Running R

Running R online, 2025/2026 version

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

UNIVERSITY OF TORONTO

2i2c JupyterHub

jupyter

Classic Jupyter Notebook

LOG IN

RStudio

RStudio

LOG IN

jupyterlab

JupyterLab

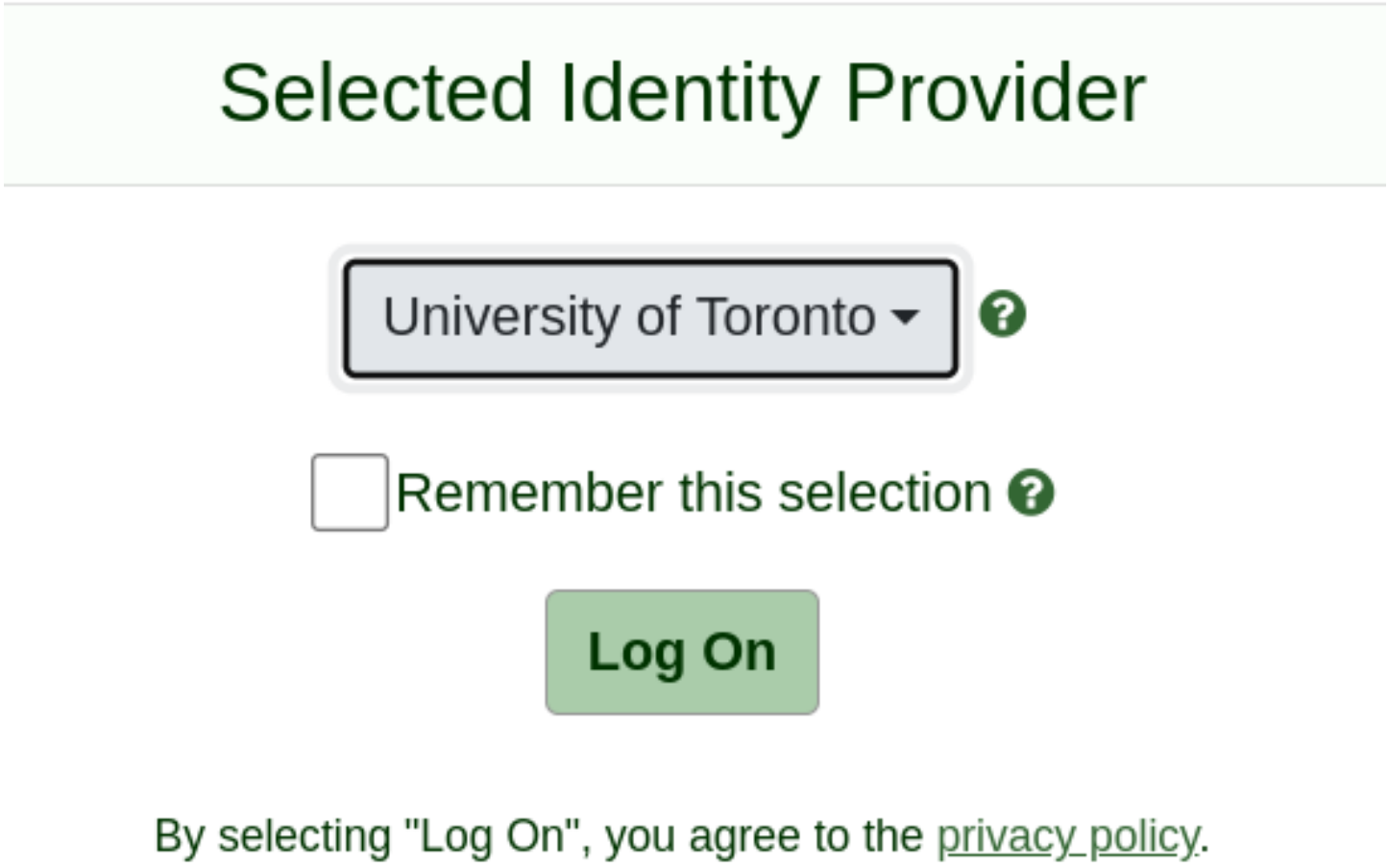
LOG IN

Welcome to U of T's JupyterHub for education

The University's 2i2c JupyterHub is an open source, web-based platform that offers a standardized computing environment. It can be accessed via your browser.
To get started, select one of the above services: Jupyter Notebook, RStudio or JupyterLab.

