

Package: Rcmdr (via r-universe)

June 2, 2026

Version 2.12.3

Date 2026-05-04

Title R Commander

Encoding UTF-8

Depends R (>= 3.5.0), grDevices, graphics, methods, stats, utils, splines, RcmdrMisc (>= 2.10.1), car (>= 3.1-0), effects (>= 4.0-3)

Imports tcltk, tcltk2 (>= 1.2-6), abind, relimp (>= 1.0-5), lme4, tools

Suggests aplpack, boot, colorspace, e1071, foreign, grid, Hmisc, knitr, lattice, leaps, lmtest, markdown, MASS, mgcv, multcomp (>= 0.991-2), nlme, nnet, nortest, readxl, rgl (>= 0.110.2), rmarkdown (>= 0.9.5), sem (>= 2.1-1)

ByteCompile yes

Description A platform-independent basic-statistics GUI (graphical user interface) for R, based on the tcltk package.

License GPL (>= 2)

URL <https://github.com/RCmdr-Project/rcmdr>, <https://www.r-project.org>,
<https://www.john-fox.ca/RCommander/index.html>

RoxygenNote 7.3.3

Config/pak/sysreqs cmake make libicu-dev libuv1-dev tcl tk tk-dev tk-table libx11-dev zlib1g-dev

Repository <https://rcmdr-project.r-universe.dev>

Date/Publication 2026-05-03 20:54:21 UTC

RemoteUrl <https://github.com/rcmdr-project/rcmdr>

RemoteRef HEAD

RemoteSha 42f2622f312d71537cfdc11c4899091123052a13

RemoteSubdir pkg

Contents

Commander	2
Commander-es	10
Index	17

Commander	<i>R Commander</i>
-----------	--------------------

Description

Start the R Commander GUI (graphical user interface)

Usage

Commander()

Details

Getting Started

For more detailed information about getting started, see *Help -> Introduction to the R Commander* from the R Commander menus or Fox (2017).

The default R Commander interface consists of (from top to bottom) a menu bar, a toolbar, a code window with script and R Markdown tabs, an output window, and a messages window.

Commands to read, write, transform, and analyze data are entered using the menus in the menu bar at the top of the *Commander* window. Most menu items lead to dialog boxes requesting further specification. I suggest that you explore the menus to see what is available.

Below the menu bar is a toolbar with (from left to right) an information field displaying the name of the active data set; buttons for editing and displaying the active data set; and an information field showing the active statistical model. There is also a *Submit* button for re-executing commands in the Script tab. The information fields for the active data set and active model are actually buttons that can be used to select the active data set and model from among, respectively, data frames or suitable model objects in memory.

Almost all commands require an active data set. When the Commander starts, there is no active data set, as indicated in the data set information field. A data set becomes the active data set when it is read into memory from an R package or imported from a text file, SPSS data set, Minitab data set, STATA data set, SAS XPORT data set; or an Excel spreadsheet. In addition, the active data set can be selected from among R data frames resident in memory. You can therefore switch among data sets during a session.

By default, commands are logged to the Script tab (the initially empty text window immediately below the toolbar), and commands and output appear in the Output window (the initially empty text window below the Script tab). Commands that don't require direct user interaction (such as interactive identification of points on a graph) are also used to create an R Markdown document in the tab of the same name. When the R Markdown tab is in front, pressing the "Generate HTML report" button compiles the document to create an html page with input and output, which opens

in a web browser. To alter these and other defaults, see the information below on configuration. Note, for example, that the **knitr** package can be used to create a LaTeX document to be compiled to a PDF report, as an alternative to — or in addition to — an R Markdown document (see the `use.knitr` option below).

Some **Rcmdr** dialogs (those in the *Statistics -> Fit models* menu) produce linear, generalized linear, or other models. When a model is fit, it becomes the active model, as indicated in the information field in the R Commander toolbar. Items in the *Models* menu apply to the active model. Initially, there is no active model. If there are several models in memory, you can select the active model from among them.

If command logging is turned on, R commands that are generated from the menus and dialog boxes are entered into the Script and R Markdown tabs in the Commander. You can edit these commands in the normal manner and can also type new commands. You can also type explanatory text in the R Markdown tab. Individual commands in the Script tab can be continued over more than one line, but the several lines of a multi-line command must be submitted simultaneously. (It is not necessary, as in earlier versions of the R Commander, to begin continuation lines with white space.) The contents of the Script and R Markdown tabs can be saved during or at the end of the session, and a saved script or R Markdown document can be loaded into the respective tabs. The contents of the Output window can also be edited or saved to a text file. Finally, editing operations also work in the Messages window.

To re-execute a command or set of commands in the Script tab, select the lines to be executed using the mouse and press the *Submit* button at the right of the toolbar (or *Control-R*, for "run", or *Control-Tab*). If no text is selected, the *Submit* button (or *Control-R* or *Control-Tab*) submits the line containing the text-insertion cursor. Note that an error will be generated if the submitted command or commands are incomplete.

Pressing *Control-F* brings up a find-text dialog box (which can also be accessed via *Edit -> Find*) to search for text in the Script tab, R Markdown tab, knitr tab, Output window, or Messages window. Edit functions such as search are performed in the Script tab unless you first click in another tab or window to make it active.

Pressing *Control-S* will save the Script tab, R Markdown tab, knitr tab, or Output window.

