How to get familiar with **RStudio** quickly

**Menu bar**

The Menu bar is composed of several tools to manage R session, code editing, project settings, and package development. Most of the utilities that sit in the Menu bar can be found in RStudio panels, or using keyboard shortcuts.
General options

From the Menu bar you can access the General options panel (Rstudio > Options), and update default settings like R default directories and backup policies ➊, font size ➋, or general layout ➌.
RStudio tips

Pane layout

Default settings may not be your taste, so here are some suggestions. Use left and right vertical frames to display code editor (Source) ➊ and R prompt (Console) ➋, then split each frame horizontally with informations on R code and objects on one side ➌ and interactive panels on the other side ➍.
RStudio tips

Version control

RStudio automatically detect Git and Subversion clients ⃣ if they are installed on your system. However, you will probably need to enable version control for projects ⃣. If you have an SSH public key (e.g., to interact with GitHub, BitTorrent, or other servers), it will be detected as well ⃣; otherwise, you can create one from this panel.

Note: In case you installed Git from Homebrew on a Mac, then you will have to update the path by editing the file ~/.rstudio-desktop/monitored/user-settings/user-settings with

\[
vcsGitExePath="/usr/local/bin/git"
\]
Reporting

Automatic reporting is controlled by Sweave or knitr. Be sure to check that the latter ❶ is enabled, and update LaTeX settings if needed, including default engine ❷ and previewer ❸.

![Screenshot of RStudio options window with Sweave selected]

Program defaults (when not in a project)

Weave Rnw files using: ❶

Typeset LaTeX into PDF using: ❷

NOTE: The Rnw weave and LaTeX compilation options are also set on a per-project (and optionally per-file) basis. Click the help icons above for more details.

LaTeX editing and compilation

☑ Clean auxiliary output after compile

☐ Enable shell escape commands

☐ Insert numbered sections and subsections

PDF preview

Preview PDF after compile using: ❸

☑ Always enable Rnw concordance (required for synctex)
File browser

RStudio features a built-in file browser that allows to navigate across your hard disk, create or delete new folder, and set any directory as the current working directory.
Package manager

RStudio includes a package manager, which allows to load package ① with a simple click, to install new package(s) ②, or to update all packages at once ③.
Online help

The help panel includes an interactive previewer with in-page or full-system searching utilities for specific topics, and a built-in viewer of R HTML documentation (Manuals, Reference, and Miscellaneous Material). It is also possible to navigate back and forth using History buttons.

Risk Factors Associated with Low Infant Birth Weight

Description
The birthwt data frame has 189 rows and 10 columns. The data were collected at Baystate Medical Center, Springfield, Mass during 1986.

Usage
birthwt

Format
This data frame contains the following columns:
low
indicator of birth weight less than 2.5 kg.
When starting from scratch, RStudio shows the R console, objects and history browser, and other tools. Note that if a panel is minimized before RStudio is closed, it will remain as is next time RStudio is started. RStudio will use the default version of R that is recognized by your system. If you have multiple versions of R, you will have to select the one to use before launching RStudio (e.g., using RSwitch for Mac users). Likewise, the default localization will be chosen. See instructions in the R Installation and Administration manual on CRAN to get help with "Internalization and Localization".
Creating a new R script file

Creating an R script file is done via the Menu bar (File ▸ New File ▸ R Script), or by clicking on the corresponding icon ❶ on the toolbar. The same approach will be used to create R Markdown file, or R slideshow. Be careful, the default keyboard shortcut is ?⇧⌘N, and not ⌘N as in other applications.

Any new R script file will be saved automatically (even if RStudio happens to crash), and a default extension ( *.R ) will be added automatically when saving manually the file.

To open an existing R script, click on the folder icon ❷ and select your file using the dialog box. RStudio can also open basic text files (with extension like, e.g., *.txt, *.dat, *.csv).
**Editing code**

The code editor is composed of a main window with its own toolbar. All R expressions are highlighted depending on the type of R objects (literals, numbers, functions, etc.), and opening/closing parentheses are matched automatically. There is also an auto-indentation mechanism, which is useful when constructing complex expressions, writing functions, or simply using `for` loop. It is possible to run any command by clicking the corresponding button; this will send the current line or a selection to the R console. Alternatively, we can use the shortcut (`⌘⏎` or `Ctrl⏎`). To run the same code again, we click on the button next to the preceding one. It is also possible to send the whole buffer to the R console by "sourcing" it silently using the corresponding button; we will need to select "Source with Echo" to display intermediate results generated by R.
Code utilities

The Code editor toolbar also features various utilities that can be used when working with R code. It is possible to get help on a specific R function by clicking on F1 ➊ when the cursor is in the middle of R function name, or when it is highlighted. With user-written functions, pressing F2 will bring the cursor back to the function directly ➋. There is also a searching utility that allows to find and/or replace ➌ specific part of R code in the current buffer. Also, code completion is available by pressing the TAB key at any time when we start writing the name of a built-in or user-written function. Notebooks ➍ are a special feature and we will talk about them later.
R object browser

When variables or functions are created in the workspace (e.g., by sending R code to the console, from the Code editor or by typing commands directly in the Console), they are listed in the Environment panel 1, under "Global Environment" (other environments can be selected, but this is not relevant to the purpose of visualizing available R objects in the user workspace). This object browser provides almost the same information than what `str()` will give when typed in the Console, i.e., the size or dimension of the R variable in case of vectors and rectangular data structure (matrix and data frame), and the first 6 elements of that object 2.
RStudio tips

R object viewer

When a data frame is available in the workspace, basic information as those returned by `dim()` are displayed in the Object browser. Moreover, it is possible to display the data in a spreadsheet-like viewer by double clicking on the line where the data frame appears, or simply on the grid-like icon at the right. From the Console, the same result could be obtained by typing `View(d)`. Finally, a summary (as produced by `str()`) of variables contained in this data frame can be viewed by clicking on the small arrow button.

From the same panel, it is possible to save an image of the R workspace, to load an existing one, or to clear the current workspace.

![Image of RStudio interface showing the Object browser and the data frame viewer.](image-url)
Importing external files

Instead of using read.table(), read.csv(), or other related commands, RStudio offers a small utility to import external text-based files. Besides a basic preview ➊ of the content of the selected file (Import Dataset ▸ From Text File), the main options from read.table() can be updated depending on the type of data source. A live preview ➋ is offered to check that the corresponding data frame will match the raw data. The name of the data frame can also be given in the corresponding text box ➌.
RStudio includes a specific panel to access R history, that is all commands that have been typed during the session. A searching utility ❼ is available to look for past commands. Any command that is available in the history can be sent to the Code editor ❷ or R Console ❸. Finally, it is possible to remove specific entries ❹ or to clean up all history ❺.