# Running R

# Running R online

Go to https://r.datatools.utoronto.ca.

You should see this:

The 2i2c JupyterHub for University of Toronto
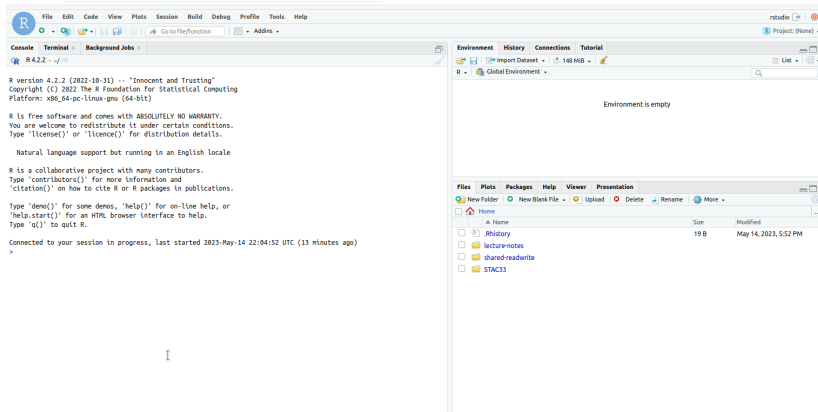
UNIVERSITY OF TORONTO JUPYTERHUB

Operated by: **2i2c** | Funded by: **University of Toronto** | Designed by: **2i2c**

**After logging in, open:** ○ Jupyter Notebook  ◉ RStudio  ○ JupyterLab

Log in to start

If you do, find the line that says "after logging in, open" and click

# How R Studio looks



If you are already signed in with UTorID and password, you might come directly here. (If you look at the URL that you are now at, you'll see that it has your email address in it.)

# Projects

- Each user has a "workspace", a place where all your work is stored.
- Within that workspace, you can have as many Projects as you like.
- To create a new Project, click on the blue New Project button.
- I recommend having one project per *course*.
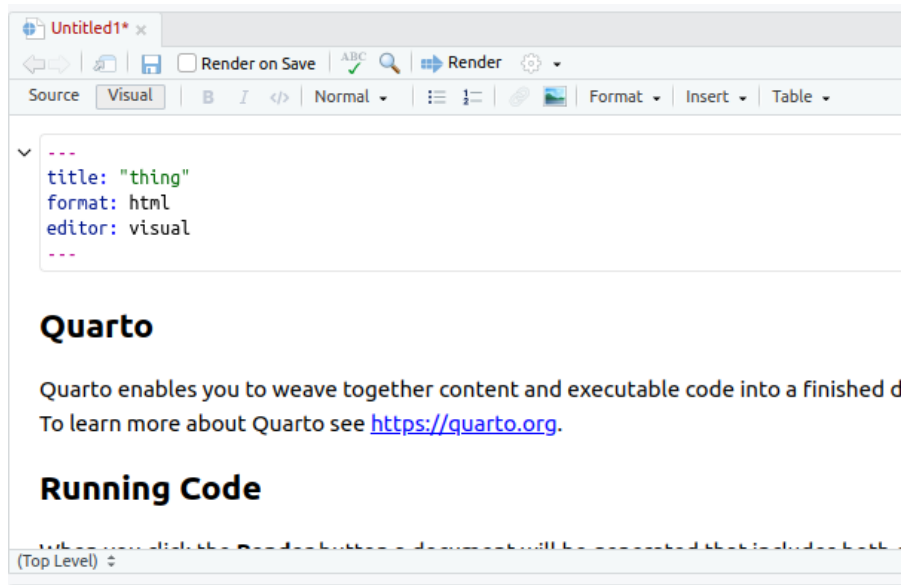- R Studio restarts where you left off.

# Make a new project

- Call it what you like. Mine is called `thing`:
- Select:
    - File,
    - New Project,
    - New Directory,
    - New Project (again),
    - give it a name and click Create Project.
- You see the name of your new project top right.

# R Notebooks

- At left of previous view is Console, where you can enter R commands and see output.
- A better way to work is via "Quarto Documents". These allow you to combine narrative, code and output in one document.
- Data analysis is always a story: not only what you did, but why you did it, with the "why" being more important.
- To create a new Quarto Document, select File, New File, Quarto Document. Give it a title. This brings up an example document as over.

# The template document



```
---
title: "thing"
format: html
editor: visual
---
```

## Quarto

Quarto enables you to weave together content and executable code into a finished d
To learn more about Quarto see https://quarto.org.

