

# Анализ данных с R (II).

© А. Б. Шипунов\*, А. И. Коробейников<sup>‡</sup>, Е. М. Балдин\*\*



---

\*dactylorhiza@gmail.com

‡asl@math.spbu.ru

\*\*E.M.Baldin@inp.nsk.su

Эмблема **R** взята с официального сайта проекта <http://developer.r-project.org/Logo/>

# Оглавление

<b>6. Графический интерфейс к R</b>	<b>3</b>
6.1. Может лучше скажем мыши нет? . . . . .	3
6.2. R Commander . . . . .	5
6.3. RKWard . . . . .	7
6.4. JGR . . . . .	8
6.5. SciViews-K . . . . .	9
6.6. Rattle . . . . .	10
6.7. PMG . . . . .	12
6.8. RPMG . . . . .	13
6.9. RWeb . . . . .	13
6.10. gnumeric . . . . .	15
6.11. Emacs . . . . .	15
6.12. Ещё редакторы и среды . . . . .	17

# Глава 6

## Графический интерфейс к R

Работа с R ориентирована на консоль, но и мышь иногда тоже хочется чем-нибудь занять, поэтому в этой статье будут рассмотрены разные дополнения к R с графическим интерфейсом, которые могут оказаться полезными в тяжёлом деле анализа данных.

### 6.1. Может лучше скажем мыши нет?

Прежде чем начать повествование хотелось бы сказать пару слов в пользу консольного интерфейса. Анализ данных — это творческий процесс. Ничего так этот процесс не продвигает, как неспешный вдумчивый ввод команд с клавиатуры. Естественно, этому вводу должен предшествовать интервал времени целиком и полностью посвящённый чтению документации, книг и статей по теме проблемы. Менюшки и кнопки отвлекают, создавая иллюзию простоты творческого процесса, требуя нажать их немедленно и посмотреть что случится. Как правило, ничего при этом не случается, то есть всё равно придётся взять книгу в руки, а затем подумать.

Консольный интерфейс в R, идеален. Он предоставляет пользователю историю команд, позволяет по TAB дополнить их, сохраняет информацию и объекты между сессиями (при условии если пользователь это захочет, естественно). Нужно поработать на удалённом компьютере? Нет проблем с консольным интерфейсом. А если воспользоваться программой `screen` (`man screen` — много интересного), то можно не бояться «разорванных» сессий и случайно закрытых терминалов.

Вы всё ещё хотите использовать GUI при анализе данных? Ну что же, дело ваше, но помните, что Вас предупреждали. ☺

