

4.1 Introduction to R: Hello World!

Julia Gallucci

Data Science Institute, University of Toronto

2023-12-11

Acknowledgements

Slides are adapted from Anjali Silva, originally from Amy Farrow under the supervision of Rohan Alexander, University of Toronto. Slides have been modified by Julia Gallucci, 2023.

Outline of Topics

Getting Set up

- ▶ R
- ▶ RStudio

R basics

File types

- ▶ Scripts
- ▶ R markdown

Hello, world!

Outline of Topics

Getting Set up

- ▶ R
- ▶ RStudio

R basics

File types

- ▶ Scripts
- ▶ R markdown

Hello, world!

Getting set up with R/RStudio

Steps for R

1. Go to <https://cloud.r-project.org/>
2. Click Download R for (your operating system).
3. For Windows, select install R for the first time. For Mac and Linux, select the download that is appropriate for your OS.
4. Download and install

Getting set up with R/RStudio

Steps for RStudio

1. Go to <https://posit.co/download/rstudio-desktop/>
2. Under Products, look under Open Source and select *RStudio.
3. Scroll down and select RStudio Desktop and DOWNLOAD RSTUDIO DESKTOP.
4. Select the DOWNLOAD button under the Free version of RStudio Desktop.
5. If the download that is “Recommended for your system” is correct, click the download button. If not, scroll down and find the version that is correct for your OS.
6. Download and install.
7. Open and test to ensure RStudio is working.

Questions?

RStudio

The screenshot displays the RStudio IDE interface. At the top is the menu bar with options: File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. Below the menu is a toolbar with various icons for file operations like Open, Save, and Print. A search bar labeled "Go to file/function" is followed by a dropdown for "Addins". The main workspace consists of several panes:

- Code Editor:** An untitled R script titled "Untitled1" is open, showing a single line of code: "1".
- Environment:** This pane shows the global environment, which is currently empty.
- Console:** The R console output is displayed, starting with the R startup message and information about the platform (R 4.1.1, x86_64-w64-mingw32/x64 (64-bit)). It also includes standard R help text and a prompt for quitting.
- Help:** A sidebar on the right provides links to R resources, RStudio support, and manuals.

Help Sidebar Content:

- R Resources:** Learning R Online, CRAN Task Views, R on StackOverflow, Getting Help with R.
- RStudio:** RStudio IDE Support, RStudio Community Forum, RStudio Cheat Sheets, RStudio Tip of the Day, RStudio Packages, RStudio Products.
- Manuals:** An Introduction to R, The R Language Definition.

Components of RStudio

The screenshot shows the RStudio IDE interface. The left pane, which is highlighted with a red border, contains the R script editor with the following text:

Your saved R files, where you can write and edit code, are in the upper left. If you do not have a file open, this section will be collapsed

Below the script editor is the R console output:

```
R 4.1.1 : ~ / 
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
```

The right pane displays the environment browser, showing the Global Environment tab with the message "Environment is empty". Below the environment browser is a navigation bar with tabs: Files, Plots, Packages, Help, and Viewer. The Help tab is currently selected. On the far right, there are two columns of links:

R Resources	RStudio
Learning R Online	RStudio IDE Support
CRAN Task Views	RStudio Community Forum
R on StackOverflow	RStudio Cheat Sheets
Getting Help with R	RStudio Tip of the Day
	RStudio Packages
	RStudio Products

Below these resources is a "Manuals" section with links to "An Introduction to R" and "The R Language Definition".

Components of RStudio

The screenshot shows the RStudio interface with several components highlighted:

- Console (Lower Left):** A red box highlights the "Console" tab in the bottom-left corner. It displays the standard R startup message and help text.
- Environment (Top Right):** The "Environment" tab is selected. It shows the "Global Environment" pane which is currently empty, indicated by the message "Environment is empty".
- Help (Bottom Right):** The "Help" tab is selected in the bottom-right corner. It provides links to various RStudio resources and documentation, such as "R Resources", "RStudio IDE Support", and "Manuals".

