

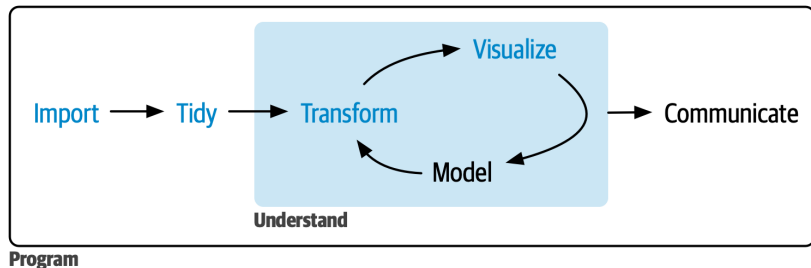
Socio-Informatics 348

R, Tidyverse, and Workflows

Dr Lisa Martin

Department of Information Science
Stellenbosch University

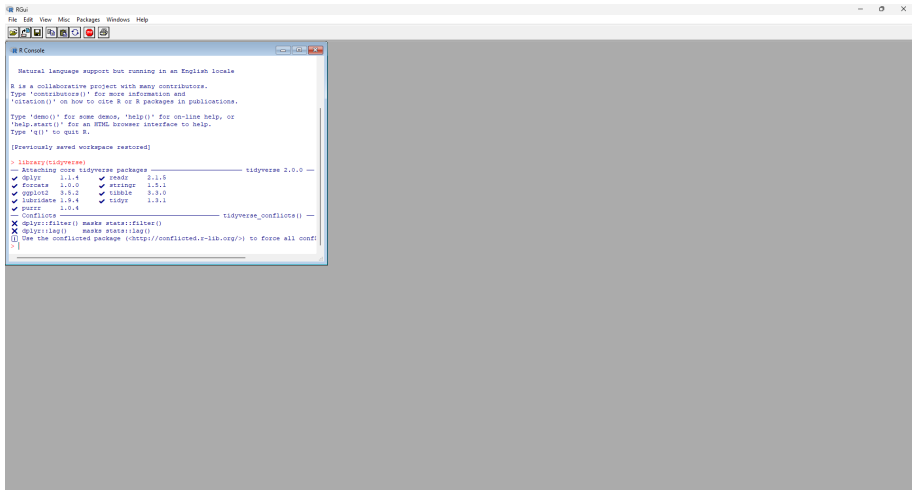
Typical Data Science Project Model



Source: R4DS, Wickham et al. (2023)

What is R? What is RStudio?

- R = programming language for data analysis
- RStudio = interface for working with R
- Why use them?
 - Reproducibility – code can be shared and reused
 - Power – handles large datasets and complex analyses
 - Open-source – free to use and modify
 - Community – large user base and extensive resources



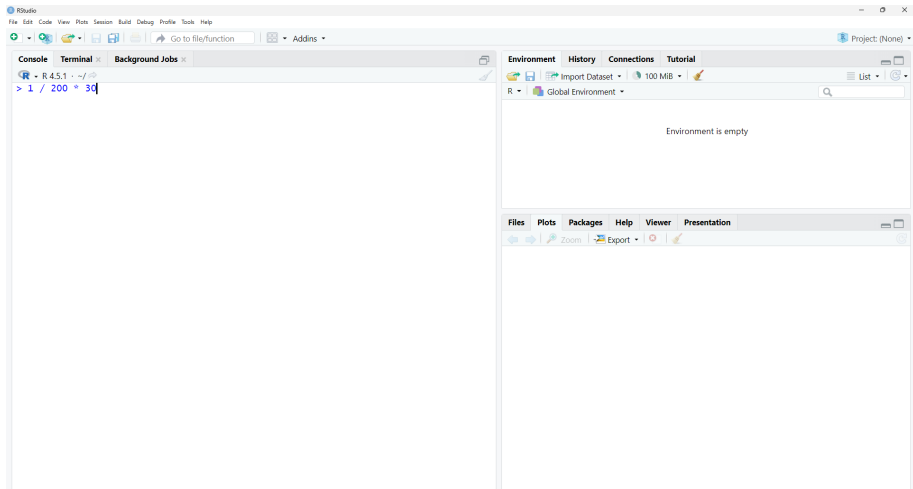
RStudio

The screenshot displays the RStudio integrated development environment (IDE) interface. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. Below the menu bar is a toolbar with icons for file operations and a search bar. The main workspace is divided into four panes:

- Console:** Displays the R version (4.5.1) and copyright information. It shows the output of the `license()` function, which includes the R license text and the message "Natural language support but running in an English locale".
- Environment:** Shows the current environment, which is empty. The toolbar includes icons for Import Dataset, Global Environment, and a search bar.
- Files:** Shows the file explorer, currently displaying the "Global Environment" pane.
- Plots:** Shows the plot pane, currently displaying the "Global Environment" pane.

