Using R for Data Processing

Getting Started in R

Glenn Williams

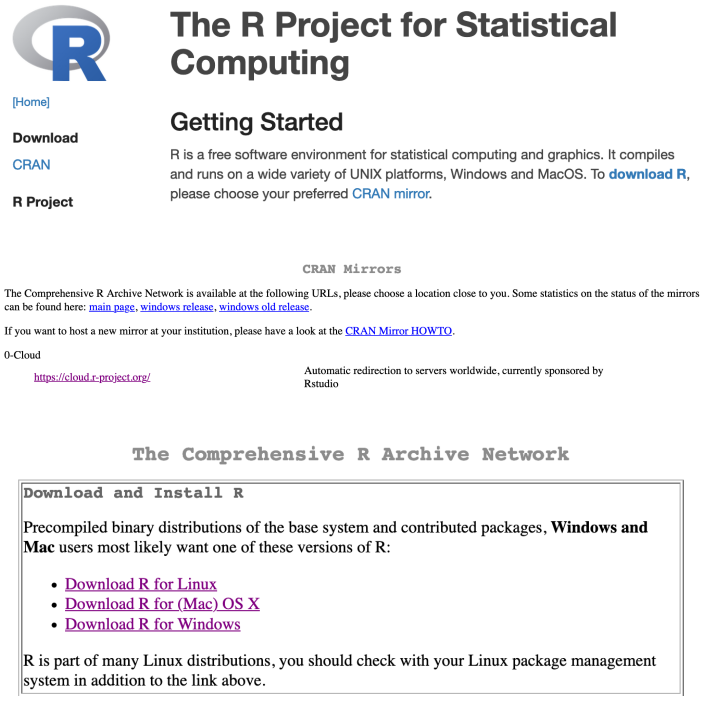
University of Sunderland

2020-09-14 (updated: 2020-09-22)

How Do I Install R?

Local Installation

- To get started, at the very least you'll have to download R from CRAN.
- Choose a mirror from which to download R. Any will do.
- Select the correct distribution for your operating system and then click through to **install R for the first time** if on Windows, or just click the most recent install version for Mac/Linux.
- You'll see a new page; For Windows, click on **"Download R**

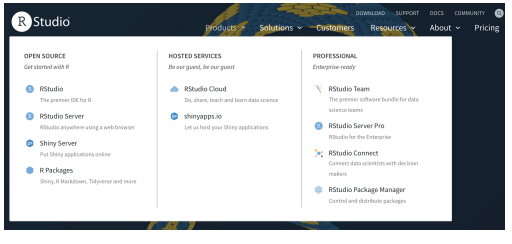


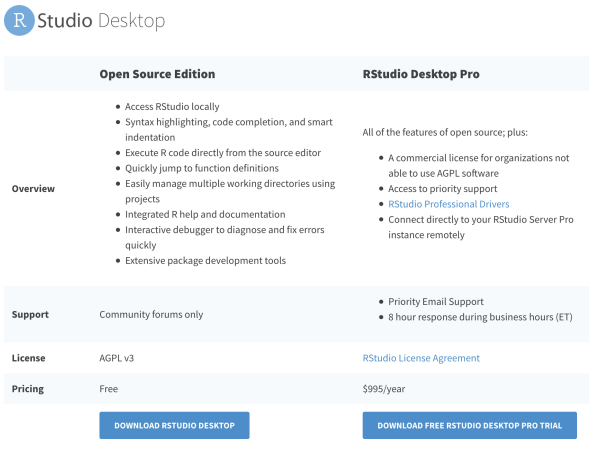
What is RStudio? How Do I Install It?

Local Installation

Integrated Development Environment for R: makes working with R and extensions to it easier.

- Download from the RStudio website.
- Select products and choose **RStudio**. Scroll down until you see **Download RStudio Desktop**.



