

# Sociology Quant Camp

Introduction to R

Module 1: Intro and basics

Monica Alexander, Statistical Sciences and Sociology

# Welcome

- Hello!
- Overview of today
  - Module 1: Intro, navigating RStudio, R scripts and R code basics
  - Module 2: Piping, the tidyverse
  - Module 3: Intro to plotting with ggplot2

# Hello!

- Statistical Sciences and Sociology departments
- Demographer
- Will be teaching SOC6302 in the Winter
- Love R
- [monica.alexander@utoronto.ca](mailto:monica.alexander@utoronto.ca)

# Introduction to R and RStudio

# Downloads

- To install R:
  - Go to <http://cran.utstat.utoronto.ca/>
  - Select the download link that is relevant to you: if you have a Mac, select “Download R for (Mac) OS X”, if you have a Windows machine, select “Download R for Windows”.
  - If you have a Mac, click on the “R-4.2.1pkg” link. If you have a Windows machine, click on the “base” link, then click on the “Download R 4.2.1 for Windows” link. Open the downloaded file and follow the install instructions on your machine.

# Downloads

- To install RStudio
  - Go to <https://rstudio.com/products/rstudio/download/>
  - Scroll down and click the “Download” button under “RStudio Desktop Open Source License Free”
  - Click the “Download RStudio” button (it will either say for Mac or Windows)
  - Open the downloaded file and follow the install instructions on your machine.

**Install check?**

# Back up plan: Rstudio cloud

- Rstudio Cloud: <https://rstudio.cloud/plans/free>
- You will need to create a free account
- Then go to: <https://rstudio.cloud/content/4459414>



# What is R?

- R is a programming language for statistical computing and graphics
- Using R is like speaking another language (but you type it)
- You may have used other programs to do statistical calculations before (Excel, SPSS)
- With R you have to give the computer typed commands in order for it to do statistics (rather than clicking buttons)
- Much more powerful methods available



# What is RStudio?

- RStudio is an integrated development environment for R
- It makes it easier to write R code and visualizes inputs and outputs
- Car analogy:
  - R is the engine
  - RStudio is the car dashboard



Untitled1 \*

Source on Save

Run

Source

1

2

3

4

5

6

7

8

# Here is some R code I am writing

x <- 3

y <- 2

my\_name <- "Monica"

x+y

|

8:1

(Top Level)

R Script

Console

Terminal

Background Jobs

R 4.2.1 · ~/

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.  
  
Natural language support but running in an English locale  
  
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.  
  
> x <- 3  
> y <- 2  
> my\_name <- "Monica"  
> x+y  
[1] 5  
> |

Environment

History

Connections

Tutorial

Import Dataset

144 MiB

List

R

Global Environment

Values

my_name	"Monica"
x	3
y	2

Files

Plots

Packages

Help

Viewer

Presentation

New Folder

New Blank File

Delete

Rename

More

Home

	Name	Size	Modified
<input type="checkbox"/>	.r		
<input type="checkbox"/>	.Renviron	169 B	Jan 20, 2022, 3:07 PM
<input type="checkbox"/>	.Rhistory	17.2 KB	Sep 1, 2022, 9:18 AM
<input type="checkbox"/>	Applications		
<input type="checkbox"/>	bin		
<input type="checkbox"/>	Desktop		
<input type="checkbox"/>	Documents		
<input type="checkbox"/>	Downloads		
<input type="checkbox"/>	Dropbox		
<input type="checkbox"/>	Library		
<input type="checkbox"/>	Movies		
<input type="checkbox"/>	Music		
<input type="checkbox"/>	OneDrive – University of Toronto		
<input type="checkbox"/>	Pictures		
<input type="checkbox"/>	Public		
<input type="checkbox"/>	src		

