

Using R with SciViews (part 3)

Visual Guide to R (3)

Ph. Grosjean <phgrosjean@sciviews.org>

May 5, 2010

1 The R reference toolbox

The **R reference** toolbox provides a series of “snippets” of R code easily configured through dialog boxes. These snippets are sorted into logical categories (data manipulation, graphs, hypothesis tests, ...). All entries use dialog boxes to configure them (selection of a pertinent subset of arguments through GUI widgets in the dialog box). All widgets in these dialog boxes have a history mechanism, so that you can retrieve previously entered values easily. The history is shared between different dialog boxes, that is, when you select a *data frame* in a dialog box to summarize your data, the same *data frame* sits on top of the history list on another dialog box, say, to plot your data... This is a convenient and natural way to define and reuse a context pertaining to your current analysis. Even better: these histories are *persistent* lists. It means that, when you exit Komodo and restart it later on, the same histories are still available!

Snippets of code represent a simple, but powerful mechanism to store solutions of any complexity (from one function call, up to a whole R script). They also contribute to lower the learning curve of R, or any R package for which you have build a carefully selected series of snippets. Indeed, snippets do show you *by the example* how to perform a given task in R, but on the contrary to `example()` or `demo()`, the traditional mechanisms in R to illustrate how some code is working, you apply snippets to **your own** data. Furthermore, snippets are easier to find than corresponding R function, because they are ranged in a tree with a logical organization (manipulate data frames, plotting, modeling, hypotheses testing, ...).

Snippets with dialog boxes are also not unlike the R Commander, RKward, or Deducer *menu-dialog box* approach, but they have an important difference: they are designed to create R code, rather than to run a process from a GUI. The distinction is subtle, but it applies at three stages:

- ▷ The dialog box visually presents the logic used in R to parameterize functions, that is: `argument = value`. Thus, the dialog box presents a series of lines, starting by an **argument** name (often, but not always exactly the same argument name as in the corresponding R function), followed by an equal sign and a text box where you can provide the **value**. As an example, a snippet for calculating the mean as `mean(x, na.rm = TRUE/FALSE)` will present a dialog box with two lines: **x** prompting for an object, and **na.rm** prompting for a logical value to indicate if one would consider missing values or not in the calculation. In Deducer, RKward, or R Commander, such a dialog box would present somewhere a checkbox with a label like “ignore missing values”. This is very nice, does not help much the end-user to make the connection in their mind between this checkbox widget and the corresponding code `na.rm = TRUE/FALSE` in R. In the SciViews snippet’s dialog

boxes, the GUI looks much closer to the final code it creates: you have simply a label with `na.rm =` followed by a text box proposing **TRUE** or **FALSE** in its history list by default (but you can type anything else, including a valid R expression returning **TRUE** or **FALSE**). The connection between the GUI and the R code is thus much more evident!

- ▷ Great care is taken to make sure that the R code produced by any snippet is clean and well-formatted. As a simple example, still considers calculating the mean of a univariate sample. In R Commander, you got the following code:

```
numSummary(iris[,"Petal.Length"], statistics=c("mean"), quan-
tiles=c(0,.25, + .5,.75,1))
```

In RKward, the situation is even worse: the corresponding code is:

```
local({
## Prepare
## Compute
vars <- list (substitute (iris[["Petal.Length"]]))
results <- data.frame ('Object'=rep (NA, length (vars)))
for (i in 1:length (vars)) {
results[i, 'Object'] <-
rk.get.description (vars[[i]], is.substitute=TRUE)
var <- eval (vars[[i]], envir=globalenv())
# fetch the real object
# we wrap each single call in a "try" statement to always con-
tinue on errors.
results[i, 'mean'] <- try (mean (var, trim = 0.00, na.rm=TRUE))
}
## Print result
rk.header ("Descriptive statistics", parameters=list (
"Trim of mean", 0.00))
rk.results (results)
})
```

Sic! How would an end-user learn the “easy” way to calculate the mean of a vector in R with such an “example” code? The corresponding SciViews snippet produces arguably the cleanest R code in this case, which is:


```
mean(iris$Petal.Length, na.rm = TRUE)
```

- ▷ Finally, and not the least important point, when you click 'OK' in a SciViews snippet's dialog box, you only cook some R code that is inserted in a script editor. You are, then, obliged to review this code and to **actively** run it (it is enough to hit **Ctrl+Enter** which runs the current line of code in R). This helps reluctant users that tend to stick to the GUI and to forget the script to realize that they are actually working with some R code.