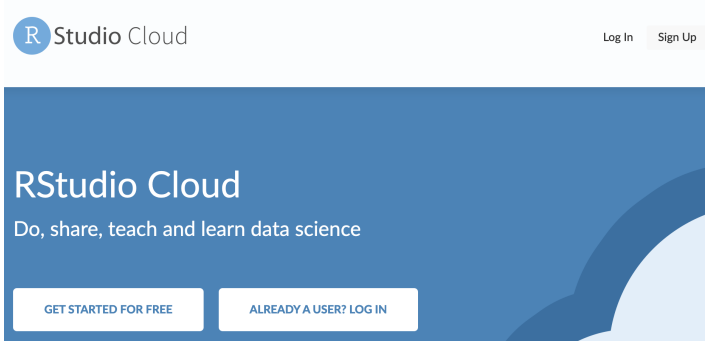


- Click the **Download** button in the free tier and select the correct installer for your operating system.

Getting Started in the Cloud

Alternatively, we can use RStudio Cloud to do everything online. For consistency.

- This may be the **easiest** route, especially if your system is locked down (e.g. on University controlled computers).
- This may be a little more limited than using R on your machine, but **most things you need will be available without effort on your behalf**.



rstudio.cloud signup page

- **Works with Chromebooks** etc. that do not allow local installs of R.
- Please note that rstudio.cloud **has usage limits**. You can only do 15 hours per month in it. If you can, get a local install.

Sign up for an account by clicking sign up on the homepage.

Starting Your First Project

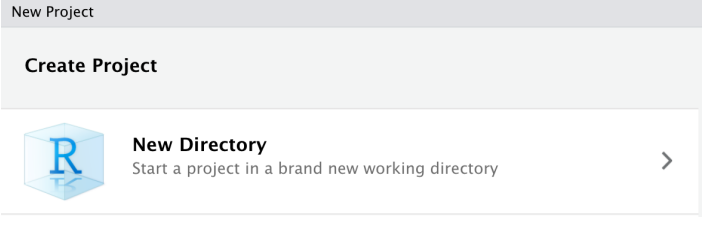
Assuming you're using a local installation of R/RStudio:

- Make a folder somewhere on your computer.
- Open RStudio.
- **Click File** → **New Project** → **New Directory** → **New Project**.
- Give this a name and save it somewhere you can access on your computer (e.g. Desktop)

Assuming you're using rstudio.cloud:

- Make a new Space and give it a name, then **click New Project**.

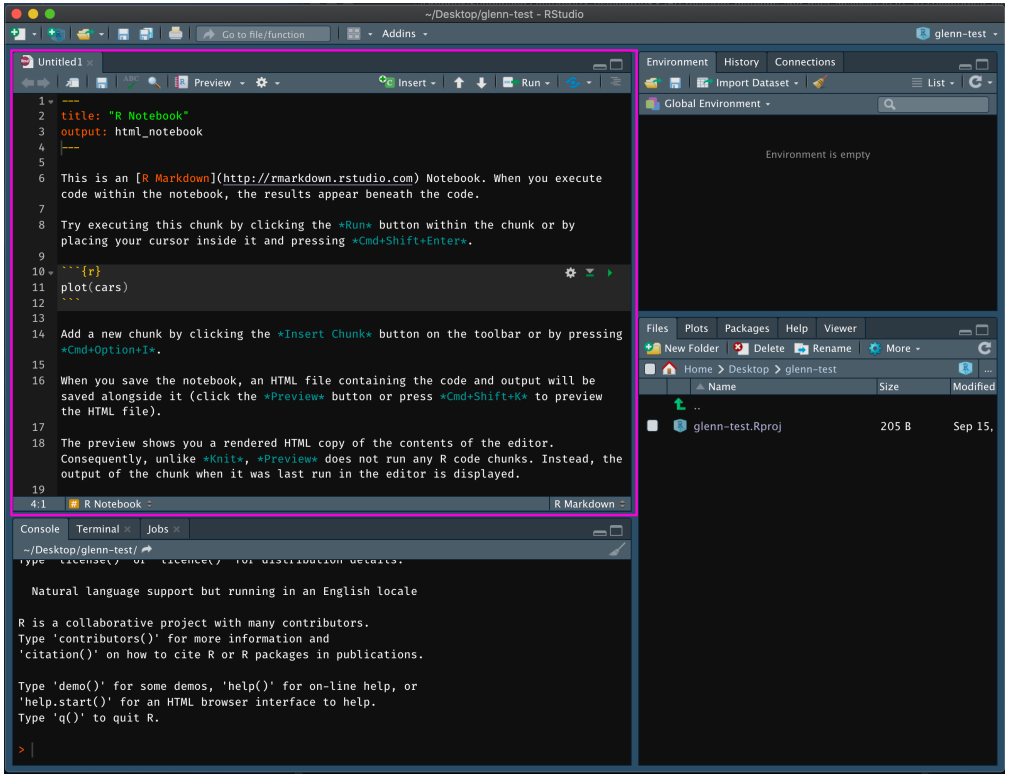
You now have a **folder and an rstudio project** for your work. You can put data and code in here, and all outputs will be saved in this place.



