REACHOUT • TEACHOUT

ROTO Summer coding workshop: R basics and applied Workshop Team

Before the Workshop

We kindly ask that you complete at least the first three sections of this guide before the in-person workshop.

- **Section 1 - Setting up a RStudio server account**
- **Section 2 - Introduction to RStudio layout**
- **Section 3 - Installing packages**

If you have additional time, we strongly suggest that you go through the following

- Section 4 shows you how to upload data into RStudio. We will review this together, but it is super simple to do on your own.
- Section 5 shares a brief outline of the workshop topics
- Section 6 provides some additional resources and tips for learning to code in R.
- Section 7 provides a guide to install R to your personal computer if you need to in the future.

In total it should only take about 45 minutes to complete. Doing so will make sure that everyone is on the same page and is ready to go!

1. Setting up an RStudio Server account to use R and RStudio

To program in R, we will use both R and RStudio. Why do we need both? In brief, R is the core programming language and software engine, but it does not have a convenient user interface. Thankfully, programmers have developed RStudio - a free application that makes R software super easy and convenient to use!

For our workshop we will be using a web-tool called **RStudio Server** which already has both R and RStudio built in! So getting set up is as easy as making a free account!

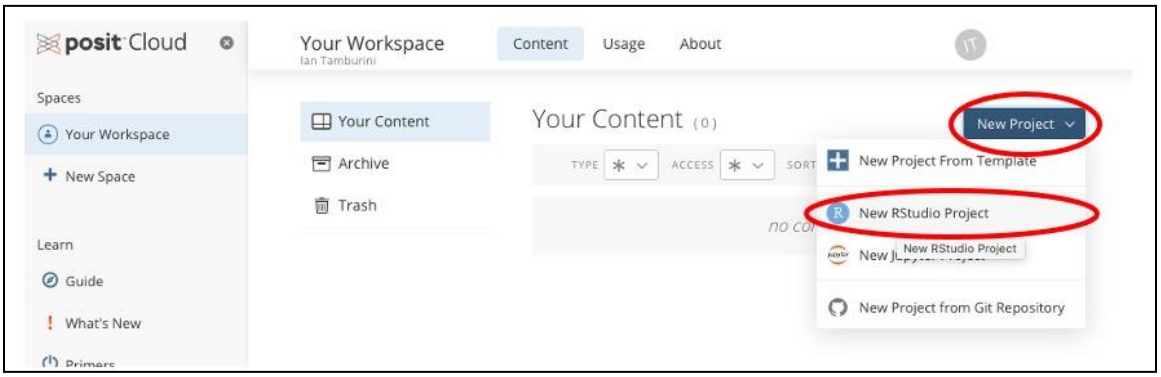
- Although RStudio Server is entirely sufficient for our workshop, in the future you may want to install R and RStudio onto your personal computer. At the end of this guide we include instructions to download both R and RStudio should you ever need to in the future. Of course, doing so is not required for the workshop.

1.1. Set up a free account for RStudio Server (required)

Go to the following website from **Poist** https://posit.cloud/ and click the blue button that says **Get Started**. On the next page under the free account column, click **learn more** and then **sign up**. Follow the onscreen instructions and be sure to record the email and password you use for your account.