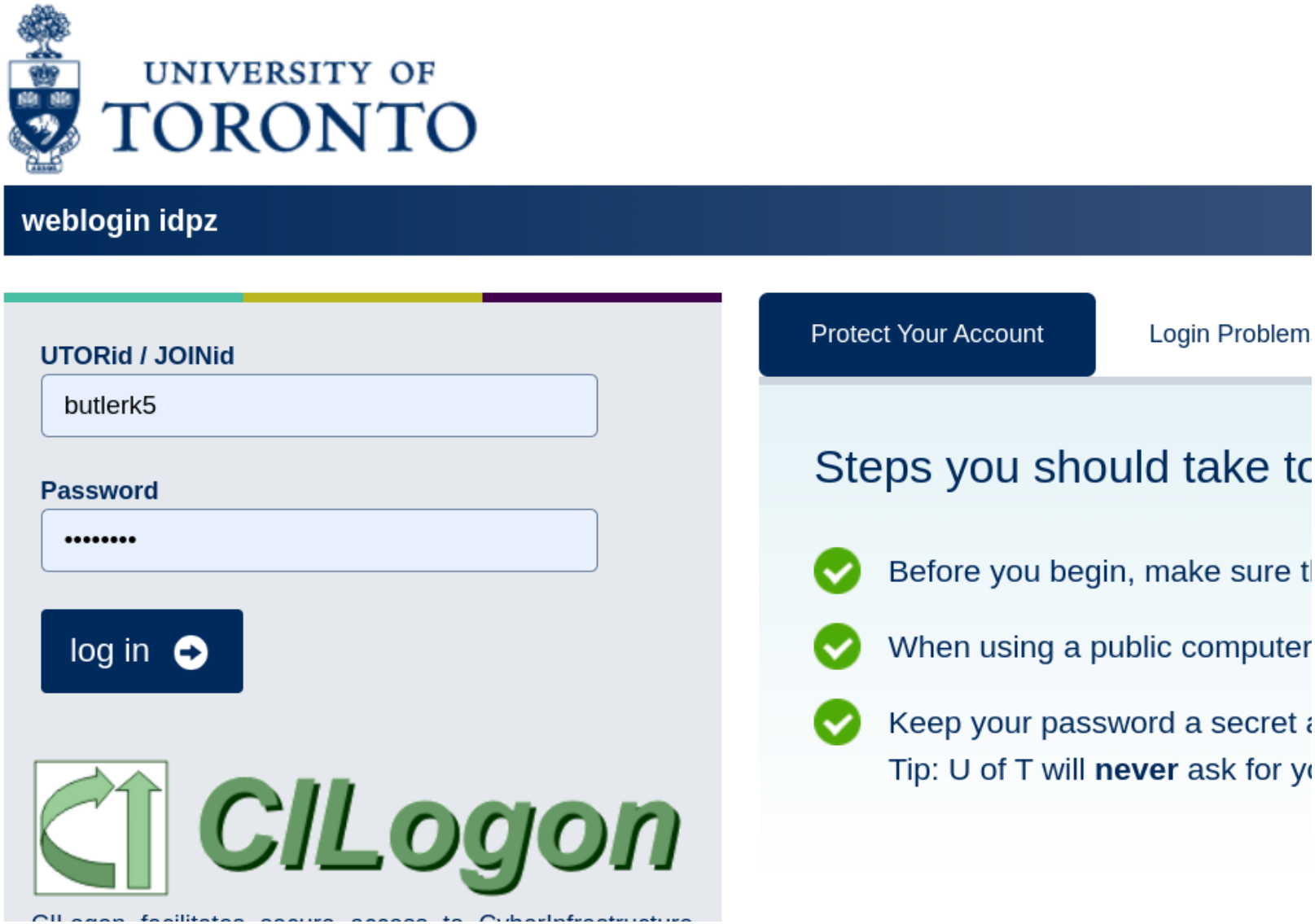
Click Log In (the blue button) under R Studio.

Log in



Click Log On, to verify that you actually are at U of T.