The bottom status bar shows the current file path and the number of lines in the file (5 / 23).

Basics



Basics

The screenshot displays the RStudio application window. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. Below the menu is a toolbar with icons for saving, opening, and other file operations, along with a search bar and an 'Addins' dropdown. The main workspace is divided into four panels:

- Console:** Shows the R prompt and the execution of the command `1 / 200 * 30`, resulting in the output `[1] 0.15`.
- Environment:** Displays the current environment as 'Global Environment' and indicates that the environment is empty.
- Files:** Shows the file explorer with a search bar and a list of files.
- Plots:** Displays the plot area with a search bar and a list of plots.

The bottom status bar shows the current page number 7 out of 23.

Basics

The screenshot displays the RStudio application window. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. Below the menu is a toolbar with icons for saving, opening, and navigating files. The main workspace is divided into four panels:

- Console:** Shows the R prompt and the following commands and output:

```
> 1 / 200 * 30
[1] 0.15
> (59 + 73 + 2) / 3
[1] 44.66667
> sin(pi / 2)
[1] 1
> |
```
- Environment:** Displays the current environment, showing 'Global Environment' with a memory usage of 100 MiB. The text 'Environment is empty' is visible.
- Files:** Shows the file explorer, currently empty.
- Plots:** Shows the plot viewer, currently empty.

The bottom status bar indicates the current slide is 8 out of 23.

Basics

The screenshot displays the RStudio environment. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. Below the menu is a toolbar with icons for saving, running, and other functions. The main workspace is divided into three panels:

- Console:** Shows the R prompt and the following commands and output:

```
> 1 / 200 * 30
[1] 0.15
> (59 + 73 + 2) / 3
[1] 44.66667
> sin(pi / 2)
[1] 1
> x <- 3 * 4
> primes <- c(2, 3, 5, 7, 11, 13)
> |
```
- Environment:** Displays the current environment, which is the Global Environment. It shows a table of values:

Values	
primes	num [1:6] 2 3 5 7 11 13
x	12
- Files:** Shows the file explorer, currently displaying the 'Global Environment'.

The bottom status bar includes icons for navigation and a zoom level of 100%.

Basics

The screenshot displays the RStudio environment. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. Below the menu is a toolbar with icons for saving, running, and other functions. The main window is divided into four panes: Console, Terminal, Background Jobs, and Environment/History/Connections/Tutorial. The Console pane shows the following R code and output:

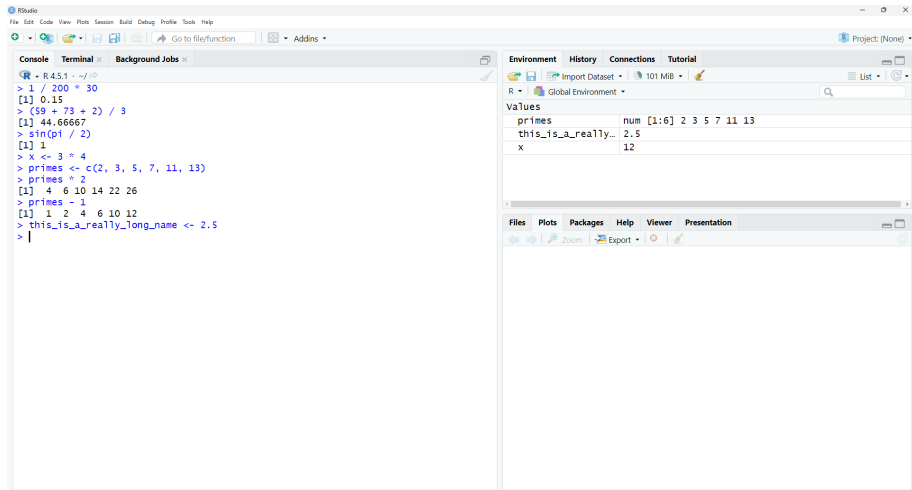
```
R - R 4.5.1 - ~/
> 1 / 200 * 30
[1] 0.15
> (59 + 73 + 2) / 3
[1] 44.66667
> sin(pi / 2)
[1] 1
> x <- 3 * 4
> primes <- c(2, 3, 5, 7, 11, 13)
> primes * 2
[1] 4 6 10 14 22 26
> primes - 1
[1] 1 2 4 6 10 12
> |
```