Editor

Where you write all your code

Environment

Shows objects that have been defined in current session

Console

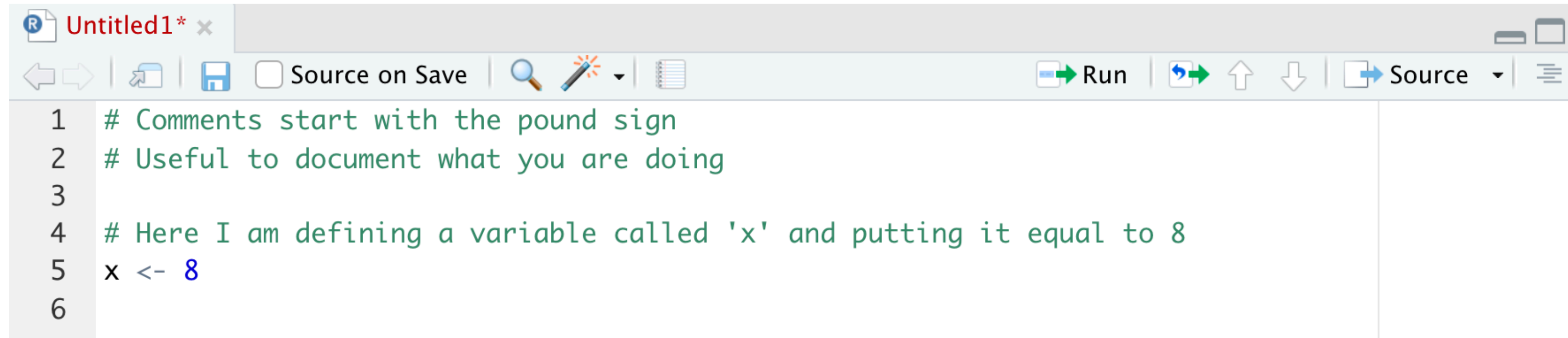
Shows the code that's been executed (run)

