

Description

R Commander GUI (графический интерфейс пользователя)

Usage

`Commander()`

Details

Введение

По умолчанию интерфейс R-commander состоит из (сверху вниз) строки меню, панели инструментов, окна скриптов, окна вывода, и окна сообщений.

Команды: чтения, записи, преобразования и анализа данных вводятся с помощью меню в панели верхней части окна *Commander*. Большинство пунктов меню ведет к диалоговым окнам запроса дополнительной информации. Я предлагаю вам изучить меню, чтобы увидеть, что имеется в наличии.

Ниже панели меню есть панель инструментов, содержащая (слева направо): информационное поле, отображающее имя текущего набора данных, кнопки для редактирования и отображения текущего набора данных, и информационное поле, показывающее текущую статистическую модель. Есть также кнопка *Выполнить* для повторного выполнения команд в окне командной строки. Информационные поля для текущего набора данных и текущей модели на самом деле являются кнопками, которые могут быть использованы для выбора текущего набора данных и модели из имеющихся, соответственно, таблиц данных или подходящих объектов моделей в памяти.

Почти все команды требуют наличия активного набора данных. Когда Commander стартует, активного набора данных нет, что указано в информационном поле набора данных. Набор данных становится активным набором данных, когда он считывается в память из R пакета или импортируется из текстового файла, набора данных SPSS, набора данных Minitab, набора данных STATA, или набора данных Excel, Access, dBase. Кроме того, активный набор данных может быть выбран из числа имеющихся в памяти R таблиц данных. Во время сессии Вы можете переключаться между наборами данных.

По умолчанию, команды записываются в окно скриптов (первоначально пустое текстовое окно сразу ниже панели инструментов); команды и их вывод появляются в окне вывода (первоначально пустое текстовое окно ниже скриптового окна); и активный набор данных подключен к пути поиска. Чтобы изменить эти и другие значения по умолчанию, см. ниже информацию о конфигурации.

Некоторые Rcmdr диалоги (в *статистике* -> *подгонка моделей* меню) создают линейные, обобщенные линейные, или иные модели. Когда модель подогнана, она становится активной моделью, что указывается в информационном поле панели инструментов R Commander. Пункты в меню *Модели* применяются к активной модели. Первоначально, активной модели нет. Если в памяти есть несколько моделей, вы можете выбрать из их числа активную.

Если включена запись команд, R команды, которые генерируются из меню и диалоговых окон, выводятся в окне скриптов Commander. Вы можете редактировать эти команды в обычным образом, можно также вводить в окно скриптов новые команды.

Отдельные команды можно продлевать на несколько строк, все строки многострочной команды должны быть выполнены одновременно. (Не является необходимым, как в предыдущих версиях R commander, начинать дополнительные линии с пробела.) Содержание окна скриптов можно сохранить во время или в конце сессии, а сохраненный скрипт может быть загружен в окно скриптов. Содержание окна вывода также может быть отредактировано или сохранено в текстовый файл.

Чтобы снова выполнить команду или набор команд, выберите строки, которые должны быть выполнены с помощью мышки и нажмите кнопку *Выполнить* в правой части панели инструментов (или для "запуска" *Control-R*, или *Control-Tab*). Если текст не выделен, кнопка *Выполнить* (или *Control-R* или *Control-Tab*) выполняет строку, содержащую текстовый курсор. Заметим, что произойдет ошибка, если выполняемая команда или команды будут неполными.

Нажатие *Control-F* выводит диалоговое окно поиска текста (которое также можно достичь через *Правка -> Найти*) для поиска текста в окнах скрипта или вывода. Функции Правки, такие как поиск, выполняются в окне скриптов, если вы не кликнете сначала в окно вывода, чтобы сделать его активным.

Нажатие *Control-S* сохранит окно скриптов или окно вывода.

Нажатие *Control-A* выделяет весь текст в окне скриптов или окне вывода.

