Installing R Packages Using RStudio

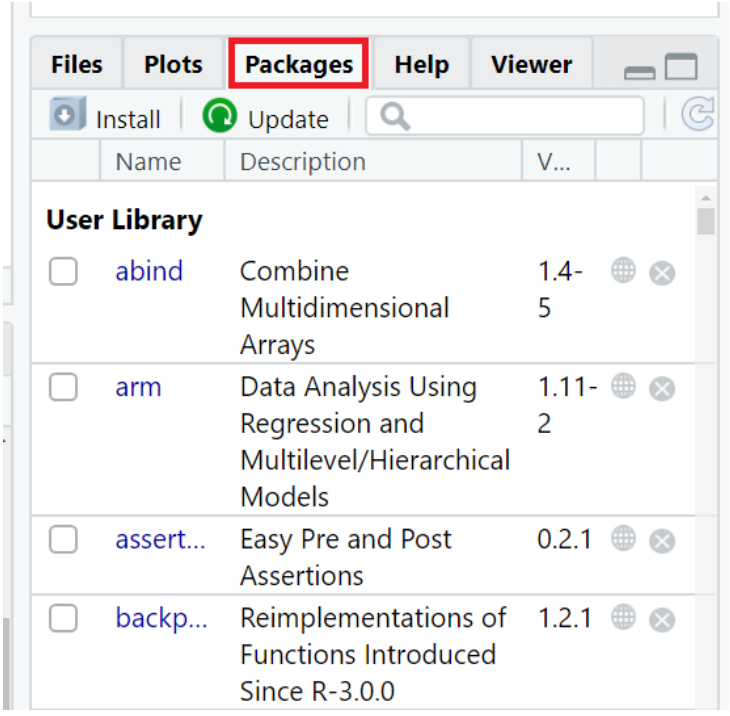
Many R functions that you will use are "built in" to the base R library, meaning that we can use those functions as long as we just have our basic R installed. However, some of the functions are built into packages (sets of functions and commands) that are not in the base R library.

In such cases, the necessary package must be downloaded and installed before we are able to use any functions in that package. Luckily, most of this process is pretty automatic; R should know where to automatically store and install the package(s). This set of R instructions will discuss how to go about locating, installing, and using non-base packages in R.

For example, in STAT 213, there are a few functions that will be used that are not in the base R library. In order to use these functions, we will need to install the **mosaic** package (and some related packages). So let's use that package as our example for this set of instructions.

Checking if a Package is Installed

Depending on the computer you're using (your own, one of the lab computers through the Virtual Desktop, etc.), a package of interest may or may not already be installed in your R library. To check a list of available packages, open RStudio and select the "**Packages**" tab in the lower-right pane. This will show the list of installed packages.

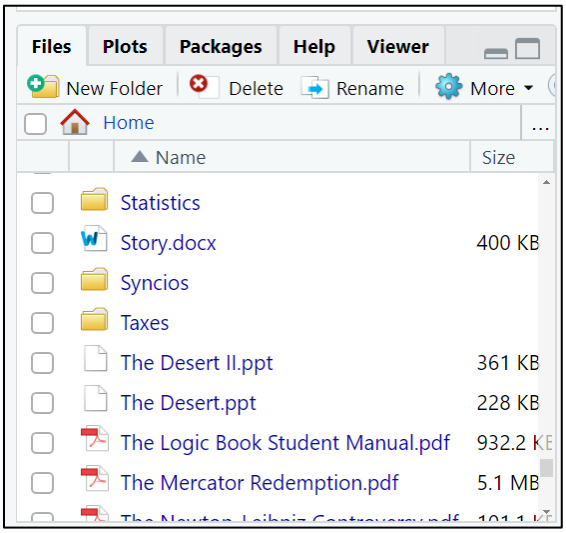


If you scroll through this list and do not see **mosaic** (or whatever package you need), then we need to install it!

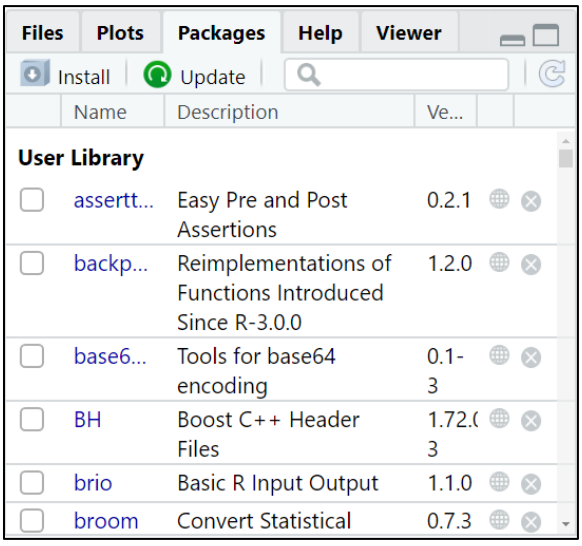
Installing a Package Using RStudio

There are a few ways to install a package of interest using RStudio, but here is one way to do it:

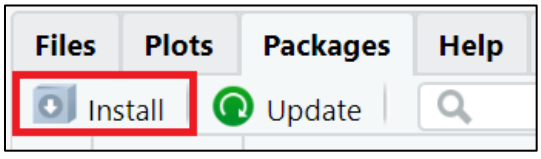
1. Open RStudio. In the lower right-hand part of the RStudio program, you should have a window that looks something like this:



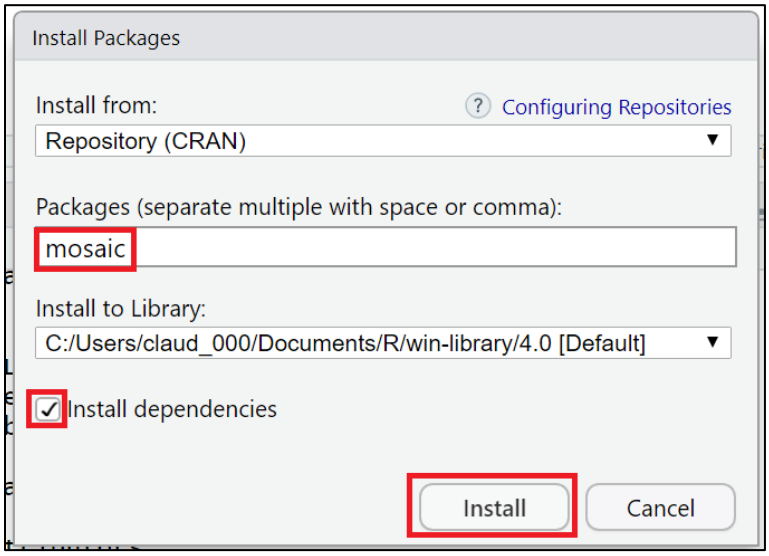
Click the "**Packages**" tab. This will show you the list of packages already in your library.



2. Click the "**Install**" button.



3. In the window that pops up, type “mosaic” (without quotes) in the “Packages (separate multiple with space or comma)” field. Check “Install Dependencies” if it isn’t checked already, then click “Install.”



The mosaic package is big and also requires several other packages for it to run; it *should* automatically install any related packages that it needs. Be patient while it installs; it may take a few minutes! **You will notice a lot of text show up in your R console window. This is normal!** It means the packages are installing.

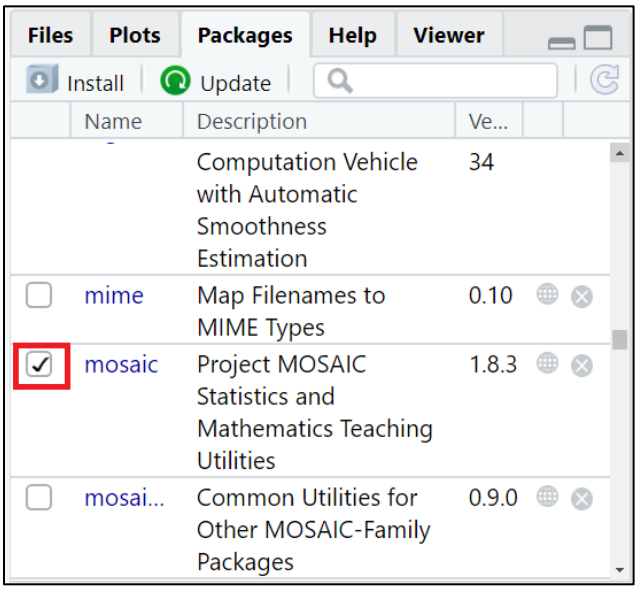
The **mosaic** package will have been successfully installed when you get a message similar to this and your cursor (the little red caret **>**) appears after it.

```
package 'mosaic' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\claud_000\AppData\Local\Temp\RtmpiiJoS5\downloaded_packages
> 
```

Loading a Package in RStudio

To load a package in RStudio, you can either type `library(packagename)` in your R console window or you can select the package(s) you want loaded in that "Packages" window.



This will load the package.

Troubleshooting

The easiest way to make sure a package is working is to just try one of the functions that is built into that package. As an example, the function `favstats()` is built into the **mosaic** package, meaning that in order to use `favstats()`, the **mosaic** package must be installed, loading, and working properly.

If you are running into an issue with a function that should be working, here is a set of troubleshooting steps you may want to try.