Pressing *Control-A* selects all of the text in the Script tab, R Markdown tab, knitr tab, Output window, or Messages window.

In addition, the following Control-key combinations work in these tabs and windows: *Control-X*, cut; *Control-C*, copy; *Control-V*, insert; *Control-Z* or *Alt-Backspace*, undo; and *Control-W*, redo.

Under Mac OS X, the *command* key may be used in place of the *Control* key, though the latter works as well.

Right-clicking the mouse (clicking button 3 on a three-button mouse, or *Control-left-clicking*) in the tabs or windows brings up a "context" menu with the *Edit*-menu items, plus (in the Script, R Markdown, and knitr tabs) a *Submit* item.

You can open a larger editor window with the document in the Markdown or knitr tab by making the corresponding selection from the *Edit* menu, the right-click context menu when the cursor is in the tab, or by pressing *Control-E* when the cursor is in the tab.

When you execute commands from the *Commander* window, you must ensure that the sequence of commands is logical. For example, it makes no sense to fit a statistical model to a data set that has not been read into memory.

Pressing a letter key (e.g., "a") in a list box will scroll the list box to bring the next entry starting with that letter to the top of the box.

You can cancel an R Commander dialog box by pressing the *Esc* key.

Most R Commander dialogs remember their state when this is appropriate, and can be restored to pristine state by pressing the Reset button.

Some R Commander dialogs have an Apply button that will execute the command generated by the dialog and then re-open the dialog in its previous state.

Exit from the Commander via the *File -> Exit* menu or by closing the *Commander* window.

Customization and Configuration

The preferred way of customizing the R Commander is to write a plug-in package: see `help("Plugins")`.

Alternatively, configuration files reside in the `etc` subdirectory of the package, or in the locations given by the `etc` and `etcMenus` options (see below).

The **Rcmdr** menus can be customized by editing the file `Rcmdr-menus.txt`.

You can add R code to the package, e.g., for creating additional dialogs, by placing files with file type `.R` in the `etc` directory, also editing `Rcmdr-menus.txt` to provide additional menus, sub-menus, or menu-items. Alternatively, you can edit the source package and recompile it.

To reiterate, however, the preferred procedure is to write an R Commander plug-in package.

A number of functions are provided to assist in writing dialogs, and **Rcmdr** state information is stored in a separate environment. See `help("Rcmdr.Utilities")` and the manual supplied in the `doc` directory of the **Rcmdr** package for more information.

In addition, several features are controlled by run-time options, set via the `options("Rcmdr")` command. These options should be set before the package is loaded. If the options are unset, which is the usual situation, defaults are used. Specify options as a list of `name=value` pairs. You can set none, one, several, or all options. The available options are as follows:

`ask.to.exit` if TRUE (the default), then the user is asked whether he or she wants to exit the **Rcmdr**; if this option is set to FALSE, then the subsequent option is also set to FALSE.

`ask.on.exit` if TRUE (the default), then the user is asked whether to save the script file, R Markdown file, and output file when the **Rcmdr** exits.

`attach.data.set` if TRUE (the default is FALSE), the active data set is attached to the search path.

`check.packages` if TRUE (the default), on start-up, the presence of all of the **Rcmdr** recommended packages will be checked, and if any are absent, the **Rcmdr** will offer to install them.

`command.text.color` Color for commands in the output window; the default is "red".

`console.output` If TRUE, output is directed to the *R Console*, and the *R Commander* output window is not displayed. The default is FALSE, unless the R Commander is running under RStudio, in which case the default is TRUE.

`crisp.dialogs` If TRUE, dialogs should appear on the screen fully drawn, rather than built up widget by widget. Prior to R 2.6.1, this option only works on the Windows version of R, but should in any event be harmless. The default is TRUE. If you encounter stability problems, try setting this option to FALSE.

`default.contrasts` Serves the same function as the general `contrasts` option; the default is `c("contr.Treatment", "contr.poly")`. When the Commander exits, the `contrasts` option is returned to its pre-existing value. Note that `contr.Treatment` is from the `car` package.

- `default.font.family` The default font for GUI elements such as menus and text labels, in the form of a Tk font family specification, given in a character string. For example, "Helvetica" specifies the sans-serif Helvetica font family. The default is taken from the `TkDefaultFont`. Normally a sans-serif font should be used.
- `default.font.size` The size, in points, of the default font. The default is 10 on non-Windows system and the size of the system font on Windows. To set the font size for R input and output, see the `log.font.size` option. The **Rcmdr** `scale.factor` option may also be used to control font size.
- `discreteness.threshold` should be a positive integer; if greater than 0 (which is the default), the maximum number of distinct values for a numeric variable to be considered discrete; if 0 (or smaller), the threshold is taken as the smallest of 100, twice the squareroot of the number of cases in the active data set (n), and 10 times $\log_{10}(n)$.
- `double.click` Set to TRUE if you want a double-click of the left mouse button to press the default button in all dialogs. The default is FALSE.
- `editDataset.threshold` If the number of values in the current data set exceed this value (the default is 10000), then the standard R data editor is used in preference to the R Commander `editDataset` editor.
- `error.text.color` Color for error messages; the default is "red".
- `etc` Set to the path of the directory containing the **Rcmdr** configuration files; defaults to the `etc` subdirectory of the installed **Rcmdr** package.
- `grab.focus` Set to TRUE for the current Tk window to "grab" the focus — that is, to prevent the focus from being changed to another Tk window. On some systems, grabbing the focus in this manner apparently causes problems. The default is TRUE. If you experience focus problems, try setting this option to FALSE.
- `help_type` This Rcmdr option takes precedence over the global R `help_type` option (see [options](#) and [help](#)), and by default is set to "html".
- `iconify.commander` If TRUE, the *Commander* window is minimized on startup; the default is FALSE.
- `length.output.stack` The R Commander maintains a list of output objects, by default including the last several outputs; the default length of the output stack is 10. `popOutput()` "pops" (i.e., returns and removes) the first entry of the output stack. Note that, as a stack, the queue is LIFO ("last in, first out").
- `length.command.stack` The R Commander also maintains a list of commands that is managed similarly; the default length of this stack is also 10.
- `log.commands` If TRUE (the default), commands are echoed to the script window; if FALSE, the script window is not displayed.
- `log.font.family` The font family to be used for text in the script window, output window, messages window, etc., specified as a character vector giving a Tk font family. This should normally be a monospaced font like "Courier". The default is taken from the `TkFixedFont`.
- `log.font.size` The font size, in points, to be used in the script window, in the output window, messages window, in recode dialogs, and in compute expressions — that is, where a monospaced font is used. The default is 10. Alternatively the **Rcmdr** `scale.factor` option may also be used to control font size.
- `log.height` The height of the script window, in lines. The default is 10. Setting `log.height` to 0 has the same effect as setting `log.commands` to FALSE.