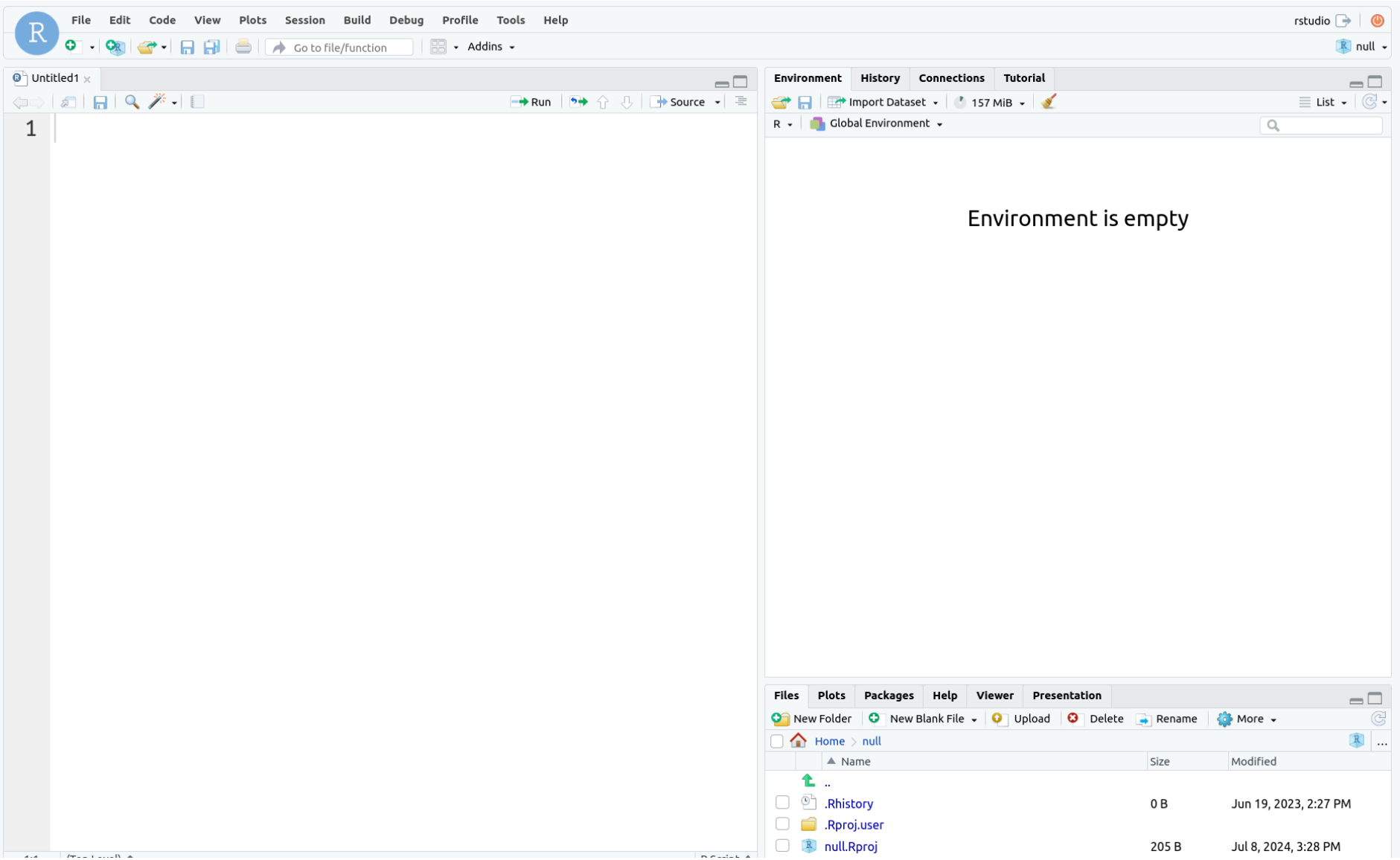
UTorID and password



as usual, but with *your* UTorID and password, not mine!

After a moment...

... gets you to R Studio:



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.
- I recommend having one project per *course*.
- R Studio restarts in project where you left off.