The Environment pane on the right shows the Global Environment with the following variables:

Variable	Value
primes	num [1:6] 2 3 5 7 11 13
x	12

The bottom of the interface features a toolbar with icons for zooming, exporting, and other utility functions.

Basics



The screenshot displays the RStudio environment. The console on the left contains the following R code and its output:

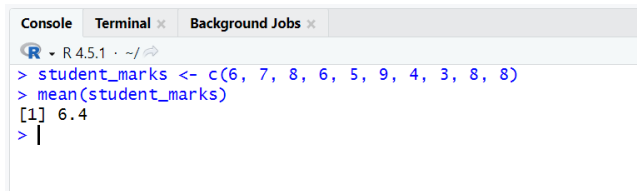
```
R > R 4.5.1 ~ /  
> 1 / 200 * 30  
[1] 0.15  
> (59 + 73 + 2) / 3  
[1] 44.66667  
> sin(pi / 2)  
[1] 1  
> x <- 3 * 4  
> primes <- c(2, 3, 5, 7, 11, 13)  
> primes * 2  
[1] 4 6 10 14 22 26  
> primes - 1  
[1] 1 2 4 6 10 12  
> this_is_a_really_long_name <- 2.5  
> |
```

The Environment pane on the right shows the following variables and their values:

Variable	Value
primes	num [1:6] 2 3 5 7 11 13
this_is_a_really...	2.5
x	12

Packages and Functions

- R has a rich ecosystem of packages
- Packages extend R's functionality
- Functions are the building blocks of R code
- R has some built-in functions, but you can also write your own
- Example: `mean()` calculates the average of a numeric vector



```
Console Terminal x Background Jobs x
R 4.5.1 · ~/
> student_marks <- c(6, 7, 8, 6, 5, 9, 4, 3, 8, 8)
> mean(student_marks)
[1] 6.4
> |
```

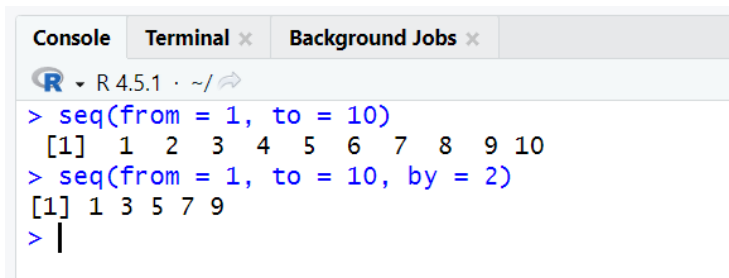
Functions

- Functions take inputs called 'arguments' and return outputs
- Example: `mean(student_marks, na.rm = TRUE)` calculates the mean of `student_marks`, ignoring NA values
- Beware! Functions can have default values for arguments
- The default value for `na.rm` is `FALSE`, meaning NA values will be included in the calculation

```
> student_marks <- c(6, 7, 8, 6, 5, 9, 4, NA, 8, 8)
> mean(student_marks)
[1] NA
> mean(student_marks, na.rm = TRUE)
[1] 6.777778
> |
```

Functions

- Arguments differ from function to function
- Some functions have multiple arguments, some have only one
- Here is another example:



The screenshot shows an R console window with three tabs: 'Console', 'Terminal x', and 'Background Jobs x'. The 'Console' tab is active, showing the R prompt and the execution of two `seq` functions. The first command is `> seq(from = 1, to = 10)`, which returns the sequence `[1] 1 2 3 4 5 6 7 8 9 10`. The second command is `> seq(from = 1, to = 10, by = 2)`, which returns the sequence `[1] 1 3 5 7 9`. The prompt `> |` is visible on the next line, indicating the user is ready to enter another command.

```
R 4.5.1 · ~/
> seq(from = 1, to = 10)
[1] 1 2 3 4 5 6 7 8 9 10
> seq(from = 1, to = 10, by = 2)
[1] 1 3 5 7 9
> |
```

- The default value of `by` in `seq` is 1, meaning the sequence will increment by 1

Scripts

Why use scripts?