- `log.text.color` Color for text in the script window; the default is "black".
- `log.width` The width of the script and output windows, in characters. The default is 80.
- `messages.height` The height of the messages window, in lines. The default is 4.
- `model.case.deletion` if TRUE (the default is FALSE), include a text box for case deletion in statistical-model dialog boxes (e.g., for linear models).
- `minimum.width` The minimum width, in pixels, for the main R Commander windows; the default is 1000.
- `minimum.height` The minimum height, in pixels, for the main R Commander windows; the default is 400.
- `multiple.select.mode` Affects the way multiple variables are selected in variable-list boxes. If set to "extended" (the default), left-clicking on a variable selects it and deselects any other variables that are selected; Control-left-click toggles the selection (and may be used to select additional variables); Shift-left-click extends the selection. This is the standard Windows convention. If set to "multiple", left-clicking toggles the selection of a variable and may be used to select more than one variable. This is the behaviour in the **Rcmdr** prior to version 1.9-10.
- `number.messages` If TRUE, the default, messages in the messages window are numbered.
- `open.graphics.devices` If TRUE (the default is FALSE), open the system graphics device and (if 3D RGL graphics are used) the RGL graphics device when the R Commander starts.
- `open.markdown.editor` If TRUE (the default is FALSE), open the R Markdown editor when the R Commander starts.
- `output.height` The height of the output window, in lines. The default is twice the height of the script window, or 20 if the script window is suppressed. Setting `output.height` to 0 has the same effect as setting `console.output` to TRUE.
- `output.text.color` Color for output in the output window; the default is "blue".
- `placement` Placement of the *R Commander* window, in pixels; the default is "", which lets the Tk window manager decide where to place the window; for example, "+20+20" should put the window near the upper-left corner of the screen, "-20+20" near the upper-right corner, though this doesn't appear to work reliably on Windows systems.
- `plugins` A character vector giving the names of **Rcmdr** plug-in packages to load when the Commander starts up. Plug-in packages can also be loaded from the *Tools* -> *Load Rcmdr plug-in(s)* menu. See [Plugins](#).
- `prefixes` A four-item character vector to specify the prefixes used when output is directed to the R console; the default is `c("Rcmdr> ", "Rcmdr+ ", "RcmdrMsg: ", "RcmdrMsg+ ")`.
- `quit.R.on.close` if TRUE, both the Commander and R are exited when the Commander window is closed. The default is FALSE, in which case only the Commander is exited (and can be restarted by the command `Commander()`).
- `RcmdrEnv.on.path` If TRUE (the default is FALSE), the environment in which R Commander state information is stored is placed on the search path. Some plug-ins, at least until they are updated, may require this setting.
- `retain.messages` If TRUE (the default), the contents of the message window are not erased between messages. In any event, a "NOTE" message will not erase a preceding "WARNING" or "ERROR".