1.1 Installation of the R reference toolbox

The R reference toolbox is part of the **SciViews-K** Komodo plugin on the [SciViews](#) web site. You install it in two easy steps in Komodo Edit/IDE (see the ReadMe file shipped with the plugin for more detailed installation instructions):

1. Install the Sciviews-K plugin in Komodo Edit/IDE.
2. After restarting of Komodo, go to menu R ▷ (Re)install R toolboxes. Restart Komodo another time, to make sure that custom toolbars provided in the toolbox are refreshed.

If your toolbox panel at right is not visible, you can display it by selecting the Komodo menu View ▷ Tabs & Sidebars ▷ Toolbox, the button  , or the shortcut Ctrl-Shift-L.

In case you have several versions of the **R reference** toolbox, eliminate old ones by right-clicking on them and selecting Delete in the context menu and confirm you want to delete them from the toolbox.

1.2 R code snippets

Most tools in the **R reference** toolkit are *snippets*. These are short pieces of code that you can insert in a document currently edited in Komodo. They may have customizable parts. In this case, a dialog box prompts for one or several values. Figs. 1.1-1.5 show a typical sequence of use of a snippet. But before you try this, you must have **R started** from within Komodo, and you must be editing **R code** in a Komodo tab (i.e., you need to be in a correct context for creating and executing R code):

- ▷ Once Komodo is started, you must start you preferred R interpreter from **within Komodo** to make sure a connection is established. To do so, go the the Komodo menu R ▷ Start R. If the menu does not exists, it means the **SciViews-K** plugin is not installed correctly. If R does not start, you would be redirected to the Komodo preference box and you can parameterize the way you want to start R in the SciViews-K ▷ R Interpreter topic in ther preferences tree at left.
- ▷ Also, since the snippet will add some R code in the currently edited file, you must make sure you are editing a **R script** (a file usually ending with a '.R' extension). The snippet of code will be inserted at the current caret location in you file. So, make sure to place it where you want your code inserted (usually, at the end on a new, empty line).
- ▷ Now, you are ready to play with your first SciViews snippet...!

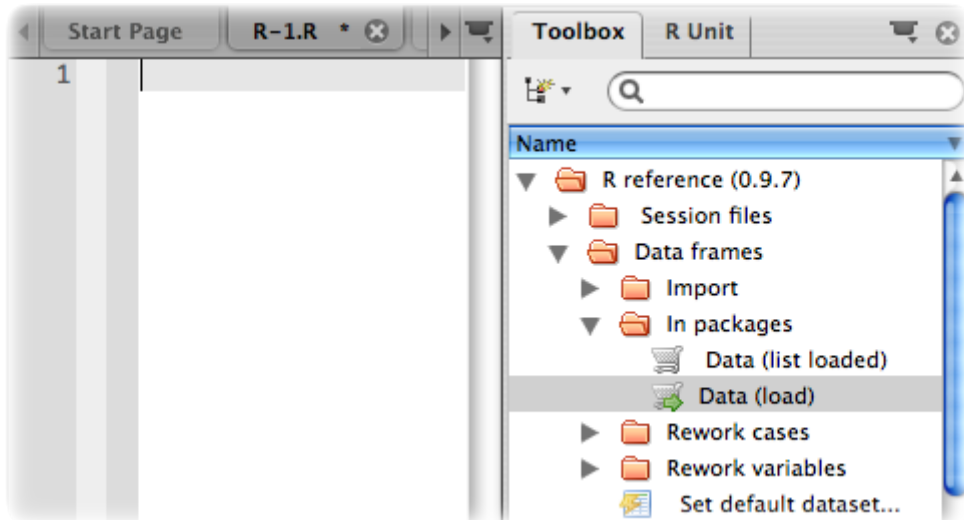


Figure 1.1: Use of snippet, step 1. Open the toolbox, navigate to the snippet you want (here R reference>Data frames>In packages>Data(select)), and double-click on it. If you have any doubt about what a snippet can do, you can select it and hit Alt-F1 to call contextual help on this particular snippet, if there is any help page available (make sure R is started from within Komodo to be able to display R help pages!).

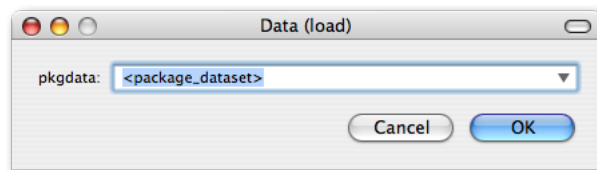


Figure 1.2: Use of snippet, step 2. If there is a configurable part in the snippet (usually), a dialog box opens, prompting for the argument(s). Here, you have to provide the name of the dataset to load.

