

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
- Will put materials here: <https://www.monicaalexander.com/teaching/>

Hello!

- Statistical Sciences and Sociology departments
- Demographer
- Will be teaching SOC6302 in the Winter
- Love R
- 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?

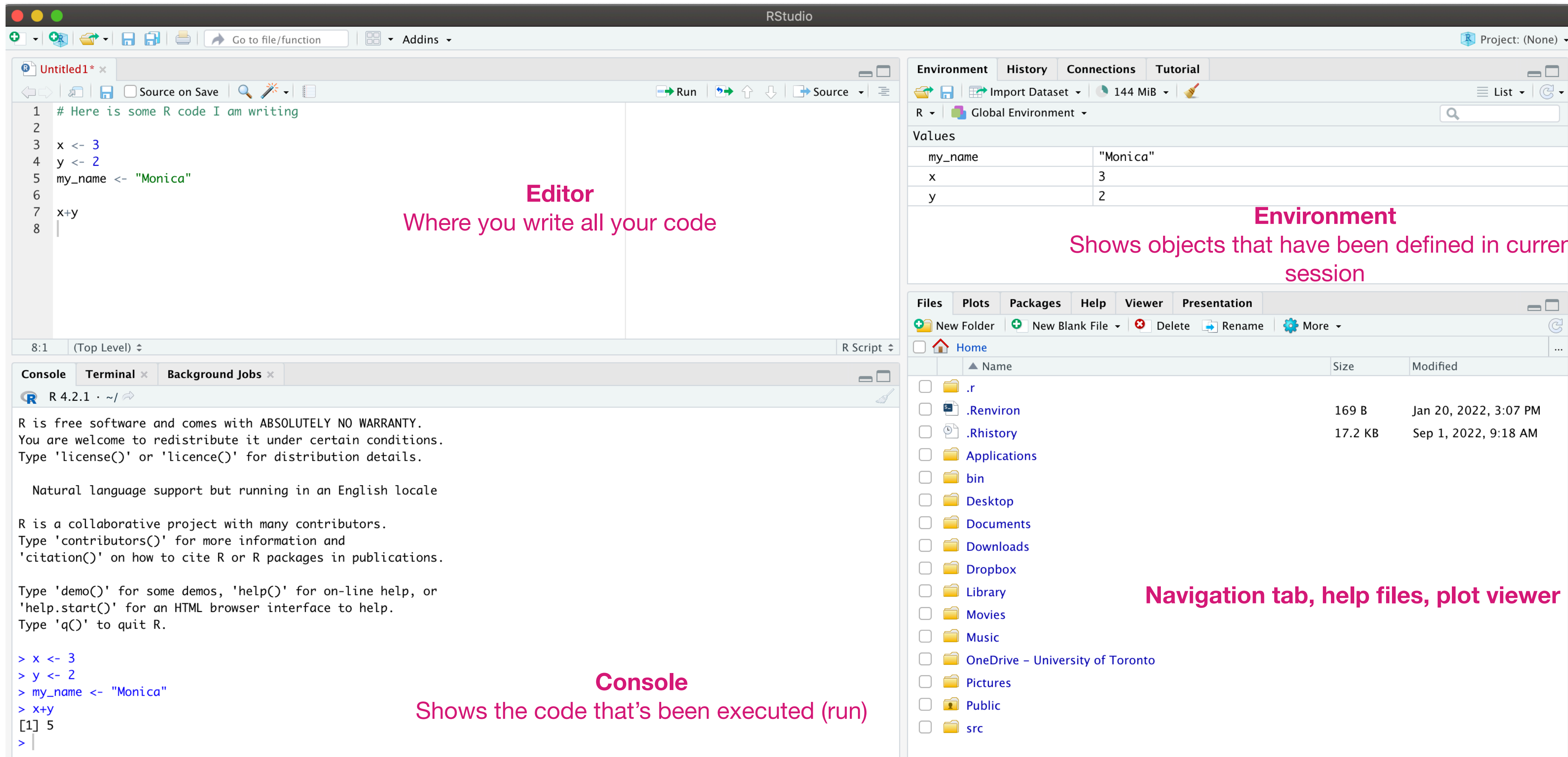
- R is a programming language for statistical computing and graphics
- Using R is like speaking another language (but you type it)
- You may have used other programs to do statistical calculations before (Excel, SPSS)
- With R you have to give the computer typed commands in order for it to do statistics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

