EmacsSpeaksStatistics

This page is about **Emacs Speaks Statistics**, also sometimes called EmacsStatisticalSystem, but usually called **ESS**. As always, you can click the page heading to list all pages related to ESS.

ESS is a package that integrates various statistical software with Emacs. S, GNU R (a free implementation of S), Stata, Julia, and SAS modes are all supported by ESS, making it possible to hook writing and programming in Emacs right into your stats. It’s pretty slick.

- ESS: http://ESS.R-project.org/
- R: http://www.r-project.org/

Consider using the prefix “ESS” for new ESS-related pages. – PaulJohnson

Basic configuration options

```
(setq ess-eval-visibly nil)
(setq ess-ask-for-ess-directory nil)
```

Completion

- company-mode ESS integration – [part of ESS]
- Auto-complete ESS integration – [part of ESS]
- Icicles Completion – Completion for ESS.
- Automatic YASnippets for function arguments – Easy insertion and editing of arguments and values of a R function.

Code Navigation And Inspection

- Describe object at point - [part of ESS]

- ess-build-tags-for-directory - [part of ESS]
- imenu-anywhere – ido and helm imenu tag selection across all buffers with the same mode
- speedbar – configure speed bar with ESS.

Debugging

- ess-tracebug – [part of ESS]

Tips and Tricks

- Configure Shift+Enter – The key sequence `` `s-<return>' `` can launch R and send commands to R (by FelipeCsaszar)
- Create Tooltips to describe objects – Tooltips are handy!
- Automatic Brackets – Configure ESS to automatically close Brackets and Parentheses (by anonymous)
- Configure Emacs for R in Windows – Concrete advice for Emacs Users Who are Forced to use MS Windows (by PaulJohnson).
- Smarter Ess Underscore behavior
- Smart lessthan: when you want to leave underscore alone
- Format your code with R parser
- Read Table from Emacs Region
- ESS can also be used to send a local buffer or some lines of a local buffer on a remote host where the statistical software is available. This can be the case for SAS. Former methods seem obsolete and following this thread, you should first execute the emacs-command shell and use from this buffer the usual ssh command (ssh user@remotehost) to connect to the remote host. Then, from this “***shell***” (the name is important; using emacs command ssh directly will give a different buffer name) buffer, execute the emacs-command “ess-sas-interactive”, it will launch sas (with -stdio) on the remote host in an interactive mode. Then move to

your local .sas buffer and use the standard ESS commands to send (on the remote host) your line, region or whole buffer, to be executed by the remote sas. Of course, the libraries and data files to be used should reside on the remote host.