Short Introduction to ESS

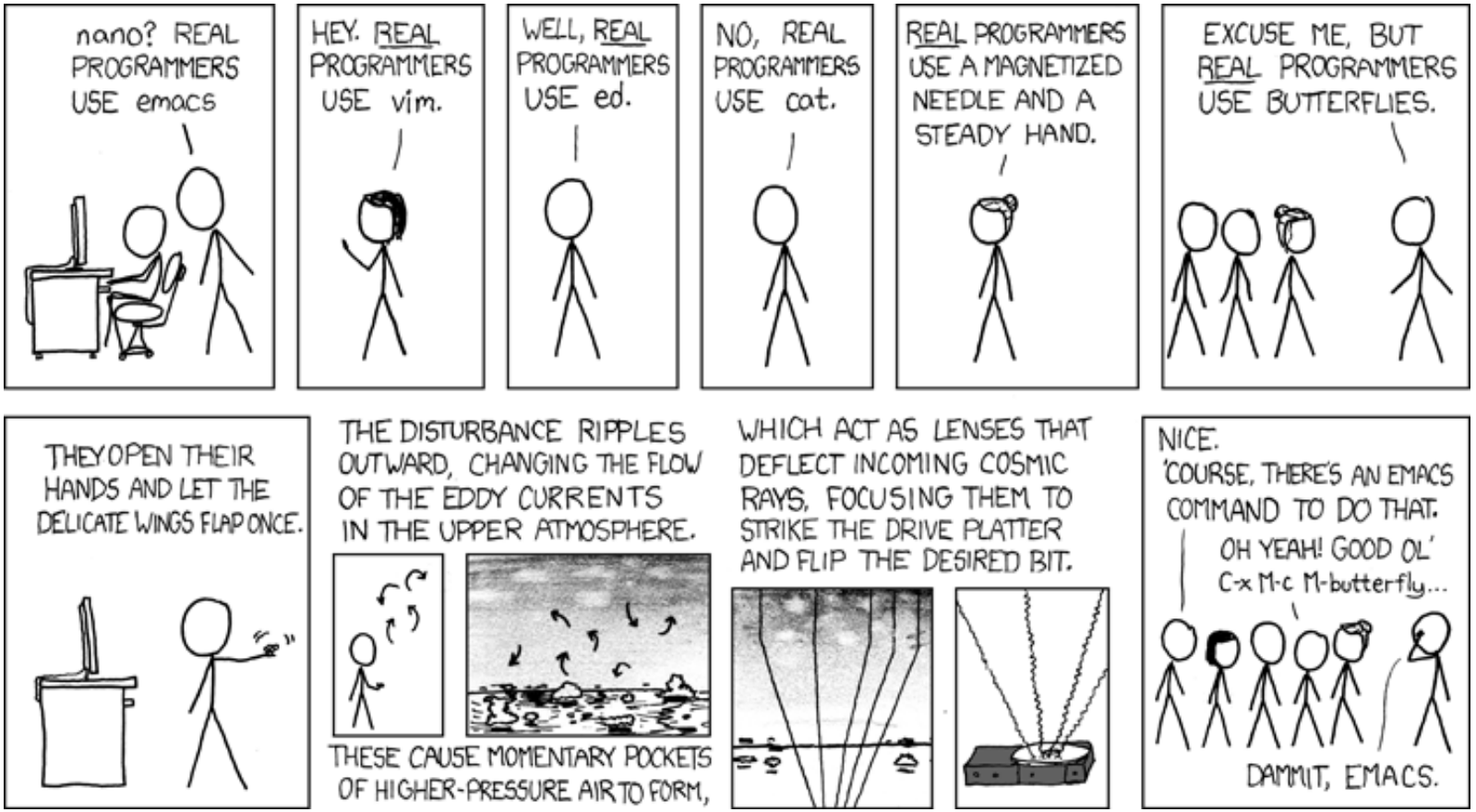
Dirk Eddelbuettel
edd@debian.org, Dirk.Eddelbuettel@R-Project.org