The main workspace area contains the following text from the R startup message:

```
R 4.1.1 · ~/Documents/RStudio  
Platform: x86_64-w64-mingw32/x64 (64-bit)  
  
R is free software and comes with ABSOLUTELY NO WARRANTY.  
You are welcome to redistribute it under certain conditions.  
Type 'license()' or 'licence()' for distribution details.  
  
R is a collaborative project with many contributors.  
Type 'contributors()' for more information and  
'citation()' on how to cite R or R packages in publications.  
  
Type 'demo()' for some demos, 'help()' for on-line help, or  
'help.start()' for an HTML browser interface to help.  
Type 'q()' to quit R.  
  
> |
```

Components of RStudio

The environment, where you can see variables that you have saved, is on the upper right.

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Untitled1 Go to file/function Addins
Source on Save Run Source
1 |

The environment, where you can see variables that you have saved, is on the upper right.

Environment History Connections Tutorial
Import Dataset 160 MiB List
R Global Environment
Environment is empty

Files Plots Packages Help Viewer Refresh Help Topic Home Find in Topic

1:1 (Top Level) R Script
Console Terminal Jobs
R 4.1.1 - /
Platform: x86_64-w64-mingw32/x64 (64-bit)
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
> |

R Resources RStudio
Learning R Online RStudio IDE Support
CRAN Task Views RStudio Community Forum
R on StackOverflow RStudio Cheat Sheets
Getting Help with R RStudio Tip of the Day
RStudio Packages RStudio Products

Manuals
An Introduction to R The R Language Definition

Components of RStudio

The lower right section contains:

- *Files*, where you see the folder structure of your working directory
- *Plots*, where plots you create are displayed
- *Packages*, an inventory of all R packages that are installed
- *Help*, where you can search information about R functions and packages

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function Addins Project: (None)
Untitled1 Source on Save Run Source Environment History Connections Tutorial Import Dataset 160 MB List Refresh Help Topic Home Find in Topic

1 | The lower right section contains:
• Files, where you see the folder structure of your working directory
• Plots, where plots you create are displayed
• Packages, an inventory of all R packages that are installed
• Help, where you can search information about R functions and packages

1:1 (Top Level) R Script
Console Terminal Jobs
R 4.1.1 · ~/
Platform: x86_64-w64-mingw32/x64 (64-bit)
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

Files Plots Packages Help Viewer Refresh Help Topic Home Find in Topic

R Resources RStudio
Learning R Online RStudio IDE Support
CRAN Task Views RStudio Community Forum
R on StackOverflow RStudio Cheat Sheets
Getting Help with R RStudio Tip of the Day
RStudio Packages RStudio Products

Manuals
An Introduction to R The R Language Definition

Interacting with R

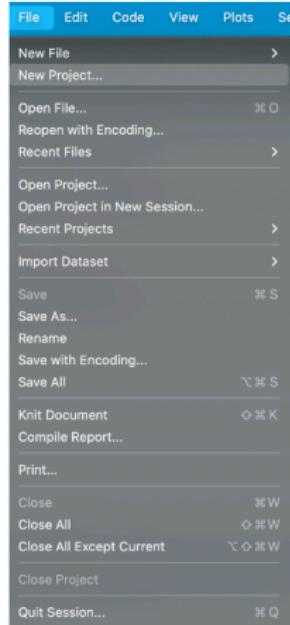
Console

- ▶ Type commands directly into the console and press ‘Enter’ to execute
- ▶ Clear console with ‘Ctrl’ + ‘L’
- ▶ If R is still waiting for you to enter more text, the console will show a +prompt.

Script

- ▶ Put cursor at the end of the line to execute OR highlight the section.
- ▶ Press ‘Ctrl’ + ‘Enter’ on Windows, Mac OR ‘Cmd’ + ‘Return’ on Mac.