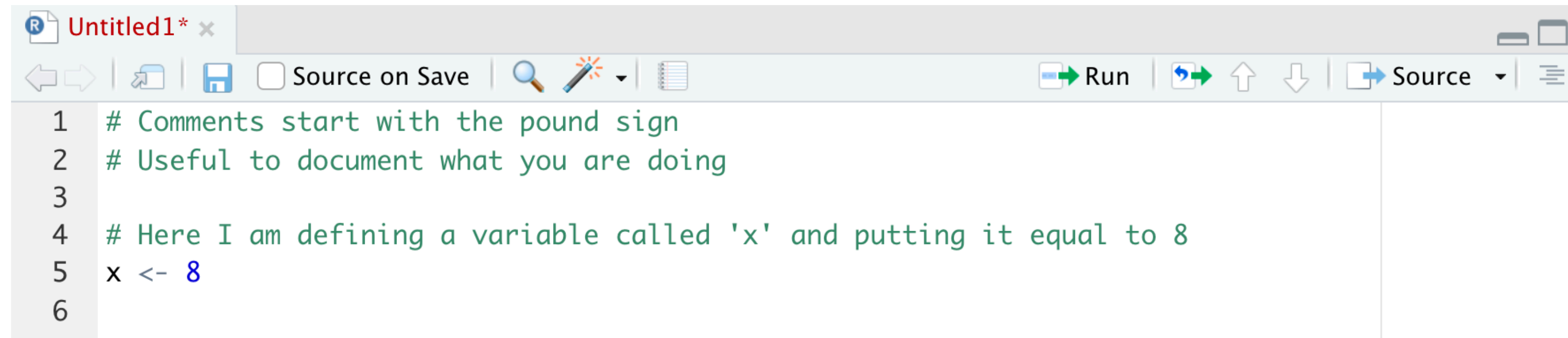




Demo: opening RStudio, opening and saving a R script

R code basics

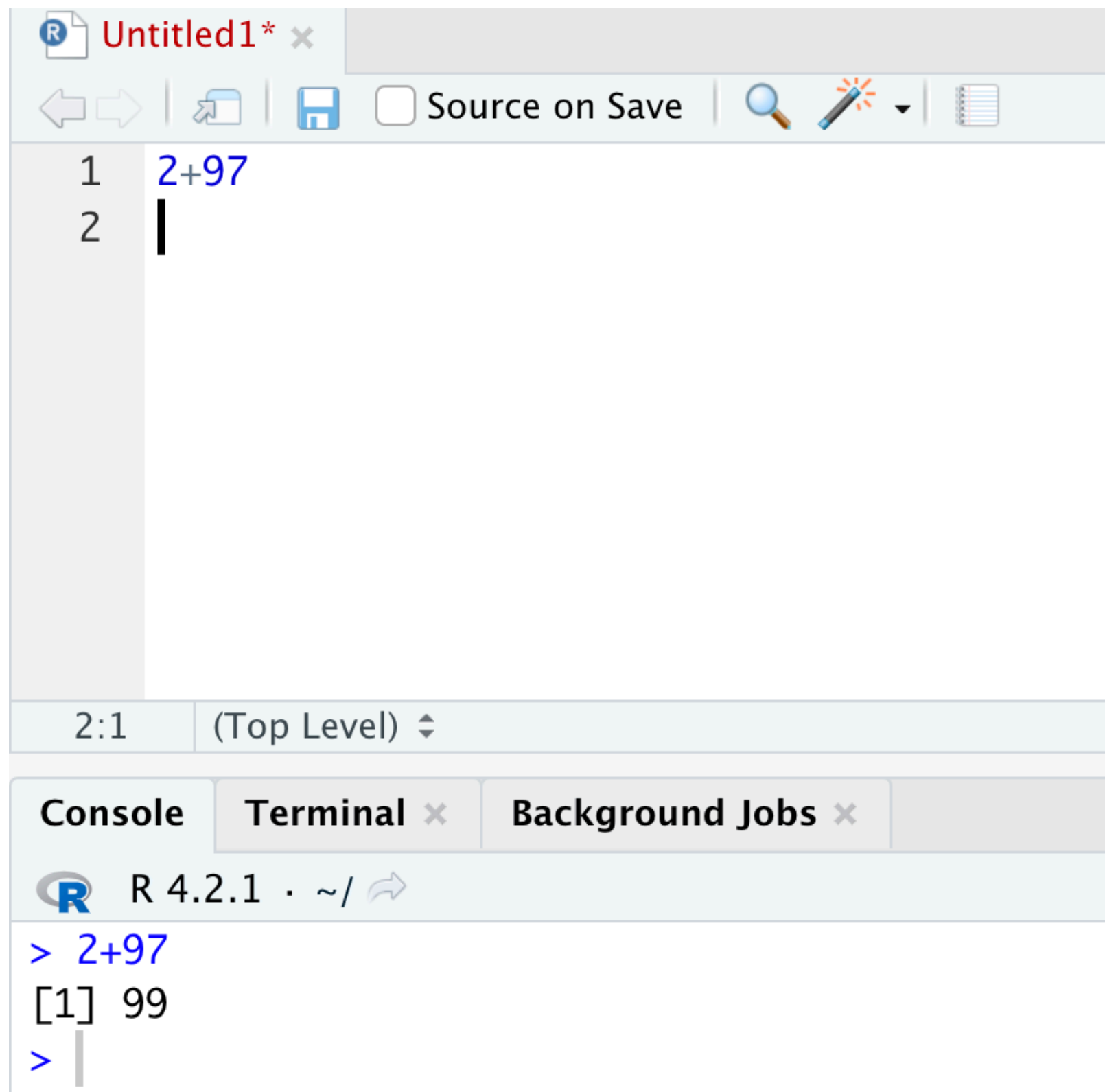
Code versus comments



The image shows a screenshot of an R script editor window titled "Untitled1*". The window has a toolbar with various icons including navigation, search, and execution. 