

Meet the toolkit

Tidyverse & Co.

Filippo Chiarello, Ph.D.

Course toolkit

Doing data science

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

Learning goals

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

- gain insight from data
- gain insight from data, **reproducibly**
- gain insight from data, reproducibly, **using modern programming tools and techniques**
- gain insight from data, reproducibly **and collaboratively**, using modern programming tools and techniques
- gain insight from data, reproducibly and collaboratively, using modern programming tools and techniques, **to answer to key intelligence topics**

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

arithmetic



load package

view data

get help

The screenshot shows the RStudio interface with the following panels and annotations:

- Environment Panel:** Shows the variable `x` with value `2`. An annotation "environment" points to this panel.
- Help Panel:** Displays the documentation for the `mean` function. An annotation "help" points to the "Description" section.
- Console Panel:** Contains the following R code and output:

```
> 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
```

 - An annotation "object assignment" points to the line `x <- 2`.
 - An annotation "access variable" points to the line `penguins$flipper_length_mm`.
 - An annotation "use function" points to the line `mean(penguins$flipper_length_mm)`.
- Data Viewer Panel:** Displays a table of penguin data with columns: species, island, bill_length_mm, bill_depth_mm, flipper_length_mm, and body_mass_g. An annotation "data viewer" points to this panel.

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