## Running Code

# About this document

- It begins with a title (that you can change).
- Most of the document is text (narrative).
- Pieces beginning with {r}, in grey, are called code cells (code chunks). They contain R code.
- Run code cells by clicking on the green "play button" at the top right of the first cell. This one does some very exciting arithmetic.

# After running the code chunk

```r
1 + 1
```

```
[1] 2
```

# Making our own document 1/2

▶ Create another new document. Give it a title of "Some random normal data", and click Create. When the document opens, delete the template that it gives you (leaving only the five or so lines that begin and end with ---).

▶ You will probably be sitting on a line that says (greyed out) Heading 2. On this line, type **Packages** (which you'll see big and bold like a title) and hit Enter a couple of times. At the top of the window, on the line with Visual, Format, Insert, and Table, it should now say Normal (which is normal text). If you want a heading, click the arrow to the right of Normal and select one. Header 2 is a good one for section titles in your document.

# Making our own document 2/2

▶ Make a new code chunk by clicking Insert (at the top of the notebook window) and selecting Code Cell and R. Inside that cell, type `library(tidyverse)`.

▶ Below that, make another "Header 2" and put "Random normal data" on that line.

▶ Make another new code cell below that, and insert two lines of code: `z <- rnorm(100)` and then `z`.

▶ Below that, make another Header 2, "A histogram", and a code cell containing `ggplot(tibble(z), aes(x=z)) + geom_histogram(bins = 10)`.

# My document

```
---
title: "Some random normal data"
format: html
editor: visual
---
```

## Packages

```{r}
library(tidyverse)
```

## Random normal data

```{r}
z <- rnorm(100)
z
```

## A histogram

```{r}
```

# Run the chunks

▶ Now run each of the three chunks in order. You'll see output below each one, including a histogram below the last one.

▶ When it works, add some narrative text before the code chunks explaining what is going to be done, and some text after describing what you see.

▶ Save the document (File, Save As). You don't need a file extension.

▶ Click Render (at the top). This makes an HTML-formatted report, which may appear in another tab of your web browser.

▶ If you want to edit anything, go back to the Quarto document, change it, save it, and run Render again. For example, you can try putting some of the text in *italics* or **bold**. (See Format.)

# The end of my (rendered) report

Then, we draw a histogram of the random normal values we generated:

```
ggplot(tibble(z), aes(x=z)) + geom_histogram(bins = 10)
```
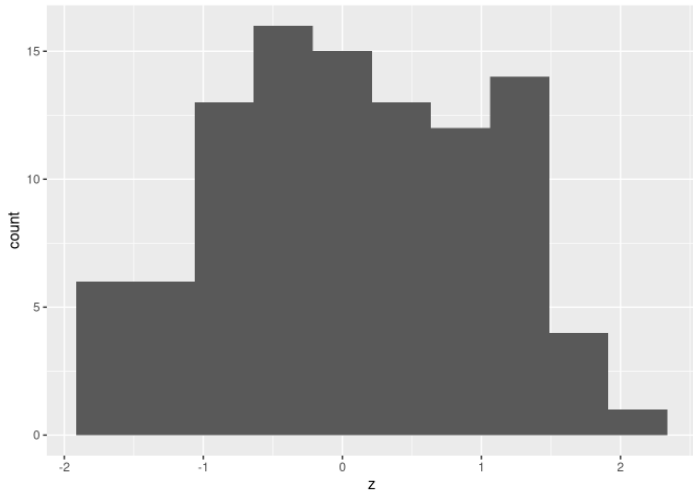


Even though we know the data to come from a normal distribution, the histogram may not

# Installing R on your own computer

- Free, open-source. Download and run on own computer.
- Three things:
  - R itself (install first)
  - R Studio (front end)
  - Quarto (for writing reports).

# Downloading R

▶ Go to https://www.r-project.org/.

# The R Project for Statistical Computing

## Getting Started

R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS. To **download R**, please choose your preferred CRAN mirror.

▶ Click Download R.

▶ R is stored on numerous "mirrors", sites around the world. The top one, "0-Cloud", picks one for you. Or you can choose one close to you (might be faster), eg. U of T:

## CRAN Mirrors

The Comprehensive R Archive Network is available at the following URLs, please choose a location close to you. Some found here: main page, windows release, windows old release.

If you want to host a new mirror at your institution, please have a look at the CRAN Mirror HOWTO.

0-Cloud

https://cloud.r-project.org/                        Automatic redirection to servers worldwide, cur

# Click your mirror

▶ Click 0-Cloud or U of T (or other mirror), get:

## Download and Install R

Precompiled binary distributions of the base system and cor

- Download R for Linux
- Download R for (Mac) OS X
- Download R for Windows

▶ Click on your operating system, eg. Windows.

