

RStudio tips

How to get familiar with [RStudio](#) quickly

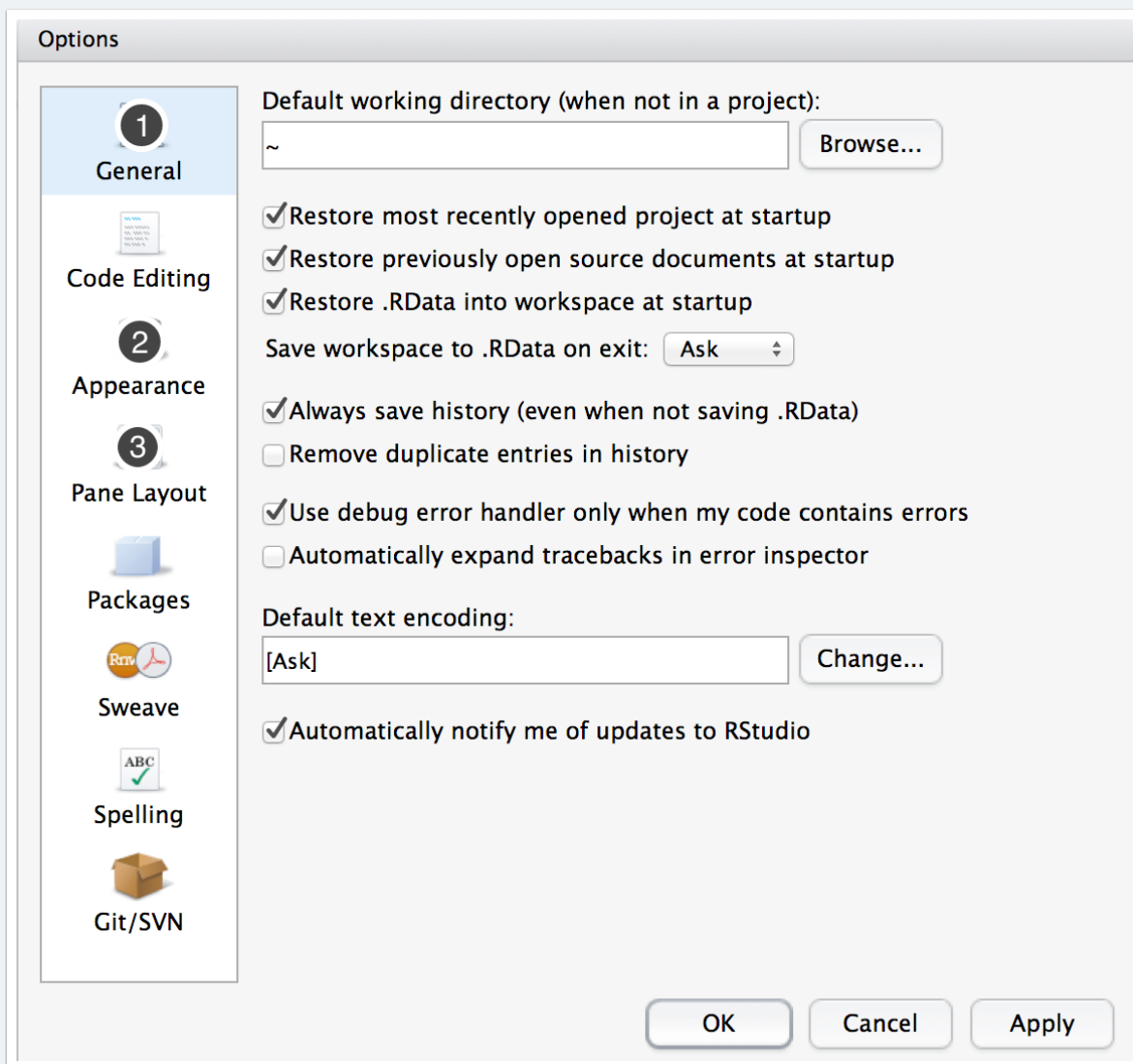
Menu bar

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.



