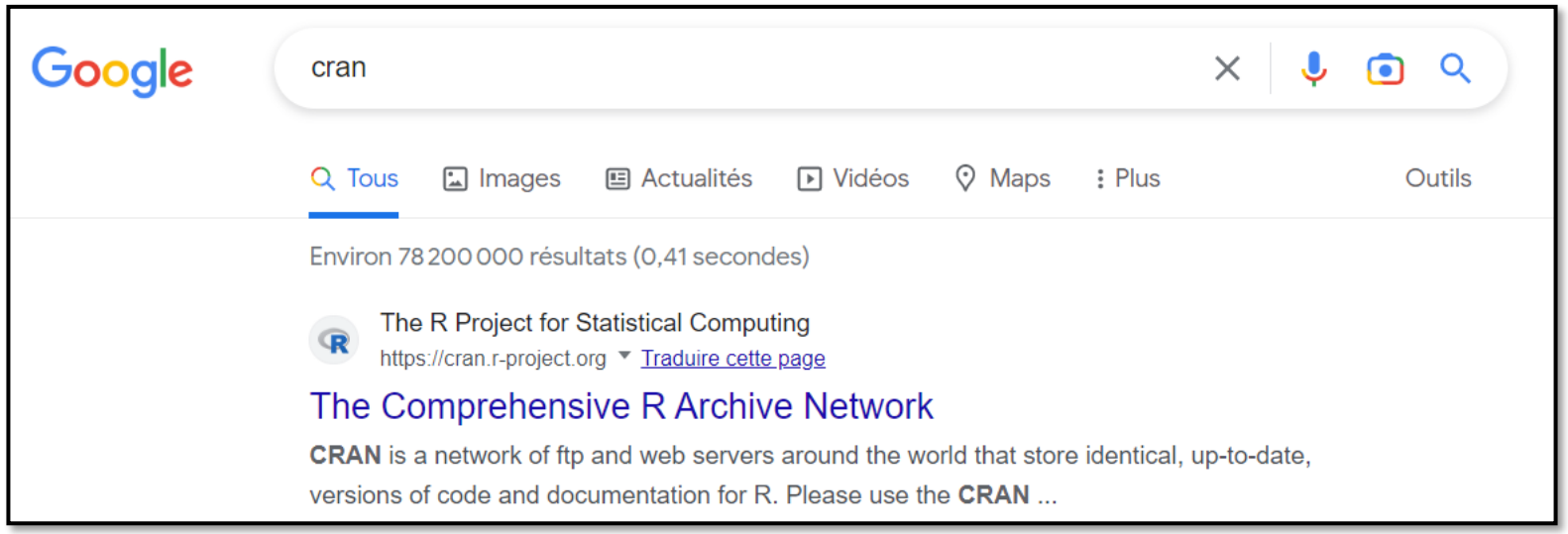
HGEN 47900 1 (Spring 2023)

Lab 0: R basics and tidyverse

Sebastien Bastide – March 24th, 2023

How to download R?