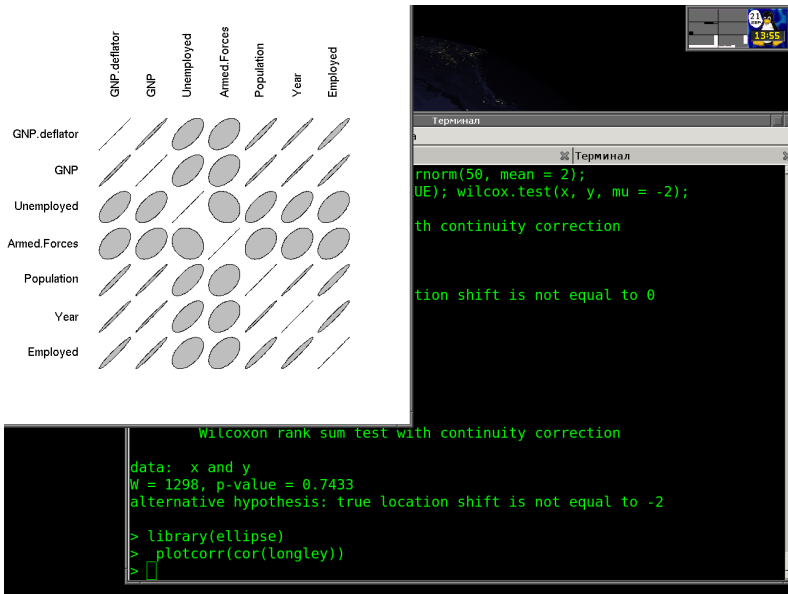


Рис. 6.1. Просто консоль

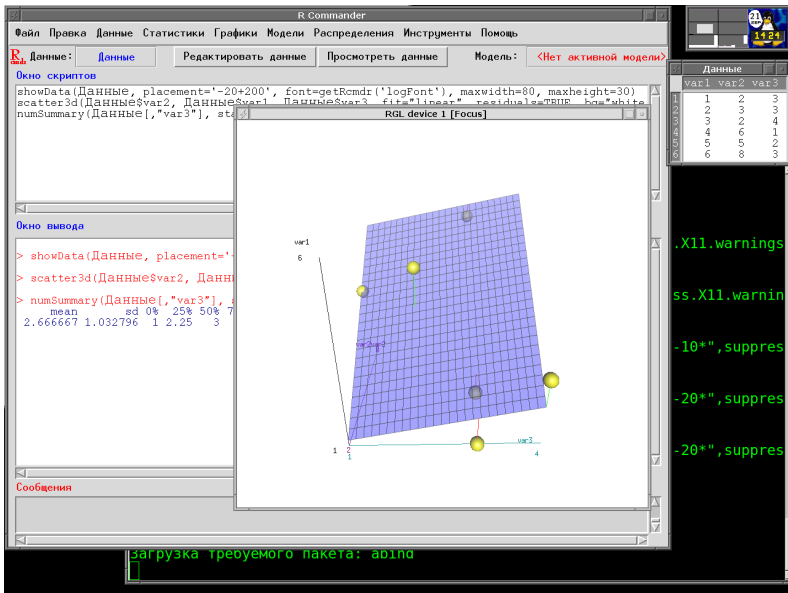


Рис. 6.2. R Commander

## 6.2. R Commander

**R Commander** или **Rcmdr** — это платформонезависимый графический интерфейс к **R**, написанный на Tcl/Tk. Домашняя страничка проекта располагается по адресу <http://socserv.mcmaster.ca/jfox/Misc/Rcmdr/>.

Автор **R Commander** Джон Фокс (John Fox) признаётся, что если говорить о программах статистического анализа, то он не является фанатом интерфейса, состоящего из меню и диалогов. С его точки зрения **R Commander** полезен в основном для образовательных целей при введении в **R**, а также в отдельных очень редких случаях для быстрого анализа. Одна из основополагающих целей, преследуемых при создании интерфейса **R Commander**, являлась мягким переводом пользователя в консоль, где можно автоматизировать свои действия более глобально,

Пакет распространяется под лицензией GNU/GPL, и поэтому доступен во всех стандартных дистрибутивах GNU/Linux. Например, для установки в Ubuntu 8.04 достаточно выполнить команду:

```
% sudo aptitude install r-cran-rcmdr
```

Пакет присутствует также и на CRAN, поэтому его установку можно произвести и силами **R**:

```
> install.packages("Rcmdr", dependencies=TRUE)
```

Обратите внимание на значение опции `dependencies`. **R Commander** зависит от довольно большого числа пакетов. Кроме стандартных **r-base** библиотек **R Commander** предполагает что в системе уже установлены пакеты **abind**, **aplpack**, **car**, **effects**, **lmtest**, **multcomp**, **relimp**, **rgl** и (только для Windows) **RODBC**, поэтому до установке самой программы необходимо убедиться, что все эти пакеты уже стоят. Установка **R Commander** с помощью системных программ установки **aptitude**/**apt-get**, как и установка опции `dependencies` в состоянии `TRUE` позволяет не беспокоиться об этих проблемах. Все необходимые пакеты будут скачены и установлены в нужном порядке.

После установки для запуска **R Commander** в сессии **R** следует выполнить команду:

```
> library(Rcmdr)
```

```
Загрузка требуемого пакета: tcltk
```

```
Загружаю интерфейс Tcl/Tk... готово
```

```
Загрузка требуемого пакета: car
```

```
Версия R Commander 1.3-9
```

При запуске **R Commander** открывается окно, разделённое на **Окно Скриптов** (Script Window), **Окно Вывода** (Output Window), информационное окно **Сообщения** (Messages) и снабжённое довольно «развесистым» меню. Многие

действия в **R Commander** можно выполнять через меню, которое достаточно легко настраивается, например, с помощью редактирования текстового файла `Rcmdr-menus.txt`. Так же можно вводить команды в **Окне Скриптов**, то есть консоль никуда не делась. При создании графиков они появляются в отдельных окнах, как и в случае **R**.

**R Commander** имеет русскую локализацию которая активизируется автоматически если выставлена русская локаль:

```
% locale
LANG=ru_RU.UTF-8
```

Основная проблема русской локализации — это кириллические шрифты, которые выставляются по умолчанию. Поэтому, если предпочтительно использовать локализованный интерфейс, то перед загрузкой **R Commander** ему с помощью команды `options` следует передать примерно следующие опции:

```
> options(Rcmdr=list(
+ default.font="-rfx-fixed-medium-r-normal-*-20*",
+ suppress.X11.warnings=TRUE))
```

Растровый шрифт семейства gfx от Дмитрия Болхивитянова (20 в конце строки — это размер шрифта по умолчанию), который указывается в качестве `default.font`, идёт в составе пакета `xfonts-bolkhov`. Вторая опция `suppress.X11.warnings` подавляет надоедливые сообщения при создании новых графических окон.

Выйти из **R Commander** можно через меню **Файл**→**Выйти**, при этом можно одновременно закрыть и сессию **R**, а можно выйти только из **R Commander**. Если сессия **R** осталась открытой, то повторный запуск **R Commander** выполняется с помощью команды:

```
> Commander()
```

Для вводного ознакомления с возможностями **R Commander** следует прочитать текст `Getting-Started-with-the-Rcmdr.pdf` получить доступ к которому можно через меню **Помощь**→**Введение в R Commander**.