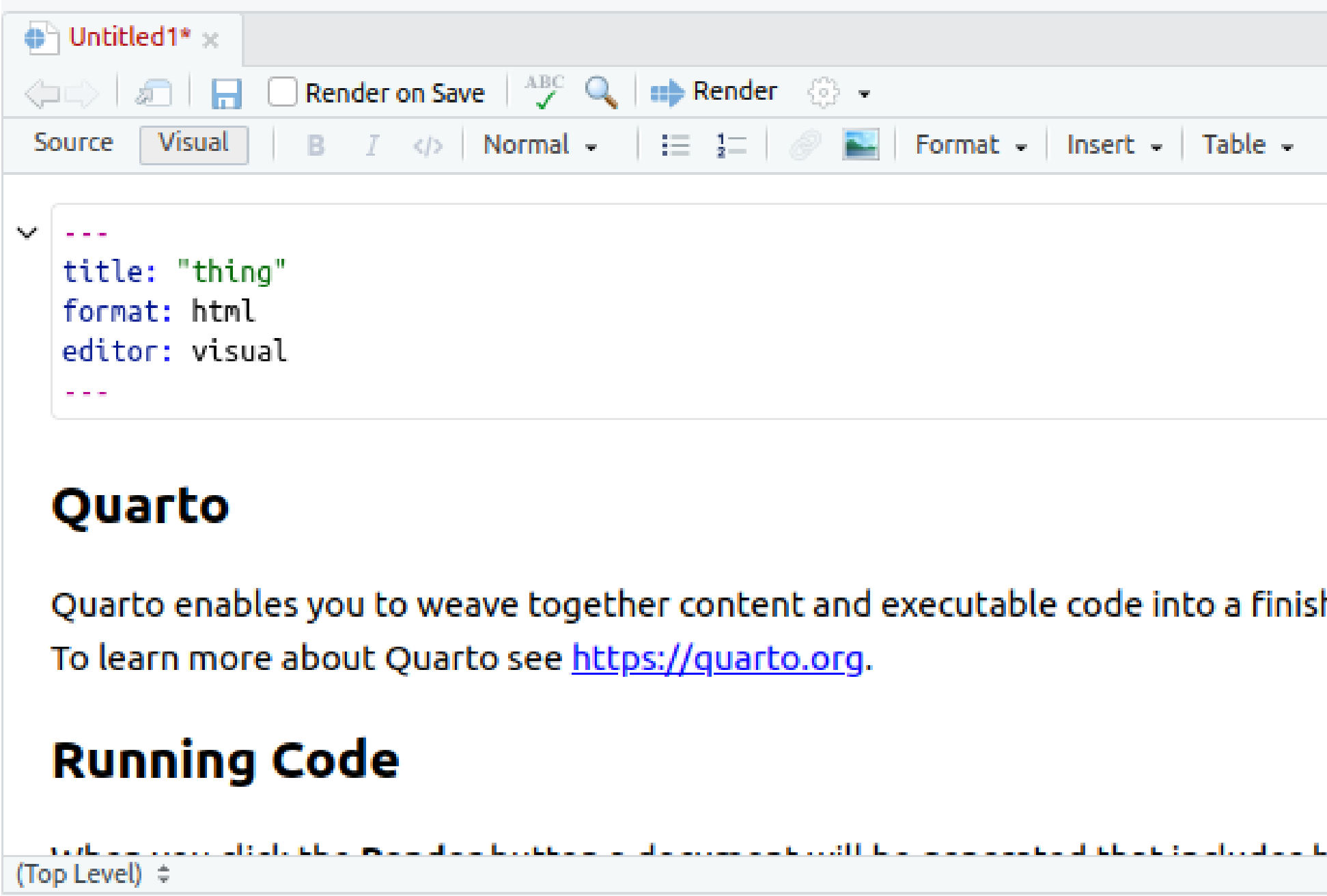
Make a new project

- 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.

Quarto documents

- 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



About this document

- It begins with a title (that you can change).
- Most of the document is text (narrative).
- Pieces beginning with `{r}`, with grey background, 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 "Chicken weights by diet", and click Create. When the document opens, delete the template that it gives you (leaving only the six lines that begin and end with ---).
- Move the cursor to the next line below those top six lines.
- Type a / (slash). This allows you to insert something.
- Start typing "heading". When you see "Heading 2" in the list, select that.
- 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, you should now see Normal (normal text).

Making our own document 2/2

- Make a new code chunk: type a slash, then select the top option "R Code Chunk".
- Inside that cell, type `library(tidyverse)`.
- Below that, make another "Heading 2" and put "Weights of chickens" on that line.
- Make another new code cell below that, and insert the line of code: `chickwts`
- Below that, make another Heading 2, "A boxplot", and another code cell containing `ggplot(chickwts, aes(x = feed, y = weight)) + geom_boxplot()`.

My document

```yaml
---
title: "Chicken weights"
author: "me"
format: html
editor: visual
---
```

Packages

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

Weights of chickens

```r
{r}
chickwts
```

A boxplot

```r
{r}
ggplot(chickwts, aes(x = feed, y = weight)) +
geom_boxplot()
```