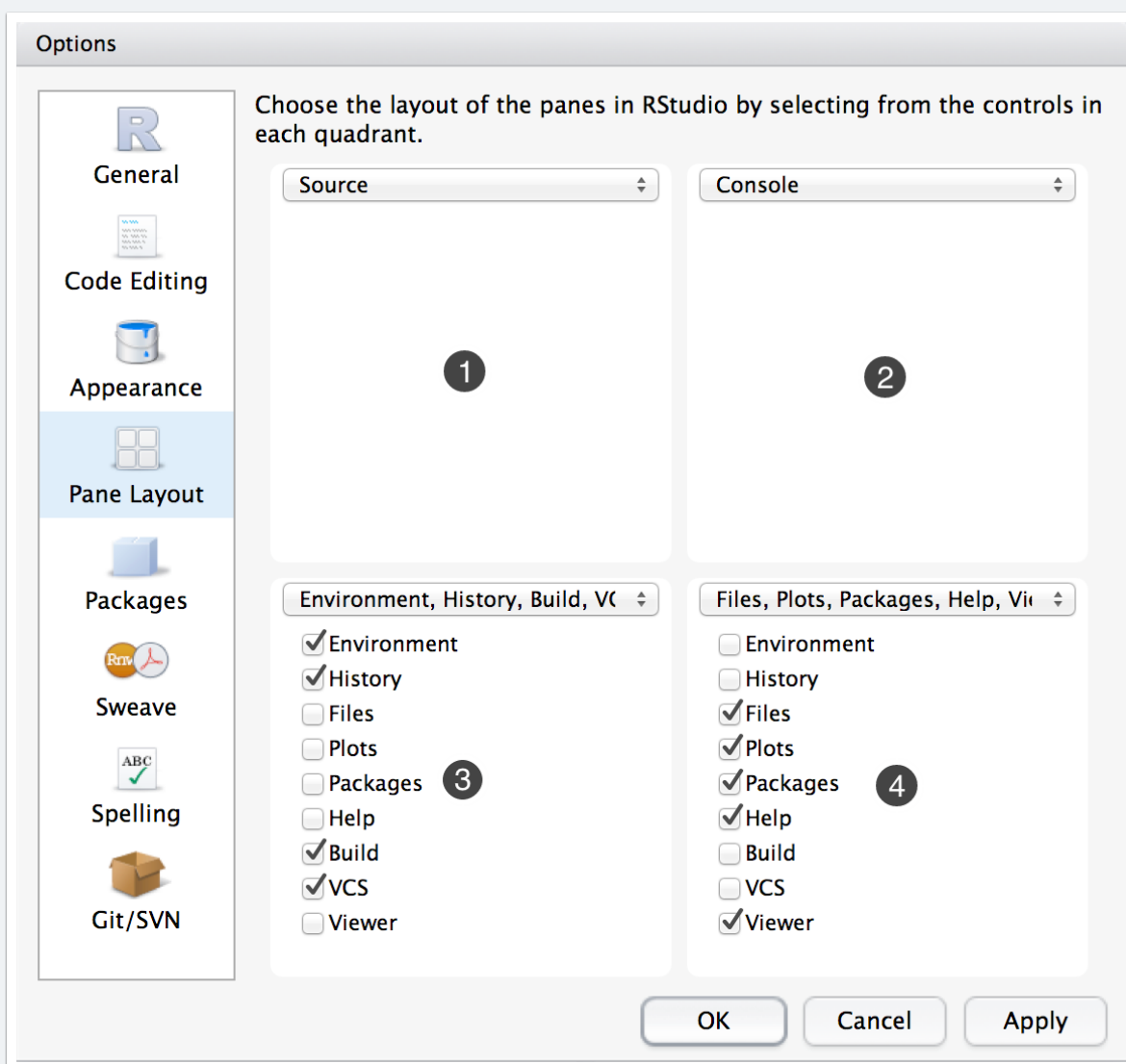
General options

From the Menu bar you can access the General options panel (Rstudio ▸ Options), and update default settings like R default directories and backup policies ❶, font size ❷, or general layout ❸.



Pane layout

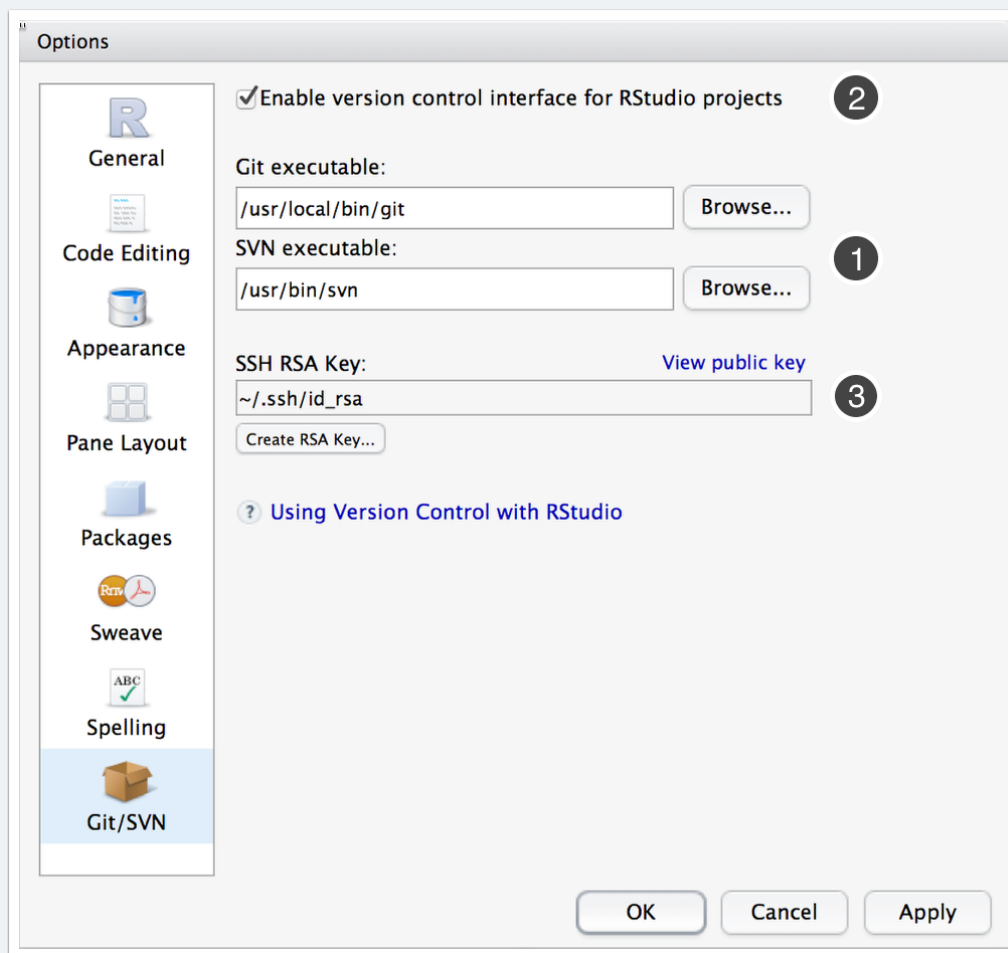
Default settings may not be your taste, so here are some suggestions. Use left and right vertical frames to display code editor (Source) ❶ and R prompt (Console) ❷, then split each frame horizontally with informations on R code and objects on one side ❸ and interactive panels on the other side ❹.



