

R

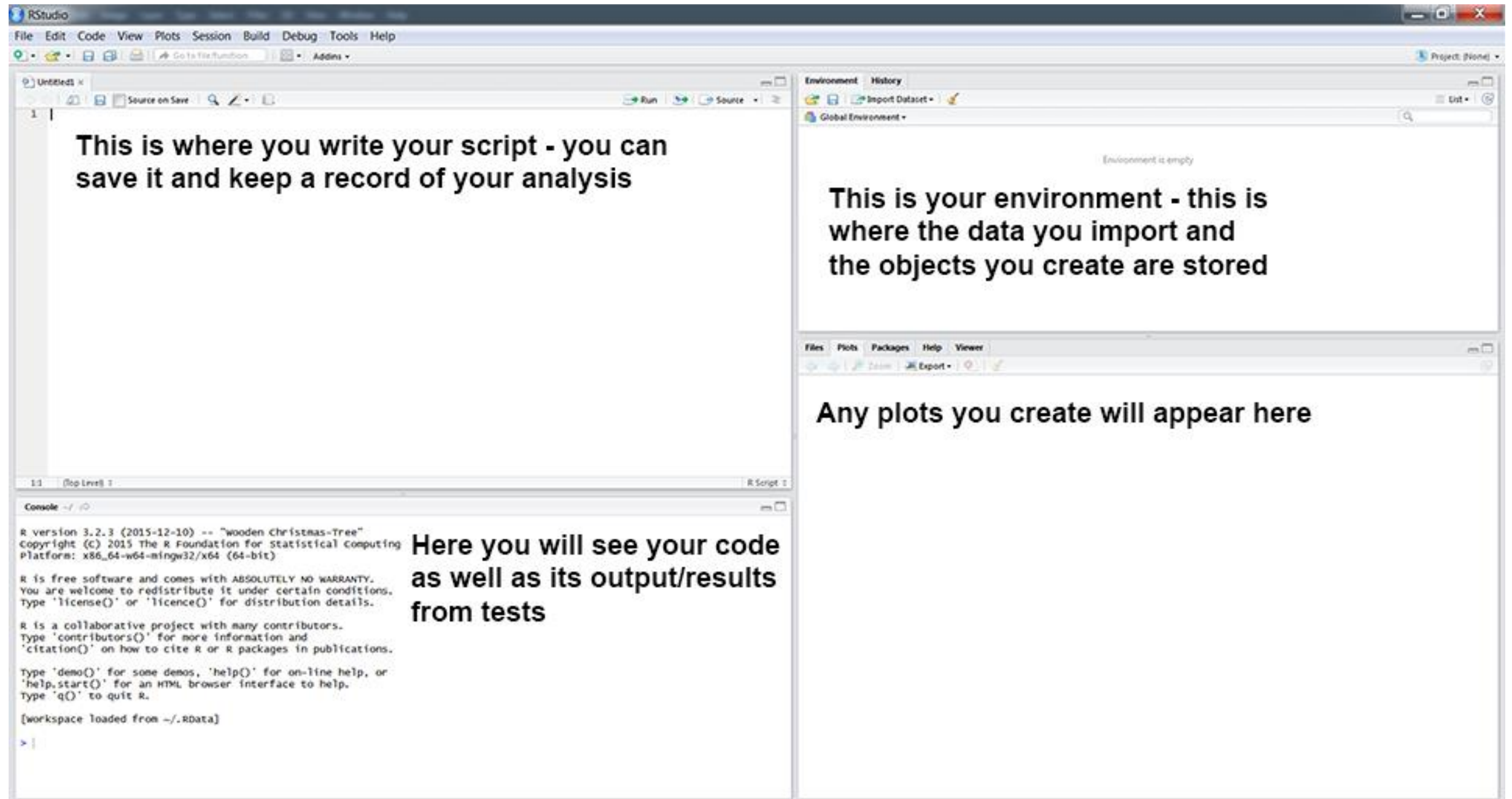
Step 0: Install R and RStudio

- First, you'll need to install R and RStudio on your computer.
- I've included some links below:
- <https://www.r-project.org/>
- <https://www.rstudio.com/products/rstudio/>
- Once you get your software installed and booted up, you should see something that looks approximately like the image below.

<https://cloud.r-project.org/>

<https://rstudio.com/products/rstudio/download/#download>

Open RStudio. Click on *File/New File/R script*.



Install Package / Library:

To install a package, type `install.packages("package-name")`.

You only need to install packages once, so in this case you can type directly in the console box, rather than saving the line in your script and re-installing the package every time.

