Installing R-studio:

Before we get started, a few quick words about what RStudio is:

It is not a point and click GUI. It doesn't let you generate graphs or do statistics by clicking on menus.

Essentially, RStudio makes working with the command line easier. R-studio:

Provides an easy way to save your work.

Keeps a copy of all the plots you generate in each R session, and makes it easy to copy these into a document.

Provides a nice way of presenting the R-help files.

Keeps track of all your variables for you.

In brief, it's just a very nice way of working with the command line in R.

We'll install R-studio and then give a brief description of how to use it. Fortunately the instructions are quite simple, regardless of the OS you're using (*warning:* if you have a really really old version of MacOS you may not be able to install RStudio - but then you should update your OS) is older.):

Make sure you've already installed R.

Go to **https://posit.co** (they recently moved to a new web site)

Click on the **Download RStudio** button (top right).

Scroll down to where it says Download RStudio, and pick the button that says "**Download**" under the free version (the paid version is only for large corporations and enterprise users that need support).
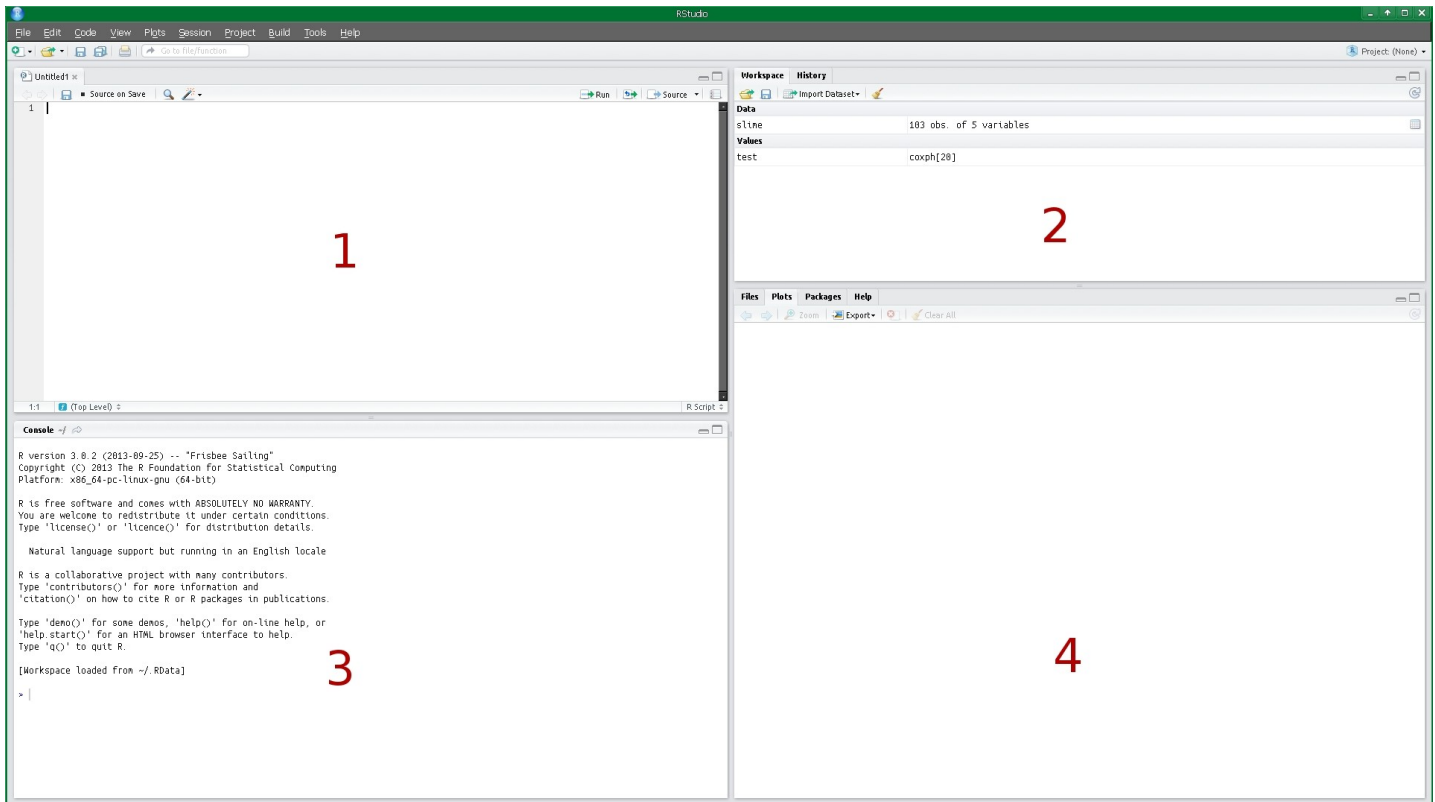
Select the right package for your OS and click on the correct download link.