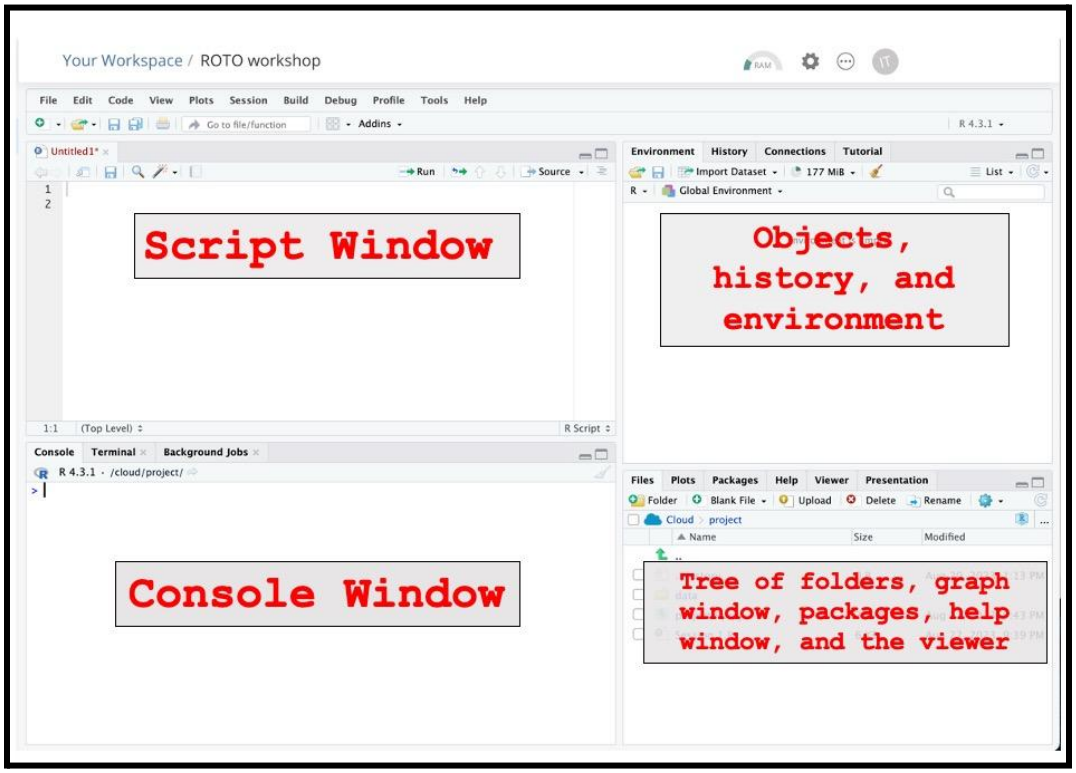
- If you have a Gmail account, you can also select the option to sign up using your Gmail!

- Once you are logged in you should see a screen similar to the image below. Click **New Project**, and in the drop down menu click **New RStudio Project**. In the next section we will have a quick look around!



2. Introduction to RStudio's layout

When you open RStudio, a screen appears with four panes in it. Let's walk through what each window is responsible for.



1) Script Window (top-left):

This is where you will write down your code so that you can run it in an organized manner. You will save the code that you write in this window as an R script (a file with a .R extension). If you are not seeing this window, at the top of the page click **File > new file > new R script** to open a fresh script.

