Customizing and extending Emacs Speaks Statistics

Frédéric Santos[1]

January 14, 2021

[1]frederic.santos@u-bordeaux.fr

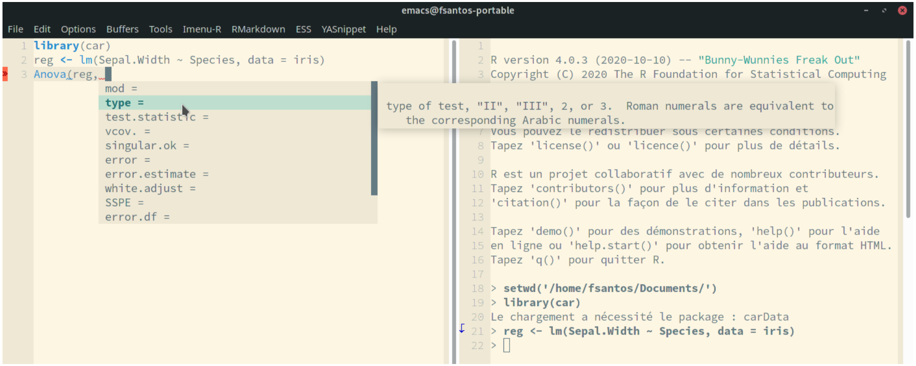
A word about `use-package`

In this document, several pieces of Emacs Lisp code will be proposed so that you can use them in your init file.
It is assumed that you use `use-package` for your init file: the Emacs Lisp code can be adapted in a straightforward manner if you do not use it.
As a reminder, this is the minimal code to add in your init file so as to use `use-package`, once it has been installed:

```
(unless (package-installed-p 'use-package)
  (package-refresh-contents)
  (package-install 'use-package))
(eval-when-compile
  (require 'use-package))
```

Documentation popups I

`company-quickhelp` allows for documentation popups, e.g. to further describe function arguments.



```
library(car)
reg <- lm(Sepal.Width ~ Species, data = iris)
Anova(reg,
```

```
type of test, "II", "III", 2, or 3.  Roman numerals are equivalent to
   the corresponding Arabic numerals.
```

```
R version 4.0.3 (2020-10-10) -- "Bunny-Wunnies Freak Out"
Copyright (C) 2020 The R Foundation for Statistical Computing
Vous pouvez le redistribuer sous certaines conditions.
Tapez 'license()' ou 'licence()' pour plus de détails.

R est un projet collaboratif avec de nombreux contributeurs.
Tapez 'contributors()' pour plus d'information et
'citation()' pour la façon de le citer dans les publications.

Tapez 'demo()' pour des démonstrations, 'help()' pour l'aide
en ligne ou 'help.start()' pour obtenir l'aide au format HTML.
Tapez 'q()' pour quitter R.

> setwd('/home/fsantos/Documents/')
> library(car)
Le chargement a nécessité le package : carData
> reg <- lm(Sepal.Width ~ Species, data = iris)
> 
```

Figure 1: Documentation popups with `company-quickhelp`.

Documentation popups II

The minimal elisp code to add to your init file is straightforward:

```
(use-package company-quickhelp
  :ensure t
  :config
  ;; Load company-quickhelp globally:
  (company-quickhelp-mode)
  ;; Time before display of documentation popup:
  (setq company-quickhelp-delay 0.3))
```

By default, the documentation popup is shown automatically. You can adjust the time before the popup shows up by customizing the variable `company-quickhelp-delay`.

Code snippets I

yasnippet is an Emacs package allowing for the expansion of whole pieces of code you often use (*snippets*) from one given abbreviation.

Key features of yasnippet

- All code snippets are stored as plain-text files in one given directory, so that they are easy to share with other people, and can be easily version controlled.
- As a corollary, it is also easy to retrieve and use large collection of snippets already available online. For instance, Andrea Crotti maintains a great collection available at `https://github.com/AndreaCrotti/yasnippet-snippets`.
- Although we only demonstrate its use within ESS and R here, note that yasnippet is not an R-specific solution, and that you can use it for any other programming language.

Code snippets II

To set up `yasnippet`, proceed through the following steps:

1. Create a directory `snippets/` at some convenient location, and add a subfolder `ess-r-mode/` in this directory.
2. Add the minimal following code in your init file:

```
(use-package yasnippet
  :ensure t
  :config
  ;; Indicate the directory containing your snippets:
  (setq yas-snippet-dirs '("path/to/your/snippets"))
  ;; Load your snippets on startup:
  (yas-reload-all)
  ;; Turn on yasnippet (minor) mode when editing R files:
  (add-hook 'ess-r-mode-hook #'yas-minor-mode))
```

Code snippets III

3. You can now fill your `snippets/ess-r-mode/` directory with your own snippets. For instance, create a file `function` (without any extension) in this directory, with the following contents:

```
#name : function
#key : fun
# --
${1:name} <- function(${2:args}) {
    ${3:body}
}
```

   Each snippet has a unique `name`, and can be triggered by typing a given `key` (followed by `TAB`). As we will see later on, the present snippet allows for the expansion of a template for defining new R functions more easily. The `yasnippet` manual gives more details about the expected syntax to define your own code snippets: `http://joaotavora.github.io/yasnippet/`.

Code snippets IV

4. Now your `snippets` directory should look like:

```
| └── snippets
|     └── ess-r-mode
|         └── function
```

   Feel free to add or retrieve (a lot!) more snippets, i.e. to add more file within the `ess-r-mode` sub-directory.