Install the package you downloaded.

A very short guide to using RStudio:

Start R studio by clicking on the icon on your desktop (or wherever it was installed).

You should see a screen similar to the following:

The red numbers won't be there - they've been added to help explain R-studio.

As you can see, there are 4 windows. Let's explain each one:

Window 1:

(Window 1 will not be visible the first time you start RStudio - click on "File" and select "New" and then "R script" and it should pop up).

This is a script window. You can type your R-commands here. You'll notice that R-studio uses "syntax highlighting". That is, it will use different colors for numbers, text, etc.

You can highlight your commands and by clicking on the little "Run" icon at the top right of the window (unfortunately hard to see in the graphic above), your commands are automatically entered on the command line.

If you want to execute your commands one at a time, just place your cursor next to the command you want to execute and click on the "Run" icon repeatedly. It'll step through all your commands one at a time.

When you're done, you can save all your commands. Just click on "File" on the top tool bar, and select "save" or "save as". You can then load your file anytime you want to work on it again.

You can have multiple script windows open at the same time (if you're working on several different projects (or homework assignments!) at the same time, you can have one script window open for each. You'll see little tabs with the names of your script at the top of the script window.

Window 2:

This one isn't that useful unless you really do a lot of work with R. Mostly what it does is list all the variables you're using (just click on them to see). You may find this a bit confusing so you might ignore this until you get really comfortable with R/RStudio.

There's also a history tab that will let you see all the commands you've executed so far. Again, this isn't terribly useful at the moment.

Window 3:

This is simply the command prompt. If you just want to type commands into R, this is the place to do it. It doesn't really do anything special except maybe use different colors for different messages and/or results.

You can easily cut and paste from here (or from the script window).

Window 4:

This is where your plots show up. The nice thing is that you can easily view multiple plots (click on the forward and backward arrows to cycle through your plots).

There's also a help tab that gets activated and shows you the help for a particular R command if you ask R for help (for example, by typing "help(mean)" at the command prompt).

General comment:

R studio is much more versatile than what is described here. It's actually something called an "Integrated Development Environment" (IDE). It's what programmers usually use when they write programs in C, Basic, Python, or other programming languages. R is actually a complete programming language, not just a statistical package.

For us, it's mostly useful as a really good way to use R.

An example using R-studio:

Let's do an example for which we'll use R-studio. We'll take some data from the Ecology lab and play with it a bit:

1 - Start up R-studio, and make sure you have a script window open.

2 - Enter the following into the script window:

```
body <- scan(nlines = 1)
17 24 25 27 33 34 37 40 41

egg <- scan(nlines = 1)
37 65 69 54 93 87 89 100 90
```

This will read in data about body weight versus number of eggs into R.

The "nlines = 1" command tells R to only read one line of data, which can be useful if you're using the script window in R studio.

3 - Now highlight all four lines and click the "Run" icon at the top right of the script window (you can also use the control key and the enter key (simultaneously) on your keyboard instead of the "Run" icon).

You should see all four lines copied automatically into the command window.

The command prompt should respond with something like "Read 9 items" (you'll also see the names of the variables pop up in the workspace window (top right)).

4 - Back in the script window, you can type things like:

mean(body)

or

sd(body)

or

median(egg)

5 - Highlight each of these one at a time or all at once and click the "Run" icon (or use control-enter)

Again, the commands are copied into the command window and executed. The results should look a little like this (assuming you've made no mistakes):

```
> mean(body)
[1] 30.88889
> sd(body)
[1] 8.115486
> median(egg)
[1] 87
```

6 - Now let's try some plots. In the script window, type:

boxplot(body)

boxplot(egg, col = "blue")

boxplot(body,egg)

plot(body,egg, ylab = "Number of eggs", xlab = "Body weight")

7 - Again, highlight this one at a time or all at once and click on "Run" or use control-enter.

You should see the plots appear in the plot window. If you highlighted everything at once, this might have happened really fast.

However, all the plots are still there. Click on the back arrow key at the top left of the plot window, and you can scroll through all the plots you generated.

If you want to see a bigger version of a particular plot, click on the zoom button. You can resize the window that pops up to any size you want.

If you want to save a particular plot, click on export and select either "Save Plot as Image...", "Save Plot as PDF...", or if you just want to move it to a document or something you can select "Copy Plot to Clipboard...".

> A window with your graph will pop up - you should be able to figure out how to save it from there (it'll be just a bit different depending on your OS).

> (If you resize this window too much, it may disappear. Not sure why that happens.)

Finally, to close R studio, select "File" then "Quit RStudio".

> If R studio offers to save anything for you, you can, **_but you should never save the "workspace image"_** _unless you really know what you're doing as that can create all kinds of problems_.

> (It'll save all your variables _and recreate_ them when you open R the next time. Not necessarily a good thing!)