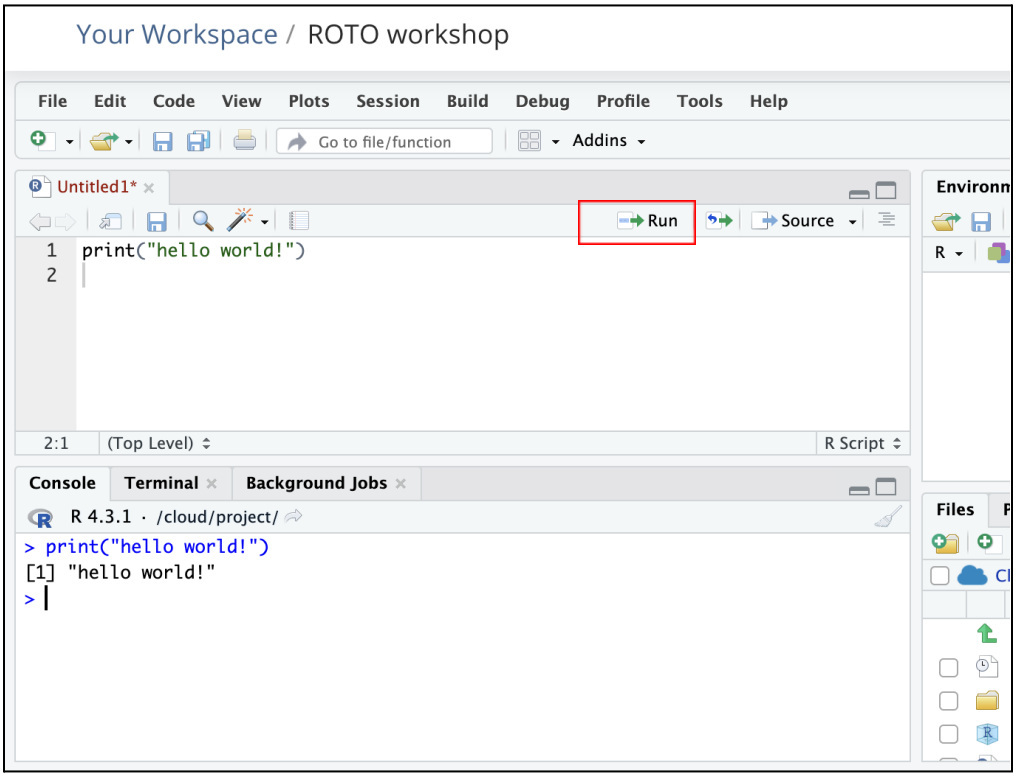
- Notice how the tab at the top of this window reads "Untitled1*". This is because you are currently writing in a blank, unsaved script.
- You can save your script by clicking the save button on the ribbon at the very top of the screen

- Once we begin writing code in our script, we can execute one line of code at a time by clicking on the **Run** button, OR by using the following keyboard shortcuts:
    - Mac - (command + enter)
    - Windows - (control + enter)

- **Type the following line of code, exactly as it appears below, anywhere in your scripts window. Then execute the line using the keyboard shortcut or the Run button. *What happens?***

```
print("hello world!")
```

… ***pay attention to what happens in the console window!***



2) Console window (bottom-left):

When we run/execute a line of code in our scripts window (by clicking **Run** or [Cmd + enter]), that code is essentially sent to the console window, indicating that the line of code is being executed. In fact you will see the code from your script appear in the console window!

You can also type code directly into the console and press enter to execute it.

- Often, after you execute a line of your code, R will produce a result that appears in the Console window, just like we saw when we ran **print("hello world!")**
- In addition to displaying executed code and results from executed code, the Console window will also show you if an error occurred upon executing a line of code. Errors appear as red text. (try misspelling print, and see what happens upon executing)

3) Environment, Objects and History window (top-right) **:**

This window has several tabs at the top. The most important tab in this window is the **Environment tab**. When we execute code, often we produce data or information that needs to be stored in memory to be accessed again later. In the Environment tab, we can conveniently view data that we have stored into the memory of our current R session. We broadly refer to these stored items as data **objects**.

- **Try executing the following code in the Scripts window, and watch what appears in the Environment window:**

```
year = 2023
```

4) Tree of folders, graph window, packages, help window, and the viewer (bottom-right). **:**

This window also has several tabs. The most important tabs are **Files** and **Plots**.

- **Files** lets us conveniently view the directory where files associated with our projects are located, such as data files and .R files (scripts).

- **Plots** displays graphs/plots/images and other graphical outputs that can be produced from executing code. Although the console can display results too, it is limited to only text outputs. Thus we rely on the Plots window to view graphical outputs.

- **Try executing the following code in the Scripts window, and watch what appears in the Plots window:**

```
hist(100)
```

3. Installing R packages

R packages are groups of functions and data sets that other R users have put together and shared with the R community. These packages increase the number of R functionalities and are key for some types of calculations and operations.

During this R coding workshop, we will be using functions from various packages which we ask you to install before the workshop.

3.1. Installing the tidyverse package

Thankfully, for our workshop we can install our packages with one simple line of code. **Type the following code into the console window of your R session, and press enter on your keyboard:**

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

- Note:
  - There are no spaces in this line of code.
  - Quotes "" around the name of our package "tidyverse" are necessary for the installation to work
  - The text is also case-sensitive; all letters must be lower-case for the command to work
  - You will probably see a bunch of status text appear in the console upon executing. Let it run to completion!