1. Go to the CRAN website (https://cran.r-project.org/).

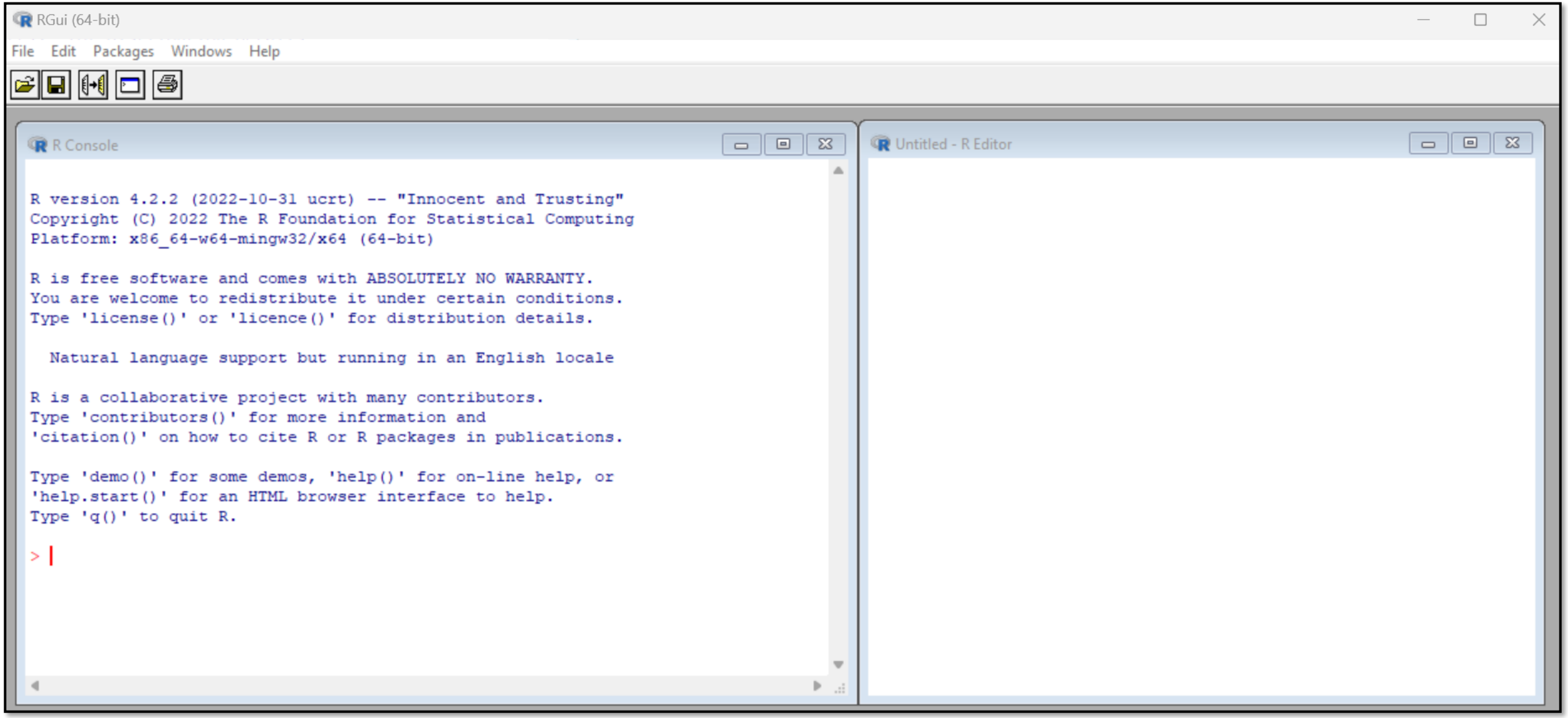


2. Download the binary/installer for your system (or follow the installation instructions):

- For Mac: https://cran.r-project.org/bin/macosx/
- For Windows: https://cran.r-project.org/bin/windows/
- For Linux (Ubuntu): https://cran.r-project.org/bin/linux/ubuntu