Дополнительно: следующие Control-key комбинации работоспособны в окне скриптов и окне вывода: *Control-X* вырезать; *Control-C* копировать; *Control-V* вставить; *Control-Z* отменить; and *Control-W* вернуть.

Правый клик мыши (нажать 3-ю клавишу трехкнопочной мыши) в окне скриптов или окне вывода выводит контекстное меню с содержанием раздела меню *Правка*, а в окне скриптов еще и с пунктом *Выполнить*.

Когда вы выполняете команды в окне *Commander*, вы должны убедиться, что последовательность команд имеет смысл. Например, бессмысленно подгонять статистическую модель для набора данных, который не считан в память.

Нажатие на клавишу с буквой (например "a") в списке промотает список до записи начинающейся с этой буквы.

Выходят из Commander через *Файл -> Выйти* меню или через закрытие окна *Commander*.

Кастомизация и конфигурация

Предпочтительный путь кастомизации R Commander это написание плагин модуля: см. `help("Plugins")`.

Кроме этого, имеются конфигурационные файлы в `etc` поддиректории модуля, или в месте заданном `etc` и `etcMenus` (см. далее).

Меню `Rcmdr` можно настроить редактируя файл `Rcmdr-menus.txt`.

Вы можете добавить код R в модуль, создать дополнительное диалоговое окно, помещая файлы с расширением `.R` в директорию `etc`, а так же редактировать `Rcmdr-menus.txt` для введения дополнительных меню, подменю, или пунктов меню. Демо содержится в файле `BoxCox.demo`. Для активации демо, переименуйте этот файл в `BoxCox.R`, и раскомментируйте соответствующую строчку меню в `Rcmdr-menus.txt`. Или же поправьте исходные тексты и пересоберите их.

Некоторое кол-во функций предусмотрено для помощи в написании диалогов, так же параметры состояния Rcmdr доступны в специальном окружении. См. `help("Rcmdr.Utilities")` и руководство в директории `doc` модуля Rcmdr для более подробной информации.

Дополнительно, некоторые возможности контролируются в процессе работы, устанавливаются командами `options("Rcmdr")`. Эти настройки должны быть установлены прежде чем модуль будет загружен. Если настройка не указывается, что обычно и происходит, используется значение по умолчанию. Задавайте настройки как список пар *имя=значение*. Вы можете задать ни одной, одну, отдельные, или все настройки. Доступны следующие настройки:

`ask.on.exit` if TRUE (the default), пользователь спрашивается следует ли сохранить скрипт и файл вывода при выходе из Rcmdr.

`attach.data.set` if TRUE (по умолчанию FALSE), текущая таблица данных подключается в путь поиска.

`check.packages` если TRUE (по умолчанию), при старте проверяется наличие рекомендованных Rcmdr модулей, и если что то отсутствует, Rcmdr предложит произвести установку.

`command.text.color` Цвет для команд окне вывода; по умолчанию "red".

`console.output` Если TRUE, вывод направляется в R Console и R Commander окно вывода не отображается. По умолчанию FALSE.

`default.contrasts` Выполняет ту же функцию, что и общие настройки эффектов; по умолчанию `c("contr.Treatment" "contr.poly")`. При выходе из Commander, настройки эффектов возвращаются к предыдущим значениям. Учтите что `contr.Treatment` находится в модуле `car`.

`crisp.dialogs` Если TRUE, диалоги должны появляться полностью прорисованными, вместо того чтобы отображать примитив за примитивом. До версии R 2.6.1, эта настройка работает только в версии R для Виндовс, однако должна быть в любом случае безопасной. По умолчанию TRUE. Если заметите какие либо сбои в работе, попробуйте выставить FALSE.

`default.font` Шрифт по умолчанию, в соответствии с нотацией X11, задается в виде строки. Если указан, его значение имеет приоритет над настройкой размера по умолчанию (ниже). Не для Windows.

`default.font.size` Размер, в пунктах, шрифта по умолчанию. Иначе используется 10 для Windows и 12 для остальных ОС. Если не было указано другого (см. предыдущий пункт), шрифт по умолчанию `"*helvetica-medium-r-normal-*-xx"`, где `xx` размер шрифта. Не для Windows.

