

Programming Toolkit for Data Science

Programming Toolkit: R

Course toolkit

Course operation

- Moodle
- Slack
- Zoom

Doing data science

- Programming:
 - R
 - RStudio
 - tidyverse
 - R Markdown
- Version control and collaboration:
 - Git
 - GitHub

Learning goals

By the end of the course, you will be able to...

- gain insight from data
- reproducibly **and collaboratively**,
- using modern programming tools and techniques

The following content is based on Mine Çetinkaya-Rundel's excellent book Data Science in a Box

Reproducible data analysis

Reproducibility checklist

What does it mean for a data analysis to be "reproducible"?

Near-term goals:

- Are the tables and figures reproducible from the code and data?
- Does the code actually do what you think it does?
- In addition to what was done, is it clear *why* it was done?

Long-term goals:

- Can the code be used for other data?
- Can you extend the code to do other things?

Toolkit for reproducibility

- Scriptability → R
- Literate programming (code, narrative, output in one place) → R Markdown
- Version control → Git / GitHub

R and RStudio

R and RStudio



- R is an open-source statistical **programming language**
- R is also an environment for statistical computing and graphics
- It's easily extensible with *packages*



- RStudio is a convenient interface for R called an **IDE** (integrated development environment), e.g. *"I write R code in the RStudio IDE"*
- RStudio is not a requirement for programming with R, but it's very commonly used by R programmers and data scientists

R packages

- **Packages** are the fundamental units of reproducible R code. They include reusable R functions, the documentation that describes how to use them, and sample data¹
- As of September 2020, there are over 16,000 R packages available on **CRAN** (the Comprehensive R Archive Network)²
- We're going to work with a small (but important) subset of these!

¹ Wickham and Bryan, [R Packages](#).

² [CRAN contributed packages](#).

Tour: R and RStudio

data viewer

| | species | island | bill_length_mm | bill_depth_mm | flipper_length_mm | body_mass_g |
|----|---------|-----------|----------------|---------------|-------------------|-------------|
| 1 | Adelie | Torgersen | 39.1 | 18.7 | 181 | |
| 2 | Adelie | Torgersen | 39.5 | 17.4 | 186 | |
| 3 | Adelie | Torgersen | 40.3 | 18.0 | 195 | |
| 4 | Adelie | Torgersen | NA | NA | NA | |
| 5 | Adelie | Torgersen | 36.7 | 19.3 | 193 | |
| 6 | Adelie | Torgersen | 39.3 | 20.6 | 190 | |
| 7 | Adelie | Torgersen | 38.9 | 17.8 | 181 | |
| 8 | Adelie | Torgersen | 39.2 | 19.6 | 195 | |
| 9 | Adelie | Torgersen | 34.1 | 18.1 | 193 | |
| 10 | Adelie | Torgersen | 42.0 | 20.2 | 190 | |
| 11 | Adelie | Torgersen | 37.8 | 17.1 | 186 | |

Showing 1 to 11 of 344 entries, 7 total columns

environment

help

Arithmetic Mean

Description

Generic function for the (trimmed) arithmetic mean.

Usage

```
mean(x, ...)
```

Default S3 method:
mean(x, trim = 0, na.rm = FALSE, ...)

Arguments

x: An R object. Currently there are methods for numeric/logical vectors and

Examples

```
x <- c(0:10, 50)
xm <- mean(x)
c(xm, mean(x, trim = 0.10))
```

[Package base version 4.0.2 [Index](#)]

arithmetic

load package

view data

get help

Object assignment

access variable

use function

```
> 2 + 2
[1] 4
> x <- 2
> x * 3
[1] 6
> library(palmerpenguins)
> View(penguins)
> penguins$flipper_length_mm
[1] 181 186 195 NA 193 190 181 195 193 190 186 180 182 191
[337] 206 189 195 207 202 193 210 198
> mean(penguins$flipper_length_mm)
[1] NA
> ?mean
> mean(penguins$flipper_length_mm, na.rm = TRUE)
[1] 200.9152
```