Если по какой-то причине было принято решение анализировать данные исключительно с помощью **R Commander**, то можно сделать так, чтобы при запуске **R** эта графическая оболочка загружалась автоматически. Для этого в файл для пользовательских настроек `~/.Rprofile` достаточно добавить следующие строки:

```
message("Запуск R Commander");
old <- getOption("defaultPackages");
options(defaultPackages = c(old, "Rcmdr"));
```

Глобальная переменная `defaultPackages` содержит информацию об автоматически загружаемых модулей при старте **R**.

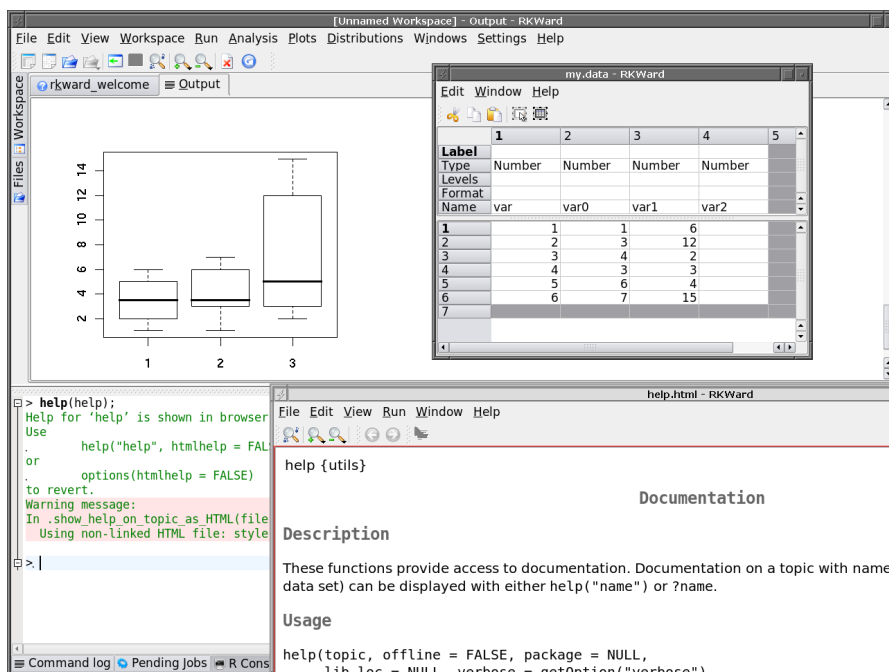


Рис. 6.3. RKWard — R и KDE

## 6.3. RKWard

**RKWard** (<http://rkwad.sourceforge.net>) — это довольно удобный интерфейс к языку программирования **R**. Разработчики **RKWard** старались совместить мощь **R** с простотой использования, подобной предоставляемой коммерческими статистическими пакетами (такими, как, например, **SPSS** или **Statistica**).

Стоит отметить, что это им вполне удалось. Для начинающих пользователей **RKWard** предоставляет широкие возможности по выбору многих стандартных процедур статистического анализа по принципу «point-and-click»: для этого достаточно выбрать соответствующий пункт меню.

Для продвинутого пользователя **RKWard** предлагает достаточно удобный редактор кода с подсветкой, автоматической расстановкой отступов, автодополнением — теми вещами, без которых в настоящее время не обходится ни одна среда программирования. Привычная консоль **R** также никуда не делась, она доступна в любое время на вкладке «R Console».

Авторы **RKWard** взяли курс на как можно более полную интеграцию функций **R** в графическую среду: присутствует браузер текущего окружения, редактор данных. Есть менеджер пакетов, умеющий не только их устанавливать, но и следить за обновлениями, прозрачная интеграция со справочной системой. Кроме