creating a new project

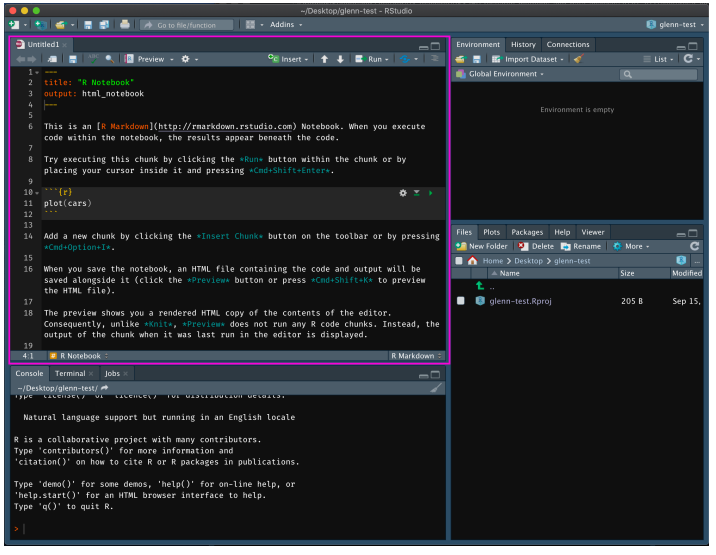
Now What?

Click File → New File → R Notebook. This will open an R notebook in the Editor.



RStudio Interface

Understanding the Editor?



The RStudio Interface

Editor (top-left): Where you **write your code**. Anything you write here can be saved to the file.

Console (bottom-left): Where we run your code. Once code is entered here you cannot edit it. **Don't work in the console!!!**

Environment (top-right): Lists any **variables avaiable in your global environment**. More on this later.

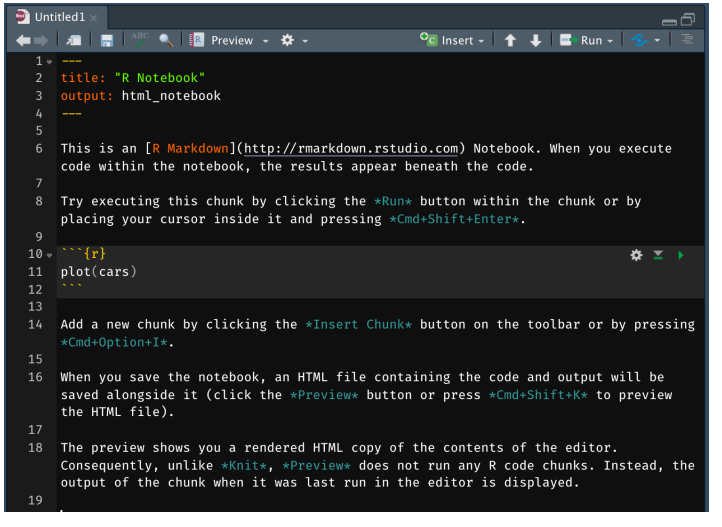
Files/Plots/Help/Viewer (bottom-right): See any plots in your **working dirctory (i.e. in your project)**, view any recently created plots or help on how to use R functions.