Once installed, you just need to load the packages using `library(package-name)`.

Today we will be using the [dplyr package](#) to provide extra commands for formatting and manipulating data. (You will learn more about the powerful features of dplyr in [a later tutorial](#).)

Many data scientist programmers and statisticians use R to design tools for analyzing data and to contribute their codes as pre-assembled collections of functions and objects called packages. Each R package is hosted at <http://cran.r-project.org>.

Available R packages are listed here:

R Package	Function
<i>library()</i>	# List available packages to load
<i>library("package")</i>	# Load the package
<i>library(help="package")</i>	# List package contents
<i>detach("package:pkg")</i>	# Unload the loaded package "pkg"
<i>install.packages("package")</i>	# Install the package



Not all packages are loaded by default, but they can be loaded/installed on demand.

R stores user-defined objects in the workspace.

At the end of an R session, you can save a snapshot of the current workspace. The workspace reloads automatically the next time R starts.



Examples:

- *getwd()* - return working directory
- *setwd()* - set working directory



Understanding Functions of R:

- There are over 1,000 functions at the core of R, and new R functions are created all the time.
- Each R function comes with its own help page. To access a function's help page, type a question mark followed by the function's name in the console.

Getting Help in R:

Here are the commands to get help for some common functions in R:

Help Command	Function
<code>help.start()</code>	# Load HTML help pages into browser
<code>help(package)</code>	# List help page for "package"
<code>?package</code>	# Display short form for "help(package)"
<code>help.search("keyword")</code>	# Search help pages for "keyword"
<code>?help</code>	# Search for more options
<code>help(package=base)</code>	# List tasks in package "base"

- What does CRAN stand for ?
 - a. Comprehensive R Network
 - b. Computational R Network
 - c. Comprehensive R Archive Network
 - d. Computational R Archive Network

Which of the following commands is used to browse and access all help documents in R?

- a. `help.start()`
- b. `help(help)`
- c. `library()`
- d. `install.packages(help)`

Which of the following commands is used for loading an installed package in R?

- a. `load("package")`
- b. `library("package")`
- c. `install.packages("package")`
- d. `library()`

Summary

- R is a programming language developed as an alternative to the S language.
- R is available across all platforms—Windows, Mac, and Linux.
- R is most useful for statistical computation and visualization.
- R has a steep learning curve, and its working with large datasets is limited by the RAM size.
- R can be downloaded from either of the following websites:
 - o CRAN
 - o RStudio
- • The RStudio program can run on a desktop or through a web browser.

- R stores user-defined objects in the workspace by allowing the user to take a
- snapshot of the current workspace and by automatically reloading it the next time R starts.
- Each R function comes with its own help page.
- Not all packages are loaded by default but can be loaded or installed on demand.

- define your **working directory**.
- This is a folder on your computer where R will look for data, save your plots, etc. To make your workflow easier, it is good practice to save everything related to one project in the same place, as it will save you a lot of time typing up computer paths or hunting for files that got saved R-knows-where. For instance, you could save your script and all the data for this tutorial in a folder called “Intro_to_R”. (It is good practice to avoid spaces in file names as it can sometimes confuse R.) For bigger projects, consider having a root folder with the name of the project (e.g. “My_PhD”) as your working directory, and other folders nested within to separate data, scripts, images, etc. (e.g. My_PhD/Chapter_1/data, My_PhD/Chapter_1/plots, My_PhD/Chapter_2/data, etc.)
- To find out where your working directory is now, run the code **getwd()**. If you want to change it, you can use **setwd()**. Set your working directory to the folder you just downloaded from GitHub:

Assignment Operator

- There are two assignment operators i.e. <- or =.
- This operator means that “x gets a value”
- Create a scalar constant x with value 2, we type
- Difference between <- and = operator

<-	=
The <- operator can be used anywhere.	The = operator is only allowed at the top level.
For example, if we assign 2 -> a is perfectly valid.	For Example, if we assign 2 = a is confusing and raises an error.

- There are several number of operators in R programming
- List of operators in R:

Arithmetic Operators

- ✓ Addition
- ✓ Subtraction
- ✓ Multiplication
- ✓ Division
- ✓ Exponentiation
- ✓ Modulo

Relational Operators

- ✓ Greater than
- ✓ Greater than equal to
- ✓ Less than
- ✓ Less than equal to
- ✓ Equal to
- ✓ Not Equal to

Logical operators

- ✓ AND operator
- ✓ OR operator
- ✓ NOT operator
- ✓ Logical AND operator
- ✓ Logical OR operator