Navigation tab, help files, plot viewer

# Demo: opening RStudio, opening and saving a R script

# R code basics

# Code versus comments

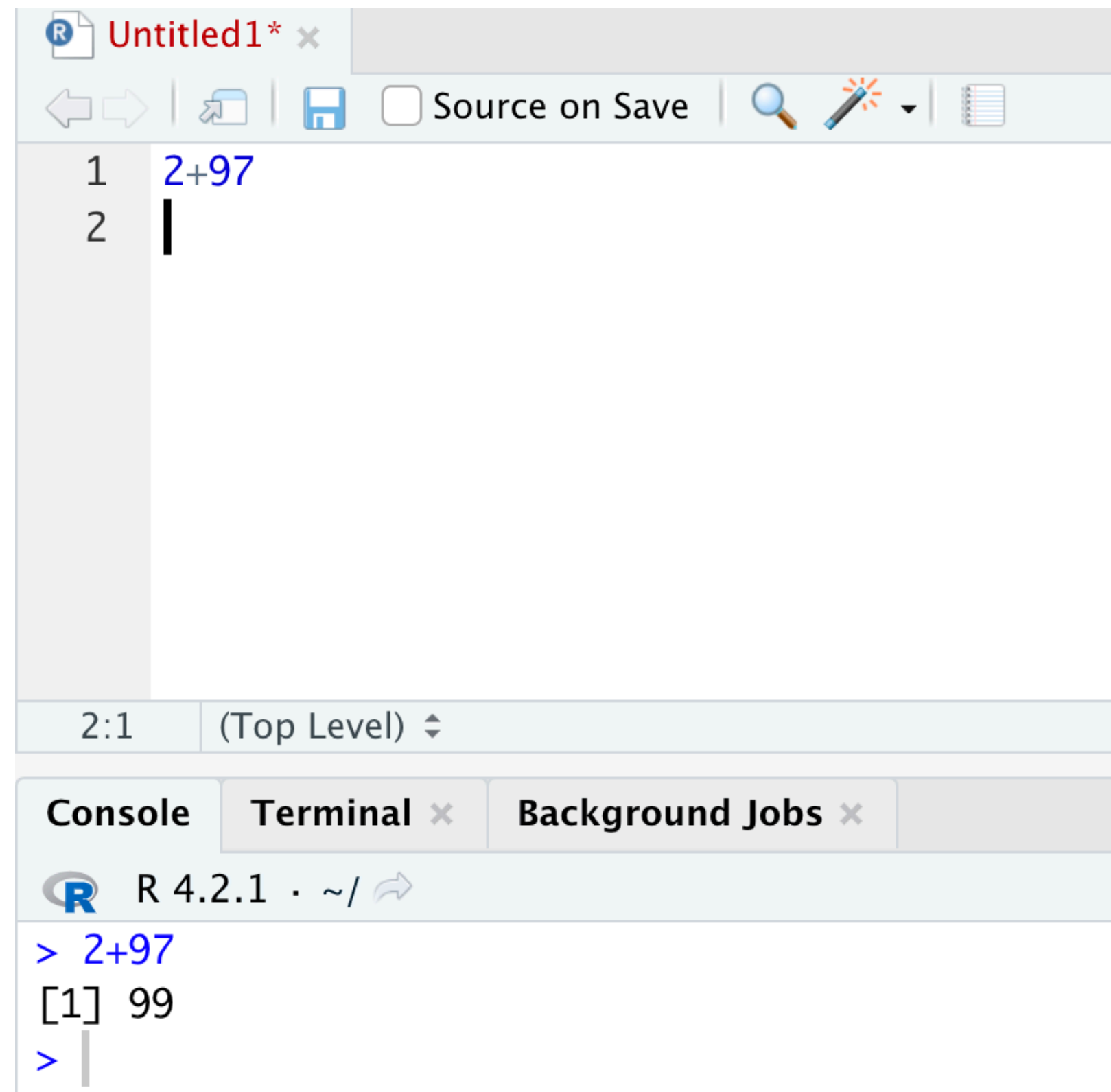


The screenshot shows the RStudio interface with a single editor window titled 'Untitled1\*'. The window contains R code with comments. The code is as follows:

```
1 # Comments start with the pound sign
2 # Useful to document what you are doing
3
4 # Here I am defining a variable called 'x' and putting it equal to 8
5 x <- 8
6
```

The interface includes a toolbar at the top with icons for navigation, saving, and running code. The 'Run' button is highlighted with a green arrow. The code is color-coded: comments are green, and the variable name 'x' and the value '8' are blue.