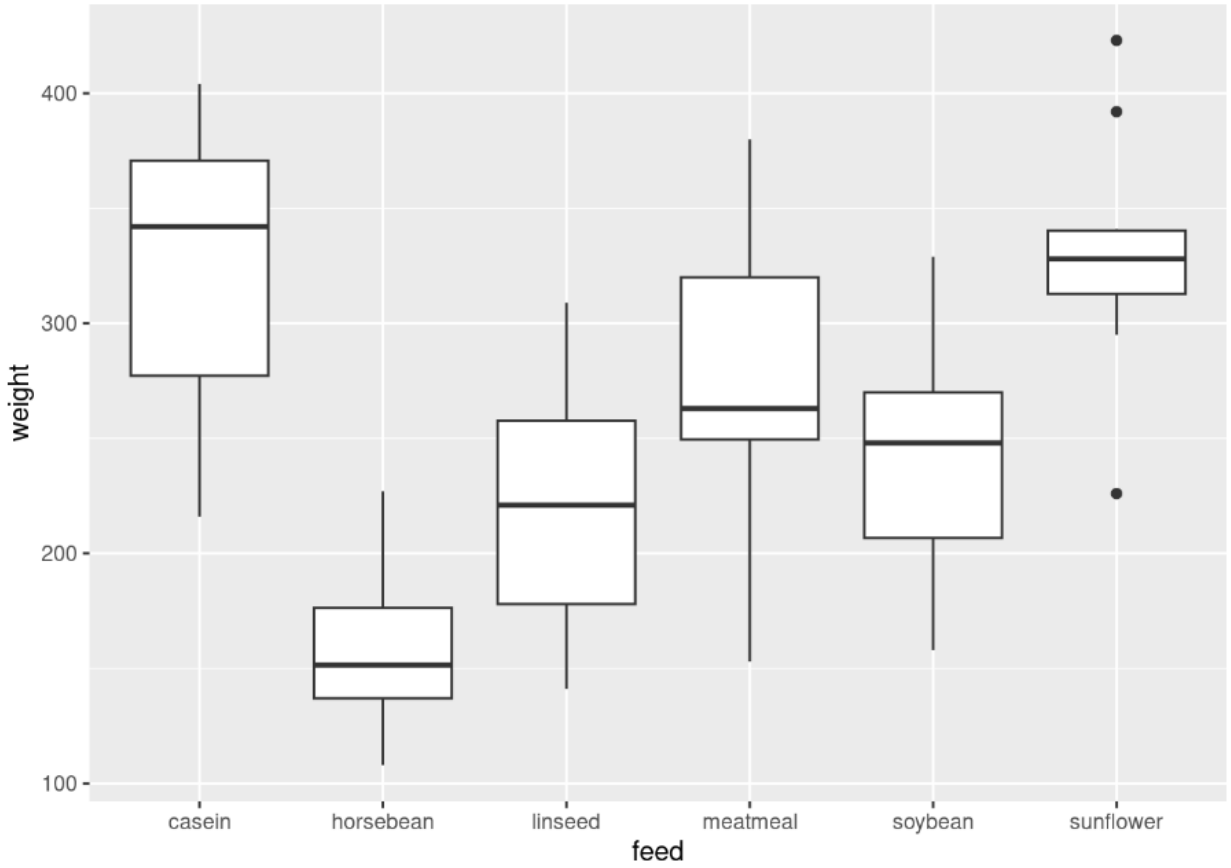
Run the chunks

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

A boxplot

```
ggplot(chickwts, aes(x = feed, y = weight)) + geom_boxplot()
```



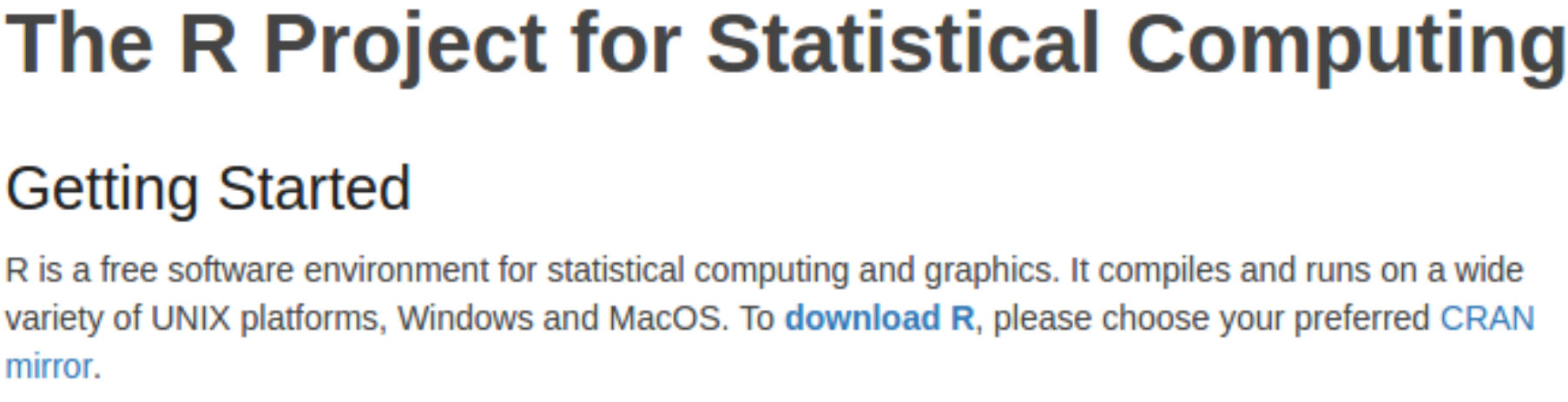
The weights of the chickens vary considerably by feed, with the chickens fed on horsebean weighing the least on average.