R Project



Good to keep data, analyses and text in a single folder

RStudio interface for this is Projects.

- ▶ File → New project; choose New Directory → New project
Enter a name for this folder (“directory”) and choose a convenient location for it. This will be your working directory (save as “DSI IntroR” on Desktop)
- ▶ Click on “Create” project, create a new file where we will type our scripts
- ▶ Go to file → New File → R script. Click the save icon on your toolbar and save your script as “script.R”

Outline of Topics

Getting Set up

- ▶ R
- ▶ RStudio

R basics

File types

- ▶ Scripts
- ▶ R markdown

Hello, world!

Location

Current location of the file, if saved:

```
getwd()
```

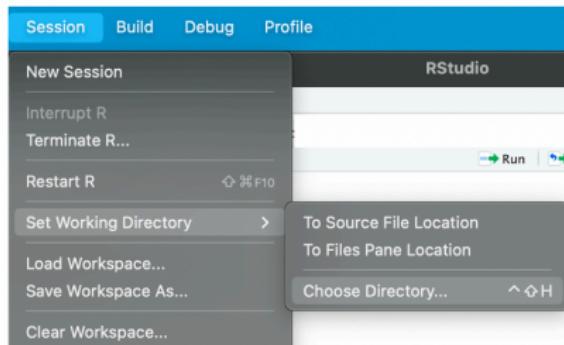
```
[1] "/cloud/project"
```

Set working directory by typing path:

```
#setwd("/Users/<Name>/Desktop")
```

Or (*recommended method*):

Session → Set Working Directory → Choose Directory...



Coding style

Limit yourself to 80 characters per line.

Use comments. Don't describe what the code does, but explain why you wrote it that way.

Use only `←` for assignment, not `=`

Never reassign reserved words/built in functions (i.e., `mean`)

Optional: You may read more

- ▶ <https://google.github.io/styleguide/Rguide.html>
- ▶ http://steipe.biochemistry.utoronto.ca/abc/index.php/RPR-Coding_style

Features of R

In R, the indexing begins from 1.

```
set_of_numbers <- c(1:10)  
set_of_numbers
```

```
[1] 1 2 3 4 5 6 7 8 9 10
```

```
set_of_numbers[1] #index 1
```

```
[1] 1
```

R is case sensitive ("X" is not the same as "x").

```
x <- 1  
X
```

```
Error in eval(expr, envir, enclos): object 'X' not found
```

Mathematical operations in R

For now we will work in the console.

To run code, hit enter.

If it runs successfully, you will see a > on the line with the cursor.

If it instead shows a +, the command was incomplete. You can finish the command and hit enter, or hit ESCAPE to start again.

```
(27 +0.4 - 2 * 7 / 11) ^ 3
```

```
[1] 17835.37
```

Basic Math Operators	Operation
x + y	Addition
x - y	Subtraction
x * y	Multiplication
x / y	Division
x ^ y	Exponent
x %% y	Modulus

Creating objects in R

<- is the assignment operator.

To assign the value 27 to the object named num_participants, you type:

```
num_participants <- 27
```

Rules for object names:

1. Must start with a letter
2. Can only contain letters, numbers, underscores, and periods
3. Typical style conventions; camelCase, snake_case

Running the name of the object will display the object.

```
num_participants <- 27  
num_participants
```

```
[1] 27
```

Built-in R Functions

R has a rich set of functions that can be used to perform almost every task for the user

Built-in functions perform many operations. They take the form:

```
function_name(argument1 = value1, argument2 = value2, ...)
```

```
sqrt(16)
```

```
[1] 4
```

```
seq(1,14)
```

```
[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14
```

Help in R

Online documentation for functions and variables in R exists.

