# Introduction to R and RStudio

## Names

There are a few different names involved:

- **R** is a coding language for statistics and data analysis
- **RStudio** is a software interface for writing and running R code
- **Posit** is the name of the company that makes RStudio
- **posit.cloud** provides a way of using RStudio in your web browser

LINK TO INSTALLING R APPENDIX. You can install R and RStudio on your own computer for free and do things that way, but using posit.cloud simplifies things immensely.

## The general workflow

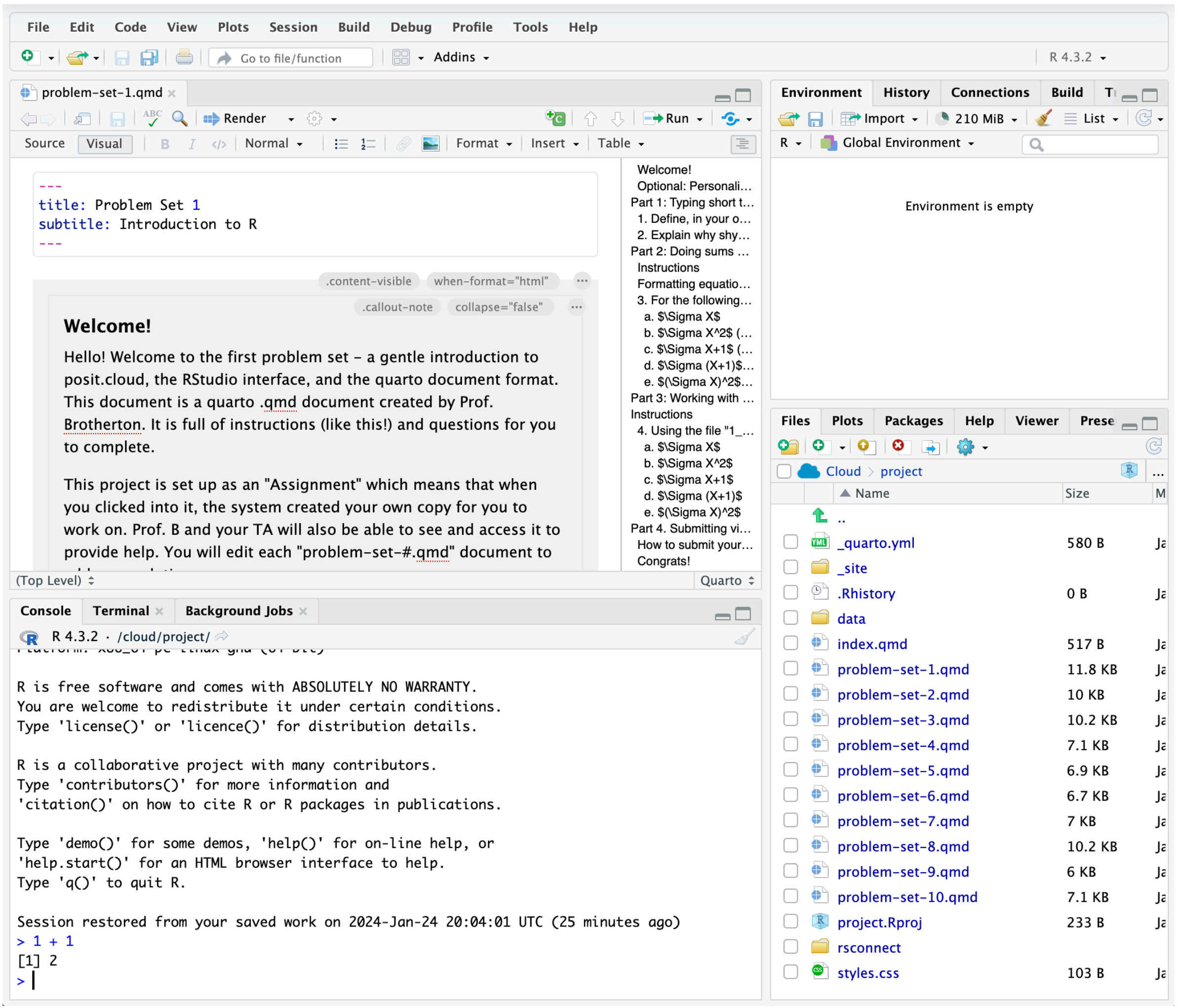
R is a programming language well-suited to interactive data exploration and analysis. It is widely used in social science research.

It might seem daunting if you've have no experience with coding, but the basic idea is that you have some data, like you are familiar with from a regular Excel or Google Sheets spreadsheet, and you perform operations on your data using functions a lot like you would in Excel/Sheets. For example, you might compute an average in Excel/Sheets by typing `=AVERAGE(A1:A10)` in a cell of the spreadsheet. In R you might type `mean(my_data$column_a)` in a code file. The specifics of the function names are different, but the basic idea is the same.