# R as a calculator

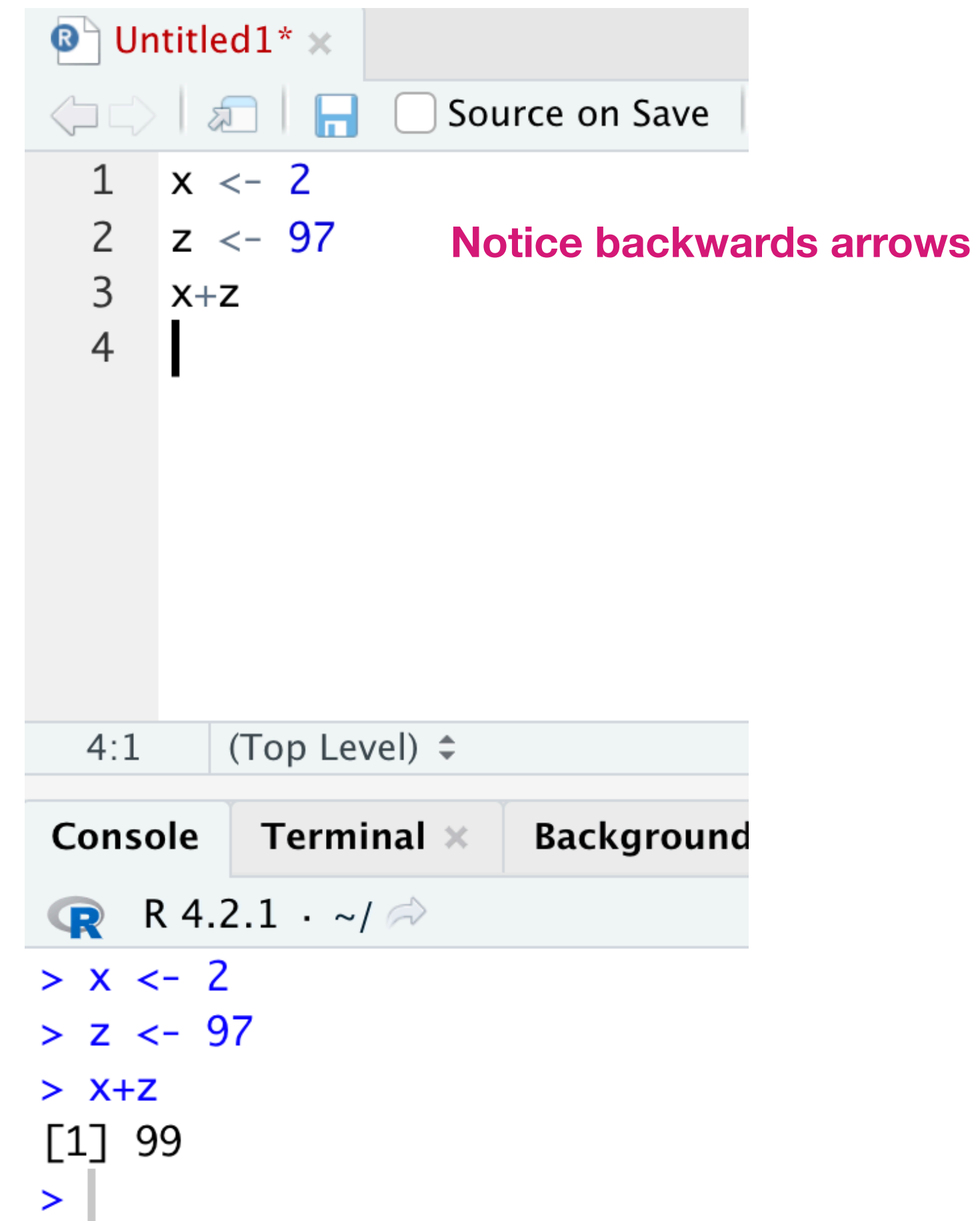


The RStudio interface shows a script editor with two lines of code: `1 2+97` and `2 |`. The console at the bottom shows the execution of the first line: `> 2+97` followed by `[1] 99`. The status bar indicates the current line is 2:1 at the top level.

```
1 2+97
2 |
```

```
> 2+97
[1] 99
> |
```

# ... versus defining objects



The RStudio interface shows a script editor with four lines of code: `1 x <- 2`, `2 z <- 97`, `3 x+z`, and `4 |`. A pink annotation "Notice backwards arrows" points to the assignment operators in the first two lines. The console shows the execution of these lines: `> x <- 2`, `> z <- 97`, `> x+z`, followed by `[1] 99`. The status bar indicates the current line is 4:1 at the top level.

```
1 x <- 2
2 z <- 97
3 x+z
4 |
```

Notice backwards arrows

```
> x <- 2
> z <- 97
> x+z
[1] 99
> |
```

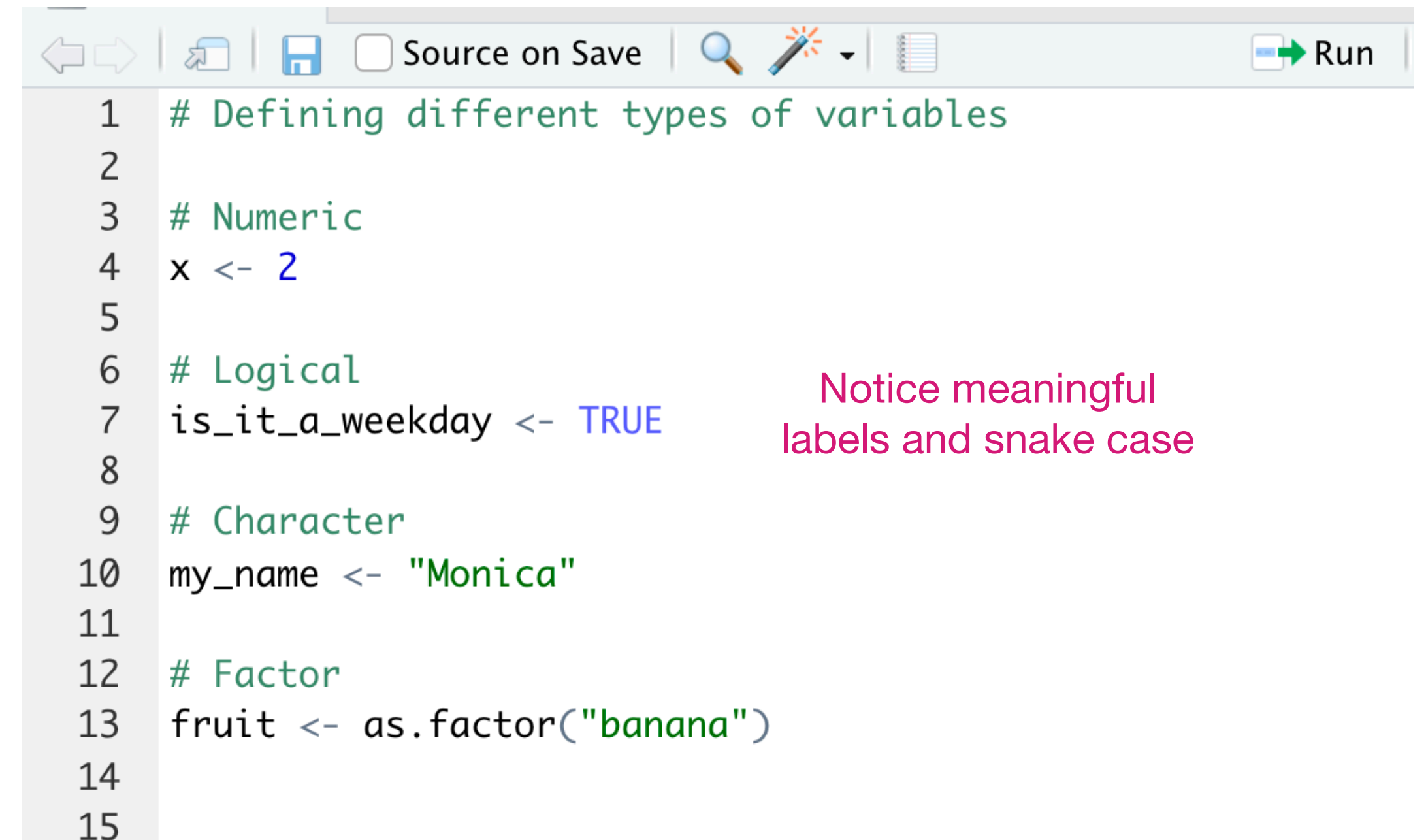
All standard mathematical operations are available