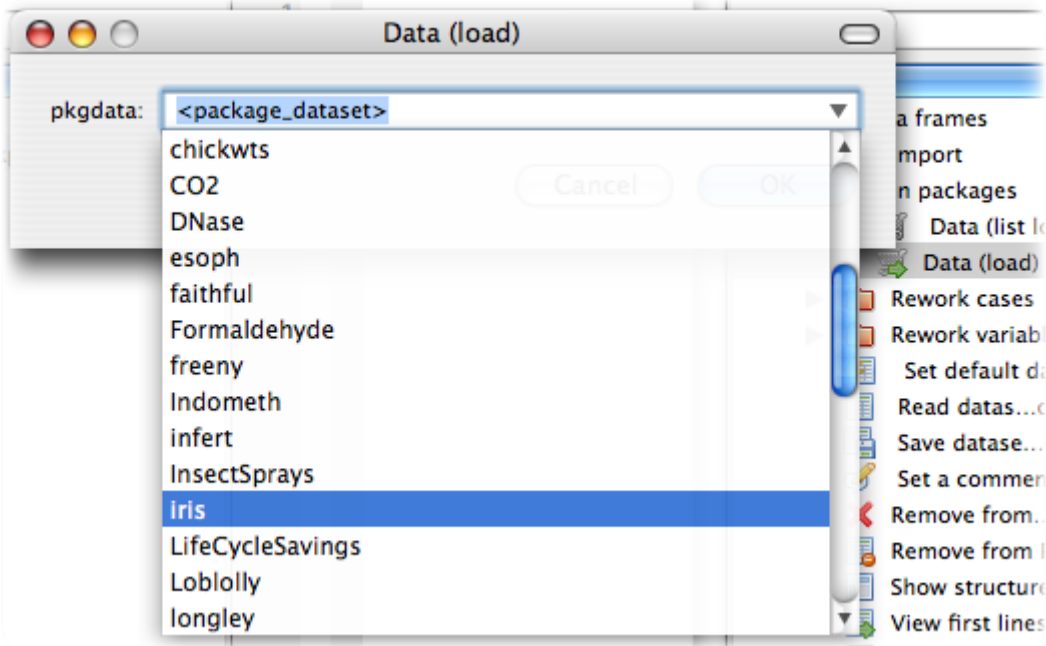


Figure 1.3: Use of snippet, step 3. Fill the entries in the dialog box and click OK. Entries have a gray arrow at right you can click to get either sensitive propositions, or a history of values you have previously entered for the same topic. Select `iris` in the dropdown list, then click the OK button in the dialog box.

1 The R reference toolbox

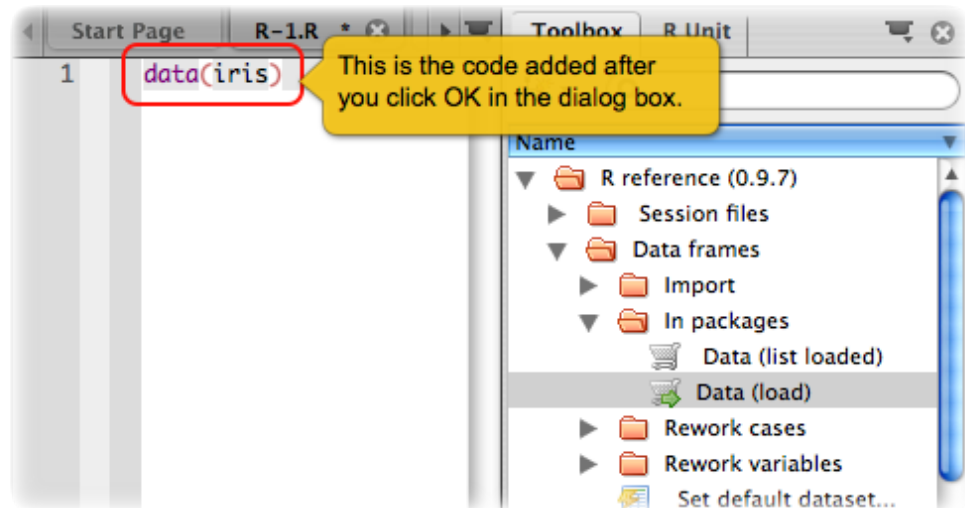


Figure 1.4: Use of snippet, step 4. The corresponding piece of code is added in the editor. You can review and correct it there if you like, or hit Ctrl-Enter to run that line of code in R (make sure that R is started before you try running any code, using the menu entry R▷Start R in Komodo).

