

Emacs enhances data analysis and programming with R

R and Emacs through ESS

A.J. Rossini

Modeling and Simulation
Novartis Pharma AG
Basel, Switzerland

14.06.2006 / useR-2006

Outline

- 1 Introduction
- 2 Using Emacs
- 3 Using ESS
- 4 Other Useful Extensions
- 5 Lisp and Emacs Lisp
- 6 Discussion

Goal 1: Efficient Program Editing

Edit, and let the computer repeat

- Getting used to weird, archaic keystrokes
- Weird archaic keystrokes which make life easier
- Leveraging menus and help for Emacs
- Folding and Object Explorers

Goal 2: Process Interaction

You enter text for the process, so edit. . .

- R vs. “emacs -f R”
- Getting Help
- R and R-2
- Practical Directory layout

Goal 3: Emacs is Extensible

Lisp is your friend

- Introduction to Lisp and Emacs Lisp
- Lisp extends Emacs
- Design of ESS
- Future Extensions??

Disclaimer

ESS is many things to many people; it is critical to understand that even for R development, the range of personal philosophies for use is high.

Emacs Overview

- EMACS = Extensible MACroS (many others!)
- One of the oldest and yet still most powerful display editors
- “Religious” doctrines: vi vs. emacs.

On-line Resources

- <http://www.emacswiki.org/>

For general information on projects related to Emacs:

<http://www.gnu.org/>

The Emacsen Problem

different religions cause stress

There are many variants or dialects of Emacs (*similar to the S language having S-PLUS and R*)

- Emacs: “classic version”
- XEmacs: “experiments” (GUIs, dynamic loading, packaging)
- SXEmacs: “splinter” (Unix-focus, streamlining XEmacs)

ESS support is best in order (Emacs, XEmacs, SXEmacs).

Most active developers use Emacs.

Installation

the bane of Emacs

Installation can be tricky:

Linux Emacs usually can be installed or is preinstalled

Microsoft XEmacs is easier to install, Emacs might be better supported.

Mac OSX Fink, or UCLA package, or... (check on the list).

Overview

- Common interface to completely different statistical tools:
R is similar to S-PLUS is different from SAS is similar to Stata is not quite like XLispStat.
- Goal is to support **interactive** programming.

History

The oldest statistically-oriented open source project ('89).

- 1989 Doug Bates and students, U Wisc, 1989 (S-PLUS)
- 1990 Mike Meyer, CMU (now Intelligent Results)
- 1993 David M Smith, Lancaster (now Insightful)
- 1995 Tony Rossini / Martin Maechler / Kurt Hornik, 1995 (start of current team) (R, SAS support)
- 1997 Most recent restructuring generalized with the concept of generic statistical activity support for interactive data analysis (unification theory).

We are still suffering from that last bit of nonsense.

Interactive Programming

Everything goes back to being Lisp

- Interactive programming (as originating with Lisp): works extremely well for data analysis (Lisp being the original “programming with data” language).
- Theories/methods for how to do this are reflected in styles for using ESS.

Good Statistical Analysis is on-line Interactive Programming

Basic Motif

encourages good practices

- Start emacs from the command line or an icon (or similar accelerator)
- Load a file using keystrokes: `C-x C-f`
- Decide you were wrong before you select a file: `C-g` (universal break)
- Change your mind and do it after all: `C-x C-f`
- The biggest headache? Differentiation between commands and text entry (which in itself is a command, i.e. **self-insert-command**).

Commands

2 approaches for commands

- **M-x (command-name)**
- keystrokes

Understanding Keystrokes.

- **C-h** means
- **M-h** means
- **M-C-h** means

Ways to Learn the Strokes.

- forced practice: painful but fast
- accelerators and menus can help

Review of Common Keystrokes

Demo and pain

Point and Mark

critical concepts to learn

- Point: where the cursor is.
- Mark: a location that you've decided (actively or inactively) is important.
- Example: Move cursor somewhere. **M-x set-mark** or **C-(space)**.
- Exchange point and mark: **M-x exchange-point-and-mark** or **C-x C-x**.

Style and Substance

Basics of commenting:

indent to right-hand side

indent at current level

indent flush left

Like:

```
my.x <- rnorm(10)
### This is flush
  ## but this is at level
                        # this is right
```

Introduction

Using Emacs

Using ESS

Other Useful Extensions

Lisp and Emacs Lisp

Discussion

Editing

Interactive

Specials

Sweave

Presentation

Wrapping lines:

Starting R

- Within emacs
- At the command line

Introduction

Using Emacs

Using ESS

Other Useful Extensions

Lisp and Emacs Lisp

Discussion

Editing

Interactive

Specials

Sweave

Philosophy: Files are “truth”

Don't believe your .RData

Philosophy: RData is true

Editing objects without source

Don't use the command line!

If you must use a command line. . .

Navigating Help Buffers

Using Transcripts

Literate Data Analysis.

Sweave originates from the Literate Data Analysis philosophy; support for editing Noweb files existed within ESS 4 years before Sweave.

Evaluation

Chunks and Threads

2 additional commands appear to support development of Sweave documents:

- `eval-chunk` (run code inside a code chunk)
- `eval-thread` (run code inside a series of code chunks)

Accelerated Sweave Development

Packages and Extensions.

Emacs has many modes:

- language-specific markup (programming, markup)
- editor behaviour (spell checking, pending delete, folding, editor emulation (vi, brief))
- IDE capabilities (object explorers, completers)
- database (edb)
- communication (mail, news, www, rdf)
- sub (inferior) process control (telnet, R, ...)
- publication (text mode conversion)
- version control interfaces
- anything you can program!

Relevant to Data analysis and R

- folding modes
- object explorers
- documentation modes

AUC-TeX

well established mode for efficient production of LaTeX documents.

DEMO.

Introduction

Using Emacs

Using ESS

Other Useful Extensions

Lisp and Emacs Lisp

Discussion

LaTeX Extensions

ECB

Muse and Planner

BiBTeX-mode and RefTeX

Citation management

X-Symbol

What you see is mostly what you get.

tex-preview

What you see is exactly what you get.

Introduction

Using Emacs

Using ESS

Other Useful Extensions

Lisp and Emacs Lisp

Discussion

LaTeX Extensions

ECB

Muse and Planner

Emacs Code Browser: overview

Introduction
Using Emacs
Using ESS

Other Useful Extensions

Lisp and Emacs Lisp
Discussion

LaTeX Extensions

ECB

Muse and Planner

ESS integrates with ECB

Introduction
Using Emacs
Using ESS

Other Useful Extensions

Lisp and Emacs Lisp
Discussion

LaTeX Extensions

ECB

Muse and Planner

Extending ECB for ESS: what you want

Muse: a documentation mode.

Wiki-like editing, hyperlinks, within text.

Planner: Task management.

Uses Muse to provide plain-text task management

Lisp.

Wiki-like editing, hyperlinks, within text.

Emacs Lisp.

Uses Muse to provide plain-text task management

Playing in the scratch buffer.

Uses Muse to provide plain-text task management

ielm: command-line usage.

Uses Muse to provide plain-text task management

Organization.

Wiki-like editing, hyperlinks, within text.

Debugging.

Uses Muse to provide plain-text task management

SLIME: some improvements for ESS.

Uses Muse to provide plain-text task management

Future?

- Continued support for R within ESS
- Object and formating support

Summary

- Emacs takes time to learn, but rewards the user.
- ESS enhances R.
- Lisp is worth learning.
- Outlook
 - ESS is the oldest open-source statistics project; yet it can always be improved!
 - Many possible extensions for supporting work modes for Statisticians and Data Analysts. More need implementation.