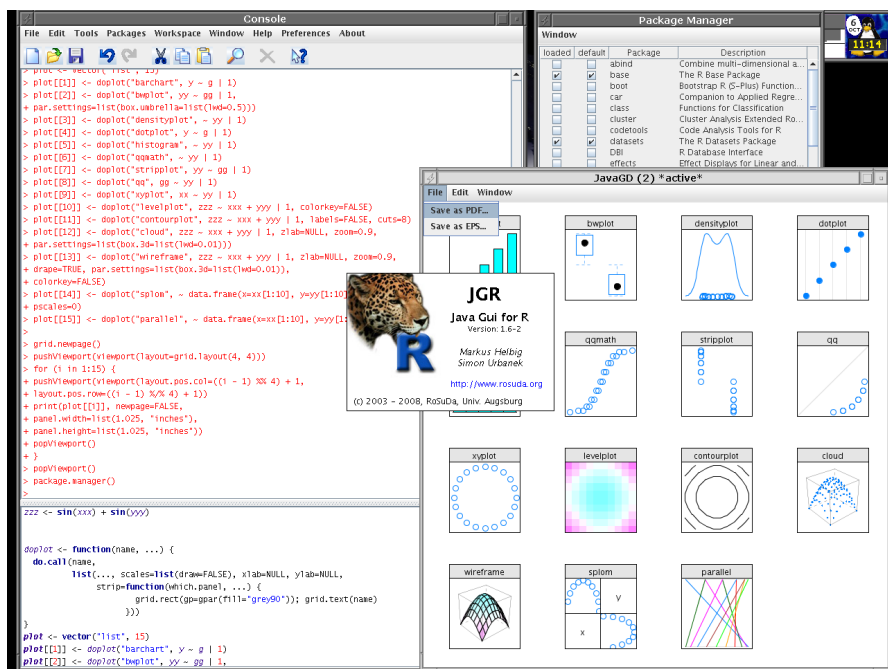


Рис. 6.4. JGR — интерфейс рисует Java.

того, **RKward** умеет «перехватывать» создание графических окон и добавлять к ним чрезвычайно удобные функции типа сохранения содержимого графического окна в файл одного из стандартных форматов, поддерживаемых **R** (PDF, EPS, JPEG, PNG).

Интерфейс **RKward** чрезвычайно гибок: пользователь может его расширять за счёт написания собственных плагинов (собственно, все встроенные средства анализа, доступные сразу же после запуска из меню — это тоже плагины, но только написанные авторами **RKward**).

Программа использует библиотеки KDE (есть и для KDE 4).

## 6.4. JGR

Предпочитаете, чтобы интерфейс рисовался с помощью Java? Тогда **JGR** (Java GUI для **R**) — это решение. **JGR** (произносится как Ягуар) впервые был представлен общественности в 2004 году и до сих пор развивается и поддерживается. Пакет **JGR** распространяется под лицензией GPL2 и его домашняя страничка доступна по адресу <http://jgr.markushelbig.org/JGR.html>.



**JGR** был «made on Mac» со всеми вытекающими, но под Linux он тоже работает. Для использования **JGR** необходимо установить Sun Java(TM) Development Kit (JDK).

```
> # Установка Sun JDK
> apt-get install sun-java6-jdk
> # Выбор Sun JDK в качестве основного из имеющихся альтернатив
> sudo update-java-alternatives -s java-6-sun
> # Настройка Java окружения для R
> sudo R CMD javareconf
```

После установки и настройки Java-окружения следует запустить сессию **R** и выполнить следующие действия:

```
> install.packages('JGR')
# В качестве зеркала выбираю Швейцарию
> library(JGR)
> JGR()
Starting JGR run script. This can be done from the shell as well,
just run /home/user/R/i486/2.6/JGR/scripts/run
```

Всё, **JGR** теперь запущен. После установки **JGR** так же можно запустить с помощью скрипта, путь до которого указывается при запуске **JGR**. В приведённом выше примере `/home/user/R/i486/2.6/JGR/scripts/run`.

В **JGR** есть встроенный текстовый редактор, который конечно, не является полноценной заменой Emacs/vim, но в нём есть подсветка синтаксиса и завершение команд по TAB. В **JGR** встроена гипертекстовая помощь, простенькая электронная таблица, добавлена возможность управлять объектами в том числе и с помощью мышки, а так же есть графический интерфейс для установки и загрузки R-пакетов. С учётом того, что этот GUI может запускаться везде, где есть Java, то на него стóит обратить внимание.

## 6.5. SciViews-K

В мире универсальных редакторов и IDE достаточно известна платформа разработки Komodo от ActiveState. Она довольно хорошо себя зарекомендовала, как удобная среда для языков с динамической типизацией: PHP, Python, Perl, Ruby.

**SciViews-K** (<http://www.sciviews.org/SciViews-K/index.html>) — это молодой и динамично развивающийся проект (первый релиз был не так давно — в июне 2008 года), добавляющий **R** к семейству языков, поддерживаемых свободным редактором с открытым исходным кодом (лицензии как у Mozilla) Komodo Edit ([http://www.activestate.com/Products/komodo\\_ide/komodo\\_edit.mhtml](http://www.activestate.com/Products/komodo_ide/komodo_edit.mhtml)).