- `retain.selections` If TRUE (the default), dialogs remember their previous state, where appropriate, as long as the data set isn't changed; some dialogs, e.g., for probabilities, retain selections even when the data set changes.
- `RExcelSupport` If TRUE (the default is FALSE), menus and output are handled by Excel.
- `rmarkdown.output` Values of several options for converting R Markdown to a document file. The default for this option is TRUE, which corresponds to `markdown.output=list(command.sections=TRUE, section.level=3, toc=TRUE, toc_float=TRUE, toc_depth=3, number_sections=FALSE, translate.rmd.headers=TRUE)`. The sub-option `command.sections` controls whether most R commands produce sections in the R Markdown document; the sub-option `section.level` controls the level of the sections that are created; the sub-option `translate.rmd.headers` controls whether the headers are translated from English into another language, if a translation is available; and the other sub-options are standard for `rmarkdown`. The `toc_float`, `toc_depth`, and `number_sections` sub-options are only effective if Pandoc is installed.
- `rmd.output.format` The output file type for R Markdown documents if pandoc is installed; one of "html" (the default), "pdf" (requires LaTeX), "docx" (Word), or "rtf" (rich text file).
- `rmd.template` The quoted path to a .Rmd file to serve as a template for R code and output. The default is to use a template included with the package.
- `scale.factor` A scaling factor to be applied to all Tk elements, such as fonts. This works well only in Windows. The default is NULL.
- `scientific.notation` Higher numbers cause ordinary (decimal) notation to be increasingly preferred to scientific notation for representing very small and very large numbers; correspond to the `scipen` option in R: see [options](#). The default is 5, while the standard default in R is 0 (where 0 means that scientific notation is used whenever the resulting printed representation of a number is smaller in scientific than in standard notation).
- `showData.threshold` a vector with 2 entries, defaulting to `c(20000, 100)`. If the number of cases in the active data set exceeds the first number (default, 20,000) or the number of variables exceeds the second number (default, 100), then `View()` rather than `showData()` is used to display the data set. The reason for the option is that `showData()` is very slow when the number of cases or variables is large; setting the threshold to `c(0, 0)` suppresses the use of `showData` altogether. It's necessary to use `showData` however for the view of the active data set to be updated dynamically when, e.g., a variable is added.
- `show.edit.button` Set to TRUE (the default) if you want an *Edit* button in the Commander window, permitting you to edit the active data set. Windows users may wish to set this option to FALSE to suppress the *Edit* button because changing variable names in the data editor can cause R to crash (though I believe that this problem has been solved).
- `sort.names` Set to TRUE (the default) if you want variable names to be sorted alphabetically in variable lists.
- `suppress.icon.images` Set to TRUE to suppress the icon images in dialog OK, Cancel, Reset, and Help buttons; the default is FALSE.
- `suppress.menus` if TRUE, the Commander menu bar and tool bar are suppressed, allowing another program (such as Excel) to take over these functions. The default (of course) is FALSE.
- `suppress.X11.warnings` On (some?) Linux and Mac OS X systems, multiple X11 warnings are generated by **Rcmdr** commands after a graphics-device window has been opened. Set this option to TRUE (the default when running interactively under X11) to suppress reporting of these warnings. An undesirable side effect is that then *all* warnings and error messages

are intercepted by the **Rcmdr**, even those for commands entered at the R command prompt. Messages produced by such commands will be printed in the Commander Messages window after the next **Rcmdr**-generated command. Some X11 warnings may be printed when you exit from the Commander.

`theme` A ttk theme to control the overall style of the Commander GUI; should be one of the themes returned by `tcltk2::tk2theme.list()`. The default theme varies by operating system, and can be discovered by entering the command `tcltk2::tk2theme()` in a fresh R session.

`title.color` Color for the titles of some widgets, such as variable-list boxes; can be given as a color name, such as "blue" or as an RGB value, such as "#0000FF". The default is the standard color for ttk label frames, unless that is "#000000" or "black", in which case "blue" is used instead.

`tkwait.commander` This option addresses a problem that, to my knowledge, is rare, and may occur on some non-Windows systems. If the Commander causes R to hang, then set the `tkwait` option to TRUE; otherwise set the option to FALSE or ignore it. An undesirable side effect of setting the `tkwait.commander` option to TRUE is that the R session command prompt is suppressed until the Commander exits. One can still enter commands via the script window, however. In particular, there is no reason to use this option under Windows, and it should not be used with the Windows R GUI with buffered output when output is directed to the R console.

`tkwait.dialog` If TRUE (the default is FALSE), R will wait until an R Commander dialog is closed. This has the disadvantage of preventing help pages from being displayed until a dialog is closed in the Mac OS X R.app and in RStudio. This was also the standard behavior of the R Commander in earlier versions and is provided for compatibility with previous behavior. If this option is TRUE, then the R Commander data editor is disabled in favor of the standard R platform-specific data editor, and the new-data-set menu item is suppressed.

`use.knitr` If TRUE (the default is FALSE), a knitr .Rnw LaTeX document is created in a tab of the main Commander window; this document can be compiled into .tex and .pdf reports via the `knit2pdf` function in the **knitr** package.

`use.markdown` If TRUE (the default is the negation of the `use.knitr` argument), an R Markdown document is created, which can be compiled into an HTML, PDF, Word, or rich text file report.

`use.rgl` If TRUE (the default), the `rgl` package will be loaded if it is present in an accessible library; if FALSE, the `rgl` package will be ignored even if it is available. The `rgl` package can sometimes cause problems when running R under X11.

`"valid.classes"` The classes of variables that the R Commander recognizes, in addition to numeric data; other variables in a data set will be suppressed. The default is "factor", "ordered", "character", "logical", "POSIXct", "POSIXlt", "Date", "chron", "yearmon", "yearqtr", "zoo", "zooreg", "timeDate", "xts", "its", "ti", "jul", "timeSeries", "fts", "Period", "hms", "difftime").

`variable.list.height` the number of items (typically variables) to display in list boxes; longer lists may be viewed by scrolling. The default is 6.

`variable.list.width` a two-item vector controlling the width of list boxes, in characters, giving the minimum and maximum width to display; the default is `c(20, Inf)`. If the widest item name falls in this range, then its number of characters determines the width of the box. *Note:* This specification works only approximately.

`warning.text.color` Color for warning messages; the default is "darkgreen".

Some options can also be set via the *File -> Options* menu, which will restart the Commander after options are set.