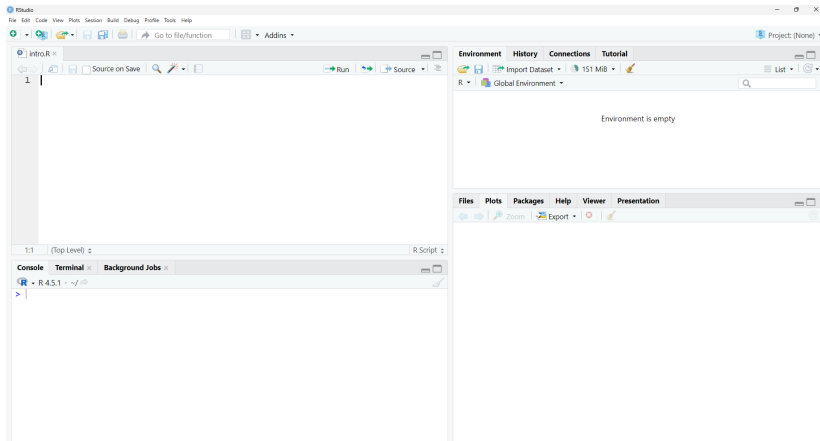
- Imagine this...

Benefits of using scripts:

- Scripts save time and effort
- Scripts allow for reproducibility
- Scripts allow for collaboration
- Scripts allow for version control
- Scripts allow for debugging
- Scripts allow for automation

Scripts

File ▷ New File ▷ R Script

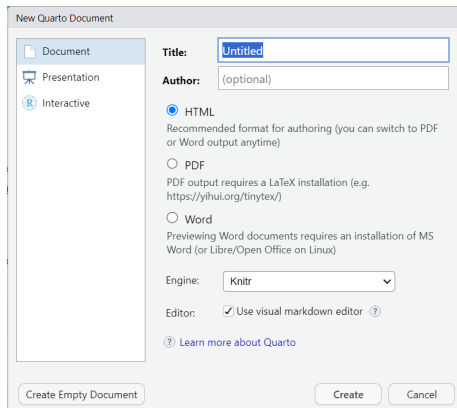


- Keep naming conventions consistent

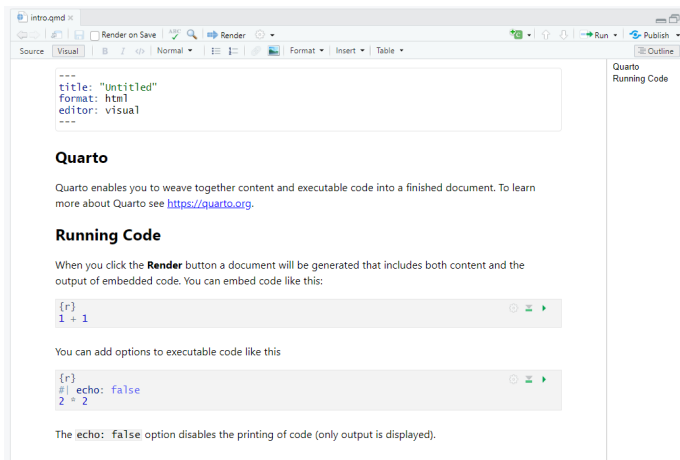
What is the Tidyverse?

- A collection of R packages for data science
- Core packages: `ggplot2`, `dplyr`, `tidyr`, `readr`, `purrr`, `tibble`, `stringr`, `forcats`
- Unified design philosophy
- Can be loaded into R with one line: `library(tidyverse)`

File ▷ New File ▷ Quarto Document



- Keep naming conventions consistent



The screenshot shows the Quarto editor interface. The top toolbar includes buttons for 'Render on Save', 'Render', 'Run', and 'Publish'. The document is titled 'intro.qmd'. The main content area displays a YAML header, a section header, a paragraph, another section header, a paragraph, and two code blocks. The right sidebar shows the 'Outline' and 'Running Code' panels.

```
---  
title: "Untitled"  
format: html  
editor: visual  
---
```

Quarto

Quarto enables you to weave together content and executable code into a finished document. To learn more about Quarto see <https://quarto.org>.