На настоящий момент возможности **SciViews-K** довольно скромны: более чем стандартный редактор кода с подсветкой, интеграция с командной строкой **R** и

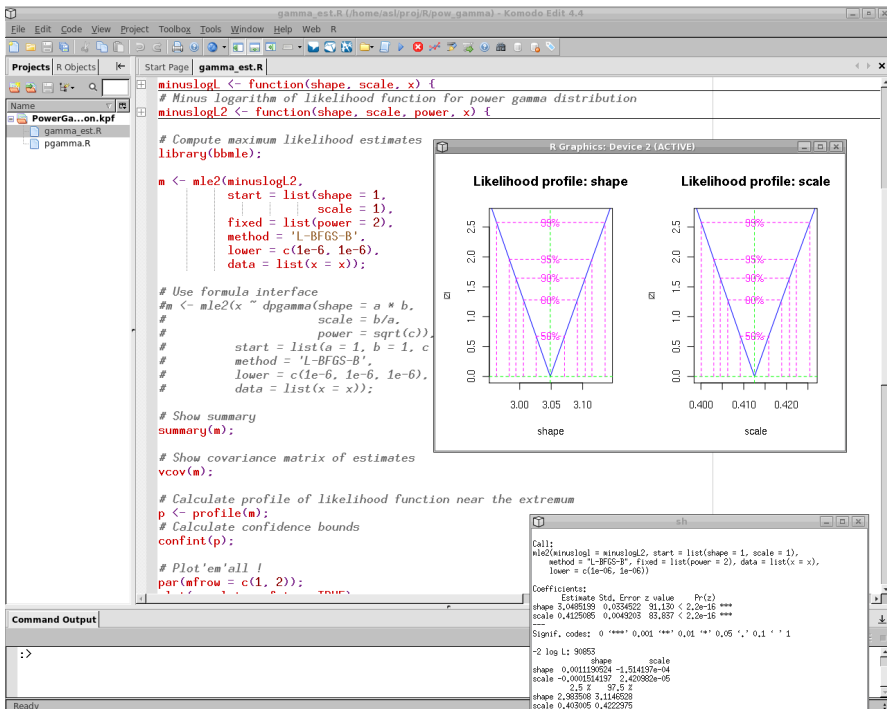


Рис. 6.5. SciViews-K — расширение для Komodo Edit.

системой справки. Кроме того, доступен просмотр текущего окружения и редактор наборов данных.

Поскольку **SciViews-K** является расширением для редактора Komodo Edit, то его установка производится в два приема: сначала надо установить собственно сам Komodo Edit, а потом уже установить **SciViews-K** как расширение к нему. Кроме того, при первом запуске **R** из под **SciViews-K** будет установлен пакет **SciViews** вместе с зависимостями.

## 6.6. Rattle

**Rattle** (<http://rattle.togaware.com/>) — это сокращение, которое расшифровывается как «the **R** Analytical Tool To Learn Easily» (легкая в освоении среда анализа **R**). По уверениям разработчиков **Rattle** он предназначен для интеллектуального анализа данных (data mining), иными словами для выявления скрытых закономерностей или взаимосвязей между переменными в больших массивах необработанных данных. **Rattle** — это среда для «разглядывания» данных человеком и программа всячески человеку в этом деле помогает. Альтернативное

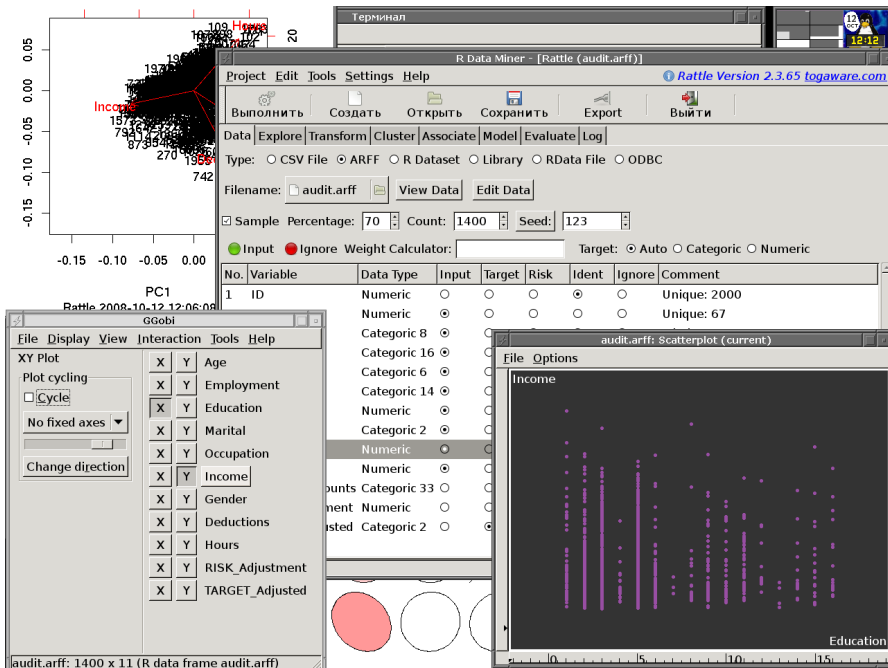


Рис. 6.6. Rattle и анализ данных в нём.

название пакета «A GNOME Data Miner Built on R» намекает, что этот пакет интегрирован в графическую среду GNOME.

