

ESS Reference Card for R

Interacting with the process

For use in a process buffer (`inferior-ess-mode`):

<code><RET></code>	Send a command
<code>C-c <RET></code>	Copy old input
<code><TAB></code>	Complete object or file name
<code>C-c C-c</code>	Break
<code>C-g</code>	interrupt Emacs' waiting for S
<code>C-a / C-e</code>	Beginning / End of command
<code>C-c C-u</code>	Delete this command
<code>C-c C-w</code>	Delete last word
<code>C-c C-r</code>	Top of last output
<code>C-c C-o</code>	Delete last output
<code>C-c C-n</code>	Next prompt
<code>C-c C-p</code>	Previous prompt

Command history (part of Menu 'In/Out')

<code>M-p</code>	Previous command
<code>M-n</code>	Next command
<code>C-c C-l</code>	List command history (& choose!)
<code>C-c M-r</code>	Previous similar command
<code>C-c M-s</code>	Next similar command

Others

<code>C-c C-v</code>	Help for object
<code>C-c M-l</code>	Load source file
<code>C-c C-x</code>	List objects
<code>C-c C-s</code>	Display search list
<code>C-c C-q</code>	Quit from S
<code>C-c C-z</code>	Switch to most recent script buffer

Inside ESS Transcripts (I + O)

Inside `*.Rout` files:

<code>M-<RET></code>	Send current line
<code><RET></code>	Send and move to next line
<code>C-<RET></code>	Copy current line into ESS buffer
<code>C-c C-w</code>	Clean Region (→ input only)

Sweave

<code>M-n s</code>	Sweave the file
<code>M-n l</code>	Run latex
<code>M-n p</code>	Postscript file
<code>M-n P</code>	PDF file

Editing source files

For use in `ess-mode` edit buffers, (`*.R` files):

<code><TAB></code>	Indent this line
<code>M-<TAB></code>	Complete filename/object
<code>M-C-/</code>	Indent region
<code>M-C-q</code>	Indent this expression (use at '{')
<code>M-C-a</code>	Beginning of function
<code>M-C-e</code>	End of function
<code>M-C-h</code>	Mark this function
<code>c-u c-u C-y</code>	Yank striped commands

Evaluation commands (Prefix `C-u`: *in/visibly*)

<code>M-C-x</code>	Evaluate region or function or para
<code>C-c C-c</code>	Evaluate region or para. or function & <i>step</i>
<code>C-<RET></code>	Evaluate region or line & <i>step</i>
<code>C-c C-l</code>	Load this buffer's source file
<code>C-c C-j</code>	Evaluate this line
<code>C-c M-j</code>	Evaluate this line and go
<code>C-c C-f</code>	Evaluate this function
<code>C-c M-f</code>	Evaluate this function and go
<code>C-c C-p</code>	Evaluate this paragraph and step
<code>C-c C-r</code>	Evaluate this region
<code>C-c M-r</code>	Evaluate this region and go
<code>C-c C-b</code>	Evaluate this buffer
<code>C-c M-b</code>	Evaluate this buffer and go

Others

<code>C-c C-v</code>	Help for object
<code>C-c C-z</code>	Switch to process buffer

General Commands

`ess-doc-map` (prefix `C-c C-d`):

<code>C-a, a</code>	A propos
<code>C-d, d</code>	D oc on object
<code>C-e, e</code>	D escribe object at point (<code>C-e</code> or <code>e</code> to cycle)
<code>C-i, i</code>	I ndex
<code>C-v, v</code>	V ignettes
<code>C-o, o</code>	D emos
<code>C-w, w</code>	W eb search (dialect dependent)

`ess-extra-map` (prefix `C-c C-e`):

<code>C-d, d</code>	D ump object into edit buffer
<code>C-e, e</code>	E valuate expression (<code>C-u</code> in temp buf)
<code>C-i, i</code>	I nstall package (in R) or library
<code>C-l, l</code>	L oad package (in R) or library
<code>C-s, s</code>	S et indentation style
<code>C-t, t</code>	B uild tags for directory

Reading help files

For use in `*help[R] (...)*` help buffers:

SPC	Next page
DEL	Previous page
n	Next section
p	Previous section
s	Skip ('jump') to a named section
s e	e.g., skip to "Examples:"
l	Evaluate one 'Example' line
r	Evaluate current region
q	Quit window
k	Kill this buffer
x	Kill this buffer and return to ESS
h	Help on another object
?	Help for this mode
a	Display apropos
i	Display index
v	Display vignettes
w	Display this help in web browser
/	Search (also on C-s)

ESS tracebug

Commands in `ess-dev-map` (prefix C-c C-t):

?	Show key help
b	Set BP (repeat to cycle)
C-b, B	Set conditional BP
C-k, k	Kill BP
C-K, K	Kill all BPs
C-n, n	Goto next BP
C-p, p	Goto previous BP
`	Show R traceback (also on C-c `)
~	Show R call stack (also on C-c ~)
C-e, e	Toggle error action (cycle)
C-d, d	Flag for debugging
C-u, u	Un-flag debugged objects
C-w, w	Watch window

Commands in `ess-debug-mode-map`
(active during debugging):

M-C	Continue
M-N	Next line
M-Q	Quit
M-U	Up frame
C-M-S-c	Continue Multiple
C-M-S-n	Next Multiple

Others

C-x ', M-g n	'next-error' (emacs)
M-g p	'previous-error'(emacs)

Package development

Evaluate your code into a given package (in `ess-dev-map`, prefix C-c C-t):

C-s, s Select package for namespaced evaluation

Keybindings in `ess-r-package-mode` (C-c C-w)
(call to `devtools` functions):

C-b, b	Build package (<code>devtools::build()</code>)
C-c c	Check package (<code>devtools::check()</code>)
C-c w	Check and build package with winbuilder (<code>devtools::check_win_devel()</code>)
C-c h	Check and build package with R-hub (<code>devtools::check_rhub()</code>)
C-d, d	Document package (<code>devtools::document()</code>)
C-i, i	Install package (<code>devtools::install()</code>)
C-t, t	Run all tests (<code>devtools::test()</code>)

Editing documentation

Inside `*.Rd` files:

C-c C-e	Insert a template of help file
<TAB>	Indent current line
C-c C-f	Prefix for font specifiers
C-c C-f <TAB>	Display list of font specifiers
C-c C-j	Insert an '\item' on next line
C-c C-n	(in example section) send current line to ESS process
C-c C-p	Preview help file in plain text

For `Roxygen` users, inside `*.R` files
(prefix C-c C-o):

C-o	Insert an <code>Roxygen</code> template
C-c	Add <code>##</code> prefix at the beginning of all selected lines
C-r	Parse R file and get Rd code
C-t	Parse R file and preview help file