When the installation is finished, the console will finally output a message indicating that it is done. Probably something like:

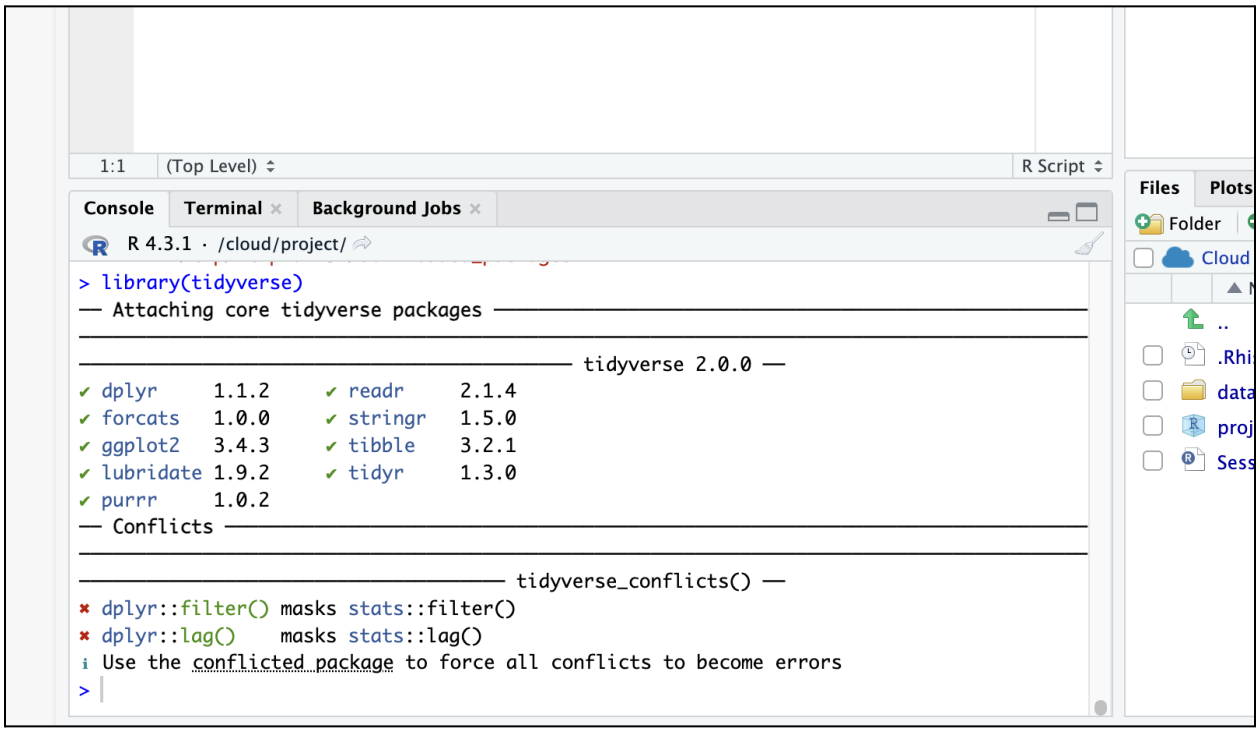
```
* DONE (tidyverse)
The downloaded binary packages are in
        /var/folders/9_/kzpplh157kg75ry1yjmyd1880000gn/T//Rtmp9o8I7K/down
        loaded_packages
```

3.2. Check that you can load the tidyverse package

Now test that you can load this library. **Type the following code into the console window of your R session, and press enter on your keyboard (no quotes this time around tidyverse):**

```
library(tidyverse)
```

If everything worked, the console will output a message that looks something like this:



The tidyverse package basically contains multiple sub-packages that are handy for data science. These sub-packages are the individual items seen in the output above (**dplyr**, **forcats**, **ggplot2**, **readr**, etc…). One smart cookie had the great idea to make them all accessible in one library called the tidyverse. This is convenient, because instead of having to load every individual library, you can just load the one tidyverse package. You can learn more about tidyverse and its sub-packages here: https://www.tidyverse.org/

REACHOUT • TEACHOUT

4. Downloading the data for our workshop and importing into RStudio Server

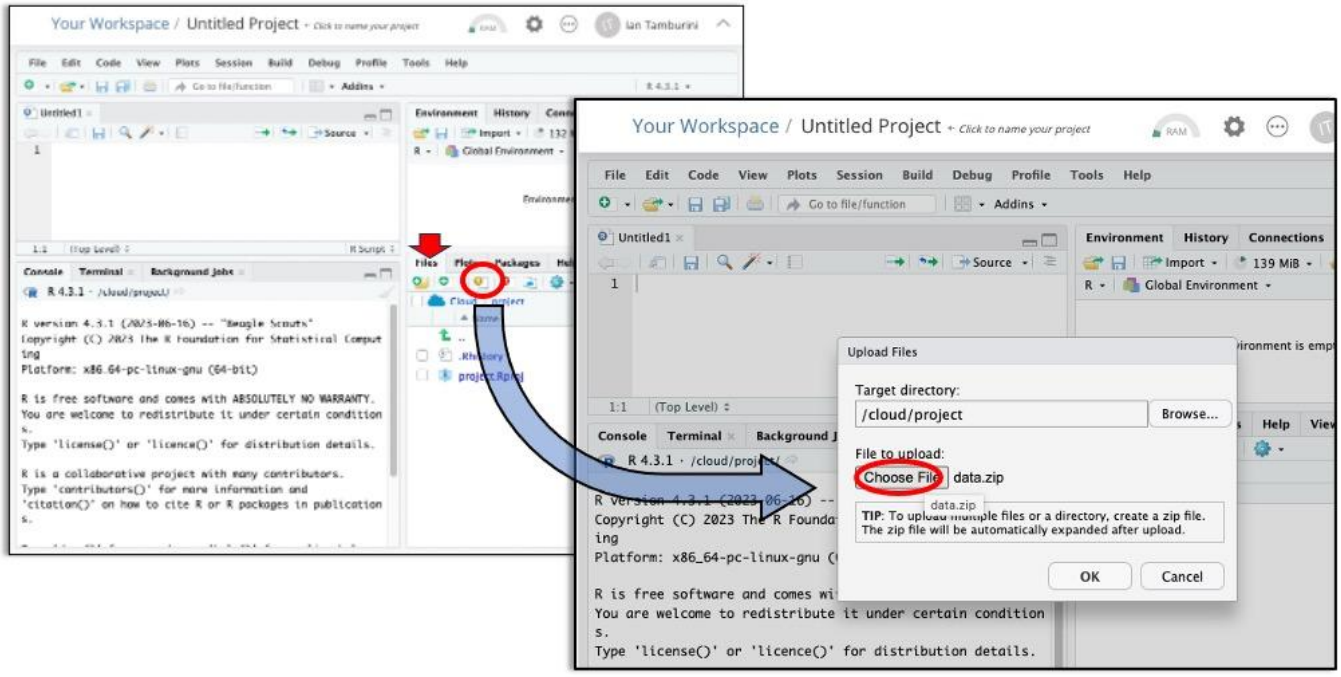
1) First, head to the following Google-drive link: https://drive.google.com/open?id=1_5a-cJ-1xMuCGiMWsxRdlMB3Jfqq0OR4&usp=drive_fs

   - Download the file to a convenient location on your computer, like the Desktop. The downloaded file should be called **data.zip**

2) Return to your RStudio Server project page (https://posit.cloud/) and open the project which was created earlier in section 2 of this tutorial.

3) Click the **upload button** under the Files tab of the bottom-right window (see image below for reference)

   - A new window will appear. Click **choose file** and navigate to where you saved data.zip, and select the file. Then click OK



You should now see a folder called data under your files tab in the bottom-right window!