Для установки **Rattle** необходимо чтобы уже присутствовали пакеты `r-base-core`, `ggobi` (визуализация данных) и `libglade2-dev`:

```
> apt-get install r-base-core ggobi libglade2-dev
```

После этого в консоли **R** следует выполнить команду:

```
> install.packages("rattle", dependencies=TRUE)
```

и откинуться на спинку кресла. Из-за большого количества зависимостей установка занимает приличный промежуток времени.

Запуск GUI производится, как обычно:

```
> library(rattle)
> rattle()
```

Простой интерфейс анализа позволяет использовать **Rattle** для обучения основам анализа. Программа активно развивается.

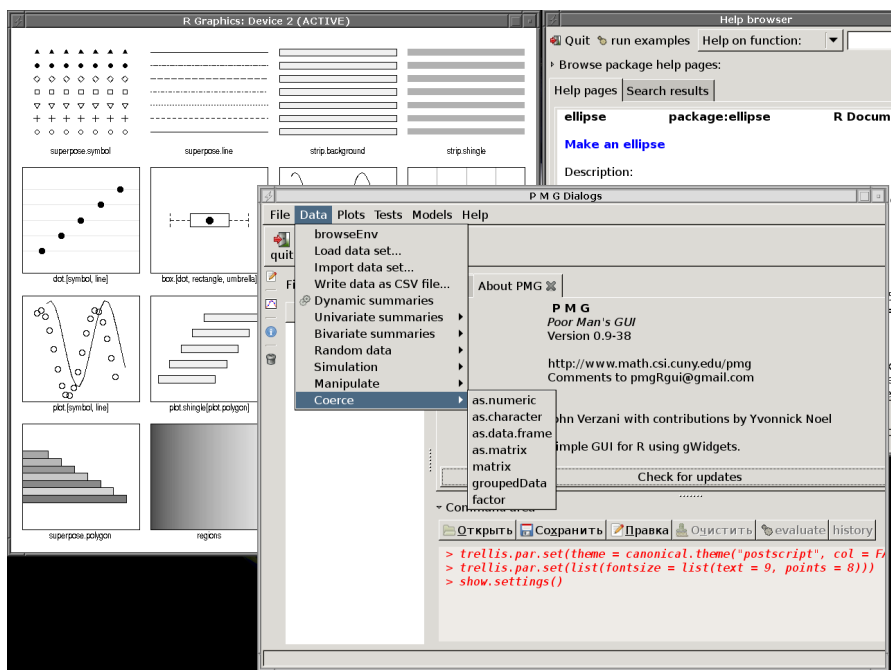


Рис. 6.7. PMG — GUI «для бедных».

## 6.7. PMG

Нужен графический интерфейс, но нет особого желания разбираться в «навороченных» GUI, тогда, возможно, имеет смысл обратить внимание на пакет **PMG** (Poor Man's Gui <http://wiener.math.csi.cuny.edu/pmg>).

Установка проста, хотя у пакета есть множество зависимостей, и проводится силами **R**. Пакет **PMG** является кроссплатформенным. Для отрисовки графического интерфейса используется библиотека **GTK** через пакет **RGtk2**, который в свою очередь можно использовать для создания своего GUI.

```
> # Установка PMG
> install.packages("pmg", dep=TRUE)
> # Запуск PMG
> library(pmg)
```

Этому графическому интерфейсу определённо есть куда развиваться.

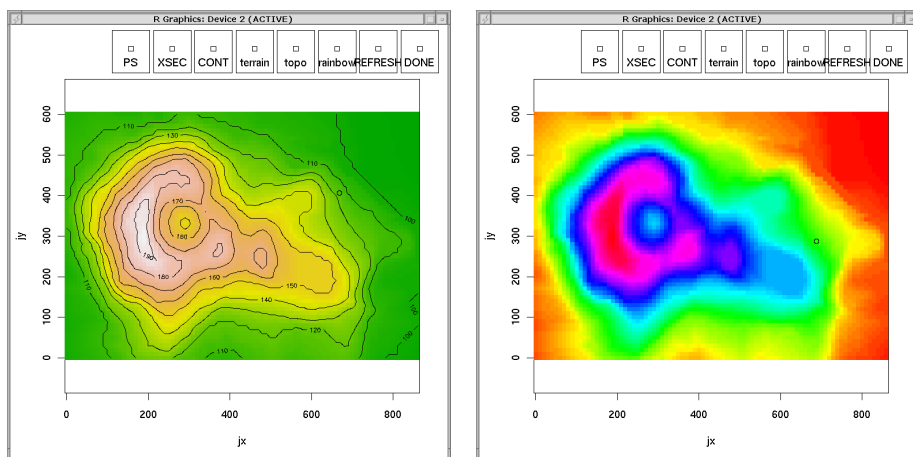


Рис. 6.8. RPMG — GUI «для действительно бедных».