Running Code

When you click the **Render** button a document will be generated that includes both content and the output of embedded code. You can embed code like this:

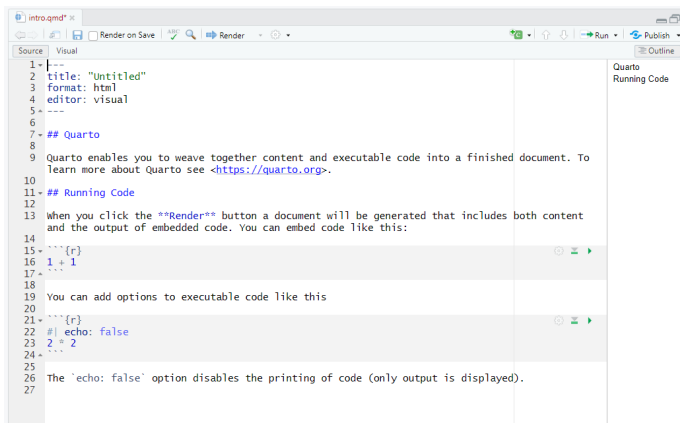
```
{r}  
1 + 1
```

You can add options to executable code like this

```
{r}  
#| echo: false  
2 * 2
```

The `echo: false` option disables the printing of code (only output is displayed).

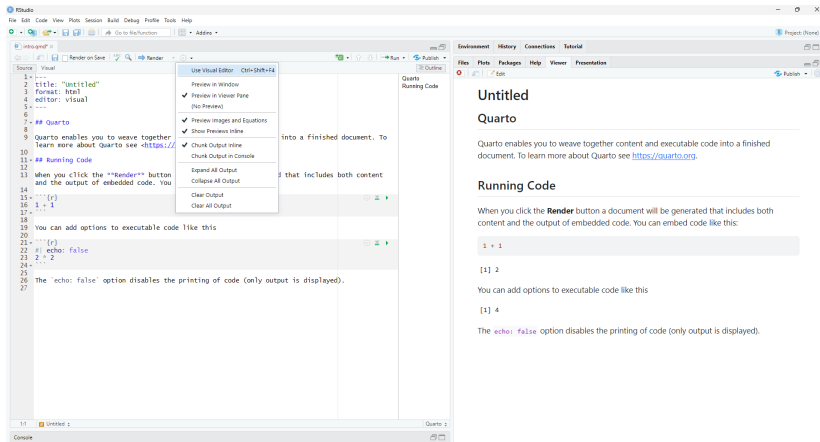
Quarto



The screenshot shows the Quarto RStudio interface with a source file named 'intro.qmd'. The interface is divided into three main panes: Source, Visual, and Outline. The Source pane shows the following code:

```
1 -|---
2 title: "Untitled"
3 format: html
4 editor: visual
5 -|---
6
7 -|## Quarto
8
9 Quarto enables you to weave together content and executable code into a finished document. To
  learn more about Quarto see <https://quarto.org>.
10
11 -|## Running Code
12
13 When you click the **Render** button a document will be generated that includes both content
  and the output of embedded code. You can embed code like this:
14
15 ```{r}
16 1 + 1
17 ```
18
19 You can add options to executable code like this
20
21 ```{r}
22 #| echo: false
23 2 * 2
24 ```
25
26 The 'echo: false' option disables the printing of code (only output is displayed).
27
```

The Visual pane shows the rendered output of the code, which is currently empty. The Outline pane shows the document structure, including the title 'Untitled' and the sections '## Quarto' and '## Running Code'.



- You won't have to render, just submit the qmd file

Best Practices from Day 1

- Always load libraries at the top
- Use scripts to save your work
- Comment your code!
- Ask for help — errors are normal (and expected)

Next Week

- No class
- Tutors will present the practical (Prac 0) over the next 2 weeks
- Focus on getting comfortable with R, RStudio and Quarto