Obtained by typing `help(function_name)` or `? function_name` at the Rprompt, where `function_name` refers to the name of function.

```
help(sqrt)
```

```
?sqrt
```



R Resources

[Learning R Online](#)

[CRAN Task Views](#)

[R on StackOverflow](#)

[Getting Help with R](#)



RStudio

[RStudio IDE Support](#)

[RStudio Community Forum](#)

[RStudio Cheat Sheets](#)

[RStudio Tip of the Day](#)

[RStudio Packages](#)

[RStudio Products](#)

R packages

Packages are collections of R functions, data, and compiled code.

Libraries are directories in R where the packages are stored.

Built-in functions are part of R standard or base packages and do not need to be downloaded.

```
library(help = "base")
```

```
library(help = "stats")
```

Some functions are not built-in. To get these, need to download packages

R packages extend R's functionality.

R packages

Popular repositories for Packages:

The Comprehensive R Archive Network (CRAN)

▶ <https://cran.r-project.org/web/packages/>

Bioconductor

▶ <https://www.bioconductor.org/packages/release/bioc/>

GitHub

▶ <https://github.com/search?q=r+packages&type=registrypackages>

Note: Depending on the source of the package download,
instructions may differ. Typically:

```
install.packages("tidyverse") #download a package  
library(tidyverse) #load it into your RStudio session
```

Tidyverse Package

Tidyverse is a package from CRAN.

(<https://cran.rproject.org/web/packages/tidyverse/index.html>)

To get more details on tidyverse package..



R packages for data science

The tidyverse is an opinionated collection of R packages designed for data science. All packages share an underlying design philosophy, grammar, and data structures.

Install the complete tidyverse with:

```
install.packages("tidyverse")
```

```
ls("package:tidyverse") # list all functions in package
```

```
[1] "tidyverse_conflicts" "tidyverse_deps"      "tidyverse_"
[4] "tidyverse_packages"   "tidyverse_sitrep"    "tidyverse_"

```

```
?tidyverse # get information on package
```

In class exercise

Basic R Operations

1. Create an object named `my_sequence` that is a sequence from 1 to 7.
2. Use an R function to take the square roots of all the numbers in the sequence. Save this new sequence as an object named `sqrt_sequence`.
3. Multiply `sqrt_sequence` by 5.

Basic R Operations

```
my_sequence <- c(1:7)  
my_sequence
```

```
[1] 1 2 3 4 5 6 7
```

```
sqrt_sequence <- sqrt(my_sequence)  
  
sqrt_sequence
```

```
[1] 1.000000 1.414214 1.732051 2.000000 2.236068 2.449490 2.
```

```
sqrt_sequence * 5
```

```
[1] 5.000000 7.071068 8.660254 10.000000 11.180340 12.24
```