## 6.8. RPMG

Что делать, если нет необходимости в сложной переливающейся графике, а нужно пара простейших интерактивных кнопочек на картинке и минимальное число зависимостей от других пакетов? В этом случае поможет **RPMG** — GUI для действительно бедных (**R**eally **P**oor **M**an's **G**UI), зато с его помощью легко можно организовать интерактивное графическое **R**-окно для личного пользования. На рис. 6.8 одно и то же окно которое меняется в зависимости от активации кнопочек (набор прямоугольничков в верхней части экрана) с помощью мышки. Для установки пакета и демонстрации его возможностей следует выполнить следующие команды:

```
> install.packages("RPMG", dep=TRUE)
> library(RPMG)
> demo(RPMG)
```

Пакет не имеет домашней страницы, но его всегда можно скачать с CRAN: <http://probability.ca/cran/web/packages/RPMG/index.html>

## 6.9. RWeb

Нет желания давать студентам в руки ничего опаснее WEB браузера. Странное желание, но и оно для преподавания статистики с использованием **R** не помеха. Простой набор perl-скриптов **Rweb** (**R** Web Based Statistical Analysis) (<http://www.math.montana.edu/Rweb/>) можно использовать как отправную точку для реализации своего интерактивного Web-проекта. На сайте проекта есть

General Rweb interface - Mozilla Firefox

Файл Правка Вид Журнал Закладки Службк Инструменты Справка

http://bayes.math.montana.edu/Rweb/Rweb.general.html

Самые популярн... Smart Bookmarks Википедия Wikipedia Gmail GDocs

## Statistical Analysis On The Web

# Rweb

To run **Rweb** just type the **R** (or **Splus**) code you want to execute into the text window and then click on the submit button. You will get a new html page with the text output of your code followed by the graphical output (if any) from your code.

Below the submit button is a text area where you can enter the URL for a Web accessible dataset and a browse button for selecting a dataset on your computer. Either way, the dataset will be read in using `read.table` with `header=T` and stored in a dataframe called `X`. The dataframe, `X`, will then be attached so you can use the variable names. Eventually I hope to add several other options for data entry ... let me know if you have any suggestions.

If you use the back button on your browser to come back to this page you can modify your old code and then resubmit it, or you can clear the text area and type in all new code. The computer time for all of this is donated by the Department of Mathematical Sciences, Montana State University.

```
x <- rnorm(100)
y <- exp(x) + rnorm(100)
result <- lm(y ~ x)
plot(x,y)
abline(result)
lines(lowess(x,y), col=2)
```

Submit Erase Everything ...

### External Data Entry

21/09/08 Воскресенье 21:55:55 Готово

Рис. 6.9. Rweb (ввод данных)

Results from Rweb - Mozilla Firefox

Файл Правка Вид Журнал Закладки Службк Инструменты Справка

http://bayes.math.montana.edu/cgi-bin/Rweb/Rweb.cgi

Самые популярн... Smart Bookmarks Википедия Wikipedia Gmail GDocs

Type "{}" to quit R.

```
Rweb-> postscript(file= "/tmp/Rout.11512.N03d.ps", onefile=FALSE)
Rweb->
Rweb-> x <- rnorm(100)
Rweb-> y <- exp(x) + rnorm(100)
Rweb-> result <- lm(y ~ x)
Rweb-> plot(x,y)
Rweb-> abline(result)
Rweb-> lines(lowess(x,y), col=2)
Rweb->
```

### Images

21/09/08 Воскресенье 21:54:55 Готово

Рис. 6.10. Rweb (результат выполнение команд)

его рабочая демонстрация. Там можно прямо в окне браузера набрать команды **R** и рассмотреть результаты в том же окне. Проект давно не обновлялся, но его код вполне обозрим и его можно легко подстроить под свои нужды.

## 6.10. gnumeric

Gnumeric (<http://www.gnome.org/projects/gnumeric>) — это табличный процессор с открытым исходным кодом. Это то, что называется электронная таблица со всеми преимуществами и недостатками этого типа программ. Gnumeric поддерживает систему подключаемых модулей, позволяющая расширять функциональность. В 2003 году сотрудник Bell Labs Дункан Лэнг (Duncan Temple Lang) написал плагин, который позволяет вызвать прямо из Gnumeric любую функцию **R**. Этот плагин был назван без особых затей **RGnumeric** и взять его можно по адресу (<http://www.omegahat.org/RGnumeric/>).

К сожалению пакет для интеграции **R** с Gnumeric представляет из себя скорее демонстрацию возможностей, нежели законченный продукт. В рамках Google of Summer Code 2008 была попытка возродить идеи заложенные в этом пакете на новом уровне <http://www.r-project.org/SoC08/ideas.html>, но похоже, проект всё ещё ждёт своего автора. В любом случае исходники доступны и, хотя просто сама их сборка представляет из себя некоторое приключение, но истинные энтузиасты не ищут лёгких путей.