Installing R on your own computer

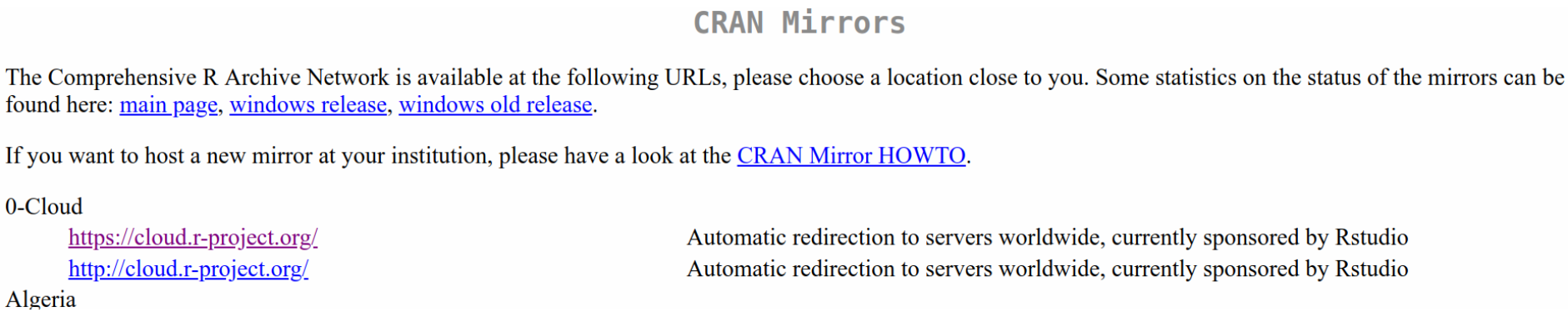
- Free, open-source. Download and run on own computer.
- Three things:
  - R itself (install first)
  - R Studio (front end)
  - Quarto (for writing reports).

Downloading R

- Go to https://www.r-project.org/.

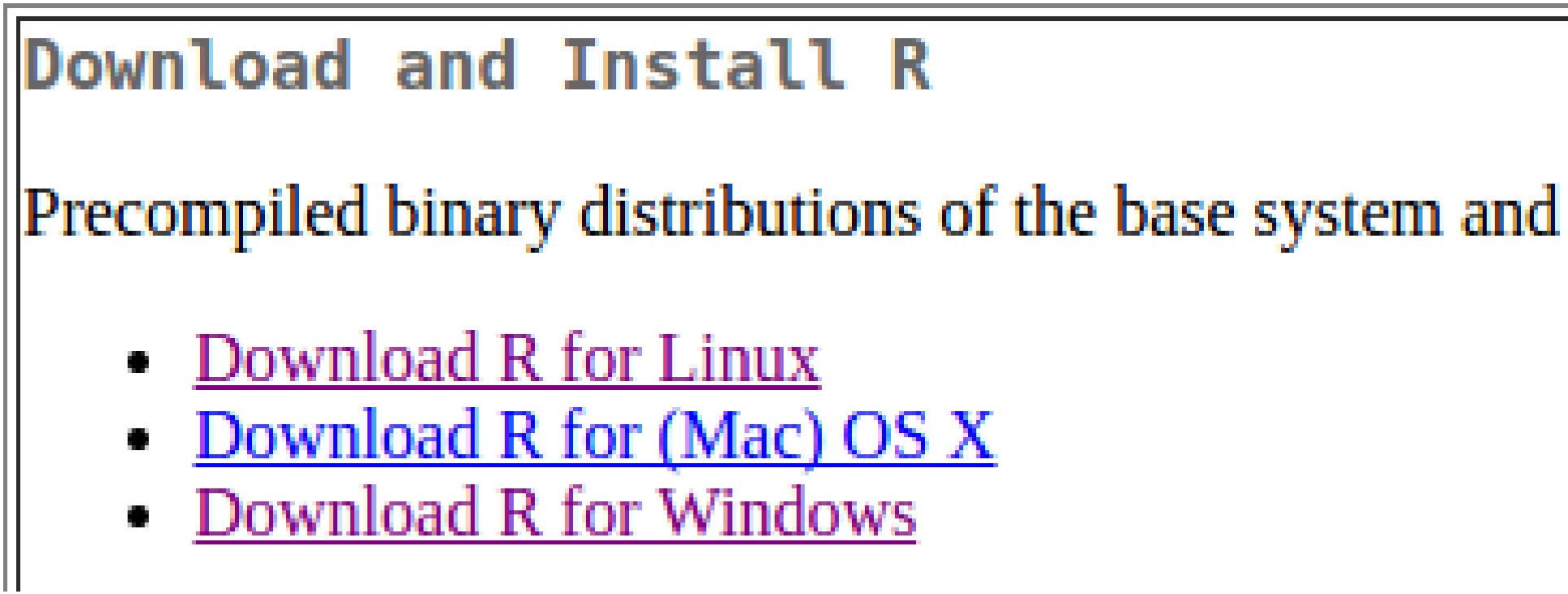


- Click Download R (the link in the first paragraph) .
- R is stored on numerous "mirrors", sites around the world. The top one, "0-Cloud", picks one for you.



