

Demography 213

Optional Material: emacs and ESS mode with R in subshell

Carl Mason
carlm@demog.berkeley.edu

September 15, 2008

abstract ESS (*Emacs Speaks Statistics*) is an emacs “mode” that makes emacs very helpful for writing R, SAS, or STATA code. The document describes how to use emacs with ESS wherein an R process runs inside emacs rather than in a separate terminal window. This is optional material. It will save you time and elbow stress in the long run, but in the short run it will add to the number of secret key combinations competing for your neurons.

1 Splitting the window and launching R

If you have an emacs buffer with a name that ends in “.r”, you are ready to start.

`C-x 2` → splits your emacs window into an upper and lower half.
Both halves initially show the same contents of your initial buffer.

With the window successfully split, it is time to launch R.

`M-x R` → You will be prompted as to what directory R should launch from. The default (which is generally what you want) is the directory from which you launched emacs. Hit Return to accept the default

At this point you should have two windows one holding your buffer for R code, hereinafter the **R code buffer** and the other with the usual blabber and an R prompt, hereinafter the **R execution buffer**.

When you are ready to quit, be sure to type `q()` in the R execution window before exiting emacs.

2 Executing R code with secret key combinations

Once you have successfully launched R as described above, you can use the R execution buffer in more or less the same way as you would use a shell window in which R is running. **BUT** the real benefit of this scheme is to execute snippets of code that you have typed into the R code buffer **without** having to change focus or move the mouse.

To execute code in the R code buffer line by line or by highlighted region, function or buffer you will need to know the secret key combinations given below. But first note the following:

- Almost all of the key combination are really double key combinations almost of which begin with `C-c`.
- You almost always execute these key combinations with the focus in the buffer that holds your R code. `C-c C-c` is the most prominent exception.

Now the secret key combinations, all of which – and more – are documented in *ESS-Mode: Emacs Speaks Statistics* by A. J. Rossini et al which you can download from the course website.

`C-c C-n` → Sends the current line (in the R code buffer) to the R process buffer for execution, AND moves the cursor to the next executable line in the R code buffer.

`C-c C-r` → Sends the currently highlighted region (in the R code buffer) to the R process buffer for execution

You can highlight a region of text either with the mouse or by anchoring a corner with `C-SPACE` and then moving the cursor to the opposite corner.

Sending a region of text to the execution buffer is very handy for testing and debugging.

`C-c C-b` → Sends the entire contents of the current (R code buffer) to the execution buffer

`C-x o` → Switches the focus to the other window. If the R code window has the focus then this will shift it to the R execution window and vice-versa

`M-C-v` → scroll the **other** window forward one screenful

`Shift+M-C-v` → scroll the **other** window backward screenful

`C-c C-c` →

kills the current command that is running wild in the R execution window. NOTE that you need to have the focus in the R execution window for this to work – so generally you will need to “C-x o” before “C-c C-c”ing.

3 Help facility

The help facility works somewhat differently in this setup.

`C-c C-v` →

will prompt in the mini buffer for which help screen to display

After you accept the suggested default or specify a different help item—and you can use `[TAB]` completion – a new buffer will be created with the requested help screen displayed in it. That new buffer will occupy the other window. Consequently either your Rcode buffer or your R execution buffer will have disappeared.

No worries. It’s still there and it’s easy to get back. After reading the help screen, you can either

1. Kill it by typing `C-c k` **with the focus in the buffer that you wish to kill** in which case the previous buffer will again occupy the window.
2. Switch back to the previous buffer via `C-x b` or `C-x C-b` in which case the help screen buffer will still be in your emacs session so you can switch back and forth to read it again.