With thanks to ESS Core
for their presentations in SVN

Short presentation
Chicago R User Group
16 December 2010

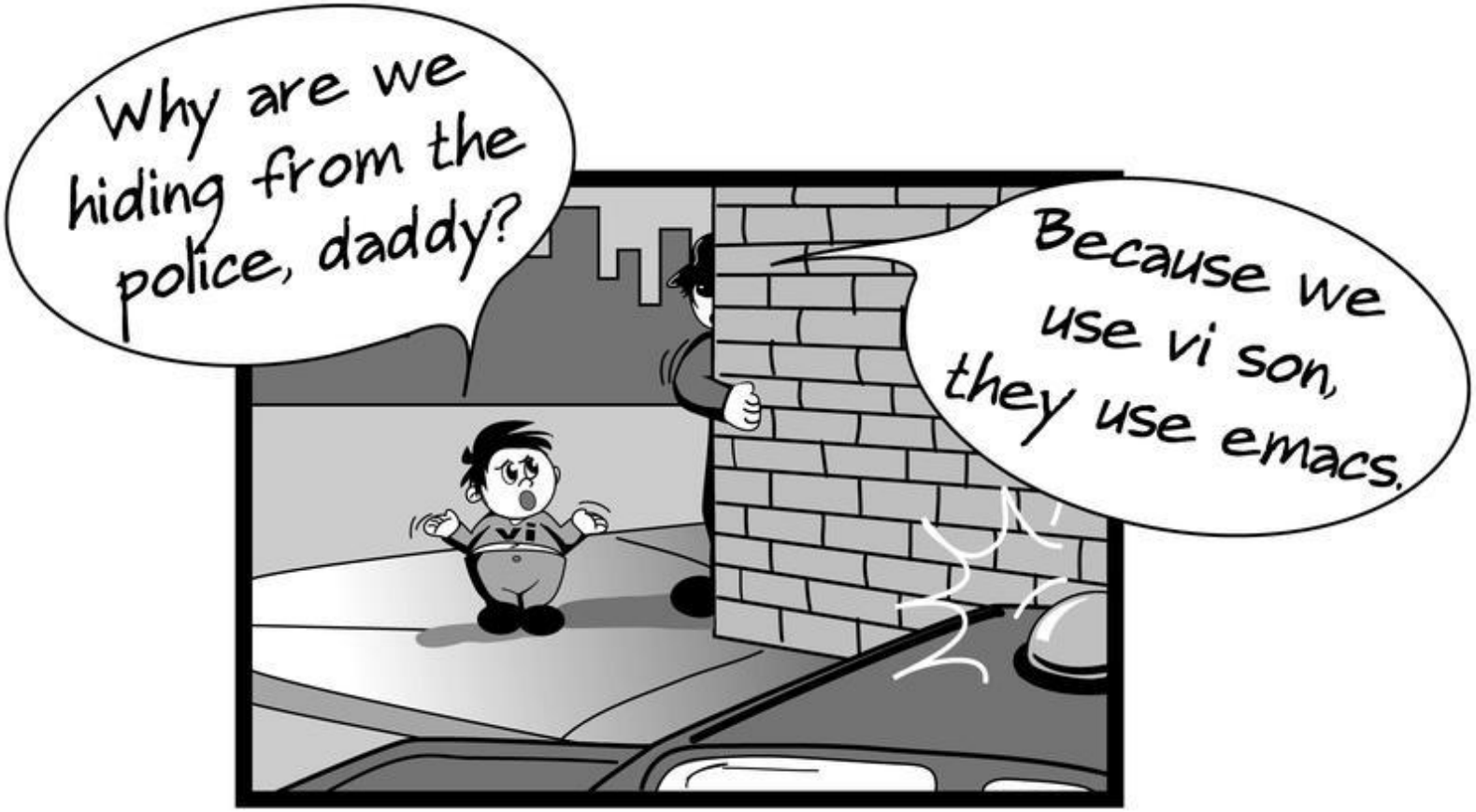
Outline

Why?



Source: http://xkcd.com/378/

Why?



Source: http://www.io.com/~dierdorf/emacsvi.html

Outline

Overview

- Emacs is the extensible, customizable, self-documenting real-time display editor.
- One of the oldest and yet still most powerful display editors
- The name **Emacs** comes from Editor (or Extensible) MACroS. (source of other amusing acronym expansions)
- Originally written in 1976 as a set of extensions to TECO (Text Editor and COrrector). It has since evolved.

Pros and Cons for Emacs

- *Why use Emacs?* It is a powerful text editor which can be extended to be a general interface to a computer (everything can be done within it).
- It is highly portable (in its own way) across many platforms.
- *Why do not use it?* It has a different user interface, which is understandable given its age (little change since 1985).
- It does not follow “modern” keybindings.