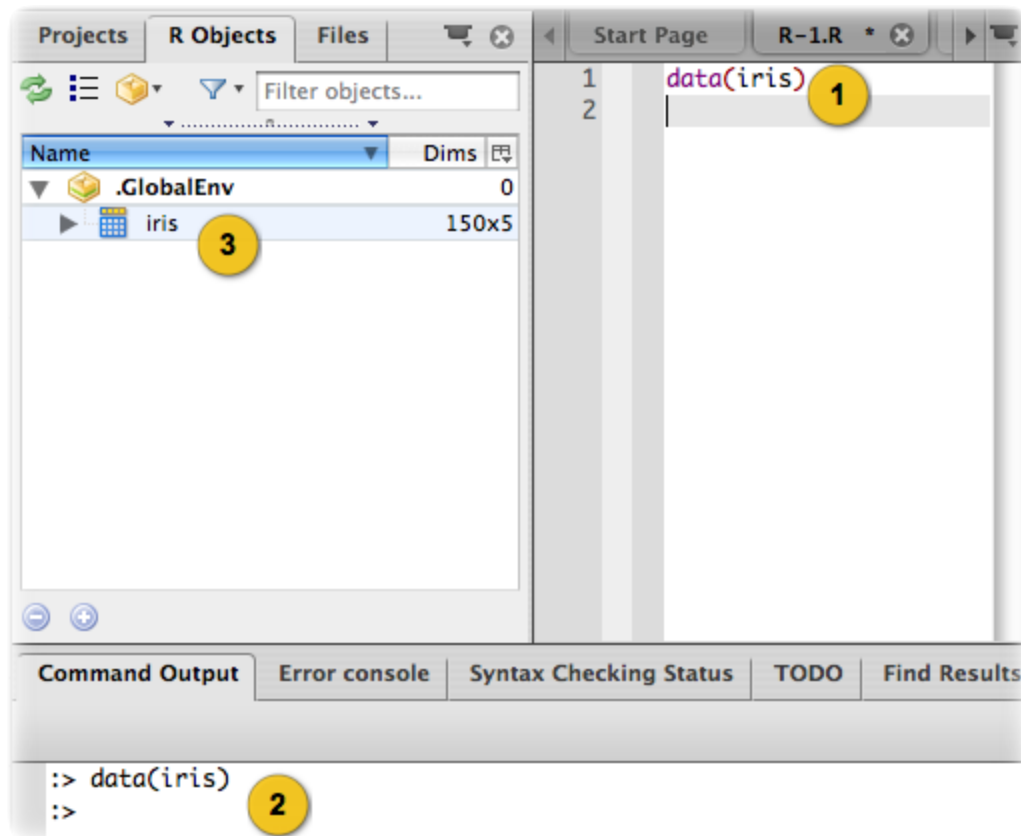


Figure 1.5: Use of snippet, step 5. After hitting **Ctrl-Enter** (1), the cursor moves to next line in the editor and the line of code is run in R. Check in the bottom pane, **Command Output**, to make sure execution is correct (2). In the present case, R outputs nothing special, but the *iris* data frame is loaded in memory, and you should see it in the **R objects explorer** in the left pane (3).

1.3 What next

You should now explore the various snippets provided in the **R reference** toolbox. But before you do so, you should also know:

- ▷ that you can speed up your work and fill history list for entries with a correct context by **making objects active**. As an illustration, if you want to make the *iris* data set active, use the shortcut **Ctrl+Shift+A** and select *iris* in the dialog box. When you click 'OK' in this dialog box, nothing seems to happen, but look at the status bar at the very bottom of the Komodo dialog box. You will read **... data: iris...** What happens is that all pertinent argument histories are recalculated, using the new context. That is, arguments prompting for a numerical variable will propose the four numerical variables of the *iris* dataset on top of the

1 The R reference toolbox

history list (`Sepal.Length` , `Sepal.Width` , `Petal.Length` and `Petal.Width`). Similarly, where you should use a factor variable as an argument, you will find `Species` , the only factor variable in the iris dataset, on top of the history list of that argument. If you change the variables in your data frame, or if you switch to another dataset, do not forget to “activate” it in order to update histories...

- ▷ that you can easily manage R scripts, custom datasets and reports in sessions from within the **R reference** toolbox. Play with entries in the `Session` files subdirectory to learn these features.
- ▷ that many snippet’s dialog boxes provide plenty of help: (1) a short description on top of the dialog box explaining what you are about to do, (2) tooltips that appear automatically when you move the mouse pointer on top of the corresponding text box, and (3) a help button at the bottom of the dialog box, that displays the corresponding R help page, a R wiki page, or another help material on the web, ...
- ▷ that you can create your own toolbox with your own snippets easily. Look at the Komodo help sections about code snippets and also inspect existing snippets in the **R reference** toolbox to learn how to write them. Inspecting and changing a snippet is very easy. Just right-click in it, and select `Properties` in the context menu, or hit `Alt+Enter` to view and edit the snippet. Your changes are automatically saved when you click 'OK'!

Enjoy using SciViews/Komodo snippets!