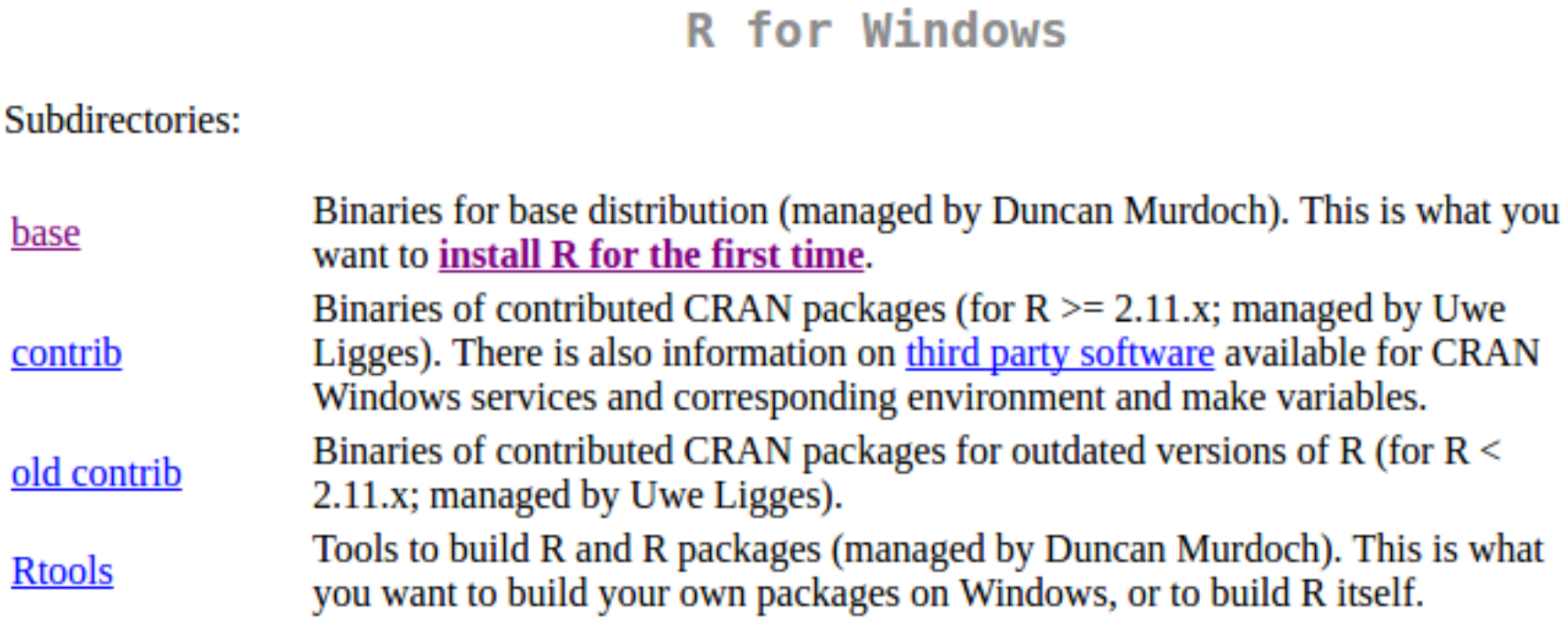
Click your mirror

- Click 0-Cloud (or other mirror), get:



- Click on your operating system, eg. Windows.

Click on Base

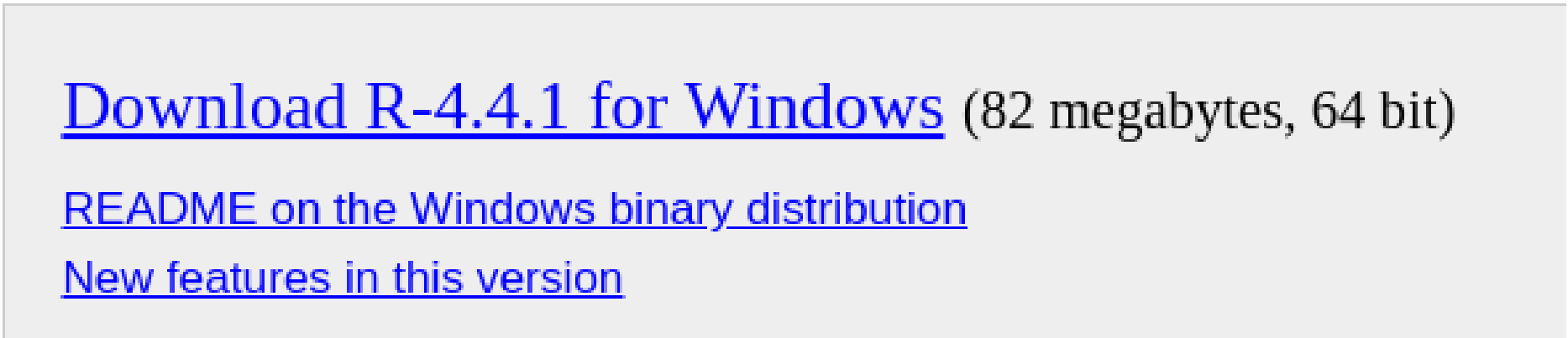


- Click on “base” here.