A short list (for now) of R essentials

- Functions are (most often) verbs, followed by what they will be applied to in parentheses:

```
do_this(to_this)  
do_that(to_this, to_that, with_those)
```

- Packages are installed with the `install.packages` function and loaded with the `library` function, once per session:

```
install.packages("package_name")  
library(package_name)
```

R essentials (continued)

- Columns (variables) in data frames are accessed with \$:

```
dataframe$var_name
```

- Object documentation can be accessed with ?

```
?mean
```

tidyverse



tidyverse.org

- The **tidyverse** is an opinionated collection of R packages designed for data science
- All packages share an underlying philosophy and a common grammar

rmarkdown

rmarkdown.rstudio.com

- **rmarkdown** and the various packages that support it enable R users to write their code and prose in reproducible computational documents
- We will generally refer to R Markdown documents (with `.Rmd` extension), e.g. *"Do this in your R Markdown document"* and rarely discuss loading the rmarkdown package



R Markdown

- Fully reproducible reports -- each time you knit the analysis is ran from the beginning
- Simple markdown syntax for text
- Code goes in chunks, defined by three backticks, narrative goes outside of chunks

Tour: R Markdown

The image shows the RStudio interface with an R Markdown document named `bechdel.Rmd` open. The document is divided into two main sections: a source editor on the left and a viewer on the right.

Source Editor (Left):

- Knit:** A yellow arrow points to the `Knit` button in the top toolbar.
- YAML:** A red bracket labeled `yaml` highlights the YAML frontmatter at the top of the document, which includes the title, author, and output options.
- Link:** A green arrow labeled `link` points to a URL in the text of the document.
- Code chunk:** A pink bracket labeled `code chunk` highlights a block of R code used to load packages.

Viewer (Right):

- Bechdel:** The rendered title of the document.
- Mine Çetinkaya-Rundel:** The rendered author name.
- Text:** The rendered text of the document, including a paragraph about the FiveThirtyEight story and a link to the story.
- Data and packages:** The rendered title of the section.
- Text:** The rendered text of the section, including a paragraph about loading packages.
- Code chunk:** The rendered R code chunk, showing the `library` function calls.
- Text:** The rendered text of the section, including a paragraph about the dataset and a paragraph about the financial variables.
- List:** A list of financial variables, including `budget_2013`, `domgross_2013`, and `intgross_2013`.

Environments

The environment of your R Markdown document is separate from the Console!

Remember this, and expect it to bite you a few times as you're learning to work with R Markdown!

Environments

First, run the following in the console

```
x <- 2  
x * 3
```

All looks good, eh?

Then, add the following in an R chunk in your R Markdown document and knit it.

```
x * 3
```

What happens? Why the error?

R Markdown help

R Markdown Cheat Sheet
Help -> Cheatsheets

R Markdown :: CHEAT SHEET

What is R Markdown?

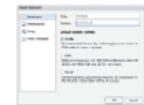


.Rmd files - An R Markdown (.Rmd) file is a record of your research. It contains the code that a scientist needs to reproduce your work along with the narration that a reader needs to understand your work.

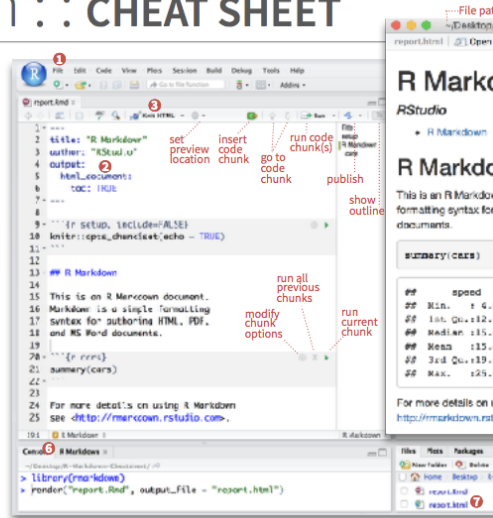
Reproducible Research • At the click of a button, or the type of a command, you can rerun the code in an R Markdown file to reproduce your work and export the results as a finished report.

Dynamic Documents • You can choose to export the finished report in a variety of formats, including html, pdf, MS Word, or RTF documents; html or pdf based slides, Notebooks, and more.