Version control

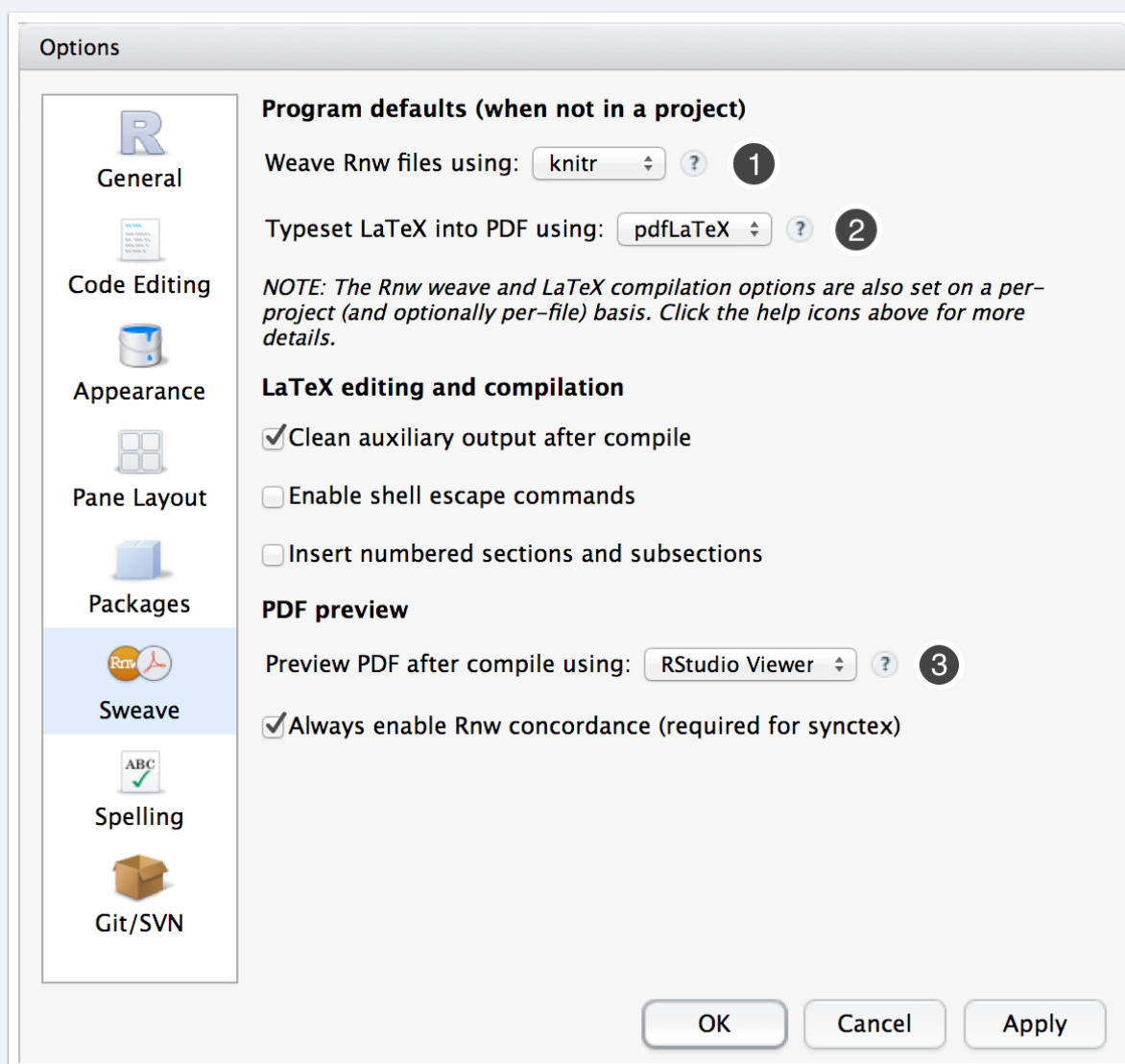
RStudio automatically detect Git and Subversion clients ❶ if they are installed on your system. However, you will probably need to enable version control for projects ❷. If you have an SSH public key (e.g., to interact with GitHub, BitTorrent, or other servers), it will be detected as well ❸; otherwise,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