The actual download

- The version number is, as I write this, 4.4.2, but there may be an update between me writing this and you reading it.
- For Windows, click something like the top link below (yours will have the latest version number):

... continued



- Then install usual way.
- For Mac, install `R-4.4.1-arm64.pkg` (Big Sur with Apple Silicon M1-3), `R-4.4.1-x86_64.pkg` (Intel), or a newer version if available.
- Or, for Linux, click your distribution (eg. Ubuntu), then follow the instructions.

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.
- Click Open Source, then go down to Download R Studio (at the bottom).
- Scroll down to left Download R Studio button. Click it.

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. See over.

Choose the right one

All Installers and Tarballs

RStudio requires a 64-bit operating system.

Linux users may need to import Posit's public code-signing key prior to installation, depending on the operating system's security policy.

OS	Download	Size	SHA-256
Windows 10/11	RSTUDIO-2024.04.2-764.EXE	262.79 MB	09E1E38A
macOS 12+	RSTUDIO-2024.04.2-764.DMG	664.40 MB	D0DDD395
Ubuntu 20/Debian 11	RSTUDIO-2024.04.2-764-AMD64.DEB	194.73 MB	87B20155
Ubuntu 22/Debian 12	RSTUDIO-2024.04.2-764-AMD64.DEB	196.64 MB	1D0BD2F5

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 (blue).
- Find your operating system and install as usual (over):

Quarto 2/2

Step 1

Install Quarto

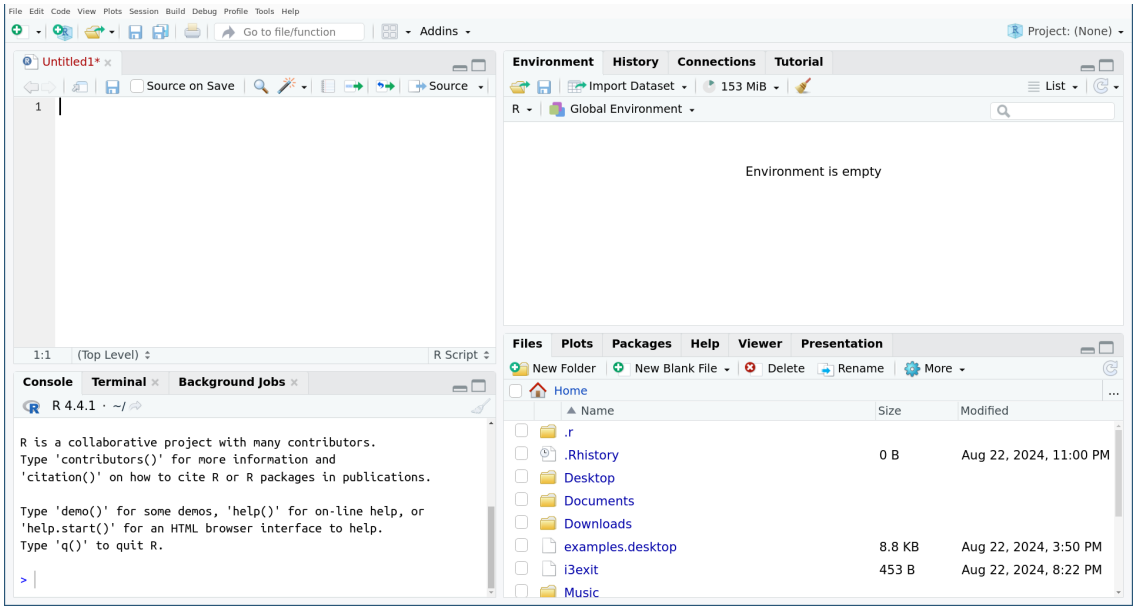
Find your operating system in the table below

Platform	Download
Ubuntu 18+/Debian 10+	quarto-1.5.54-linux-amd64.deb
Linux x86 Tarball	quarto-1.5.54-linux-amd64.tar.gz
Linux Arm64	quarto-1.5.54-linux-arm64.deb
Linux Arm64 Tarball	quarto-1.5.54-linux-arm64.tar.gz
RHEL 7 Tarball	quarto-1.5.54-linux-rhel7-amd64.tar.gz
Mac OS	quarto-1.5.54-macos.pkg
Windows	quarto-1.5.54-win.msi
	Release notes and more downloads…

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.

Install Tidyverse

- 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.