

Meet the toolkit:  
programming

Data Science in a Box  
[datasciencebox.org](http://datasciencebox.org)



# Course toolkit

## Course operation

- Course website
- Moodle

## Doing data science

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



# Learning goals

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



# Learning goals

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

- gain insight from data



# Learning goals

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

- gain insight from data
- gain insight from data, **reproducibly**



# Learning goals

By the end of the course, 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**



# Learning goals

By the end of the course, 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



# Learning goals

By the end of the course, 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 **(with literate programming and version control)** and collaboratively, using modern programming tools and techniques



# Reproducible data analysis



# Reproducibility checklist

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



# 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 data<sup>1</sup>
- As of September 2022, there are over 16,000 R packages available on **CRAN** (the Comprehensive R Archive Network)<sup>2</sup>
- We're going to work with a small (but important) subset of these!

<sup>1</sup> Wickham and Bryan, R Packages.

<sup>2</sup> CRAN contributed packages.



# Tour: R and RStudio

The screenshot illustrates a workflow for data analysis in RStudio:

- data viewer**: A data frame titled "penguins" is displayed in the Data Viewer pane, showing columns for species, island, bill length, bill depth, flipper length, and body mass.
- arithmetic**: Basic arithmetic operations like addition and multiplication are shown in the Console.
- load package**: The palmerpenguins package is loaded using the command `> library(palmerpenguins)`.
- view data**: The penguins dataset is viewed using `> View(penguins)`.
- get help**: Help for the mean function is accessed using `> ?mean`.
- Object assignment**: An object named "x" is assigned the value 2 using `> x <- 2`.
- access variable**: The flipper length variable is accessed using `penguins$flipper_length_mm`.
- use function**: The mean function is used to calculate the mean flipper length, with and without NA values removed.
- environment**: The variable "x" is listed in the Global Environment pane with a value of 2.
- R Documentation**: The help page for the `mean` function is displayed, providing details on its usage, arguments, and examples.



# 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)
```



# 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
```