5. Outline of our workshop

This R workshop will be divided into several sections. Our goal is to get you familiar with some of the core principles of data analysis in R. We plan to first cover some general R fundamentals and then focus on how to work with the most common data structure in R, the data frame (basically a glorified data table). We will learn how to analyze our data by producing plots.

Session 1:

- Getting oriented in RStudio
- Common data structures in R:
    - One-dimensional data: vectors
    - Two-dimesional data: matrices and **data frames**

- Principles in editing, subsetting, and performing calculations on these data structures

Session 2:

- Loading data from an outside source into R
- Manipulating data frames to extract specific information
- Practical examples of calculations and operations on data frames
- Reshaping data
- Combining separate data frames
- Basic plotting

Session 3:

- More advanced plotting
- Open work:
    - Choose from several data sets and work in small groups to analyze these data and generate plots (with the support of workshop instructors)

6. Additional Resources

Why is R so great?

The reason that programming in R is so popular is it provides the flexibility to do whatever you want. While R was initially designed for statistical analysis in the 1990s, it has since developed significantly to be able to serve data science applications across an extremely wide range of domains. R has a vast ecosystem of packages contributed by the community that serve diverse disciplines including bioinformatics, machine learning, economics and finance, and countless others. This ecosystem allows you to quickly access pre-built functions and tools to solve specific problems. It's no wonder that programmers skilled in R are in demand among tech companies and the healthcare industry (woz-u.com, 2021).

R for data science (tutorial link here):

This tutorial is a fantastic and easy-to-follow crash course to learn data science basics in R. It covers essential principles and skills - data importing, tidying, transforming and plotting. Following this tutorial in your own R session will be highly impactful in your own learning, and all said and done is not a big time investment considering the ***huge*** reward in learning. **Very well worth your time and cannot recommend this resource enough!**

Keep calm and Google on!

Consulting online resources is a necessary part of coding and improving our programming skills. When we run into errors, or simply have forgotten syntax for code, us programmers ***routinely*** use Google when coding. Would you believe me if I said that **the more advanced you become in programming the *more* you will use Google** and the web as a resource? It's very true!

When you are web-searching to troubleshoot and develop code, you will frequently encounter community forums (such as StackOverflow) where users post issues they are facing, and other members of the community help and provide solutions. It will shock you how frequently others have encountered and solved the very same issues you are facing.

7. Installing both R and RStudio to your personal computer

7.1. Installing R to your personal computer

R is available through the webpage of The Comprehensive R Archive Network (CRAN). The top section of the web provides the links to install R for Linux, macOS, and Windows.

The instructions below are extracted from the RStudio Education Github page, and they provide specific instructions to download R for the three operating systems.

7.1.2 Windows

To install R on Windows, follow these simple steps. First, click on the "Download R for Windows" link. Next, click on the "base" link. Then, select the first link at the top of the page. This link will show the current version of R available for download. The installer program will download automatically, and you can run it to install the latest version of R for Windows. Follow the installation wizard that appears to complete the process.

7.2.3. macOS

To install R on a Mac, simply click on the "Download R for Mac" link, and then select the appropriate package link (the most current release). An installer will guide you through the easy installation process, giving you the option to customize your installation if desired. However, the default settings will be sufficient for most users.

Note that the latest version (R-4.3.1) has a version for Apple silicon (M1/M2) Macs and another version for older Intel Macs. Make sure to use the link that suits your computer.

1.2.4. Linux

Many Linux systems have R preinstalled, but if yours is outdated, it's best to get the latest version. To do this, visit the CRAN website and download the files to build R from the source. Look for the "Download R for Linux" link and navigate to the specific Linux version you want to install.

7.2. Installing RStudio to your personal computer (not required)

If you're looking to write in R, RStudio is a helpful application that you can use on Windows, Mac OS, or Linux. The user interface remains consistent across all platforms. You can easily download RStudio for free by clicking the "Download RStudio" button and following the simple instructions.