Step 1

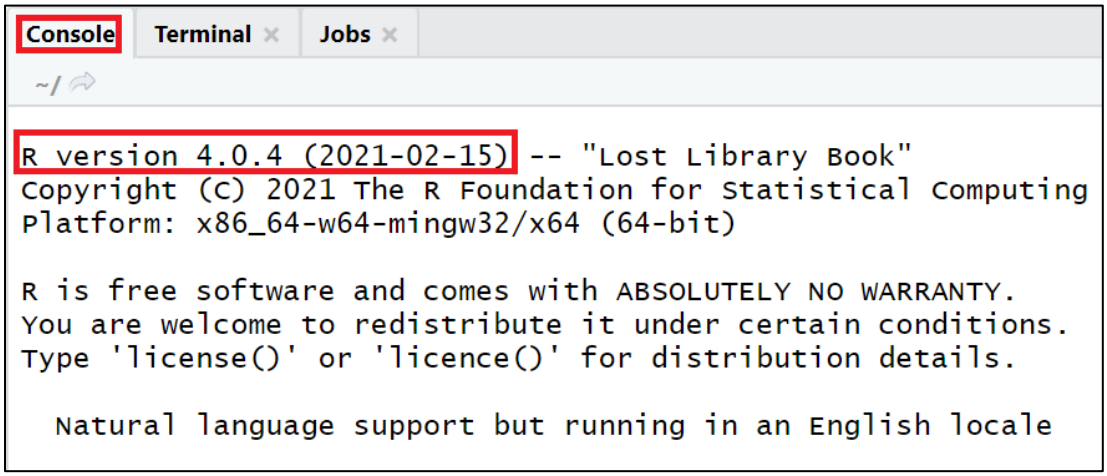
First, **make sure the package that you need is installed**. To check a list of installed packages, open RStudio and select the "**Packages**" tab in the lower-right pane. This will show the list of installed packages. If you do not see the package that you need in the list, install it (see the "Installing a Package Using RStudio" section above).

Step 2

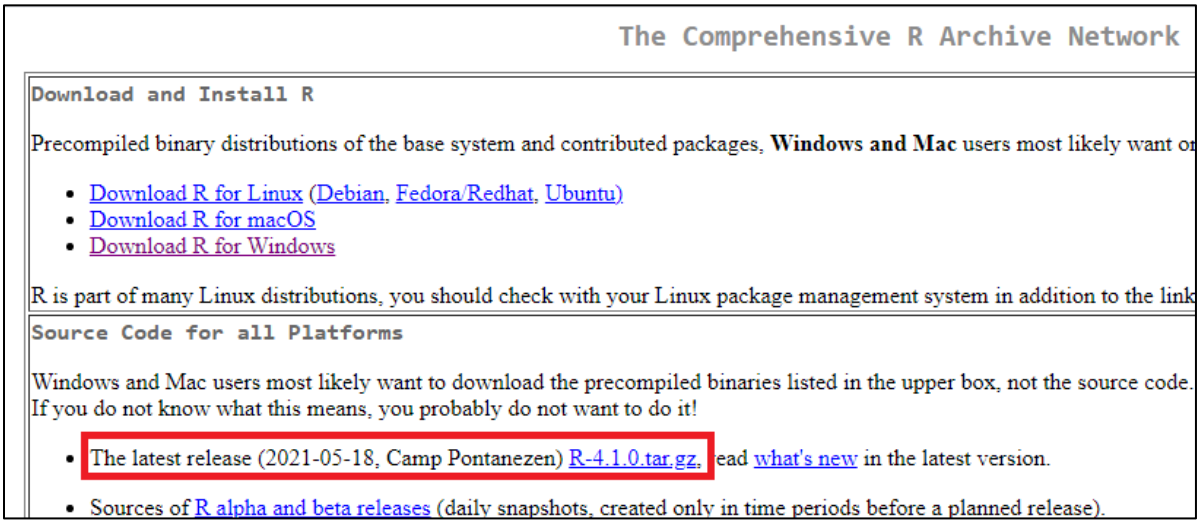
If the package that you need is installed, but your function is still not working, **make sure the package is loaded into your R session**. To load a package, you can either type `library(packagename)` in your R console window or you can select the package(s) you want loaded in that "Packages" window. Once the package is loaded, you should be able to use any functions from it.

Step 3

If the package is installed and loaded but your function is still not working, **check to see if you have the latest version of R installed.** To check your version of R, open RStudio and look at the top row of the text that automatically appears in the R console pane. This tells you what version of R you're running. In the following screenshot, I am running version 4.0.4. [Note: the version of RStudio you're running shouldn't matter!]



To check if this is the latest version of R, visit the R homepage at https://www.r-project.org/. The latest release version will be listed on the main page. In the following screenshot, it shows that the latest version as of the writing of this guide is version 4.1.0.



Some packages may not work with older versions of R. So if you do not have the latest version, you can get it by just re-installing R from https://www.r-project.org/ (if you need to refer to the "Installing R" guide). You do not have to uninstall the older version you have; it will just be overwritten by the newest version.

Once you have the latest version of R, try re-installing the package you need, load it into your R session, and try running a function from that package. This typically solves the problem!

Every once and a while, though, R can just be a little buggy. So if you're still having trouble, just closing and re-opening R can sometimes fix the problem!