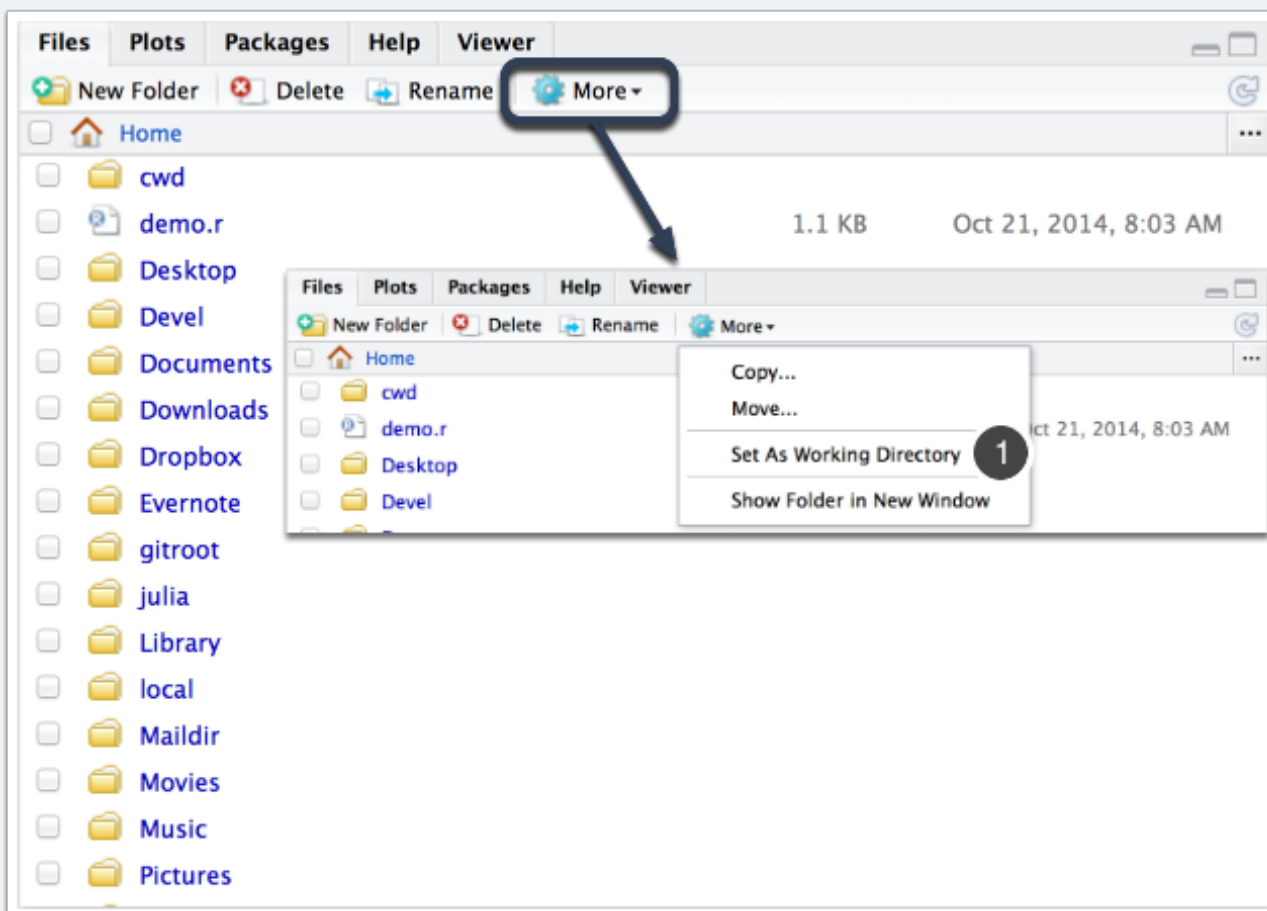
Reporting

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.



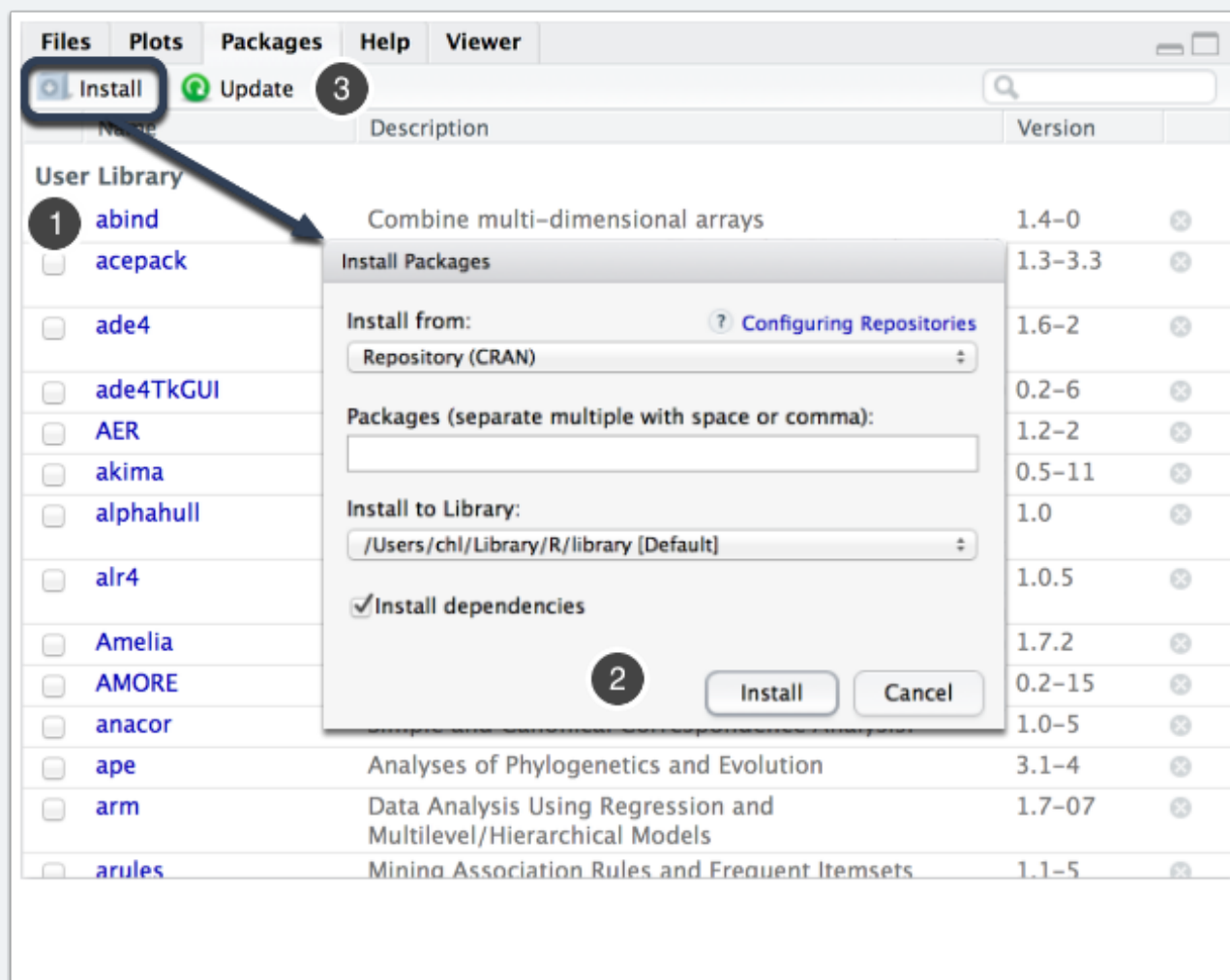
File browser

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 ❶.