If you want always to launch the R Commander when R starts up, you can include the following code in one of R's start-up files (e.g., in the `Rprofile.site` file in R's `etc` subdirectory):

```
local({
old <- getOption("defaultPackages")
options(defaultPackages = c(old, "Rcmdr"))
})
```

R Commander options can also be permanently set in the same manner. For more information about R initialization, see `?Startup`.

Warning

The R Commander Script window does not provide a true console to R, and may have certain limitations. I don't recommend using the R Commander for serious programming or for data analysis that relies primarily on scripts — use a programming editor instead. If you encounter any problems with the Script tab, however, I'd appreciate it if you brought them to my attention.

Platform-Specific Issues

Under Windows, the **Rcmdr** package can be run under the *Rgui* in the SDI (single-document interface) mode, or under `rterm.exe`. You might experience problems running the **Rcmdr** under ESS with NTEmacs or XEmacs, or under other R consoles. The R Commander can be run under the *Rgui* in MDI (multiple-document interface) mode but it is relatively inconvenient to do so and isn't recommended.

Occasionally, under Windows, after typing some text into a dialog box (e.g., a subsetting expression in the Subset Data Set dialog), buttons in the dialog (e.g., the OK button) will have no effect when they are pressed. Clicking anywhere inside or outside of the dialog box should restore the function of the buttons. As far as I have been able to ascertain, this is a problem with Tcl/Tk for Windows. I have not seen this behavior in some time and the problem may have been solved.

Under Mac OS X Mavericks and later, the R Commander may appear to freeze or hesitate when run under *R.app* if the *R.app* window is hidden and "app nap" is turned on. It is recommended that app nap be turned off for *R.app*, which can be most conveniently done via the R Commander *Tools* menu. The app nap setting is permanent until changed and so the current setting will apply whether or not the R Commander is used. When R is first installed, app nap will be on for *R.app*. The **tccltk** package requires that X Windows is installed under Mac OS X, and as a consequence the **Rcmdr** package, which depends on **tccltk**, will not load if X Windows is absent. X Windows for Mac OS X may be obtained from <https://www.xquartz.org/>.

Note

On startup, the R Commander sets `options(na.action=na.exclude)`; this is done so that observation statistics such as residuals can be properly added to the active data set when there are missing values. The option is reset to its pre-existing value when the Commander exits. Some functions may not work properly when the default `na.action` is set to `na.exclude`.

This version should be compatible with the **RExcel** package, which can use the R Commander menus.

Author(s)

John Fox

References

Fox, J. (2017) *Using the R Commander: A Point-and-Click Interface for R*. Chapman and Hall/CRC Press.

Fox, J. (2005) The R Commander: A Basic Statistics Graphical User Interface to R. *Journal of Statistical Software*, **14(9)**: 1–42.

Fox, J. (2007) Extending the R Commander by "plug in" packages. *R News*, **7(3)**: 46–52.

See Also

[Plugins](#), [Rcmdr.Utilities](#), [knit](#), [knit2pdf](#)

Examples

```
options(Rcmdr=list(log.font.size=12, default.contrasts=c("contr.Sum", "contr.poly")))
```

Commander-es

R Commander

Description

Inicia la GUI (Interfaz Gráfica de Usuario) de R Commander

Usage

```
Commander()
```

Details**Empezando**

La interfaz por defecto de R Commander consiste en (de arriba a abajo) una barra de menús, una barra de herramientas, una ventana de instrucciones, una ventana de salida y una ventana de mensajes.

Las instrucciones para leer, escribir, transformar y analizar datos se ejecutan usando la barra de menú de la parte superior de la ventana de *R Commander*. La mayor parte de los items de este menú le guiarán mediante ventanas de diálogo, preguntando más allá de la especificación. Es aconsejable explorar el menú para ver las opciones disponibles.

Bajo la barra de menú se encuentra la barra de herramientas con (de izquierda a derecha) un campo de información que muestra el nombre del conjunto de datos activos, botones para editar y mostrar el conjunto de datos activos y un campo de información mostrando el modelo estadístico activo. Bajo la ventana de instrucciones hay un botón Ejecutar para realizar las órdenes indicadas en la ventana de instrucciones. Los campos de información para los datos y el modelo activo son botones

que pueden ser usados para seleccionar éstos entre, respectivamente, conjuntos de datos o modelos disponibles en memoria.

La mayor parte de las órdenes requiere un conjunto de datos activos. Cuando se ejecuta R Commander no hay conjunto de datos activos, como está indicado en el campo de información del conjunto de datos activos. Un conjunto de datos llega a ser un conjunto de datos activos cuando éste es leído en la memoria desde un paquete R o importado desde un archivo de texto, conjunto de datos SPSS, conjunto de datos Minitab, conjunto de datos STATA, Excel, Access o dBase. Además el conjunto de datos activos puede ser seleccionado desde conjuntos de datos R residentes en memoria. Los datos pueden ser elegidos de entre todos los conjuntos para cada sesión.

Por defecto, las órdenes son registradas en la ventana de instrucciones (la ventana de texto vacía inmediatamente después de la barra de herramientas); las órdenes y las salidas aparecen en la ventana de resultados (la ventana de texto vacía después de la ventana de instrucciones) y el conjunto de datos activos es adjuntado a la ruta de búsqueda. Para alterar éstos y otros parámetros por defecto, puede consultar la información pertinente en configuración.