### Separation of data and code

A major difference between working with data in Excel vs. R is the separation of data from code. Rather than writing functions to manipulate or analyze data directly in your spreadsheet, code is written in a separate code file, which references **but does not modify** the source data file (unless you tell it to).

## RStudio Interface


File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
R 4.3.2
problem-set-1.qmd
Render
Run
Source Visual
Normal
Format Insert Table
(Top Level)
Quarto
Console Terminal Background Jobs
Environment History Connections Build
Import 210 MiB List
R Global Environment
Files Plots Packages Help Viewer Prese
Cloud > project

```
---
title: Problem Set 1
subtitle: Introduction to R
---
```

.content-visible when-format="html"

.callout-note collapse="false"

**Welcome!**

Hello! Welcome to the first problem set – a gentle introduction to posit.cloud, the RStudio interface, and the quarto document format. This document is a quarto .qmd document created by Prof. Brotherton. It is full of instructions (like this!) and questions for you to complete.

This project is set up as an "Assignment" which means that when you clicked into it, the system created your own copy for you to work on. Prof. B and your TA will also be able to see and access it to provide help. You will edit each "problem-set-#.qmd" document to

Welcome!
Optional: Personali…
Part 1: Typing short t…
1. Define, in your o…
2. Explain why shy…
Part 2: Doing sums …
Instructions
Formatting equatio…
3. For the following…
a. $\Sigma X$
b. $\Sigma X^2$ (…
c. $\Sigma X+1$ (…
d. $\Sigma (X+1)$…
e. $(\Sigma X)^2$…
Part 3: Working with …
Instructions
4. Using the file "1_…
a. $\Sigma X$
b. $\Sigma X^2$
c. $\Sigma X+1$
d. $\Sigma (X+1)$
e. $(\Sigma X)^2$
Part 4. Submitting vi…
How to submit your…
Congrats!

```
R 4.3.2 · /cloud/project/

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.

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.

Session restored from your saved work on 2024-Jan-24 20:04:01 UTC (25 minutes ago)
> 1 + 1
[1] 2
>
```

Environment is empty

| Name | Size | M |
|---|---|---|
| .. | | |
| _quarto.yml | 580 B | Ja |
| _site | | |
| .Rhistory | 0 B | Ja |
| data | | |
| index.qmd | 517 B | Ja |
| problem-set-1.qmd | 11.8 KB | Ja |
| problem-set-2.qmd | 10 KB | Ja |
| problem-set-3.qmd | 10.2 KB | Ja |
| problem-set-4.qmd | 7.1 KB | Ja |
| problem-set-5.qmd | 6.9 KB | Ja |
| problem-set-6.qmd | 6.7 KB | Ja |
| problem-set-7.qmd | 7 KB | Ja |
| problem-set-8.qmd | 10.2 KB | Ja |
| problem-set-9.qmd | 6 KB | Ja |
| problem-set-10.qmd | 7.1 KB | Ja |
| project.Rproj | 233 B | Ja |
| rsconnect | | |
| styles.css | 103 B | Ja |

RStudio is the interface we'll use to write and run R code and see its output. The basic interface has 4 panels, each with a few tabs:

- Top-left: Code editor / data viewer
  - Open, edit, and save code documents
  - Execute code within files
  - View data
  - You can have multiple 'tabs' open at once,
- Bottom-left: R console
  - You can type code directly and run it by pressing enter.

  - You won't be saving your code as a document like when you type in in the editor, so this is useful for testing something simple out

- Top-right: Environment
  - As you execute code you may be creating objects like sets of numbers of data.frames. Those objects will appear here.
  - You can click the name of some objects, like data.frames, and it will open a view of the data as a tab in the editor pane
- Bottom-right: Files/folders, Plots, Viewer, help window
  - You can navigate the file tree

### Running code

Writing some code in an .R document does not cause it to be executed automatically. You need to run the code yourself. There are several ways of doing so.

## Additional packages

The R language has many functions built in. Generally speaking, you can find a way to do pretty much anything you would like to do using just 'base' R.

However there are many common tasks that are a bit tedious or unintuitive to do using base R. One of R's strengths is how extensible it is: anyone can write their own functions, turn the code into an R package, and make that package available to other R users.

### Tidyverse

Actually, the tidyverse package is a container for multiple individual packages. The whole family of tidyverse packages are written with a consistent syntax and logic.

### Installing packages

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

install.packages("lme4")
```

Packages only need to be installed on your system once (or once per project in posit.cloud, since every cloud project represents a brand new virtual system).

## Using installed packages

If you are just using one function from a package as a one-off, you can use the double-colon `::` operator in the form `package::function()`, i.e.

```
lme4::lmer(...)
```

If you will be using a package's functions repeatedly, it can be preferable to activate the entire package using the `library()` function.

```
library(tidyverse)

library(lme4)

lmer(...)
```

Note that a package only needs to be installed once on your system (or in a new posit.cloud project), but if you are using the `library()` method to activate the package, it must be done every time you have a new 'session' in R. It is good practice to include all `library(...)` calls together near the top of your code file.