Package installation & loading

1. Install and load the faraway library

Package installation & loading

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

Outline of Topics

Getting Set up

- ▶ R
- ▶ RStudio

R basics

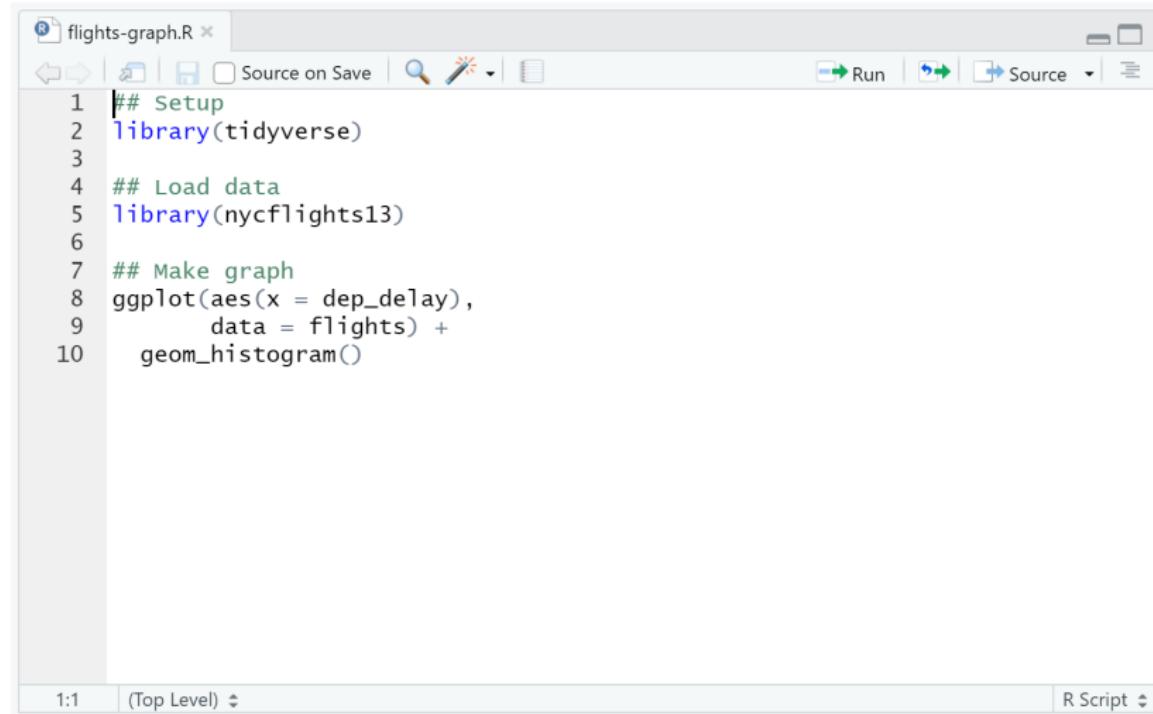
File types

- ▶ Scripts
- ▶ R markdown

Hello, world!

R Scripts

Files where you can save and edit your code



The screenshot shows the RStudio interface with an R script file open. The title bar says "flights-graph.R". The code editor contains the following R code:

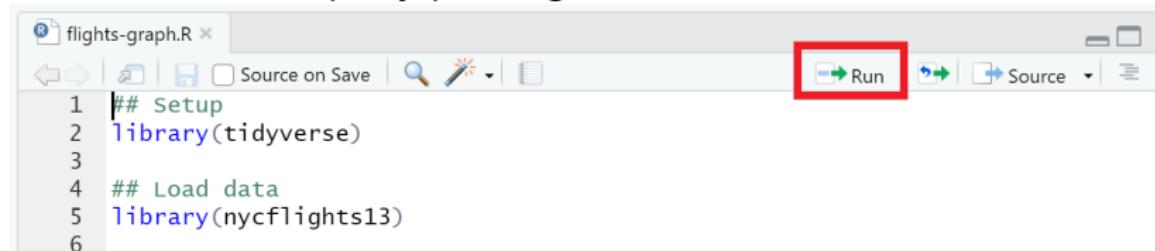
```
1 ## Setup
2 library(tidyverse)
3
4 ## Load data
5 library(nycflights13)
6
7 ## Make graph
8 ggplot(aes(x = dep_delay),
9         data = flights) +
10    geom_histogram()
```

The status bar at the bottom shows "1:1 (Top Level) R Script".

R Scripts

Running the code

Run the entire script by pressing Run



The screenshot shows the RStudio interface with a script file named "flights-graph.R" open. The code in the editor is:

```
1 ## Setup
2 library(tidyverse)
3
4 ## Load data
5 library(nycflights13)
6
```

The "Run" button in the toolbar is highlighted with a red box. The toolbar also includes other buttons for back, forward, search, and source.

Run the command where your cursor is located by pressing

Cmd/Ctrl + Enter

Run a section of commands by highlighting them and pressing

Cmd/Ctrl + Enter

R Scripts

Diagnostics

When working in a script, RStudio will mark syntax errors. If you hover over the red x, you can see what the problem is.

```
7 ## Make graph
8 ggplot(aes(x = dep_delay),
9         data = flights) +
10    geom_histogram(
```

unexpected end of document
unmatched opening bracket '('

R Markdown

R Markdown files combine code chunks with the results of those chunks and text and support formats like PDF, HTML, word files and slideshows.