Algunos diálogos de Rcmdr (éstos en Estadísticos -> Ajuste de modelos) generan el modelo lineal, modelo lineal generalizado y otros modelos. Cuando un modelo es ajustado, se convierte en el modelo activo, indicado en el campo de información de la barra de herramientas de R Commander. Los items del menú Modelos se aplican al modelo activo. Inicialmente, no hay modelo activo. Si hay varios modelos en memoria, puede elegir el modelo activo de entre ellos.

Si el registro de instrucciones está activo, las órdenes de R generadas desde los menús y los cuadros de diálogos, son introducidas en la ventana de instrucciones de R Commander. Se pueden editar estas órdenes de manera normal y se pueden escribir otras nuevas en la ventana de instrucciones. Las órdenes individuales pueden ser continuadas en más de una línea, pero cada línea después de la primera debe ser indentada con uno o más espacios o tabuladores. El contenido de la ventana de instrucciones puede ser almacenado durante o al final de la sesión y un conjunto de instrucciones guardado puede ser cargado en la ventana de instrucciones. El contenido de la ventana de resultados puede ser editado o guardado en un archivo de texto.

Para volver a ejecutar una orden o un conjunto de ellas, se seleccionan las líneas que se desean ejecutar usando el ratón y se presiona el botón Ejecutar, situado a la derecha de la barra de herramientas (o Control-R, para ejecutarlos). Si no hay texto seleccionado el botón Ejecutar (o Control-R) envía el contenido de la línea que contiene el cursor de inserción. Observar que se generará un error si la orden o las órdenes enviadas son incompletas.

Presionando Control-F se abre un cuadro de diálogo de búsqueda de texto (también es accesible vía Editar -> Buscar) para buscar el texto en la ventana de instrucciones o la ventana de resultados. Las búsquedas son realizadas en la ventana de instrucciones a menos que primero pulse en la ventana de resultados para activarla.

Presionando Control-S se guardará el conjunto de instrucciones o la ventana de resultados.

Presionando Control-A se selecciona todo el texto del conjunto de instrucciones o de la ventana de resultados.

Pulsando el botón derecho del ratón (el tercer botón en un ratón de tres botones) en el conjunto de instrucciones o en la ventana de resultados se abre el menú contextual con los items del menú Editar, más un item Ejecutar (en la ventana de instrucciones).

Cuando ejecute órdenes en la ventana de R Commander, debe asegurarse que la sentencia sea lógica. Por ejemplo, no tiene sentido ajustar un modelo estadístico de un conjunto de datos que no ha sido leído en memoria.

Presionando una letra (ej. "a") en un cuadro con una lista se recorrerá ésta hasta la siguiente entrada que comience con esa letra desde el principio del cuadro.

Salir de R Commander se realiza mediante Fichero -> Salir o cerrando la ventana de R Commander.

Personalización y configuración

Los archivos de configuración están en el subdirectorio etc de cada paquete o en la localización dada por etc y en las opciones de etcMenus (mirar abajo).

Los menús de Rcmdr pueden ser personalizados editando el archivo Rcmdr-menus.txt.

Algunas funciones (ej. histograma) que normalmente no crean salida visible cuando se ejecutan desde la consola sí lo harán - a menos que se evite - cuando se ejecuten desde la ventana de instrucciones de R Commander. Tal salida puede ser suprimida listando los nombres de estas funciones en el archivo log-exceptions.txt.

Puede añadir código R al paquete, ej., para crear diálogos adicionales, colocando archivos con extensión .R en el directorio etc, además puede editar Rcmdr-menus.txt para proporcionar menús adicionales, submenús o items. Una demostración de esto se proporciona mediante el archivo BoxCox.demo. Para activar la demo, renombre el archivo a BoxCox.R y descomente la correspondiente línea del menú en Rcmdr-menus.txt. De forma alternativa, puede editar el código del paquete y recompilarlo.

Algunas funciones son proporcionadas para ayudar a escribir diálogos y la información del estado de Rcmdr en un emplazamiento separado. Mirar help("Rcmdr.Utilities") y el manual suministrado en el directorio doc del paquete de Rcmdr para mayor información.

Además, varias características son controladas mediante opciones, en tiempos de ejecución, establecidas por la orden options("Rcmdr"). Estas opciones deben ser establecidas antes de cargar el paquete. Si las opciones no están establecidas, que es la situación normal, serán usados los parámetros por defecto. Las opciones se especifican como una lista de pares name\$=values. Puede no establecer, establecer una, varias, o todas las opciones. Las opciones disponibles son las dadas a continuación:

`attach.data.set` Si es TRUE (por defecto FALSE), el conjunto de datos activo es fijado como la ruta de búsqueda.

`check.packages` Si es TRUE (por defecto), al arranque, la presencia de todos los paquetes recomendados de Rcmdr serán comprobados y si alguno no está instalado, Rcmdr preguntará si deben instalarse.

`command.text.color` El color de las órdenes en la ventana de resultados es, por defecto, "red".

`console.output` Si es TRUE la salida será dirigida a la consola de R y la ventana de salida de R Commander no se mostrará. Por defecto es FALSE.

`contrasts` Ofrece la misma función que la opción general contrasts; por defecto es c("contr.Treatment", "contr.poly"). Cuando se sale de Commander la opción contrasts vuelve a su valor preexistente. Observe que contr.Treatment es del paquete car.