# R essentials (continued)

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

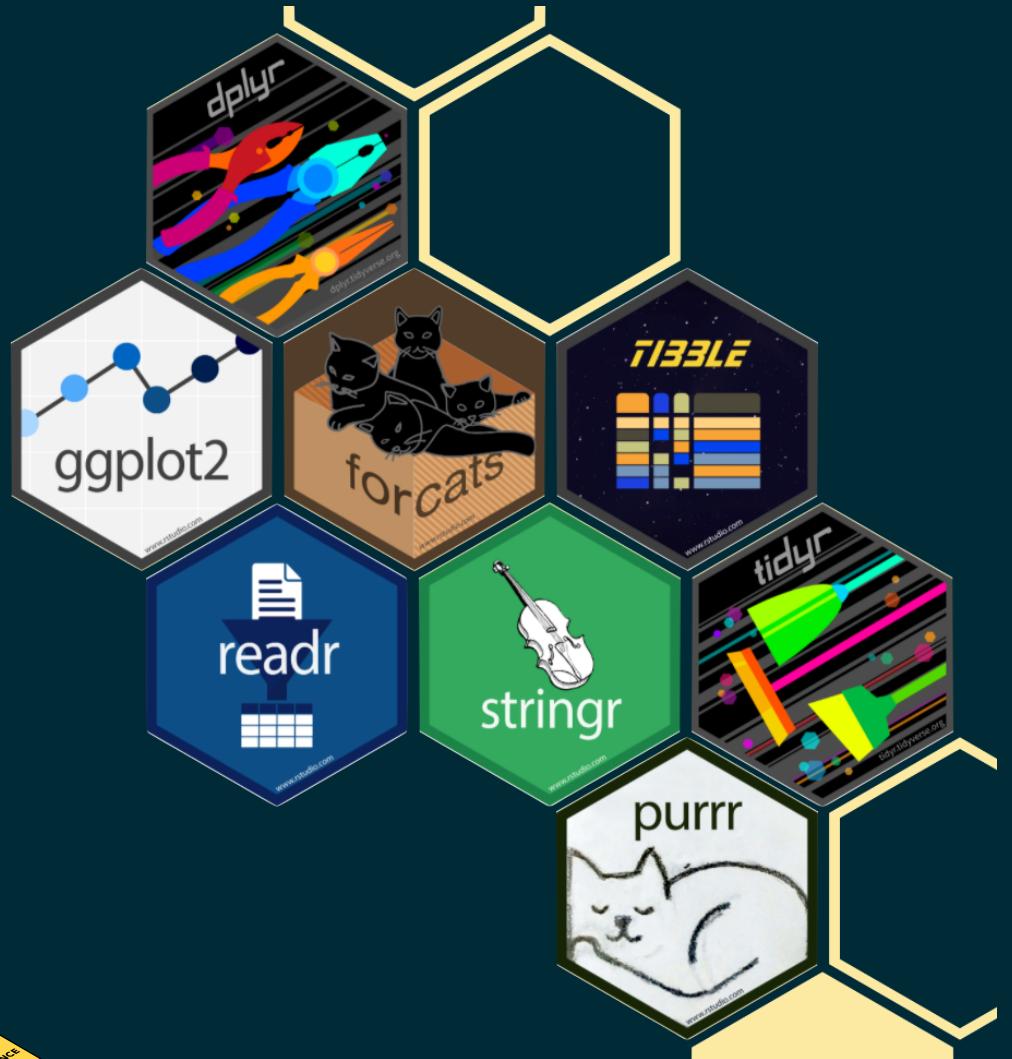
```
dataframe$var_name
```

- Object documentation can be accessed with ?

```
?mean
```



# tidyverse



[tidyverse.org](https://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](https://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



# 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 screenshot shows the RStudio interface with an R Markdown file open in the left pane and its rendered output in the right pane.

**Annotations:**

- knit:** A yellow arrow points to the "Knit" button in the RStudio toolbar.
- link:** A green arrow points to the URL link in the rendered output: <https://fivethirtyeight.com/features/the-dollar-and-cents-case-against-hollywoods-exclusion-of-women/>.
- code chunk:** A pink bracket highlights the code chunk starting at line 16: ````{r load-packages, message=FALSE}```
- yaml:** A red bracket highlights the YAML front matter at the top of the file.

**R Markdown File Content (bechdel.Rmd):**

```
1 ---  
2 title: "Bechdel"  
3 author: "Mine Çetinkaya-Rundel"  
4 output:  
5   html_document:  
6     fig_height: 4  
7     fig_width: 9  
8 ---  
9  
10 In this mini analysis we work with the data used  
11 in the FiveThirtyEight story titled ["The  
12 Dollar-And-Cents Case Against Hollywood's  
13 Exclusion of Women"](https://fivethirtyeight.com/features/the-dollar-and-cents-case-against-hollywoods-exclusion-of-women/). Your task is to fill in  
14 the blanks denoted by `___.`.  
15  
16 ```{r load-packages, message=FALSE}  
17 library(fivethirtyeight)  
18 library(tidyverse)  
19 ...
```

**Viewer Pane Content:**

## Bechdel

Mine Çetinkaya-Rundel

In this mini analysis we work with the data used in the FiveThirtyEight story titled ["The Dollar-And-Cents Case Against Hollywood's Exclusion of Women"](#). Your task is to fill in the blanks denoted by \_\_\_\_.

## Data and packages

We start with loading the packages we'll use.

```
library(fivethirtyeight)  
library(tidyverse)
```

The dataset contains information on 1794 movies released between 1970 and 2013. However we'll focus our analysis on movies released between 1990 and 2013.

```
bechdel90_13 <- bechdel %>%  
  filter(between(year, 1990, 2013))
```

There are \_\_\_\_ such movies.