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
```

R as a calculator

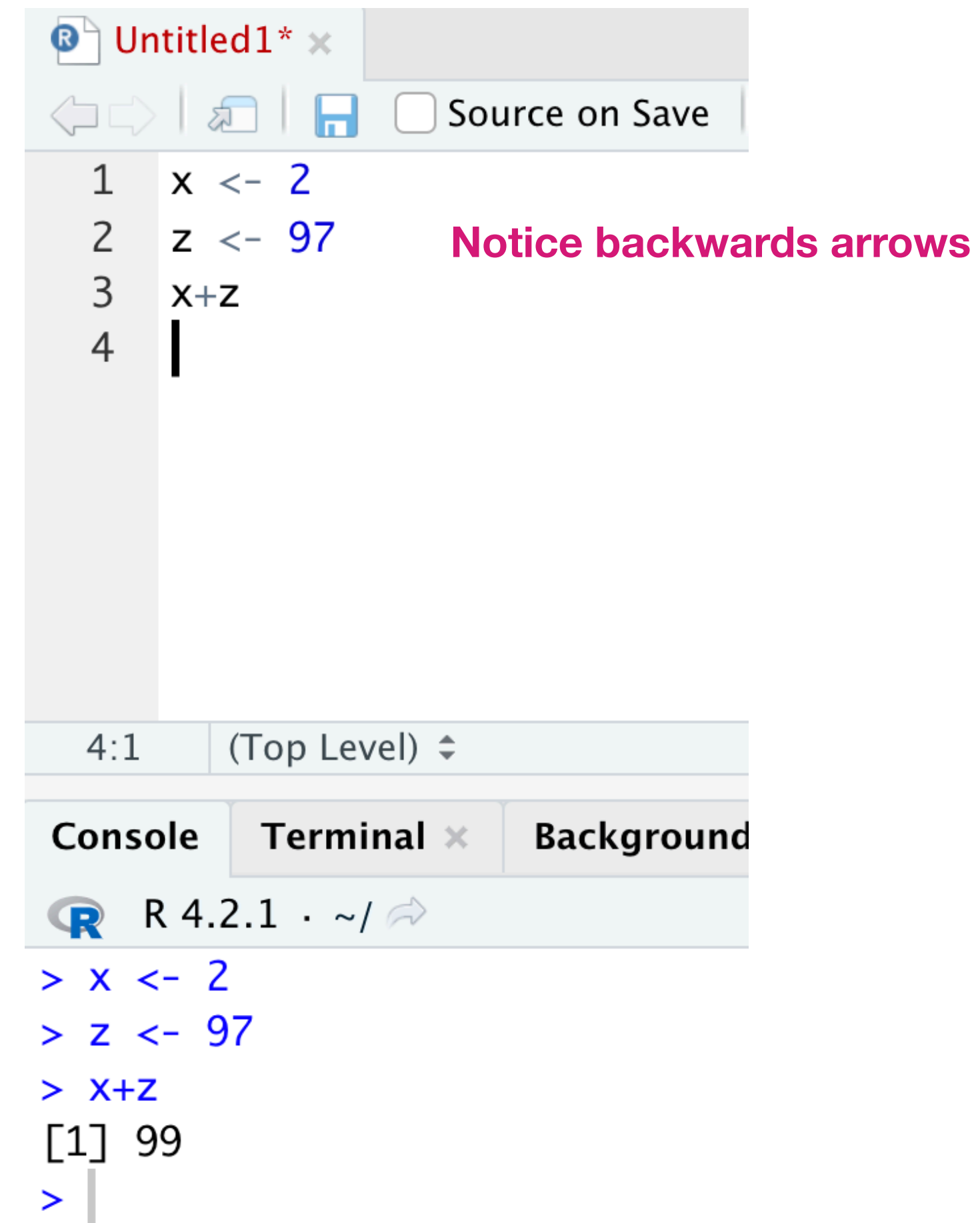


The screenshot shows the R IDE interface. The editor window contains two lines of code: `1 2+97` and `2 |`. The console window below shows the execution of the first line, resulting in `[1] 99`.