The screenshot shows the RStudio interface with the following details:

- Title Bar:** The file is titled "flights-graph.Rmd*".
- Toolbar:** Includes standard icons for back, forward, save, knit, run, and other document operations.
- Code Editor:** Displays the R Markdown code. The code includes a YAML header, setup code, library imports, and a histogram chunk.

```
1 ---  
2 title: "Flights Graph"  
3 author: "Name"  
4 date: "25/11/2021"  
5 output: pdf_document  
6 ---  
7  
8 ```{r setup, include=FALSE}  
9 knitr::opts_chunk$set(echo = TRUE)  
10 ## Setup  
11 library(tidyverse)  
12 ## Load data  
13 library(nycflights13)  
14 ````  
15  
16 A histogram displaying *flight departure delays*:  
17  
18 ```{r, flights.histogram}  
19 ggplot(aes(x = dep_delay),  
20         data = flights) +  
21     geom_histogram()  
22
```
- Status Bar:** Shows the file name "# Flights Graph" and the format "R Markdown".

R Markdown

YAML header

contains the document information and settings are specified

The screenshot shows the RStudio interface with an R Markdown file named "flights-graph.Rmd". The YAML header, which contains metadata like title, author, date, and output type, is highlighted with a red rectangle. The main body of the document includes code chunks for setup, data loading, and a histogram plot.

```
flights-graph.Rmd* ×
1 ---[red box]
2 title: "Flights Graph"
3 author: "Name"
4 date: "25/11/2021"
5 output: pdf_document
6 ---
7
8 + ``{r setup, include=FALSE}
9 knitr::opts_chunk$set(echo = TRUE)
10 ## Setup
11 library(tidyverse)
12 ## Load data
13 library(nycflights13)
14 +
15
16 A histogram displaying *flight departure delays*:
17
18 + ``{r, flights.histogram}
19 ggplot(aes(x = dep_delay),
20         data = flights) +
21     geom_histogram()
22
23 # Flights Graph ♦ R Markdown ♦
```

R Markdown Chunks

Text goes in between the code chunks. This text can be formatted with basic markdown syntax.

You can write in code chunks the same way you would write in a script

```
flights-graph.Rmd* | Knit on Save | ABC | Knit | Run |
```

```
1 ---  
2 title: "Flights Graph"  
3 author: "Name"  
4 date: "25/11/2021"  
5 output: pdf_document  
6 ---  
7  
8 ```{r setup, include=FALSE}  
9 knitr::opts_chunk$set(echo = TRUE)  
10 ## Setup  
11 library(tidyverse)  
12 ## Load data  
13 library(nycflights13)  
14 ...  
15  
16 A histogram displaying *flight departure delays*:  
17  
18 ```{r, flights.histogram}  
19 ggplot(aes(x = dep_delay),  
20         data = flights) +  
21     geom_histogram()  
22 ...
```