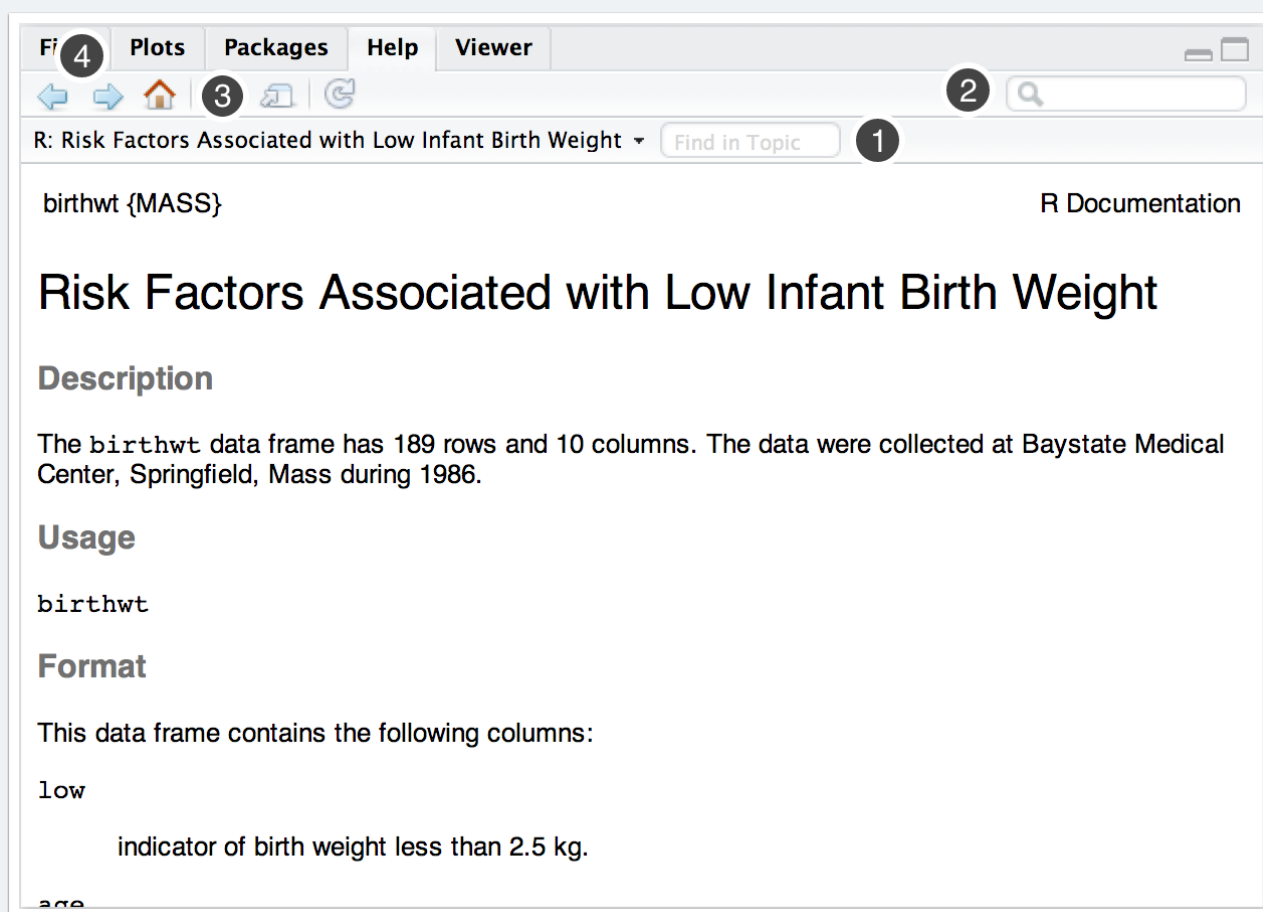
Package manager

RStudio includes a package manager, which allows to load package ❶ with a simple click, to install new package(s) ❷, or to update all packages at once ❸.