Making a Notebook

Now, we can make a notebook to get started in R. Go to **File** → **New File** → **R Notebook**. Give your file a title by changing `title:` at the top. Then save it with a sensible name in your folder.

Notebooks are made up of:

- **Markdown text**: which allows you to **write in plain English** (or other languages). You **add decoration** to text (e.g. italics, bold, links) using markdown commands. They have some examples in the text for you.
- **R code chunks**: These only accept R code. Press the play button to run the code.

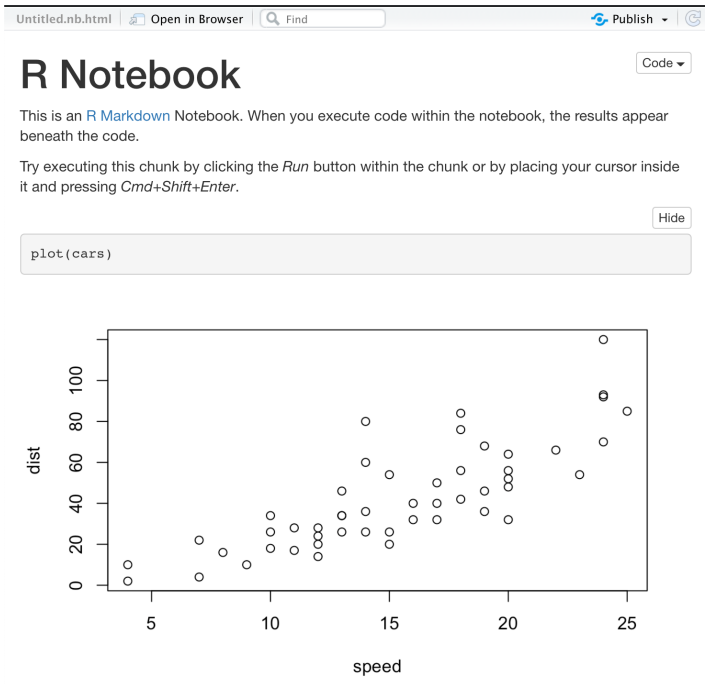


An R Notebook

When you save your notebook, you can **preview it**. It will create an HTML file that contains your markdown text and r code with output made pretty.

Preview Your Notebook

Make sure you press play on your code chunks. When done, save your work (Command/Ctrl + S) and **click Preview**. You should see this:



Avoiding Repetition

In **R code chunks**, we can code things up by hand, or make **functions** that allow us to repeat a sequence of commands easily. For example, let's say we want to add 1 to several numbers.

We could do it by hand:

```
1 + 1
```

```
## [1] 2
```

```
2 + 1
```

```
## [1] 3
```

```
3 + 1
```

```
## [1] 4
```

Or we could write a function, and apply this to our numbers:

```
add_one <- function(x) {x + 1}

add_one(1)
```

```
## [1] 2
```

```
add_one(2)
```

```
## [1] 3
```

```
add_one(3)
```

```
## [1] 4
```

Making Life Easy with Pre-built Functions

Some users know that functions they write can be useful to others, so they put them together in a **package**.

- The two packages (or pacakge of packages) we'll use are `tidyverse` and `here`.
    - **Tidyverse**: has several functions for making working with data and creating plots easier.
    - **Here**: makes your R-scripts read and write files relative to where the project is. This means you can write scripts that work on any PC, and not just your own.

This will be made clear once we start using these packages.

You only need to **install a package once per computer** (or cloud project):

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

But, you **must load the packages (or libraries) every time you start R**:

```
library("tidyverse")
library("here")
```

What Next?

Now we know the basics, we'll perform some calculations in R.