


# 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.

## Selected Identity Provider

University of Toronto ▼

☐

Remember this selection ?

**Log On**

By selecting "Log On", you agree to the [privacy policy](#).

---

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

# UTorID and password



UNIVERSITY OF  
TORONTO

weblogin idpz

UTorid / JOINid

Password

log in ➡



CILogon facilitates secure access to Scholar features

Protect Your Account

Login Problem

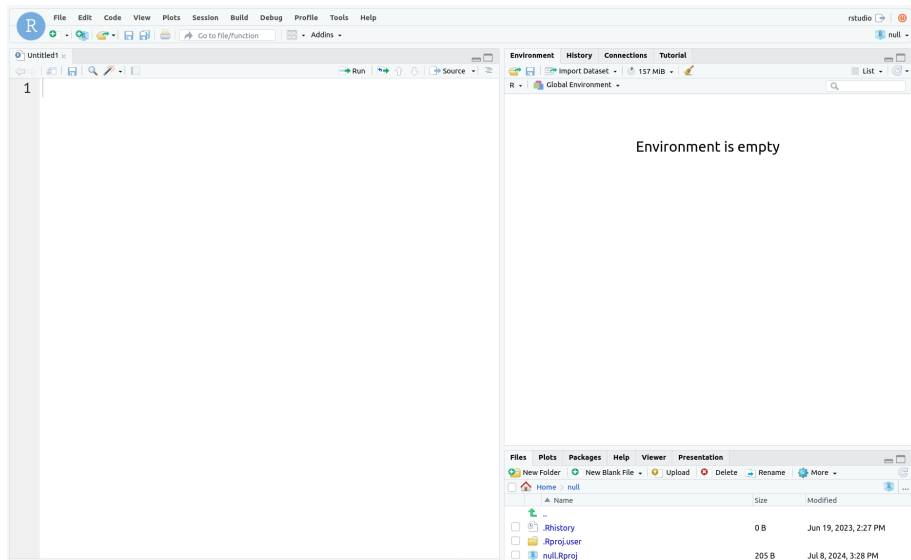
## Steps you should take to

- ✓ Before you begin, make sure th
  - ✓ When using a public computer
  - ✓ Keep your password a secret a
- Tip: U of T will **never** ask for yo

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

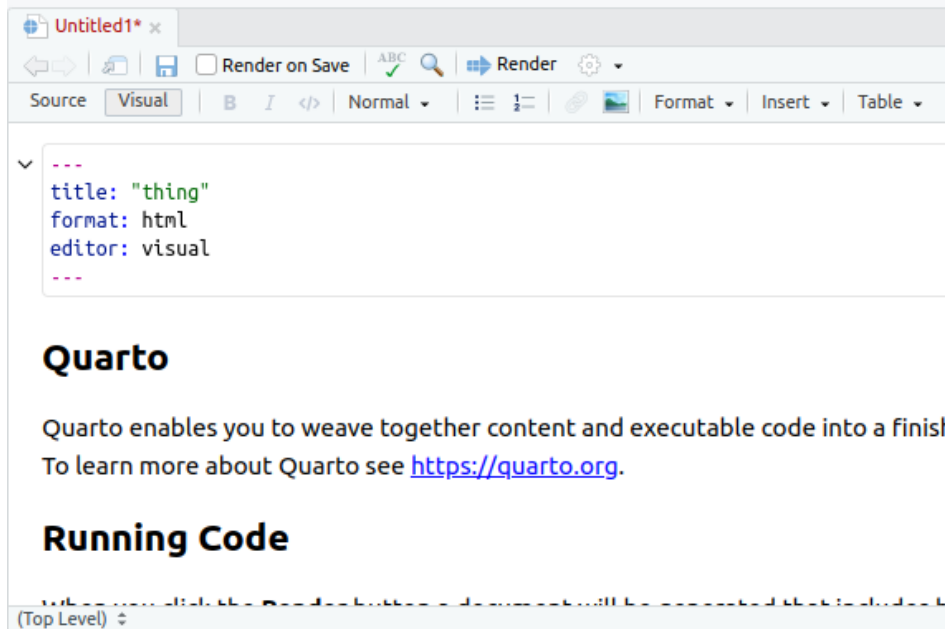
- Call it what you like. Mine is called thing:
- Select:
  - ▶ File,
  - ▶ New Project,
  - ▶ New Directory,
  - ▶ New Project (again),
  - ▶ give it a name and click Create Project.
- You see the name of your new project top right.

