

Installing & Configuring R

Downloading R

1. Go to the R home page at www.r-project.org.
2. From the navigation bar on left side of the page, click on CRAN (underneath "Download, Packages").
3. From the list of mirror sites, pick one in the United States (note that once you have made a choice, the nav bar changes).
4. Pick your operating system (use the list of precompiled binaries at the top of the page).
5. For Mac, pick the latest package, for instance R-2.14.1.pkg, and tell the browser to save the file. For Windows, pick "base" and on the next page, pick the download link and save the file.

Installing R on a Macintosh

Note: this was tested on OSX 10.7.2

1. Double click the pkg file, and run the installer (the defaults are fine). You are done.

Installing R on the Windows Operating System

Note: this was tested on Windows XP Professional Version 2002 Service Pack 2

1. Double click the exe file, which is actually an installer, not the program. Answer "run" to any security questions about an unknown publisher.
2. Selected "English" as the language.
3. Follow the installation wizard instructions. The defaults are fine.
4. When the installation process is done, right-click on the R icon (shortcut) which was installed on the desktop; select properties. On the shortcut tab, after the last quotation mark in the target box, type " --sdi" (without quotes; make sure there is a space between the quotation mark and the first -). Click OK.

Adding Rstudio

R has a GUI (graphical user interface) that is essentially a full IDE (integrated development environment). All the tools you need are present and work in a coordinated way (though the Mac version has more goodies). However, many new users have found the Rstudio program to be a great way to get started. Rstudio is a different IDE for R which is very nicely designed. You don't need it, but you might get hooked on it. Simply go to www.rstudio.org and follow the download instructions.

Upgrading R

New versions of R are issued once a year in April. When a new version is released, authors of packages may need to make changes to those packages so they will run with the new version of R. You may not want to upgrade your version of R right away, as package authors need time to catch up. Generally, to upgrade R, just re-install it as described above (it is possible to keep the old version on your computer, but I'll not provide instructions for this). If you have additional packages installed, you will need to update them as well.

Customizing Your R Installation There are many ways in which you might wish to customize your R installation, but only a couple of them should be considered for new(er) users. One is that you may need to add additional packages so that you have capabilities beyond the base installation. Another is that you might want to configure the R start up process so that you don't have to repeat a bunch of commands each time you run R, and so that the R environment looks and operates the way you like every time you run it.

Customization 1: Downloading and Installing Packages

One of R's virtues is that anyone can extend it by writing additional packages, and there are more than 3,000 packages available. Additional packages can come from many places (called repositories). The steps are done from within R. An important point to understand is that packages are kept in a library on your computer hard drive. They are not automatically available in a particular R session however. They must be "checked out" or loaded from the library before you can use them.

From CRAN on any platform using the command line ...

CRAN stands for Comprehensive R Archival Network. It's the main place where add on packages are stored.

1. Launch R.
2. In the console, type `> chooseCRANmirror()`. Select a mirror near you.
3. In the console, type `> install.packages("name_of_package")`.
4. In the console, type `> library("name_of_package")`.
5. In the future, you do not need to install the package each time you wish to use it. Simply start R and give the command `> library("package_name")`.

From Github on any platform using the command line ...

Github is another very popular place to store packages. It's where all the cool kids hang out.

1. Launch R.
2. In the console, type `> chooseCRANmirror()`. Select a mirror near you.
3. In the console, type `> install.packages("devtools")`.
4. In the console, type `> library("devtools")`.
5. In the console, type `> install_github("package_name", "user_name")`. "package_name" and "user_name" will have to be provided to you.
6. In the future, you do not need to install the package each time you wish to use it. Simply start R and give the command `> library("package_name")`.

Customization 2: Using an Rprofile File Not recommended until you get comfortable.

An .Rprofile file (note the leading "dot") is a plain text document that is read by R during the start up process. It can contain any valid R commands to be executed; these then affect the behavior of R. A common use is to load a bunch of packages that you routinely use so they don't have to be loaded every time you start R. Both operating systems give some headaches: on the Mac, files beginning with a period are invisible. With Windows, there can be problems both with the leading period and in keeping Windows from adding on an unwanted extension.

On a Mac...

1. Go to the "Applications" folder and find "TextEdit"; launch it.
2. Type in the commands you want. These must be valid R commands just like you would use in the console.
3. Save the file as "Rprofile.txt" in your highest level folder, typically "Users/yourname". This is a visible file for convenience.
4. Now go to "Applications/Utilites" and find the application "Terminal"; launch it.
5. In the Terminal window, type "pwd" to see what the working directory is. If it is "Users/yourname" you can proceed as follows. Type "cp Rprofile.txt .Rprofile" without the quotes. This tells the operating system to copy Rprofile.txt to the hidden file .Rprofile Quit "Terminal"
6. If the current working directory is not "Users/yourname", you have to navigate to that folder. Depending upon what the current working directory is, you use "cd new_path" to get to where you want to go. "cd .." takes you up one directory level.
7. Restart R to make the Rprofile take effect.
8. Finally, to edit your Rprofile file in the future, you can find Rprofile.txt, double-click it, edit it, save it, and repeat steps 4-8.

On a Windows Machine...

1. First, with Windows Explorer open, go to the "Tools → Folder Options... → View" tab. Make sure that "Shown hidden files and folders" is turned on. Make sure that "Hide extensions for known file types" is turned off.
2. Launch "Notepad".
3. Type in the commands you want. These must be valid R commands just like you would use in the console.
4. Save the file as ".Rprofile" in whatever directory is specified in the shortcut as "Start in" with no extension. To avoid the extension, be sure to save the file as "all files" type.
5. Restart R to make the Rprofile take effect.
6. To edit your Rprofile file in the future, you can find .Rprofile, double-click it, edit it, save it, then restart R.

I should note that the true operations "under the hood" of Rprofile files are more complex than I have led you to believe, but these instructions should work for typical users.