**Demo: mathematical operations,  
defining objects**



# Types of variables in R

- Numeric
- Logical (TRUE/FALSE)
- Character
- Factor (categories with levels defined)



The screenshot shows an R script editor window with a toolbar at the top containing icons for navigation, saving, and running. The script defines four variables: a numeric variable 'x', a logical variable 'is\_it\_a\_weekday', a character variable 'my\_name', and a factor variable 'fruit'. The code is as follows:

```
1 # Defining different types of variables
2
3 # Numeric
4 x <- 2
5
6 # Logical
7 is_it_a_weekday <- TRUE
8
9 # Character
10 my_name <- "Monica"
11
12 # Factor
13 fruit <- as.factor("banana")
14
15
```

Notice meaningful labels and snake case

**Demo: different variable types,  
checking variable types**

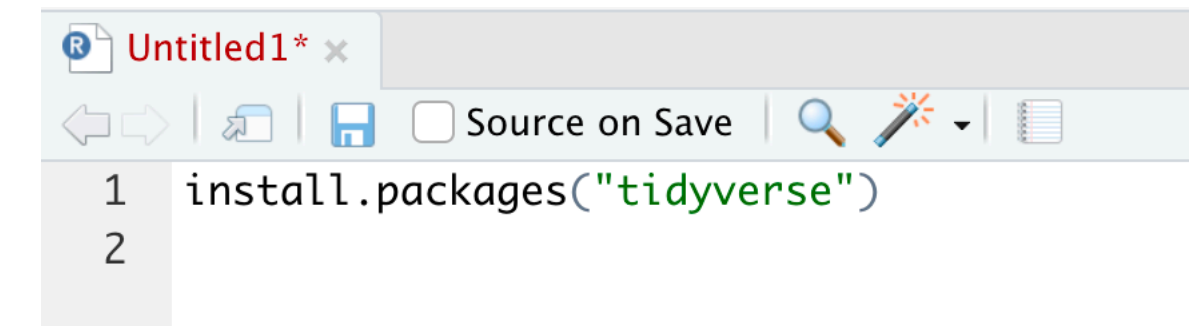
# Packages

- People have written R Packages, which are add ons to base R that increase functionality
- Phone analogy
  - R/RStudio is a phone
  - Packages are apps
- One package that is very useful is the `tidyverse`
- Has graphing capabilities, tidyverse grammar (more later)