## 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



The screenshot shows the Quarto editor interface. At the top, there's a tab labeled "Untitled1\*" with a close button. Below the tab is a toolbar with various icons: back, forward, copy, paste, save, a checkbox for "Render on Save", a keyboard shortcut "ABC", a magnifying glass, a "Render" button with a blue arrow, a settings gear, and a dropdown arrow. Below the toolbar is a menu bar with "Source", "Visual" (highlighted), "B" (bold), "I" (italic), "</>" (code), "Normal" (font style), a list icon, a table icon, a link icon, a landscape icon, "Format" (dropdown), "Insert" (dropdown), and "Table" (dropdown). The main editing area shows a code block with a collapse icon (v) on the left. The code is a YAML frontmatter template:

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

Below the code editor, the rendered HTML output is displayed. It starts with a large heading "Quarto".

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

Below this is another heading "Running Code".

At the bottom left, there is a "(Top Level)" dropdown menu.

## 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

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

## Packages

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

## Weights of chickens

```
{r}  
chickwts
```

## A boxplot

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

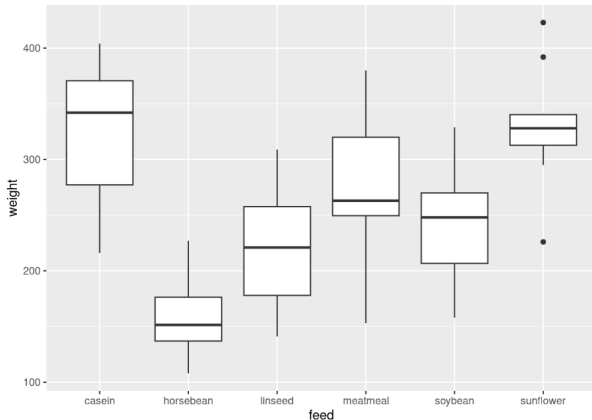
# Run the chunks

- 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/>.

## 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 (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.

#### CRAN Mirrors

The Comprehensive R Archive Network is available at the following URLs, please choose a location close to you. Some statistics on the status of the mirrors can be 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/>

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

Algeria

Automatic redirection to servers worldwide, currently sponsored by Rstudio

Automatic redirection to servers worldwide, currently sponsored by Rstudio

## Click your mirror

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

### Download and Install R

Precompiled binary distributions of the base system and

- [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 Murdoch). This is what you want to **install R for the first time**.

[contrib](#)

Binaries of contributed CRAN packages (for R  $\geq$  2.11.x; managed by Uwe Ligges). There is also information on [third party software](#) available for CRAN Windows services and corresponding environment and make variables.

[old contrib](#)

Binaries of contributed CRAN packages for outdated versions of R (for R  $<$  2.11.x; managed by Uwe Ligges).

[Rtools](#)

Tools to build R and R packages (managed by Duncan Murdoch). This is what you want to build your own packages on Windows, or to build R itself.

- 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):

[Download R-4.4.1 for Windows](#) (82 megabytes, 64 bit)

[README on the Windows binary distribution](#)

[New features in this version](#)

... continued

[Download R-4.4.1 for Windows](#) (82 megabytes, 64 bit)

[README on the Windows binary distribution](#)

[New features in this version](#)

- 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	<a href="#">RSTUDIO-2024.04.2-764.EXE</a> <small>±</small>	262.79 MB	<a href="#">09E1E38A</a>
macOS 12+	<a href="#">RSTUDIO-2024.04.2-764.DMG</a> <small>±</small>	664.40 MB	<a href="#">D0DDD395</a>
Ubuntu 20/Debian 11	<a href="#">RSTUDIO-2024.04.2-764-AMD64.DEB</a> <small>±</small>	194.73 MB	<a href="#">87B20155</a>
Ubuntu 22/Debian 12	<a href="#">RSTUDIO-2024.04.2-764-AMD64.DEB</a> <small>±</small>	196.64 MB	<a href="#">1D0BD2F5</a>

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

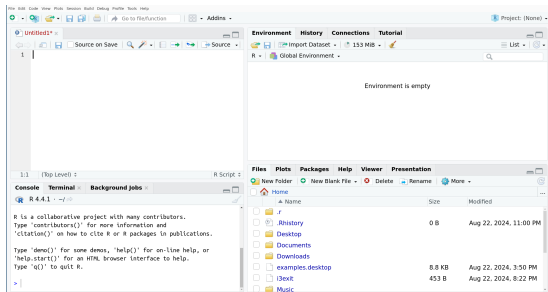
Find your operating system in the table below

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

# 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.