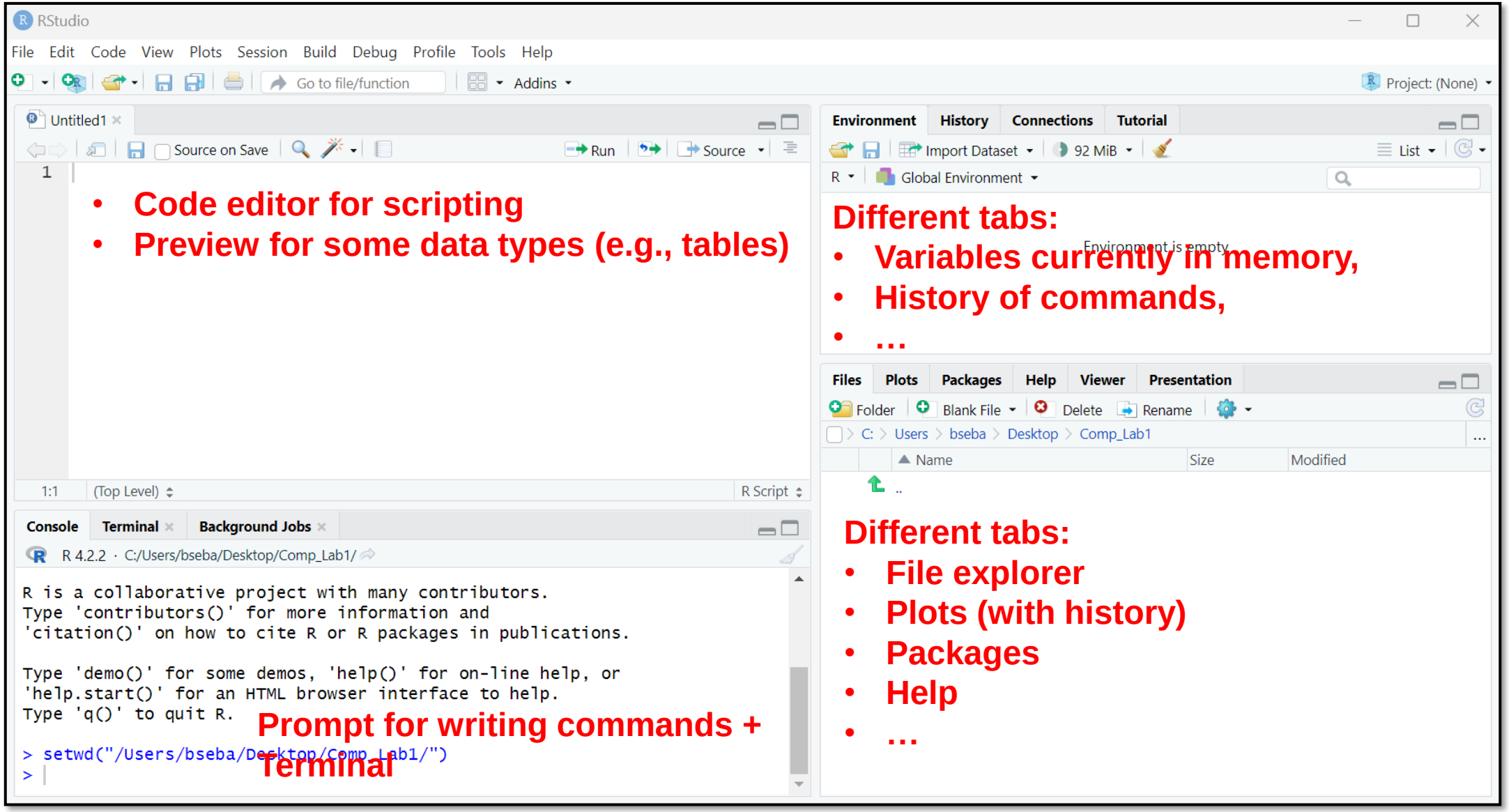
Why use RStudio?

R is a command line-based program, but the default code editor is ugly.