Workflow



- 1 **Open a new .Rmd file** at File ► New File ► R Markdown. Use the wizard that opens to pre-populate the file with a template
- 2 **Write document** by editing template
- 3 **Knit document to create report**; use knit button or `render()` to knit
- 4 **Preview Output** in IDE window
- 5 **Publish** (optional) to web server
- 6 **Examine build log** in R Markdown console
- 7 **Use output file** that is saved along side .Rmd



render

Use `rmarkdown::render()` to render/knit at cmd line. Important args:

| | | | |
|-------------------------------|---|--------------------|--|
| input - file to render | output_options - List of render options (as in YAML) | output_file | params - list of params to use |
| output_format | | output_dir | |

Embed code with knitr syntax

INLINE CODE

Insert with ``r <code>``. Results appear as text without code.

Built with 'r getRversion()' ➡ Built with 3.2.3

CODE CHUNKS

One or more lines surrounded with `'''{r}'''` and `'''`. Place chunk

options within curly braces, after `r`. Insert with 

```
## {r echo=TRUE}
```

```
getRversion()
'''
```

GLOBAL OP'1

Set with knitr::opts

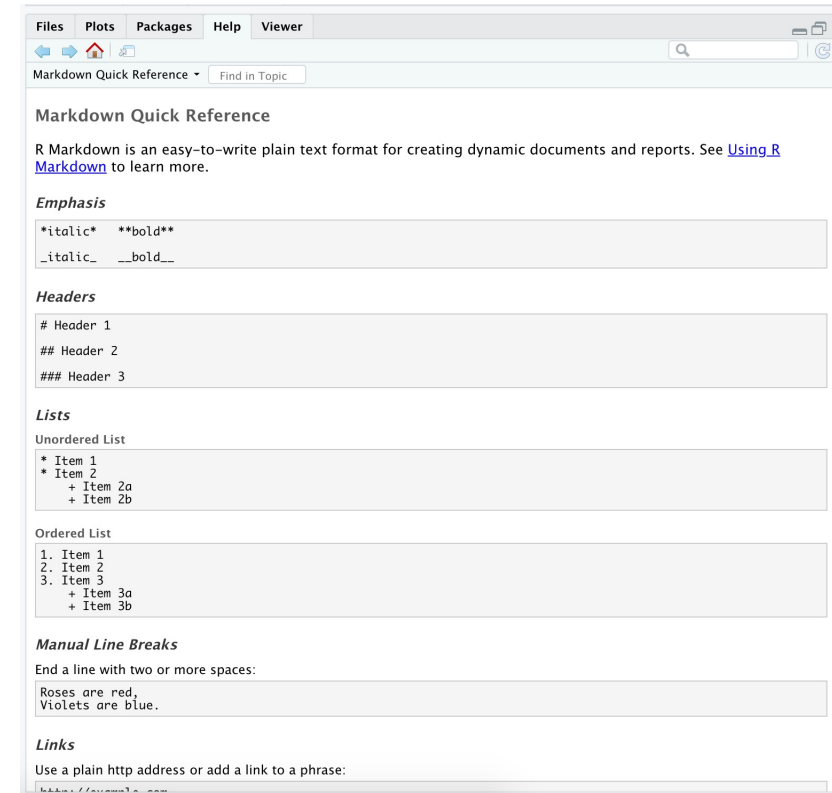
```
```{r include=FA}
```

```
knitr::opts_chunk$set(
```

...

# Markdown Quick Reference

## Help -> Markdown Quick Reference



# How will we use R Markdown?

- Every assignment / report / project / etc. is an R Markdown document
- You'll always have a template R Markdown document to start with

## *Your turn:* AE 02 – Bechdel + R Markdown

- [The Bechdel test](#) asks whether a work of fiction features at least two women who talk to each other about something other than a man, and there must be two women named characters.
- Go to [RStudio Cloud](#) and start the assignment AE 02 – Bechdel + R Markdown.
- Open and knit the R Markdown document `bechdel.Rmd`, review the document, and fill in the blanks.