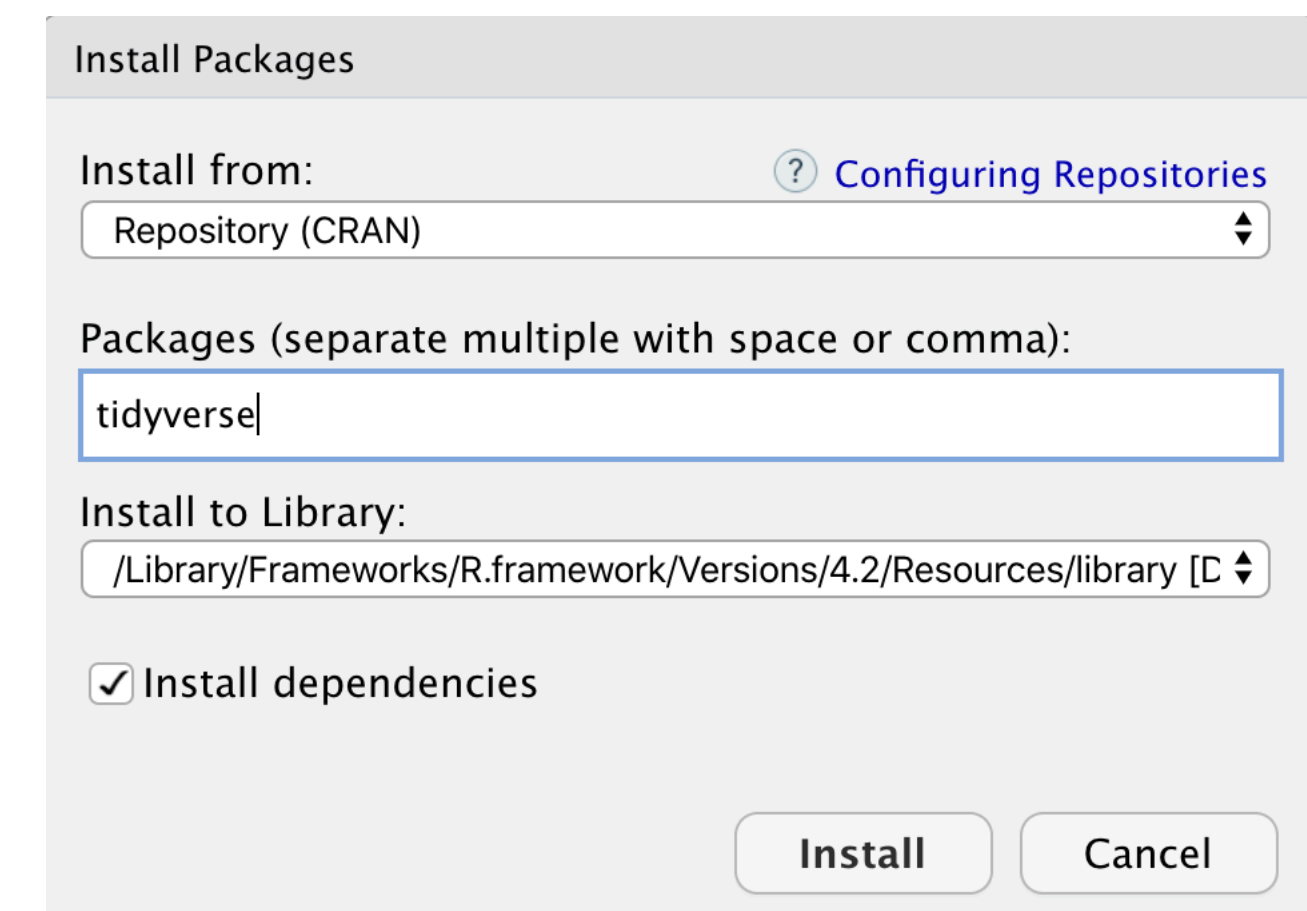
# Installing packages

- Via code
  - Or using menu (Tools -> Install packages...)
  - Once a package is installed, don't need to do it again!
- 
- To use the functionality in a package, need to load it in at the start of your code using the `library` function



```
1 install.packages("tidyverse")
2
```