The financial variables we'll focus on are the following:

- budget\_2013 : Budget in 2013 inflation adjusted dollars
- domgross\_2013 : Domestic gross (US) in 2013 inflation adjusted dollars
- intgross\_2013 : Total International (i.e. worldwide) gross in 2013 inflation

# 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?



# 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

```
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

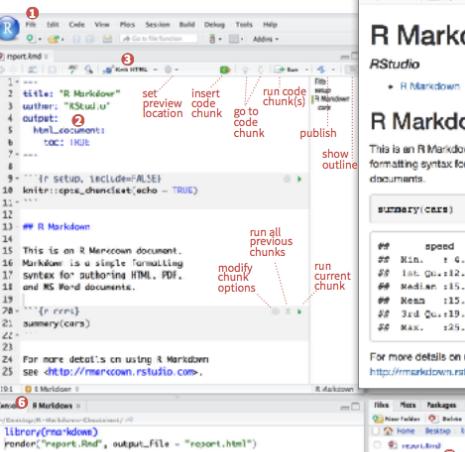
- Open a new .Rmd file at File > New File > R Markdown. Use the wizard that opens to pre-populate the file with a template
- Write document by editing template
- Knit document to create report; use knit button or render() to knit
- Preview Output in IDE window
- Publish (optional) to web server
- Examine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

### Embed code with knitr syntax

INLINE CODE  
Insert with `r code`. Results appear as text without code.  
Built with `r getRVersion()`

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 OPTS  
Set with knitr opts\_chunk  
`{{` (r include=FALSE)  
`knitr::opts\_chunk\$set(...)`  
`}}



Markdown Quick Reference  
Help -> Markdown Quick Reference

### Markdown Quick Reference

R Markdown is an easy-to-write plain text format for creating dynamic documents and reports. See [Using.R Markdown](#) to learn more.

#### Emphasis

\*italic\* \*\*bold\*\*  
italic bold

#### Headers

# Header 1  
## Header 2  
### Header 3

#### Lists

Unordered List  
\* Item 1  
\* Item 2  
+ Item 2a  
+ Item 2b

Ordered List  
1. Item 1  
2. Item 2  
3. Item 3  
+ Item 3a  
+ Item 3b

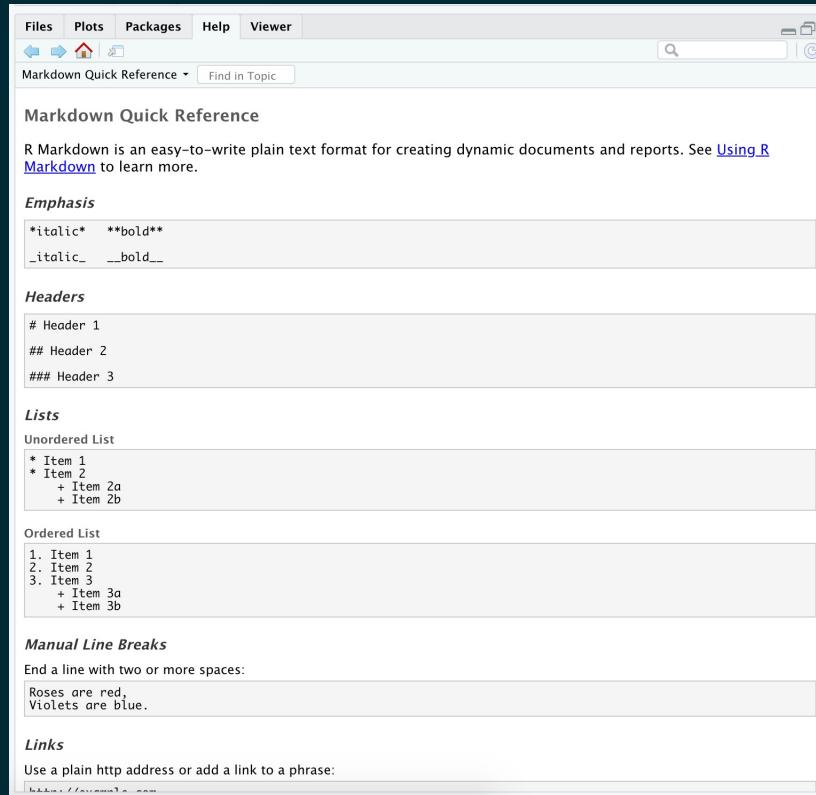
#### Manual Line Breaks

End a line with two or more spaces:  
Roses are red,  
Violets are blue.

#### Links

Use a plain http address or add a link to a phrase:

http://datasciencebox.org



# 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
- The amount of scaffolding in the template will decrease over the semester



# What's with all the hexes?



Mitchell O'Hara-Wild, useR! 2018 feature wall