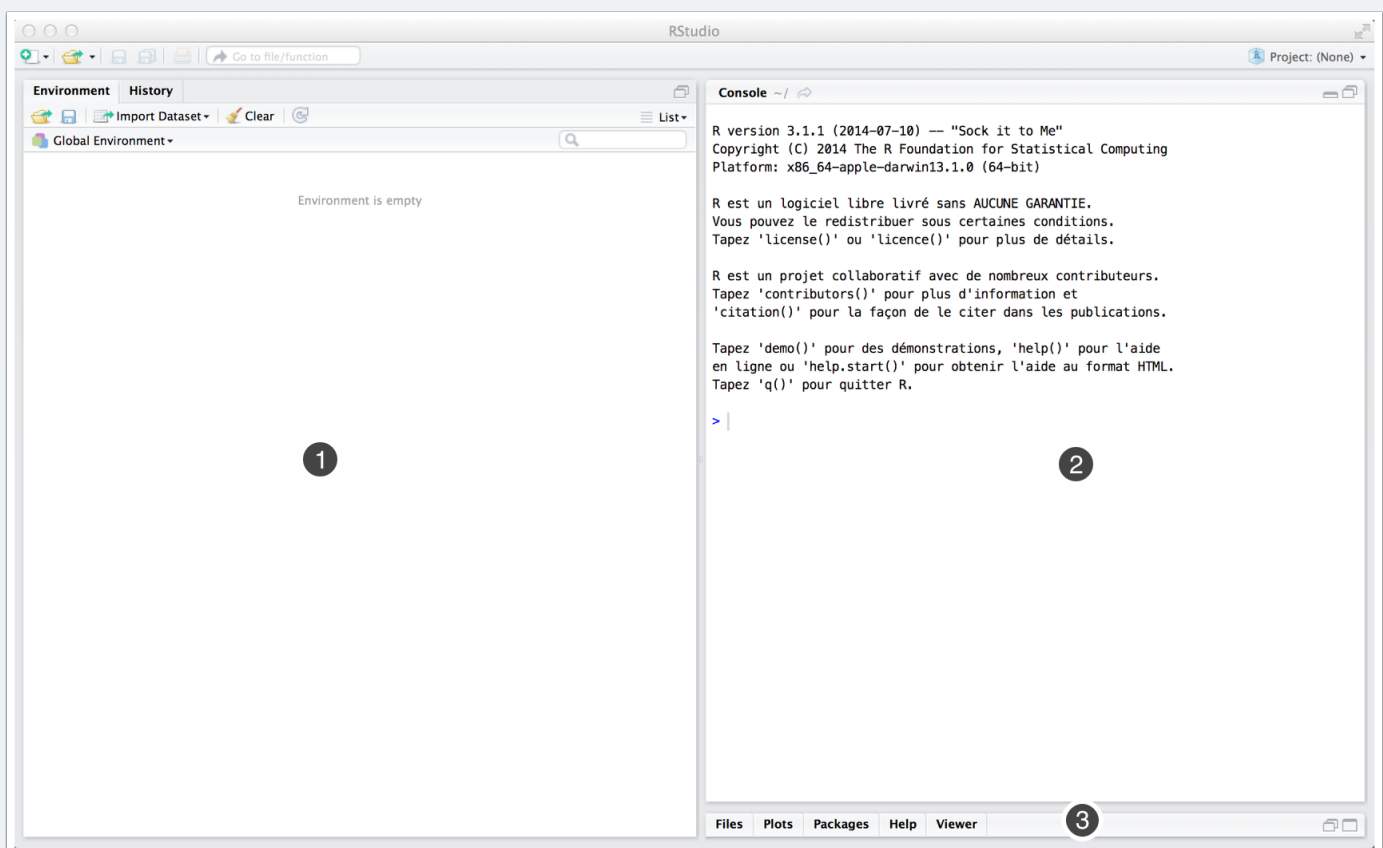
Online help

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



RStudio UI

When starting from scratch, RStudio shows the R console ❷, objects and history browser ❶, and other tools ❸. 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".

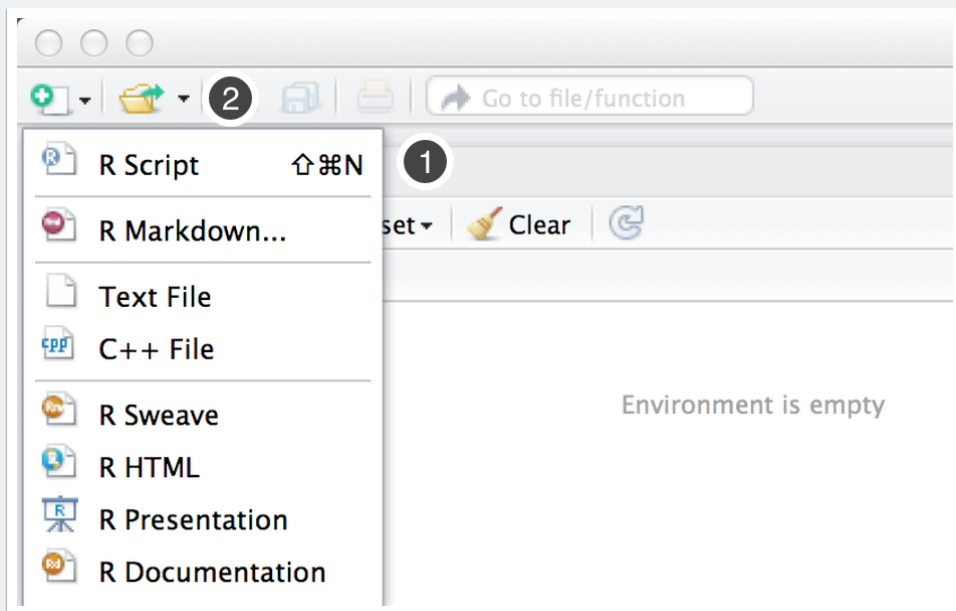


Creating a new R script file

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.

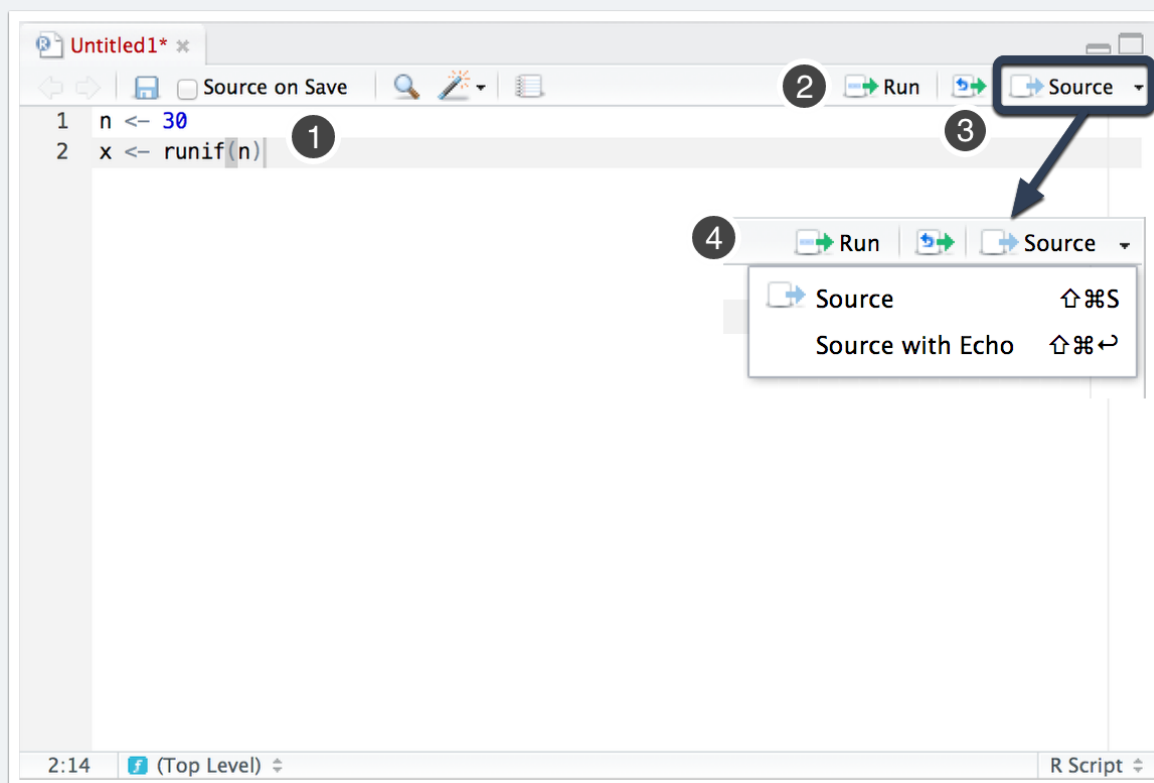
Any new R script file will be saved automatically (even if RStudio happens to crash), and a default extension (`*.R`) will be added automatically when saving manually the file.