`crisp.dialogs` Si es TRUE, los diálogos deben aparecer en la pantalla dibujada completamente, más que acumular dispositivo a dispositivo. Esta opción debería afectar sólo a versiones Windows de R, pero debe en cualquier caso ser inofensivo. Por defecto es TRUE bajo versiones Windows de R 2.1.1 y superiores y FALSE si no. Si está trabajando en Windows y encuentra que se incrementan los problemas de estabilidad, pruebe establecer esta opción a FALSE.

- `default.font` La fuente por defecto, como la especificación de la fuente de X11, dada en cadena de caracteres. Si está especificado, este valor toma precedencia sobre el tamaño de la fuente por defecto (abajo). Esta opción es sólo para sistemas no-Windows.
- `default.font.size` Tamaño, en puntos, por defecto de la fuente. Por defecto es 10 para sistemas Windows y 12 para otros sistemas, salvo especificación de lo contrario (mirar el ítem anterior). La fuente por defecto es `"*helvetica-medium-r-normal-*-xx"`, donde `xx` es por defecto el tamaño de la fuente. Esta opción es sólo para sistemas no-Windows.
- `double.click` Establecer a TRUE si quiere que un doble click con el botón izquierdo del ratón sirva para pulsar el botón por defecto en todos los diálogos. Por defecto es FALSE.
- `error.text.color` Color de los mensajes de error; por defecto es `"red"`.
- `etc` Establece la ruta del directorio que contiene los archivos de configuración de Rcmdr; por defecto el subdirectorio `etc` del paquete Rcmdr instalado.
- `grab.focus` Establecer a TRUE para "mantener" el enfoque en la ventana actual de Tk, esto es, para prevenir que el enfoque sea cambiado a otra ventana Tk. En algunos sistemas, mantener el enfoque de esta forma, puede causar problemas. Por defecto es TRUE. Si experimenta problemas de enfoque, intente establecer esta opción a FALSE.
- `load.at.startup` Vector de caracteres de nombres de los paquetes que deben ser cargados cuando el paquete Rcmdr es cargado; por defecto se carga sólo el paquete `car`. Otros paquetes requeridos serán cargados cuando se necesiten. Si esto está disponible, el paquete `car` será cargado cuando Commander se inicie en cualquier caso.
- `log.commands` Si es TRUE (por defecto), los comandos son repetidos en la ventana de instrucciones; si es FALSE, la ventana de instrucciones no se muestra.
- `log.font.size` Tamaño de la fuente, en puntos, que es usado en la ventana de instrucciones, en la ventana de resultados, en diálogos recodificados y en expresiones de cálculo, esto es, donde es usada una fuente monoespacio. Por defecto es 10 para sistemas Windows y 12 para otros sistemas.
- `log.height` La altura de la ventana de instrucciones, en líneas. Por defecto es 10. Estableciendo `log.height` a 0 tiene el mismo efecto que establecer `log.commands` a FALSE.
- `log.text.color` Color del texto de la ventana de instrucciones; por defecto es `"black"`.
- `log.width` La anchura de la ventana de instrucciones y la de salida, en caracteres. Por defecto es 80.
- `multiple.select.mode` Afecta a la forma en la que múltiples variables son seleccionadas en una caja de listas de variables. Si se establece a `"extended"` (por defecto), el botón izquierdo en una variable selecciona ésta y deselectiona cualquier otra variable que estuviera seleccionada; Control+botón izquierdo acciona la selección (y puede ser usado para seleccionar variables adicionales); Mayúsculas+botón izquierdo extiende la selección. éste es el convenio estándar de Windows. Si lo establece a `"multiple"`, el botón izquierdo acciona la selección de una variable y puede ser usado para seleccionar más de una variable. éste es el comportamiento de Rcmdr antes de la versión 1.9-10.
- `output.height` Altura de la ventana de resultados, en líneas. Por defecto es dos veces la altura de la ventana de instrucciones o 20 si la ventana de instrucciones es suprimida. Establecer `output.height` a 0 tiene el mismo efecto que `console.output` a TRUE.
- `output.text.color` Color de la salida en la ventana de resultados, por defecto es `"blue"`.
- `placement` Emplazamiento de la ventana de R Commander, en píxeles; por defecto es `"$-40+20$"`, lo que pone la ventana cerca de la esquina superior derecha de la pantalla.

