What is RStudio

R is an open-source software environment for statistical computing and graphics that runs on Windows, Mac OS, and many UNIX platforms. Unlike many other programs, users interact with R through the issuance of commands on a command line rather than through a graphical user interface. While such an interface may be unusual for many users, it's primary strength is the ability for a user to develop scripts of commands to perform various analyses that can then be easily repeated.

RStudio is an open-source integrated development environment (IDE) that serves as a front-end "on top" of R that eases the user's interaction with R by providing some of the conveniences of a GUI and, more importantly, a means for efficiently constructing and running R scripts. Among other conveniences, RStudio provides a four-panel layout that includes a feature-rich source-code editor (includes syntax highlighting, parentheses completion, spell-checking, etc.), a tight link to the R console, a system for examining objects saved in R, an interface to R help, and extended features to examine and save plots. More information about RStudio can be found at www.rstudio.com/ide/docs/.

RStudio Design

RStudio is organized around a four-panel layout (Figure 1). The upper-left panel is the R *Script Editor*. R commands will be typed in this panel and submitted to the R *Console* in the lower-left panel. For most applications, you will be typing R commands in the *Script Editor* and submitting them to the *Console*; you will not be typing commands directly into the *Console*. The *Script Editor* essentially acts as a high-level text editor whereas the *Console* is essentially the R program.

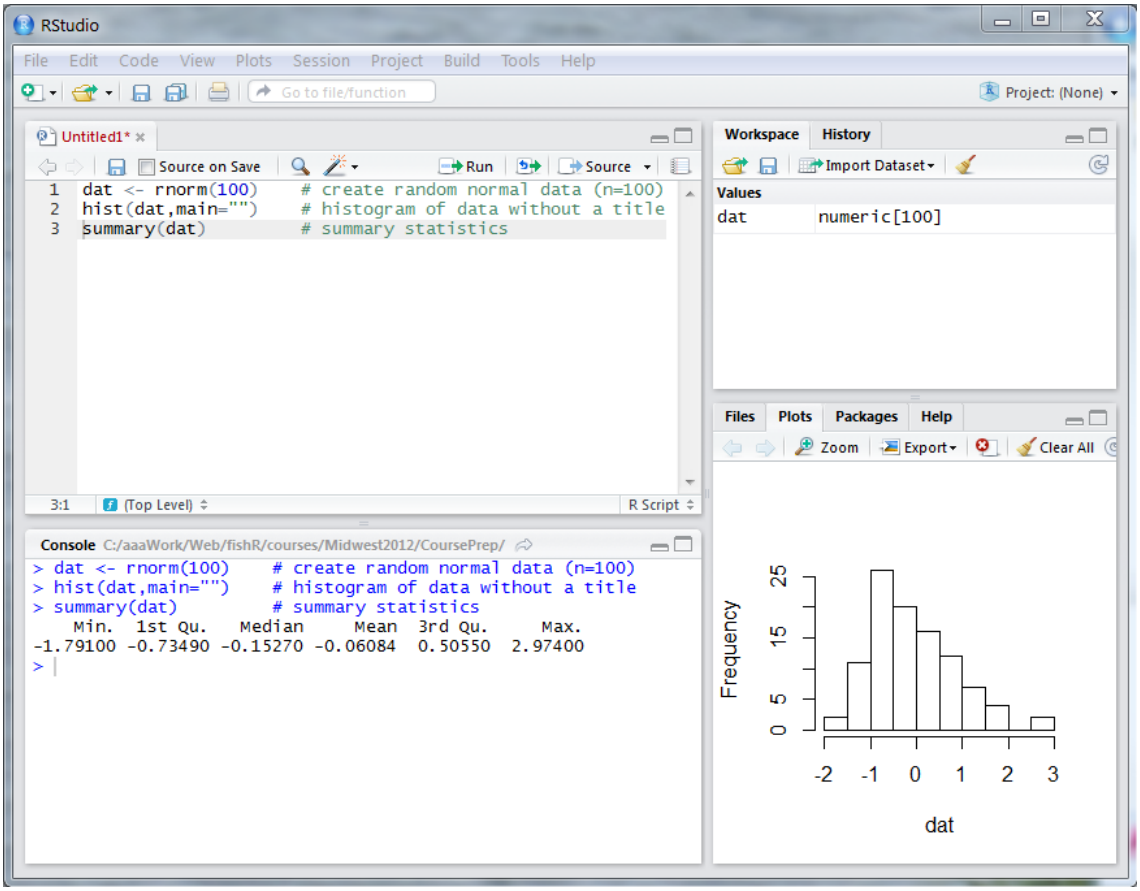


Figure 1. Example of the RStudio layout with the *Script Editor* in the upper-left panel, *Console* in the lower-right panel, the *Workspace* tab shown in the upper-right panel, and the *Plot* tab shown in the lower-right panel.