OR



Install Packages

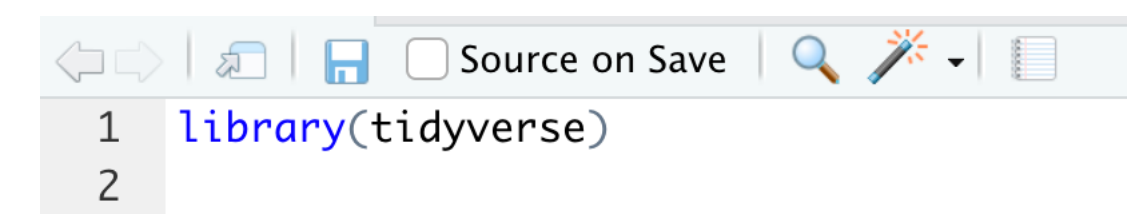
Install from: [? Configuring Repositories](#)  
Repository (CRAN)

Packages (separate multiple with space or comma):  
tidyverse

Install to Library:  
/Library/Frameworks/R.framework/Versions/4.2/Resources/library [D]

☒ Install dependencies

Install Cancel



```
1 library(tidyverse)
2
```

# Different types of objects in R

- Single values
- Vectors:
  - contain two or more values
  - Defined with the `c()` function (“concatenate”)
  - Values must be of the same type
- Data frames (tibbles)
  - Closest thing to a dataset that we deal with
  - Each column is a different variable, each row is an observation
  - Columns (variables) can be different types

```
library(tidyverse)

# single value
x <- 2
color <- "red"

# vector
my_numbers <- c(0,3,1,4,2)
my_names <- c("Monica", "Rohan", "Edward", "Hugo")

# tibble
my_dataset <- tibble(
  respondent = c("A", "B", "C"),
  age = c(16,92, 45)
)
```

We can define columns of a tibble using vectors

# Demo: defining different types of objects

# Functions

- Do stuff to your variables!
- Have already seen some: `as.factor()`, `c()`, `tibble()`
- Examples:
  - `mean()`, `median()`
  - `min()`, `max()`
  - `length()`, `dim()`
  - `paste()`
  - `is.numeric()` etc

```
1 library(tidyverse)
2
3 x <- 4
4 is.numeric(x)
5 is.character(x)
6
7 my_numbers <- c(0,3,1,4,2)
8 my_names <- c("Monica", "Rohan", "Edward", "Hugo")
9
10 mean(my_numbers)
11 length(my_names)
12
13 my_dataset <- tibble(
14   respondent = c("A", "B", "C"),
15   age = c(16,92, 45)
16 )
17
18 dim(my_dataset)  Stands for dimensions
19
```

10:17 (Top Level) ⌵

Console Terminal x Background Jobs x

R 4.2.1 · ~/

```
> library(tidyverse)
>
> x <- 4
> is.numeric(x)
[1] TRUE
> is.character(x)
[1] FALSE
>
> my_numbers <- c(0,3,1,4,2)
> my_names <- c("Monica", "Rohan", "Edward", "Hugo")
>
> mean(my_numbers)
[1] 2
> length(my_names)
[1] 4
>
> my_dataset <- tibble(
+   respondent = c("A", "B", "C"),
+   age = c(16,92, 45)
+ )
>
> dim(my_dataset)
[1] 3 2
> |
```