`double.click` Установите TRUE если хотите нажимать кнопки во всех диалогах двойным нажатием левой кнопки мыши. По умолчанию FALSE.

`error.text.color` Цвет сообщения об ошибке; по умолчанию "red".

`etc` Установить путь к директории содержащей конфигурационные файлы Rcmdr; по умолчанию `etc` поддиректория установленного модуля Rcmdr.

`grab.focus` Значение TRUE означает захват фокуса текущим Tk окном — блокирование захвата фокуса другими Tk окнами. На некоторых инсталляциях это может вызвать проблемы. По умолчанию TRUE. Если с фокусом будут проблемы, попробуйте изменить на FALSE.

length.output.stack R Commander ведет список объектов вывода, по умолчанию запоминая несколько последних выводов; по умолчанию длина очереди выводов 10. **popOutput()** “pops” (возвращает и удаляет) первую запись очереди вывода. Дисциплина очереди LIFO (“последний добавляется, первый выводится”).

load.at.startup Вектор строка имен загружаемых при старте Rcmdr пакетов; по умолчанию загружается только пакет **car**. Остальные пакеты загружаются по зависимостям. Если пакет **car** присутствует, он всегда загружается при старте Commander.

log.commands Если TRUE (по умолчанию), команды отображаются в окне скриптов; если FALSE, окно скриптов не выводится.

log.font.size Размер шрифта, в пунктах, для использования в окне скриптов, в окне вывода, в диалогах и в вычисляемых выражениях — если используется моноширинный шрифт. По умолчанию 10 в Windows и 12 в других ОС.

log.height Высота окна скриптов, в линиях. По умолчанию 10. Установив **log.height** в 0 даст тот же результат как установка **log.commands** в FALSE.

log.text.color Цвет для текста в окне скриптов; по умолчанию “black”.

log.width Ширина окна скриптов и окна вывода, в символах. По умолчанию 80.

multiple.select.mode Изменяет способ выделения нескольких переменных в списках-выборах. Если установлено “extended” (по умолчанию), левый клик на переменной выбирает ее и снимает выделение с остальных выбранных переменных; Control-левый-клик продолжает выделение (и может добавить в выделение отдельные переменные); Shift-левый-клик расширяет выделение. Это стандартное поведение для Windows. Если выбрано “multiple”, левый клик выбирает переменную и может пометить больше чем одну переменную. Это поведение Rcmdr до версии 1.9-10.

output.height Высота окна вывода в строках. По умолчанию двойная высота окна скриптов, или 20 если окно скриптов отсутствует. Выставить **output.height** в 0 действует так же, как установка **console.output** в TRUE.

output.text.color Цвет вывода в окне вывода, по умолчанию “blue”.

placement Положение окна *R Commander* в пикселях; по умолчанию 40+20”, что выводит окно вблизи верхнего правого угла экрана.

plugins Вектор строка названий плагинов Rcmdr для загрузки при старте Commander. Подключаемые пакеты могут быть загружены через меню *Tools -> Load Rcmdr plug-in(s)*. См. [Plugins](#).

suppress.menus если TRUE, подавляются меню и строка инструментов Commander, позволяя другим программам (таким как Excel) взять на себя эти функции. По умолчанию (естественно) установлено FALSE.

suppress.X11.warnings На некоторых Linux системах, выводятся множественные X11 предупреждения в ответ на Rcmdr команды после открытия графического окна. Установить ее в TRUE (по умолчанию при интерактивном запуске в X11 до версии R 2.4.0) означает подавить вывод ошибок. Нежелательным побочным эффектом является то, что подавляются все предупреждения и ошибки Rcmdr, даже команды набираемые в R. Сообщения от этих команд выводятся в окне сообщений Commander после следующей сгенерированной Rcmdr команды. Некоторые X11 предупреждения могут выводиться при выходе из Commander. Это характерно для версий R до 2.4.0, и соответственно выставлено значение по умолчанию.

retain.messages Если TRUE (по умолчанию FALSE), содержание окна сообщений не стирается между сообщениями. В любом случае, "NOTE"сообщение не сотрет предыдущий "WARNING"или "ERROR".