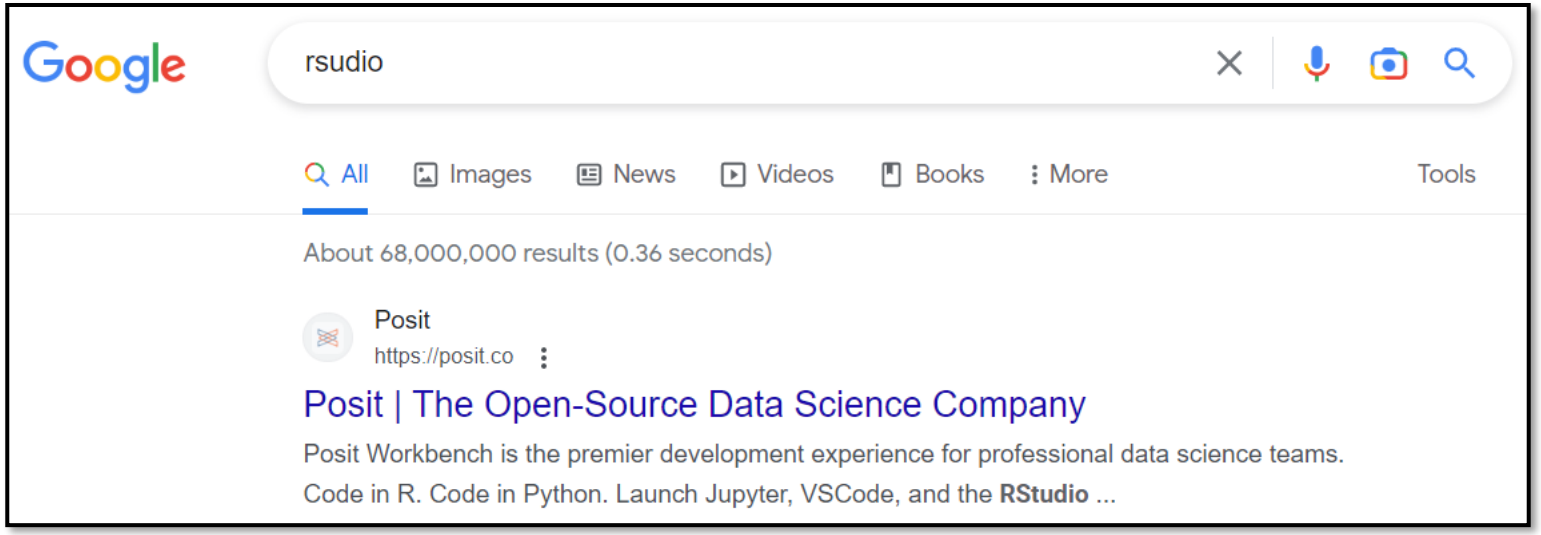
Why use RStudio?

Rstudio has a much better GUI and additional features.



How to download RStudio?

1. Go to the Posit website (https://posit.co/).



2. Download the RStudio binary/installer for your system (or follow the installation instructions):

- For Mac: https://download1.rstudio.org/electron/macos/RStudio-2022.12.0-353.dmg
- For Windows: https://download1.rstudio.org/electron/windows/RStudio-2022.12.0-353.exe
- For Linux (Ubuntu): it depends on your distribution (see https://posit.co/download/rstudio-desktop/)

OK, now what?

You can change the theme of RStudio: Tools > Global Options > Appearance
I like Cobalt for two reasons:

- It prevents my eyes from dying (the dark background helps in a light environment).
- It highlights specific elements of the R syntax and makes code easier to read.
- (Also, you are now a hackerman)

```r
# Create a function that sorts a vector vec using bubble sort
# Input:  a numeric vector
# Output: a sorted numeric vector
bubble_sort <- function(vec) {
  len_vec <- length(vec)
  swap <- TRUE
  # As long as there's been a swap in the previous step, re-iterate through the vector
  while(swap) {
    swap <- FALSE
    # Iterate through the vector and swap numbers to put them in ascending order
    for (i in 1:(len_vec-1)) {
      if (vec[i+1] < vec[i]) {
        smaller  <- vec[i+1]
        vec[i+1] <- vec[i]
        vec[i]   <- smaller
        swap <- TRUE
      }
    }
  }
  return(vec)
}

test_vec <- c(3, 6, 4, 7, 8, 6, 5, 3)
bubble_sort(test_vec)

```

Some shortcuts to remember

1. **`ctrl(Cmd) + Enter`**: **Runs current line of code or highlighted code.**
2. **`ctrl(Cmd) + Shit + N`**: Creates a new script.
3. **`ctrl(Cmd) + I`**: Reindent lines.
4. **`ctrl(Cmd) + Shift + A`**: Reformat code.
5. **`ctrl(Cmd) + Shift + C`**: Comment/uncomment lines.