R Markdown

Running the code

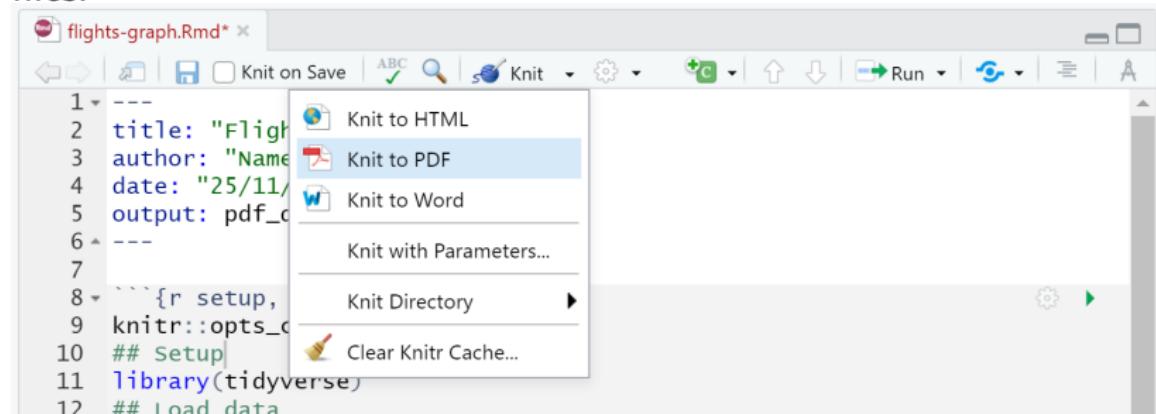
Like with a script, you can use the Run button Cmd/Ctrl + Enter.
Each chunk also has an arrow you can press to run the code in
that chunk.

The output will display below the code chunk rather than in the
console or the Plots section.

R Markdown

Knitting to .Rmd file

To present your work, you can knit your R Markdown file to a more common file type, including PDFs, Word documents, and html files.



The screenshot shows the RStudio interface with a file named "flights-graph.Rmd" open. The code editor window displays the following R Markdown code:

```
1 ---  
2 title: "Flight Data Analysis"  
3 author: "Name"  
4 date: "25/11/2023"  
5 output: pdf_document  
6 ---  
7  
8 ````{r setup,  
9 knitr:::opts_chunk$set(  
10 ## Setup}  
11 library(tidyverse)  
12 ## Load data
```

A context menu is open over the code editor, specifically over the "Knit" button in the toolbar. The menu options are:

- Knit to HTML
- Knit to PDF
- Knit to Word
- Knit with Parameters...
- Knit Directory
- Clear Knitr Cache...

In class exercise

R Script

1. Create a new R script.
2. Write code to calculate the average of numbers 1 to 10.
3. Run the script.
4. Save the script.

R Script

The screenshot shows the RStudio interface with the following details:

- Top Bar:** Shows three tabs: "IntroR-0.qmd", "IntroR-1.qmd", and "R_script.R".
- Toolbar:** Includes icons for file operations (New, Open, Save, Print), search, and code editing.
- Code Editor:** Displays the following R code:

```
1 average <- mean(1:10)
2 average
3
```
- Right Panel:** A large, empty white area.
- Bottom Status Bar:** Shows "3:1" and "(Top Level) ⌂".
- Bottom Right Corner:** Shows "R Script ⌂".

R Markdown

1. Create a new R Markdown file.
2. Load the tidyverse and faraway libraries.
3. Load the dataset “broccoli” by calling the function data()
4. Print out the broccoli dataset in the R Markdown file.
5. Save the file.
6. Knit the R Markdown file to PDF.

R Markdown

```
---
```

```
title: "R_markdown"
output: html_document
date: "2023-05-19"
---
```

```
```{r}
#install.packages("tidyverse")
#install.packages("faraway")
library(tidyverse)
library(faraway)
````
```

```
```{r}
data("broccoli")
broccoli
|``
```

Description: df [36 x 4]

	wt	grower	box	cluster
	<dbl>	<fctr>	<fctr>	<fctr>
1	352	1	1	1
2	369	1	1	2
3	383	1	1	3
4	339	2	1	1
5	367	2	1	2
6	328	2	1	3
7	376	3	1	1
8	359	3	1	2
9	388	3	1	3
10	365	1	2	1

1–10 of 36 rows

Previous 1 2 3 4 Next

R Markdown

Chunk 2

Questions?

# Outline of Topics

## Getting Set up

- ▶ R
- ▶ RStudio

## R basics

## File types

- ▶ Scripts
- ▶ R markdown

**Hello, world!** (*Live coding*)