```
1 2+97
2 |

2:1 (Top Level)
Console Terminal x Background Jobs x
R 4.2.1 · ~/
> 2+97
[1] 99
> |
```

... versus defining objects



The screenshot shows the R IDE interface. The editor window contains 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 lines 1 and 2. The console window below shows the execution of these lines, resulting in `[1] 99`.

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

4:1 (Top Level)
Console Terminal x Background
R 4.2.1 · ~/
> 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)

```
← → | 📄 | 💾 |  Source on Save | 🔍 | ✨ | 📄 | ▶ Run
```

```
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

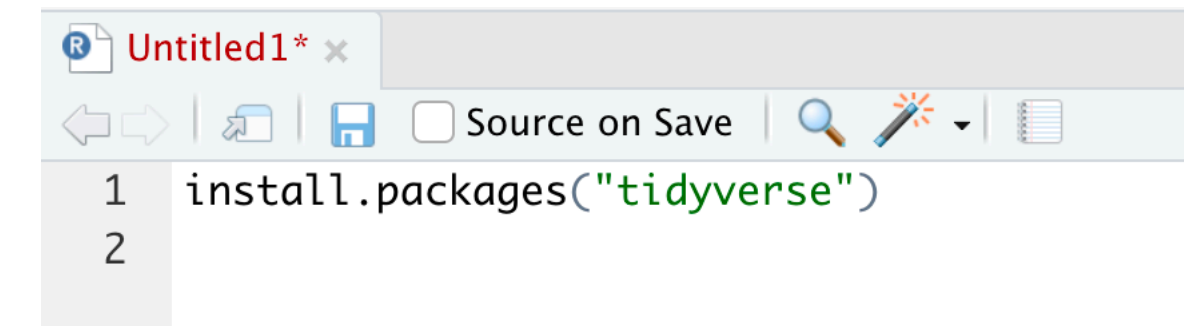
- People have written R Packages, which are add ons to base R that increase functionality
- Phone analogy
 - R/RStudio is a phone
 - Packages are apps
- One package that is very useful is the `tidyverse`
- Has graphing capabilities, tidyverse grammar (more later)