# Demo: functions

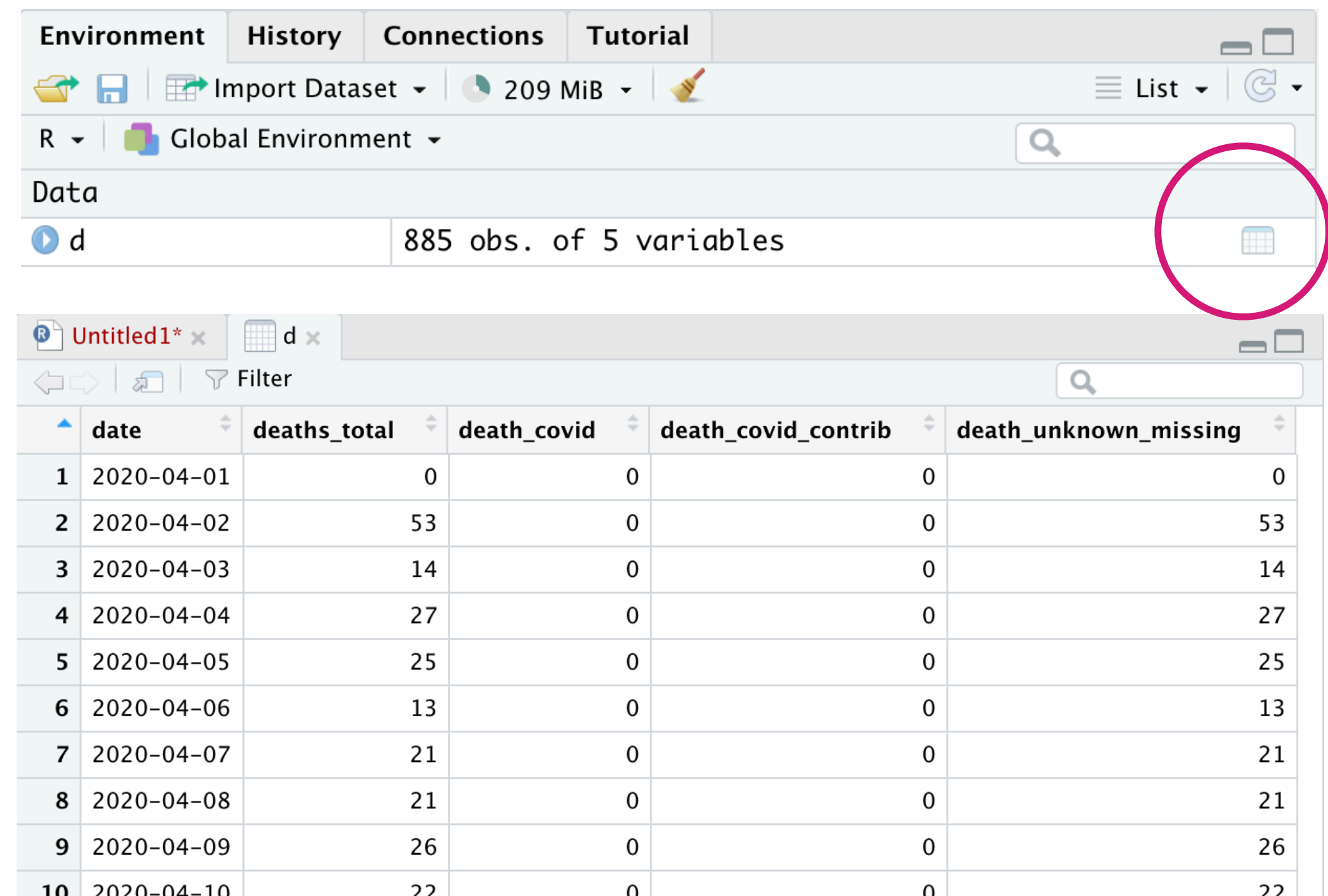


# Opening files

- One of the most powerful ways we can use R is to analyze and visualize data
- Need to be able to read in files of different formats (csv, excel, Stata...)
- Can use a variety of `read_*()` functions e.g. `read_csv()`
- Note these are from the `tidyverse` package so need to make sure this has been loaded in
- Data are from <https://data.ontario.ca/en/dataset/deaths-involving-covid-19-by-fatality-type>

Need to be careful with file paths!

```
Untitled1* x
1 library(tidyverse)
2
3 d <- read_csv("deaths_fatality_type.csv")
```



Environment History Connections Tutorial

Import Dataset 209 MiB

R Global Environment

Data

d 885 obs. of 5 variables

	date	deaths_total	death_covid	death_covid_contrib	death_unknown_missing
1	2020-04-01	0	0	0	0
2	2020-04-02	53	0	0	53
3	2020-04-03	14	0	0	14
4	2020-04-04	27	0	0	27
5	2020-04-05	25	0	0	25
6	2020-04-06	13	0	0	13
7	2020-04-07	21	0	0	21
8	2020-04-08	21	0	0	21
9	2020-04-09	26	0	0	26
10	2020-04-10	22	0	0	22

**Demo: opening files, setting  
working directory, viewing datasets**

# Where to get help

- Lots of good, free online sources
  - R for Data Science: <https://www.tidyverse.org/learn/>
  - Telling stories with data: <https://tellingstorieswithdata.com/>
  - Tidyverse skills for data science: <https://jhudatascience.org/tidyversecourse/intro.html>
- Google/Stack Overflow
- Email
- Practice, practice, practice; don't be afraid of mistakes