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`).



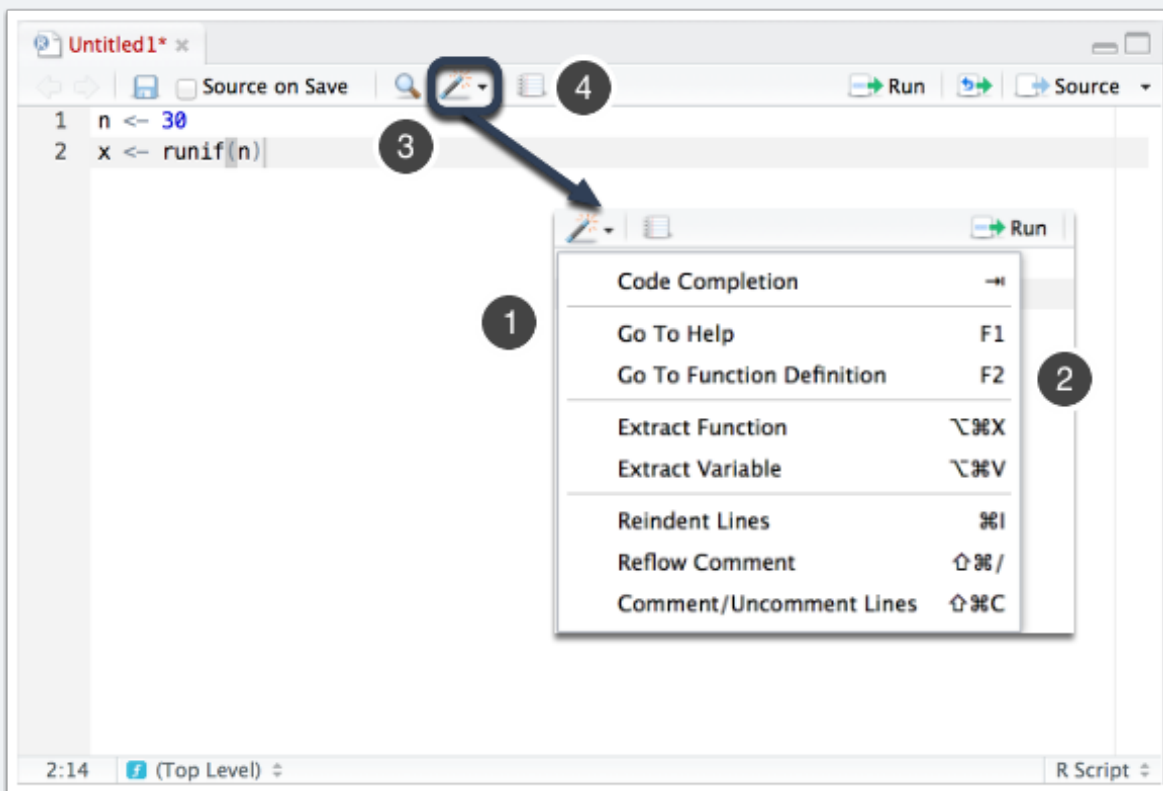
Editing code

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 ❶. There is also an auto-indentation mechanism, which is useful when constructing complex expression, writing functions, or simply using `for` loop. It is possible to run any command by clicking the corresponding button ❷; this will send the current line or a selection to the R console. Alternatively, we can use the shortcut $\text{⌘} \leftarrow$ (or $\text{Ctrl} \leftarrow$). To run the same code again, we click on the button next to the preceding one ❸. It is also possible to send the whole buffer to the R console by "sourcing" it silently using the corresponding button ❹; we will need to select "Source with Echo" to display intermediate results generated by R.