## 6.11. Emacs

Что это мы всё о GUI, да о GUI. Пора поговорить о **emacs**. Да, да про тот самый программный продукт в который встроена программа управления кофеваркой. Говорят она работает, причём очень не плохо. Посмотрим как **emacs** справляется с **R**.

**ESS** (<http://ess.r-project.org>) — это специализированная интерактивная **R**-среда или мода для GNU Emacs/ХEmacs. **ESS** является сокращением от фразы «Emacs Speaks Statistics», которую можно перевести как «Emacs говорит на языке статистики». Пакет **ESS** поддерживает не только систему статистического анализа **R**, но и другие диалекты языка S (S 3/4, S-PLUS 3/4/5/6/7), а так же SAS, XLispStat, Stata и BUGS. При установке в пакета **ESS**, например, так:

```
|> sudo aptitude install ess
```

и добавлении строчки для инициализации **ESS** в `~/.emacs`

```
;; подключаем ESS
(require 'ess-site)
```

редактору Emacs становятся доступны две дополнительные моды:

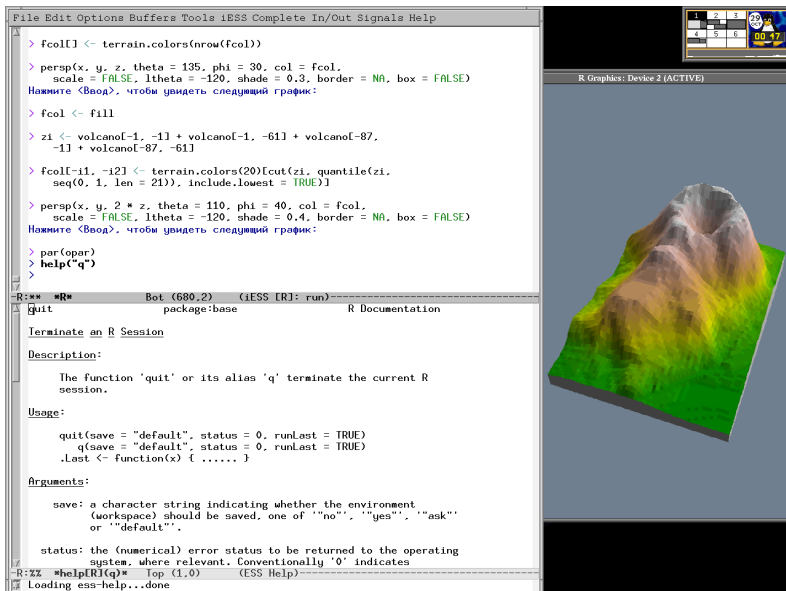


Рис. 6.11. ESS делает R частью emacs.

- 1) мода ESS, которая предназначена для редактирования исходных файлов с использованием команд R (включая дополнительную поддержку для редактирование Rnw-файлов), и
- 2) мода iESS — замена обычной R консоли.

Интерпретатор R запускается, как это не странно с помощью команды R: **Alt+x**, а затем R и перевод строки. При запуске спрашивается о местоположении рабочей директории для запускаемой сессии R, а затем отображается стандартное R приглашение. Можно параллельно запустить несколько сессий, также есть возможность запустить R-сессию на удалённом компьютере (команда `ess-remote`). Работа в этой моде почти ничем не отличается от работе в консоли, за исключением наличия стандартных возможностей редактирования текста в Emacs и дополнительного вспомогательном меню iESS. Из этого меню можно получить доступ к таким интересным возможностям как, например, редактирование объектов R (`^c ^d`). Emacs — это прежде всего текстовый редактор, поэтому подобные особенности делают анализ данных комфортнее.

ESS-мода инициализируется автоматически при редактировании файлов с расширением R. Emacs предоставляет пользователю подсветку синтаксиса, выравнивание исходного кода по TAB, исполнение отдельных строк, фрагментов и всего файла в интерпретаторе R. Дополнительное меню ESS позволяет получить доступ к этим возможностям. Редактирование Rnw-файлов (смесь L<sup>A</sup>T<sub>E</sub>X



и **R** команд для быстрого составления отчётов с помощью пакета **Sweave** — подробнее об этом было рассказано в разделе «Мастера отчётов» в главе «Данные и графики») чуть более мудрёно: мода **ESS** активируется только когда курсор оказывается внутри фрагментов с **R**-командами.

Если есть навыки работы в Emacs, то **ESS** — это верный выбор.

## 6.12. Ещё редакторы и среды

**R** широко известен хоть и в довольно узких кругах, поэтому почти все уважающие себя текстовые редакторы или среды разработки поддерживают его в той или иной степени. Кроме уже упомянутого Emacs к ним относится и Vim, и jEdit, и bluefish, и SciTE, и, естественно, монстр Eclipse имеет специальный плагин для поддержки **R** (<http://www.walware.de/goto/statet>). В этой главе рассказ коснулся только свободных пакетов, но наверняка не сложно будет найти поддержку и в коммерческих аналогах.