The upper-right panel contains two tabs – *Workspace* and *History*. Many items listed under the *Workspace* tab can be double-clicked to open them for viewing as a tab in the *Script Editor*. The *History* tab simply shows all of the commands that you have submitted to the *Console*.

The lower-right panel contains four tabs – *Files*, *Plots*, *Packages*, and *Help*. The *Plots* tab will show the high-level plots produced by commands submitted to the *Console*. One can cycle through the history of constructed plots with the arrows on the left side of the plot toolbar and plots can be saved to external files using the "Export" tab on the plot toolbar (Figure 1).

A list of all installed packaged is seen by selecting the *Packages* tab (Figure 2). A package can be loaded by selecting the box to the left of the package name, though I suggest that packages be loaded via the `library()` function (to be discussed elsewhere) in an R script rather than through this dialog box. Help for each package can be obtained by clicking on the name of package. The help will then appear in the *Help* tab.

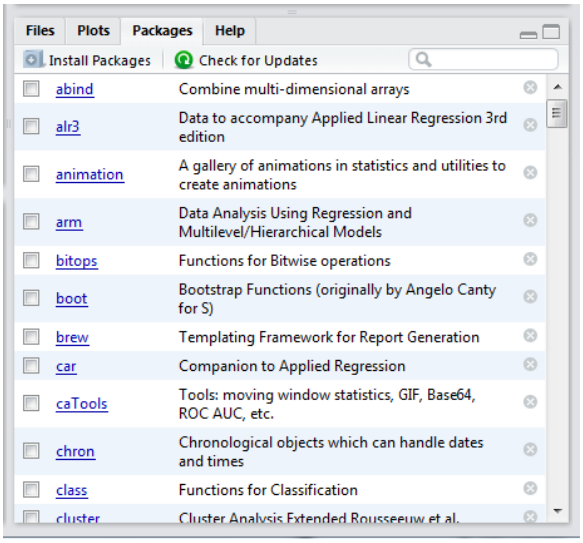


Figure 2. Example of the RStudio *Packages* tab.

Basic Usage

Our primary interaction with RStudio will be through developing R scripts in the *Script Editor*, submitting those scripts to the *Console*, and viewing textual or tabular results in the *Console*, and graphical results in the *Plot* panel. In this section, I briefly introduce how to construct and run R scripts in RStudio.

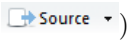
One opens a blank file for an R script by selecting the "New" icon ( ) and then `R Script`; selecting the `File` menu, `New` submenu, and `R Script` item; or with `<CTRL>` + `<Shift>` + `N`. In the ensuing tab of the *Script Editor*, type the three lines exactly as shown below[1].

```
dat <- rnorm(100)     # create random normal data (n=100)
hist(dat,main="")     # histogram of data without a title
summary(dat)          # summary statistics
```

One must now "submit" these commands to the *Console* to perform the requested calculations. These commands can be submitted in a variety of ways:

- Put the cursor on the first line in the *Script Editor* and press the "run" icon (Run). This will submit the first line to the *Console* and move the cursor to the second line in the *Script Editor*. Pressing the "Run" icon will now submit the second line. And so on.

[1] For the moment, don't worry about what these lines "do."

- Select the "down arrow" on the "Source" icon () and select `Source with Echo` (alternatively, press `<CTRL>` + `<Shift>` + `<Enter>`). This will simultaneously submit all commands to the *Console*.
- Select all commands in the *Script Editor* that you wish to submit and then press the "run" icon.

The RStudio layout after using the first method is shown in Figure 1.

The R Script in the *Script Editor* should now be saved by selecting the `File` menu and the `Save` item (alternatively, pressing `<CTRL>` + `S`). RStudio can now be closed (do NOT save the workspace) and re-opened. The script can then be re-opened (choose the `File` menu and the `Open file ...` submenu if the file is not already in the *Script Editor*) and re-submitted to the *Console* to exactly repeat the analyses[2].

[2]Note that the results of commands are not saved in R or RStudio; rather the commands are saved and re-submitted to re-perform the analysis.