- `plugins` Vector de caracteres con los nombres de paquetes de plugins de Rcmdr a cargar cuando Commander arranque. Los paquetes plugins también pueden ser cargados desde el menú Herramientas -> Cargar paquete(s).
- `suppress.menus` Si es TRUE, la barra de menús y de herramientas de R Commander son suprimidas, permitiendo que otro programa (como Excel) asuma esas funciones. Por defecto (por supuesto) es FALSE.
- `suppress.X11.warnings` En (algunos) sistemas Linux X11 se generan múltiples advertencias por las órdenes de Rcmdr, después de abrir la ventana del dispositivo gráfico. Establecer esta opción a TRUE (por defecto cuando arranca interactivamente bajo X11 antes de la versión de R 2.4.0) suprime la aparición de estas advertencias. Un efecto secundario indeseable es que entonces todas las advertencias y mensajes de error son interceptados por Rcmdr, incluso para las instrucciones introducidas en los avisos de R. Los mensajes producidos por tales órdenes serán impresos en la ventana de mensajes de R Commander después de la siguiente orden generada en Rcmdr. Algunas advertencias de X11 puede ser impresas al salir de R Commander. Este problema sólo se aplica a versiones de R anteriores a 2.4.0 y el valor por defecto de la opción es establecido por consiguiente.
- `retain.messages` Si es TRUE (por defecto FALSE), el contenido de la ventana de mensajes no es borrado entre mensajes. En cualquier caso, un mensaje "NOTE" no borrará un anterior "WARNING" o "ERROR".
- `RExcelSupport` Establecido como TRUE (por defecto es FALSE), los menús y salidas son dirigidas a Excel.
- `scale.factor` Factor de escala aplicado a todos los elementos Tk, como las fuentes. Esto funciona bien sólo en Windows. Por defecto es NULL.
- `showData.threshold` Si el número de variables en el conjunto de datos activos excede este valor (por defecto, 100), entonces `edit()`, más que `showData()`, es utilizado para exhibir el conjunto de datos. Un inconveniente es que el control no se devuelve a Commander hasta que la ventana de edición sea cerrada. La razón de esta opción es que `showData()` es muy lento cuando el número de variables es grande; fijando el umbral a 0 suprime el uso en conjunto de `showData`.
- `show.edit.button` Fijar a TRUE (por defecto) si quiere un botón `Editar` en la ventana de Commander, que permita editar el conjunto activo de datos. Los usuarios de Windows pueden desear establecer esta opción a FALSE para suprimir el botón `Editar` porque cambiando los nombres de las variables en el editor de datos se puede causar que R falle (aunque este problema se cree solucionado).
- `sort.names` Fijar a TRUE (por defecto) si se quiere ordenar alfabéticamente el nombre de las variables en una lista de variables.
- `tkwait` Esta opción trata un problema que, en mi conocimiento, es raro y puede ocurrir en algunos sistemas no Windows. Si R Commander causa que se cuelgue R, entonces establezca la opción `tkwait` a TRUE; o conserve la opción en FALSE e ignórela. Un indeseable efecto secundario de establecer la opción `tkwait` a TRUE es que el aviso de órdenes de la sesión de R es suprimido hasta salir de R Commander. Uno sin embargo todavía puede introducir órdenes por la ventana de instrucciones. En particular, no hay razón para usar esta opción bajo Windows y no se debería usar con la GUI de R en Windows con salida protegida cuando la salida esté dirigida a la consola de R.
- `use.rgl` Si es TRUE (por defecto), el paquete `rgl` será cargado si está presente en una librería accesible, si es FALSE, el paquete `rgl` será ignorado aunque esté disponible. El paquete `rgl` puede a veces causar problemas cuando se arranca R bajo X11.

`warning.text.color` Color de los mensajes de advertencia; por defecto es "darkgreen".

Muchas opciones pueden también ser establecidas mediante el menú *Archivo -> Opciones*, que reiniciará R Commander después de que las opciones sean establecidas.

Si quiere lanzar R Commander cuando inicie R, puede incluir la siguiente instrucción en uno de los ficheros de inicio de R (por ejemplo, en el fichero `Rprofile.site` de la carpeta etc de R):

```
local({
old <- getOption("defaultPackages")
options(defaultPackages = c(old, "Rcmdr"))
})
```

Las opciones de R Commander puede ser establecidas de forma permanente de la misma forma. Para más información sobre el inicio de R, véase `?Startup`.

Avisos

La ventana de instrucciones de R Commander no proporciona una verdadera consola a R y tiene ciertas limitaciones. No se recomienda usar R Commander para la programación seria o el análisis de datos que confíe primordialmente en instrucciones - usar un editor de programación en su lugar. Por ejemplo, para declaraciones de composiciones de R incluidas entre llaves "`{ }`", incluyendo definición de funciones, no serían analizadas ni ejecutadas correctamente, aunque si las líneas después de las primeras que estén indentadas. Puede ejecutar declaraciones de composiciones desde la ventana de instrucciones separando los comandos dentro de las llaves por puntos y comas.

Problemas Conocidos

Ocasionalmente, bajo Windows, después de teclear algún texto en un cuadro de diálogo (ej. subconjunto de expresiones en el diálogo de subconjunto de conjunto de datos), algunos botones en el diálogo (ej. el botón Aceptar) pueden no tener efecto cuando sean presionados. Pulsando en cualquier sitio, dentro o fuera del cuadro de diálogo, debería restaurarse las funciones de los botones. Por lo que se ha podido comprobar, éste es un problema con Tcl/Tk de Windows.

Note

Bajo Windows, el paquete `Rcmdr` puede también funcionar bajo de `Rgui` en modo SDI (interfaz de único documento) o bajo `rterm.exe`; puede ser que experimente problemas ejecutando `Rcmdr` bajo ESS con `NTEmacs` o `XEmacs`.

Author(s)

John Fox (de la versión inglesa)

Manuel Muñoz Márquez (traductor-mantenedor) <manuel.munoz@uca.es>

Véase <https://knuth.uca.es/R/doku.php?id=equipotraduccion>

See Also

[Plugins](#)

Examples

```
options(Rcmdr=list(log.font.size=12, contrasts=c("contr.Sum", "contr.poly")))
```

Index

* **misc**

Commander, [2](#)

Commander-es, [10](#)

Commander, [2](#)

Commander-es, [10](#)

help, [5](#)

knit, [10](#)

knit2pdf, [8](#), [10](#)

options, [5](#), [7](#)

Plugins, [4](#), [6](#), [10](#), [15](#)

Rcmdr.Utilities, [4](#), [10](#)

rmarkdown, [7](#)