shelly Cooper"
subtitle: with RMarkdown
output:
  pdf_document: default
  html_document: default
---
```

IMPORTANT: Do NOT change any of the tabs/indents in the YAML header. If you copy something from the internet, pay attention to this!

Anatomy of RNotebook

The anatomy of all `.Rmd` files (RNotebook or RMarkdown):

1. Document Information

2. Formatted Text

3. Code Chunks

Formatted Text

You need to tell R how you want your text to be formatted:

- Headers
- Bolded text
- Italicized text
- Hyperlinks
- Bullet/numbered lists

Headers

The number of # indicates what size and level your header should be.

Header 1

Header 1

Header 2

Header 2

Header 3

Header 3

Header 4

Header 4

Header 5

Header 5

Headers

Headers

Bold & Italics

Bold text

- ****bold text ****
- **__bold text __** (2 underscores)

Italicized text

- ** italicized text **
- *_ italicized text _*

To ***combine***, pick ***** for one and **_** for the other

- ****_combine_****
- **__**combine**__**
- ***__combine__***
- **__*combine*__**

Hyperlinks

- The word(s) you want to be the link go inside square brackets []
- Immediately after, it's a pair of parentheses () that contains the actual link.

[Google](www.google.com) is my friend!

Google is my friend!

Bullet Lists - Unordered

- First line must end with a `:` (colon)
- Must have an empty line
- Must have a space after the bullet

Unformatted

Brazilian States:

- Rio Grande do Sul
- Parana
- Rio de Janeiro

Formatted

Brazilian States:

- Rio Grande do Sul
- Parana
- Rio de Janeiro

Bullets can be `-` (dashes), `+` (plus), or `*` (asterisk), but all come out looking like what you see here.

Bullet Lists - Ordered

Same thing, but now with numbers

Unformatted

Brazilian States:

1. Rio Grande do Sul
2. Parana
3. Rio de Janeiro

Formatted

Brazilian States:

1. Rio Grande do Sul
2. Parana
3. Rio de Janeiro

Bullet Lists - Nested

You can have organized, nested lists. Go to the next line, and press **2 spaces**. Then put your new bullet symbol.

- Do NOT press `tab`. For whatever reason, R doesn't like it for Markdown.
- If you still are stuck, try 4 spaces -- that should work

Unformatted

Brazilian States & Capitals:

1. Rio Grande do Sul
 - * Porto Alegre
 - This is the best!
2. Parana
 - * Curitiba
3. Rio de Janeiro
 - * Rio de Janeiro

Formatted

Brazilian States & Capitals:

1. Rio Grande do Sul
 - Porto Alegre
 - This is the best!
2. Parana
 - Curitiba
3. Rio de Janeiro
 - Rio de Janeiro

Anatomy of RNotebook

The anatomy of all `.Rmd` files (RNotebook or RMarkdown):

1. Document Information

2. Formatted Text

3. Code Chunks

Code Chunks

- This is what makes RMarkdown so cool!
- Type your code directly into a code chunk and work with it just like you would a `.R` script file
- When you're done, click `knit` at the top to generate your pretty report
 - All code chunks will be executed (unless you say otherwise...see next lecture)

Code Chunks

```
```${r}  
code goes here
answer <- 1 + 2
```
```

Each code chunk:

- **Starts and ends** with three backticks
 - if you don't have the ending 3, you're gonna have a bad time
- Has `{r}` at top next to the first 3 backticks
- Has gray background
- Looks like normal code
- Runs like normal code

Making Code Chunks

To make a code chunk:

- Use the `insert` button
- Manually type the backticks & `{r}`
- Keyboard shortcuts
 - PCs: `ctrl + alt + i`
 - Macs: `cmd + opt + i`



A finished product

Step 1: Data Preparation

Let's first get our data ready for an analysis

```
``{r}
# load the packages
library(knitr)
library(psych)
library(ggplot2)

# set the working directory
setwd("~/Desktop/rSkillLab/")

# import data
midus <- read.csv("midus.csv")
````
```

## # Step 2: Analysis

Now let's get the mean of the age variable.

```
``{r}
mean_age <- mean(midus$age, na.rm = TRUE)
````
```

Step 1: Data Preparation

Let's first get our data ready for an analysis

```
# load the packages
library(knitr)
library(psych)
library(ggplot2)
```

```
##
## Attaching package: 'ggplot2'
```

```
## The following objects are masked from 'package:psych':
##
##      %+%, alpha
```

```
# set the working directory
setwd("~/Desktop/rSkillLab/")

# import data
midus <- read.csv("midus.csv")
```

Step 2: Analysis

Now let's get the mean of the age variable.

```
mean_age <- mean(midus$age, na.rm = TRUE)
```

Code Chunks

All code will run in a code chunk

- It's very literal!
- We can't see `midus` or `mean_age`
 - These are stored as objects
 - If you want to see them, you need to tell R to show them to you
- We needed to import `midus`
 - If it was in your Environment *before* knitting the file AND you did *not* import your data, it would fail
 - Whatever data you use, you need to import it!

Viewing data.frames

To help make data.frames readable for humans, use the `kable()` function

- comes from the `knitr` package, although surprisingly, you don't need to manually load this one

1. Data Preparation

Let's first get our data ready for an analysis

```
# load the packages
library(knitr)
library(psych)
library(ggplot2)

# set working directory
setwd("~/Box Sync/Brazil 2019/")

# import data
midus <- read.csv("midus.csv")

# view the first 6 rows of the midus dataset
# head(midus) gets the first 6 rows
# kable() makes the output pretty
kable(x = head(midus), caption = "Midus Dataset")
```

Midus Dataset

| ID | sex | age | BMI | physical_health_self | mental_health_self | self_esteem | life_satisfaction | hostility | heart_self | heart_father |
|-------|--------|-----|--------|----------------------|--------------------|-------------|-------------------|-----------|------------|--------------|
| 10001 | Male | 61 | 26.263 | 2 | 4 | 42 | 7.750 | 5.5 | No | No |
| 10002 | Male | 69 | 24.077 | 5 | 5 | 34 | 8.250 | 6.0 | No | Yes |
| 10005 | Female | 80 | NA | 4 | 4 | 49 | 9.333 | 4.0 | No | No |
| 10006 | Female | 60 | NA | 3 | 3 | NA | NA | NA | No | Yes |
| 10010 | Male | 55 | NA | 4 | 3 | 28 | 8.250 | 8.0 | No | Yes |
| 10011 | Female | 52 | 25.991 | 5 | 4 | 41 | 7.000 | 5.5 | No | No |

Next up

Changing parameters in the YAML header and in code chunks to make the output just right.