Installation and on-line Resources

Installation:

- Windows / OS X: Really nice Emacs and ESS bundles at http://vgoulet.act.ulaval.ca/en/ressources/emacs/
- Linux: `sudo apt-get install emacs23 ess`

Resources:

- Emacs Wiki: http://www.emacswiki.org/

Outline

An email from a long time ago

Still rings very true today

```
From: David M Smith <D.M.Smith@lancs.ac.uk>
To: s-news@utstat.toronto.edu
Subject: Emacs S-mode
Date: Mon, 5 Feb 96 10:03:48 GMT

Without wishing to inflame this somewhat evangelical debate even
further, but since it seems several people are wondering, I thought
I'd point out just what benefits are available in using S-mode under
Emacs compared to the usual Unix "Splus -e" or S-plus for Windows:

* Much improved command-line and history facility, including
  recall of commands containing a particular string, and
  saving of history between sessions

* Completion of S object names with the TAB key (like
  tcsh/bash) to reduce typing

* Automatic recording of session transcripts (input and
  output) and facility for re-executing commands from saved
  transcripts
```

An email from a long time ago

Still rings very true today

```
* A specialised mode for editing S functions, including
  automatic indentation, coloured syntax highlighting, and
  error location.  You can easily edit more than one function
  at a time (and continue to use the S command line in the
  meantime), and do partial or line-by-line evaluation of
  functions for testing/debugging

* Facility for running more than one S session simultaneously

If it seems like these facilities would make your life that little bit
better, and you have the time to invest in learning Emacs and
installing S-mode, then it may be time well invested.  If, when using
S-plus, you never think to yourself "Gee, I wish doing *this* was
easier!" then it's probably not worth it.  Nonetheless, more
information about S-mode is available from

http://www.maths.lancs.ac.uk:2080/~maa036/elisp/S-mode/

# Dave

--
David M. Smith, Department of Mathematics and Statistics, Lancaster University
<D.M.Smith@lancaster.ac.uk>  x3952  http://www.maths.lancs.ac.uk:2080/~maa036/
```

Overview

ESS augments Emacs for R

- Emacs Speaks Statistics (or Emacs Statistical System)
- Oldest active statistically-focused OSS project (1989).
- Supports R, S-PLUS, SAS, Stata, BUGS, JAGS.
- Current team (n=9) spans the US and Europe.
- The new new thing (not mentioned today): org-mode and babel as an Sweave alternative.

Philosophy: Files are “truth”

Don’t believe your .RData

- This is a generally decent approach.
- Saving objects in binary format in special files can be a reasonable strategy
- Work from an R file, *submitting* commands to the R process. This saves the source, and allows for structuring of the input, along with having a transcript of the order of commands to understand reproducibility aspects.
- From here, easy to re-factor your commands into functions, packages, ...

Philosophy: RData is true

Editing objects without source

- This is a dangerous approach; corruption of the RData file is more difficult to fix using an editor or other external tools than corruption of text files, usually.
- Trusting .RData to have what you've done is rather suboptimal strategy (quite prone to user mistakes)

Outline

Outline

Emacs: Here's why

- One *consistent* interface across essentially all computer interaction: shell, directory operations, email, bash, sql, make, latex, html/css, info docs, perl, python, octave/matlab, ..., c/c++, and of course R – and consistent colouring and indentation really help.
- Never ever loose a single keystroke. Whatever you enter goes to a (temp. file) and can be recovered should disaster strike.
- Remote access via ssh using tramp-mode: `/user@host.com:~/dir/file.txt` opens the file as if it was local.
- Another trick: *daemon mode*! One emacs in the back, same session available graphically (via x11) or over tty (ssh/putty, screen, ...) and persists

ESS: Here's why

- Interactive mode rocks with better history, searching, command recall, object completion, file completion, ...
- Interactive sessions can be saved as transcripts and re-executed piece by piece
- Working from a script (or 'log') file is the right philosophy: submit lines, regions, functions, ..., revise, resubmit, ... and you keep a full history.
- Easy access to help, easy navigation of help, easy execution of examples.
- Easy Rd documentation.
- Roxygen support.
- Easy Sweave editing.

Conclusion and some further reading

- Emacs makes for a bloody nice editor for shell, environment, code, help, mail, ...
- ESS ties right into it and gives us ways to edit, view, execute, review, document, ... R code
- A few resources (from the ESS sources) are at `http://dirk.eddelbuettel.com/code/ess/`
  - ESS manual
  - ESS Reference card
  - ESS useR 2006 tutorial
  - ESS paper (JCGS, 2002)
- There will be an ESS tutorial by ESS Core member Stephen Eglen at UseR 2011.