# Click on Base

R for Windows

Subdirectories:

| | |
|---|---|
| base | Binaries for base distribution (managed by Duncan Mu... want to **install R for the first time**. |
| contrib | Binaries of contributed CRAN packages (for R >= 2.1... Ligges). There is also information on third party softw... Windows services and corresponding environment and... |
| old contrib | Binaries of contributed CRAN packages for outdated v... 2.11.x; managed by Uwe Ligges). |
| Rtools | Tools to build R and R packages (managed by Duncan... you want to build your own packages on Windows, or... |

▶ Click on "base" here.

# The actual download

▶ The version number is, as I write this, 4.3.0, but there may be an update between me writing this and you reading it.

▶ For Windows, click something like the top link below:



## Download R-4.3.0 for Window

README on the Windows binary distribu

New features in this version

▶ Then install usual way.
▶ For Mac, install `R-4.3.0-arm64.pkg` (Big Sur and newer), `R-4.3.0-x86_64.pkg` (older), or a newer version if available.

# Now, R Studio

▶ Go to https://www.rstudio.com/. You will be redirected to `posit.co`, which is the new name of the company that makes R Studio.

▶ Top right, click Download R Studio (the blue link).

▶ Scroll down to the non-blue Download R Studio.

▶ There are two choices. You want the free one on the left.

▶ Click left-side Download.

# Find the one for you

▶ We already installed R, so no need to do that.

▶ Scroll down to All Installers, and click the installer for your machine (Windows, Mac, several flavours of Linux). Install as usual.

| OS | Download | Size |
|---|---|---|
| Windows 10/11 | RSTUDIO-2023.03.1-446.EXE ⤓ | 208.07 MB |
| macOS 11+ | RSTUDIO-2023.03.1-446.DMG ⤓ | 381.89 MB |
| Ubuntu 18/Debian 10 | RSTUDIO-2023.03.1-446-AMD64.DEB ⤓ | 137.83 MB |
| Ubuntu 20/Debian 11 | RSTUDIO-2023.03.1-446-AMD64.DEB ⤓ | 137.83 MB |
| Ubuntu 22 | RSTUDIO-2023.03.1-446-AMD64.DEB ⤓ | 138.32 MB |

# Quarto

The last thing we need is Quarto, so that we can render documents (and thus hand in assignments).

▶ Go to https://quarto.org/.
▶ Click on one of the Get Started links.
▶ Find your operating system and install as usual:

**Step 1**

Install Quarto

Find your operating system in the table below

| Platform | Download | Size |
|---|---|---|
| Ubuntu 18+/Debian 10+ | quarto-1.3.353-linux-amd64.deb | 84.57 MB |
| Linux x86 Tarball | quarto-1.3.353-linux-amd64.tar.gz | 85.59 MB |
| Linux Arm64 | quarto-1.3.353-linux-arm64.deb | 84.41 MB |
| Linux Arm64 Tarball | quarto-1.3.353-linux-arm64.tar.gz | 85.36 MB |
| RHEL 7 Tarball | quarto-1.3.353-linux-rhel7-amd64.tar.gz | 83.82 MB |
| Mac OS | quarto-1.3.353-macos.pkg | 105.27 MB |
| Windows | quarto-1.3.353-win.msi | 77.83 MB |

# Running R

- All of above only done once.
- To run R, run R Studio, which itself runs R.

# How R Studio looks when you run it
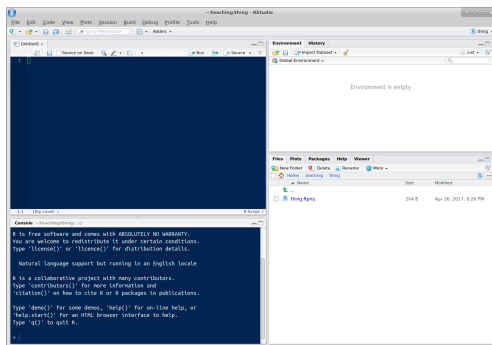


▶ that is, just the same as the online one.

▶ First time you run R Studio on your machine, click on Console
  window, and, next to the >, type
  `install.packages("tidyverse")`. Let it do what it needs
  to. (You need to do this on your machine. On
  `r.datatools.utoronto.ca`, it's already been done.)

# Projects

▶ A project is a "container" for code and data that belong together.

▶ Goes with a folder on some computer.

▶ File, New Project. You have option to create the new project in a new folder, or in a folder that already exists.

▶ Use a project for a collection of work that belongs together, eg. data files and Quarto documents for assignments. Putting everything in a project folder makes it easier to find.

▶ Example: use a project for (all) assignments in a course, a different document within that project for each one.