Installing and loading R packages

R comes with a very large library of packages. Packages are typically hosted either:

- On the CRANrepository, and can be installed with:

  `install.packages("packageName")`

  e.g., **`install.packages("BiocManager")`**

- On the Bioconductor (https://www.bioconductor.org/) repository, and can be installed with

  `BiocManager::install("packageName")`

NOTE: here, "**`install`**" is function from the **`BiocManager`** package.

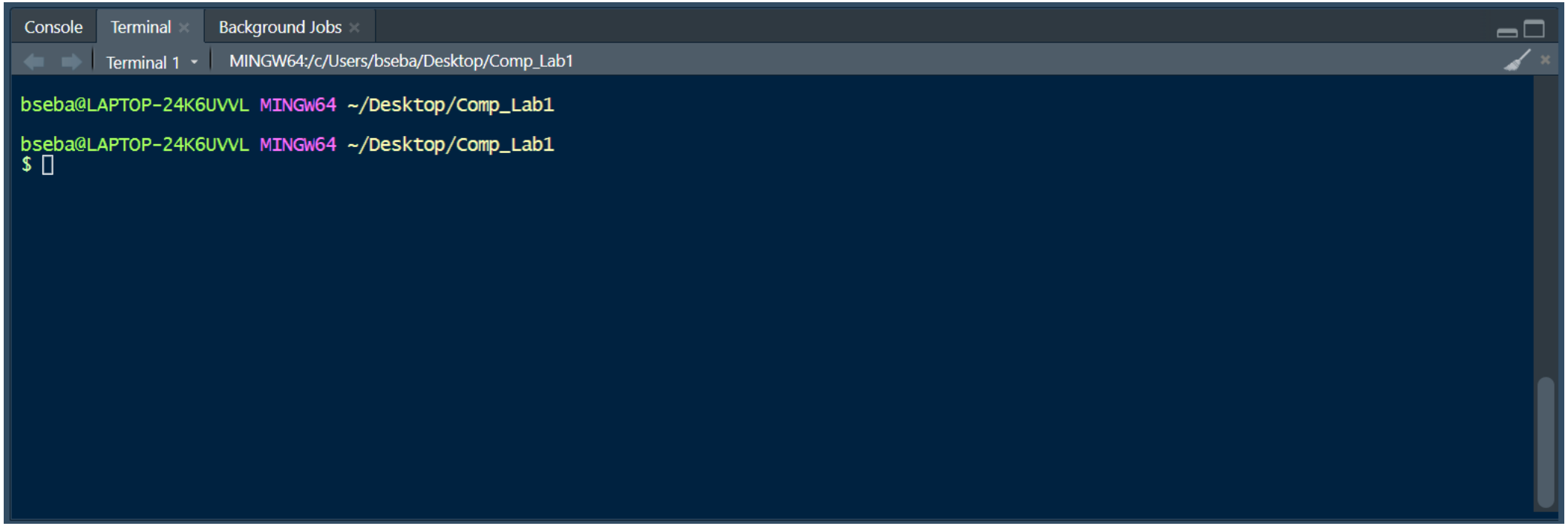
To use functions from packages, you normally first need to load them using

`library(packageName)`

However, you can use a function directly, without loading the entire package, by telling R where to find the function. This is done by adding "**`packageName::`**" in front of the function's name.

Using the RStudio terminal

RStudio comes with a terminal



On Windows, it uses Git Bash (a Unix shell) which includes a bunch of useful commands (**`cd`**, **`ls`**, **`mv`**, **`cp`**, **`rm`**, **`ssh`** , **`nano`** , **`vim`** , **`touch`** …). We'll learn how to use those

Just in case: Rtools

To install some packages on windows, you may need to install Rtools:
https://cran.r-project.org/bin/windows/Rtools/