RExcelSupport Если TRUE (по умолчанию FALSE), меню и вывод обрабатывается Excel.

scale.factor Коэффициент масштабирования, который будет применяться ко всем Tk элементы, таким как шрифты. Это хорошо работает только в Windows. По умолчанию NULL.

showData.threshold Если кол-во переменных в текущем наборе данных превышает некоторое значение (по умолчанию 100), тогда для отображения данных используется **edit()** вместо **showData()**. Основной недостаток этого — то, что вернуться в Commander можно только закрыв окно edit. Причина такой замены очень медленная работа **showData()**, когда количество переменных велико; установка в 0 полностью подавляет использование **showData**.

show.edit.button Установите TRUE (по умолчанию) если Вам нужна кнопка *Edit* в Commander, делающая возможным редактирование активного набора данных. Пользователи Windows устанавливают ее в FALSE для удаления кнопки *Edit* поскольку изменение названий переменных редакторе данных может вызвать падение R (хотя я считаю, что эта проблема все же была решена).

sort.names Установите TRUE (по умолчанию) если хотите отсортированный по алфавиту список названий переменных.

tkwait Эта настройка решает редкую проблему, встречающуюся на не-Windows системах. Если Commander заводит R, тогда установите **tkwait** в TRUE; в других случаях выставите FALSE или не обращайте внимания. Неприятный побочный эффект опции **tkwait** в положении TRUE это то, что команды в окне R подавлены до выхода из Commander. Однако вводить команды в окно R можно. Эта опция не имеет смысла в Windows, ее не надо использовать с Windows R GUI с буферизованным выводом когда вывод направляется в консоль R.

use.rgl Когда TRUE (по умолчанию), загружается модуль **rgl**, если присутствует; если FALSE, модуль **rgl** игнорируется даже если доступен. Модуль **rgl** иногда доставляет проблемы при запуске R в X11.

variable.list.height кол-во переменных для отображения окнах списках; более длинные списки просматриваются прокруткой. По умолчанию 4.

warning.text.color Цвет предупреждений; по умолчанию "darkgreen".

Многие настройки также могут быть установлены через меню *Файл -> Настройки*, что рестартует Commander после установки настроек.

Если Вы хотите всегда загружать R Commander при старте R, Вы можете включить следующий код в загрузочные файлы R (например, в файл **Rprofile.site** в R поддиректории etc):

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

Настройки R Commander могут быть установлены подобным образом. Для большей информации о старте R, см. **?Startup**.

Предупреждение

Скриптовое окно R Commander не предоставляет настоящую R консоль, и имеет определенные ограничения. Я не рекомендую использование R Commander для серьезного программирования или анализа данных основанного на использовании скриптов — используйте специализированный редактор. Если Вы заметите какую либо проблему в использовании окна скриптов, я был бы благодарен за Ваши замечания.

Известные проблемы

Иногда, в Windows, после ввода текста в окне (например, выделение подмножества в диалоговом окне Подмножество данных), Кнопки в диалоговом окне (например, кнопка ОК) не будут откликаться на нажатие. Нажатие в любом месте, внутри или вне окна диалога должно восстановить функции кнопок. Насколько мне удалось выяснить, это проблема с Tcl/Tk для Windows.

Note

Эта версия совместима с SciViews, который в настоящее время работает только под Windows: <http://www.sciviews.org/SciViews-R>; см. [Rcmdr.sciviews-specific](#). В Windows, модуль Rcmdr может быть запущен под Rgui в SDI (single-document interface) режиме, или в `rterm.exe`; Вы можете столкнуться с проблемами запустив Rcmdr в ESS используя NTEmacs или XEmacs.

Author(s)

John Fox (jfox@mcmaster.ca)

See Also

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

Examples

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