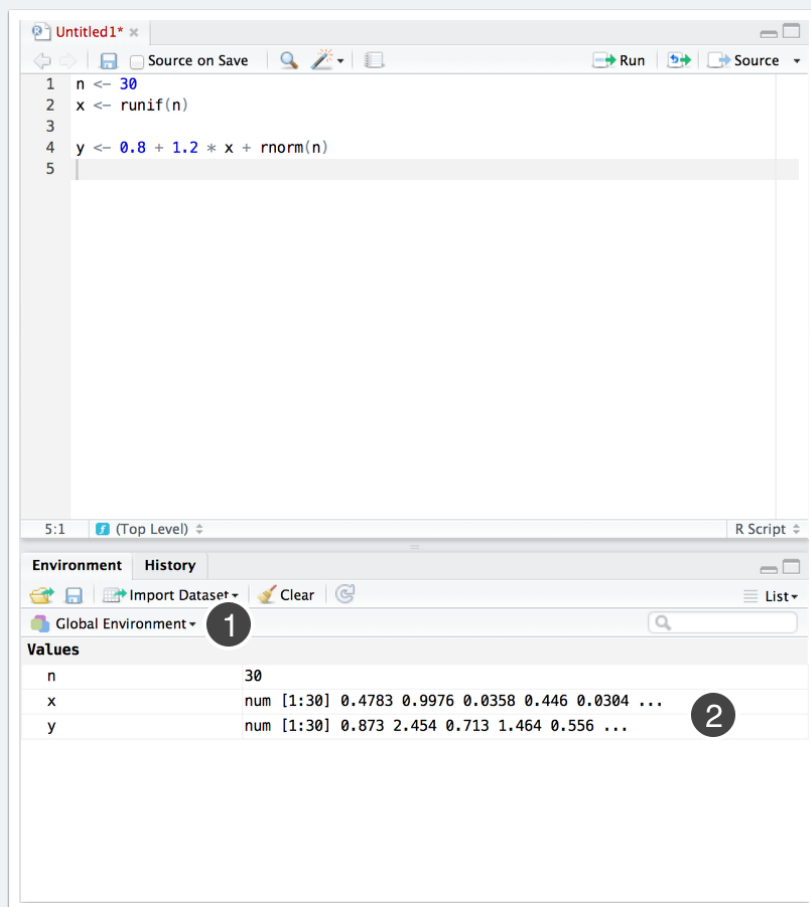
Code utilities

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 ❶ 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 ❷. There is also a searching utility that allows to find and/or replace ❸ 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 ❹ are a special feature and we will talk about them later.



R object browser

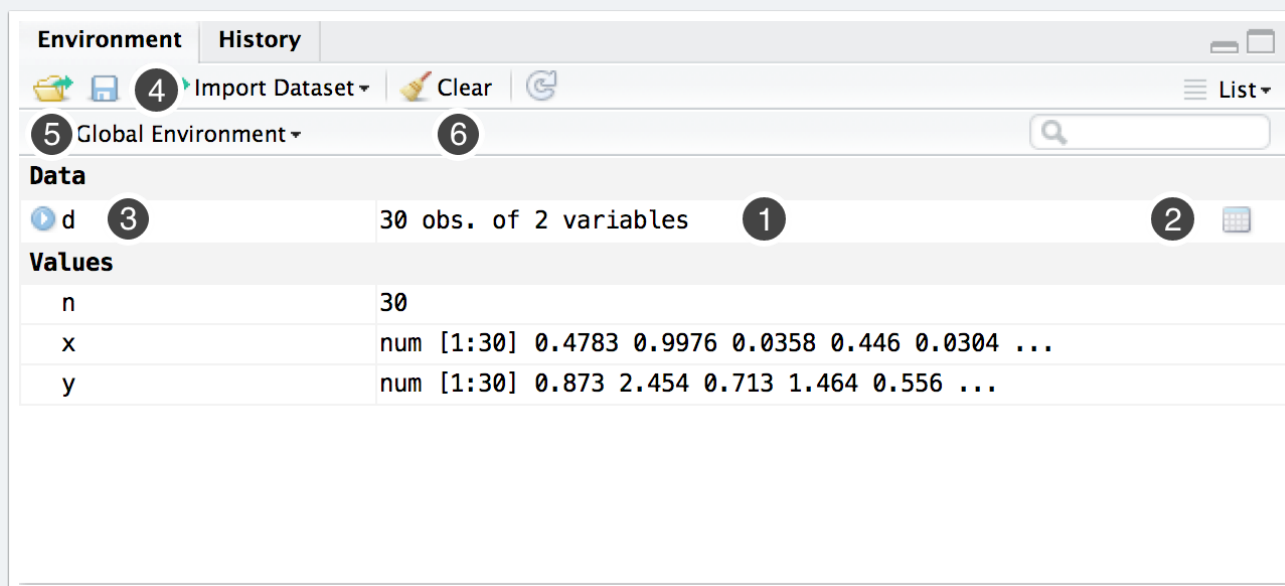
When variables or functions are created in the workspace (e.g., by sending R code to the console, from the Code editor or by typing commands directly in the Console), they are listed in the Environment panel ❶, under "Global Environment" (other environments can be selected, but this is not relevant to 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 ❷.



R object viewer

When a data frame is available in the workspace, basic information as those returned by `dim()` ❶ 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 ❷. 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 ❸.

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



The screenshot shows the RStudio Environment pane with the following elements:

- Environment** and **History** tabs.
- Buttons: Import Dataset (❹), Clear (❻), and List (☰).
- Global Environment (❺) with a search bar (❶).
- Data** section showing a data frame `d` (❸) with 30 observations and 2 variables (❶). A grid icon (❷) is visible to the right.
- Values** section showing the first few rows of data for variables `n`, `x`, and `y`.

Variable	Value
n	30
x	num [1:30] 0.4783 0.9976 0.0358 0.446 0.0304 ...
y	num [1:30] 0.873 2.454 0.713 1.464 0.556 ...

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 ❶ 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 ❷ 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 ❸.

The screenshot shows the 'Import Dataset' dialog box in RStudio. The 'Name' field is set to 'birthwt' (marked with ❸). The 'Input File' section shows a preview of the raw data (marked with ❶). The 'Data Frame' section shows the resulting data frame with columns V1-V10 (marked with ❷). The 'Import' button is visible at the bottom right.

V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
0	19	182	2	0	0	1	0	2523	
0	33	155	3	0	0	0	3	2551	
0	20	105	1	1	0	0	1	2557	
0	21	108	1	1	0	0	1	2594	
0	18	107	1	1	0	0	1	2600	
0	21	124	3	0	0	0	0	2622	
0	22	118	1	0	0	0	1	2637	
0	17	103	3	0	0	0	1	2637	
0	29	123	1	1	0	0	1	2663	
0	26	113	1	1	0	0	0	2665	
0	19	95	3	0	0	0	0	2722	
0	19	150	3	0	0	0	1	2733	
0	22	95	3	0	0	1	0	2751	
0	30	107	3	0	1	0	1	2750	
0	18	100	1	1	0	0	0	2769	
0	18	100	1	1	0	0	0	2769	

R history manager

RStudio includes a specific panel to access R history, that is all commands that have been typed during the session. A searching utility ❶ 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.