Installing packages

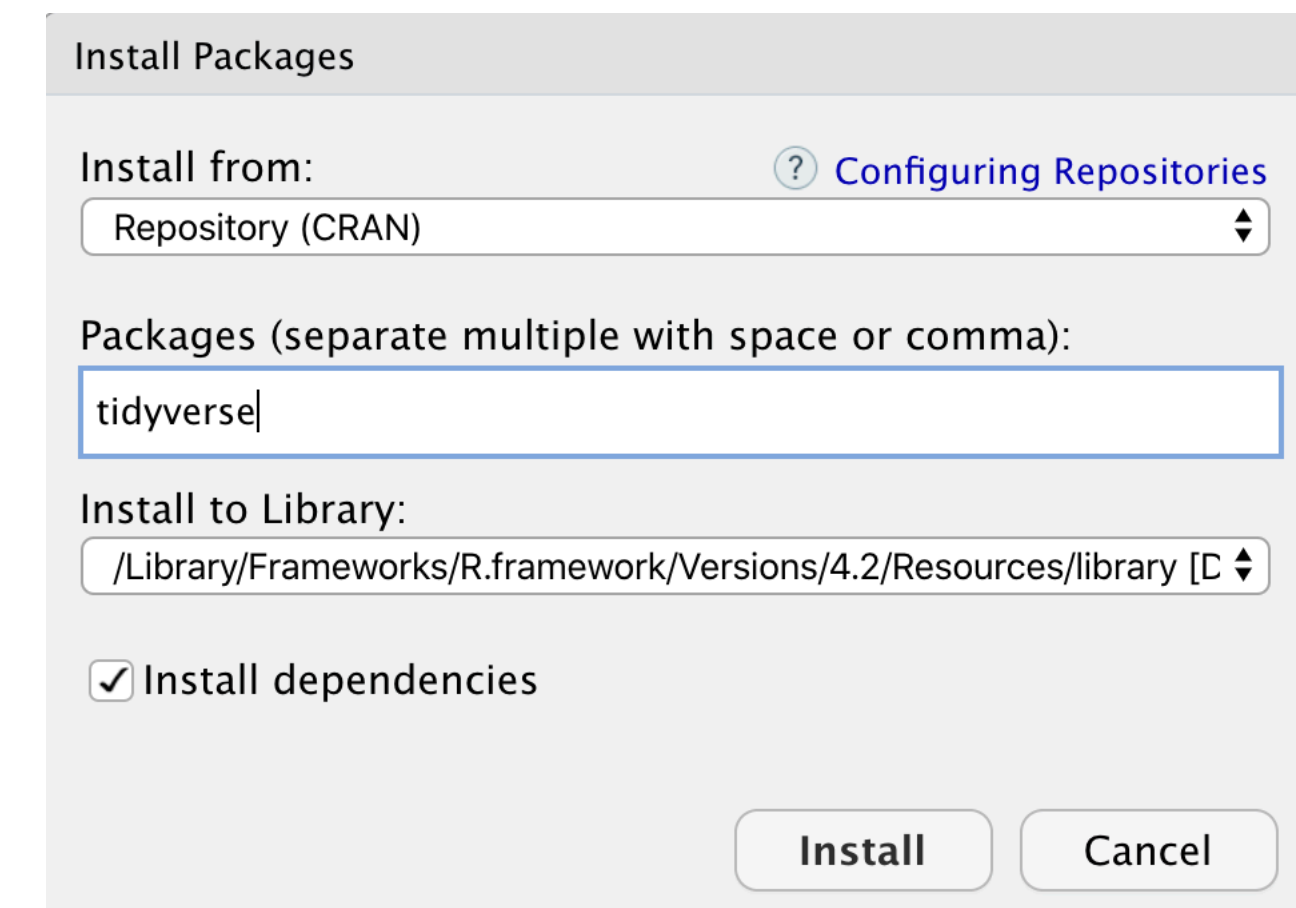
- Via code
- Or using menu (Tools -> Install packages...)
- Once a package is installed, don't need to do it again!

- 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

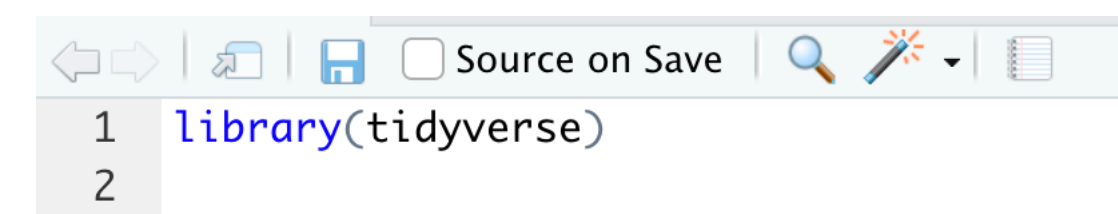
Install from: [? Configuring Repositories](#)
Repository (CRAN)

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

Need to be careful with file paths!

- 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>

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

The screenshot shows the R Studio Environment pane. At the top, there are tabs for Environment, History, Connections, and Tutorial. Below the tabs, there are icons for Import Dataset, a memory usage indicator (209 MiB), and a List button. The Environment pane shows the R Global Environment with a search bar. Under the 'Data' section, the variable 'd' is listed with '885 obs. of 5 variables'. A red circle highlights a small grid icon next to the variable name.

The screenshot shows the R Studio Data Viewer for the dataset 'd'. It displays a table with 10 rows and 5 columns. The columns are 'date', 'deaths_total', 'death_covid', 'death_covid_contrib', and 'death_unknown_missing'. The data shows daily counts from April 1st to April 10th, 2020.

	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