Setting up R, vimtex and Ultisnips in vim on a Mac

Ronald (Ryy) Glenn Thomas

2024-01-23

Table of contents

Figure 1: vim setup

1 Introduction

Start by installing vim (neovim), R tex vimtex ultisnips

See post "Setting up a minimal neovim…" for details on installing plugins with Neovim.

2 Sections

1. vim sections
2. ALE, lSP, completion, lintr, fix

- R
- julia
- python

2. ftplugins
3. ultisnips
    - dynamic

4. vimtex

- completion

3 Install and configure plugins (source file ~/.vimplugins)

```
```

4 .vimrc

```
syntax enable
filetype plugin indent on
let mapleader = ","
let maplocalleader = " "
source ~/.vimplugins
if $COLORTERM == 'truecolor'
  set termguicolors
endif
colorscheme lightning
set completeopt=menu,menuone,popup,noinsert,noselect
set complete+=k
set dictionary=/usr/share/dict/words
highlight Pmenu  guifg=Black guibg=cyan gui=bold
highlight PmenuSel  gui=bold guifg=White guibg=blue
set gfn=Monaco:h14
set encoding=utf-8
set lazyredraw
set autochdir
set number relativenumber
set clipboard=unnamed
set textwidth=80
set colorcolumn=80
set cursorline
set hlsearch
set showmatch
set hidden
set noswapfile
set ignorecase
set smartcase
set gdefault
```

```
nnoremap <leader>1 <C-w>:b1<CR>
nnoremap <leader><leader> <C-w>w
nnoremap <leader>a ggVG
nnoremap <leader>m vipgq
nnoremap <leader>t :tab split<cr>
nnoremap <leader>v :edit ~/.vimrc<cr>
nnoremap <localleader><leader> <C-u>
nnoremap <localleader><localleader> <C-d>
nnoremap Q @@
noremap - $
noremap : ;
noremap ; :

inoremap <F10> <C-x><C-k>
inoremap <F12> <C-x><C-o>
inoremap <expr> <TAB> pumvisible() ? "\<C-n>" : "\<TAB>"
inoremap <silent> <Esc> <Esc>`^

tnoremap <F1> <C-\><C-n>
tnoremap <leader>1 <C-w>:b1<CR>
tnoremap <leader><leader> <C-w>w
tnoremap <leader>b <C-w>:Buffers<cr>
tnoremap ZD quarto::quarto_render(output_format = "pdf")<CR>
tnoremap ZO source("<C-W>"#")
tnoremap ZQ q('no')<C-\><C-n>:q!<CR>
tnoremap ZR render("<C-W>"#")
tnoremap ZS style_dir()<CR>
tnoremap ZX exit<CR>
tnoremap ZZ q('no')<C-\><C-n>:q!<CR>
```

5 Practical application

1) set analysis goal: Logistic regression of Palmer Penguins data set predicting gender.

Start with a barebones system. i.e. only vim, R and latex installed.

Step one: add the minimum to vim to allow rmarkdown development. Simplest approach: open vim with empty analysis file.
p.Rmd

```
> cd ~/prj/qblog/posts/setup_R_vimtex_ultisnips/penguins
> vim -u .myvimrc p.Rmd
# first line for analysis is to load the penguin data
# enter insert mode and type first R command
inside_vim_normal_mode> i
inside_vim_insert_mode> library(palmerpenguins)
# exit insert mode
inside_vim_insert_mode> C-c
# yank (copy) line into register (unnamed register ")
inside_vim_normal_mode> yy
# open a terminal inside vim and run the R repl
inside_vim_command_mode> term
inside_vim_term> R
# paste last yank (stored in register ") from p.Rmd buffer to R repl
inside_vim_terminal_running_R> C-w ""
```

Next we want to add some plugins to .myvimrc to allow ultisnips snippets for rmd files. Install the vim-plug Vim plugin manager

```
sh> curl -fLo ~/.vim/autoload/plug.vim --create-dirs \
    https://raw.githubusercontent.com/junegunn/vim-plug/master/plug.vim

in vimrc> call plug#begin

in vimrc> call plug#end()

2) mkdir ~/sbx/penguins
3) mvim peng.Rmd
4) type rheader on the first line and hit TAB. The will match the
snippet string in Ultisnips and insert a text template with YAML markup
and latex text (for bibtex use). The snippet has X tabstops to allow
customization of the text block. Enter text in the first block
indicating the project name and then hit C-j to navigate to the next tab
stop: the title. Repeat the process to provide the project specific
information in the YAML and bibliography call. NB: don't leave insert
mode when navigating between tabstops or the ultisnip process with exit.
```

6 Rmd template

6.1 YAML header

The RMD file contains a YAML metadata header delineated with the lines “—” above and below. For this example we want to generate a pdf formatted output file.

The YAML can be as simple as one line specifying the output as pdf.

```
---
output: pdf_document
---
```

Which results in a simple output file as follows:

NB. to invoke file completion in vim for the rmd (or quarto) change the vim filetype using the command:

```
:set filetype=tex
```

then enter, e.g., `\includegraphics{` or `\input{` followed by `C-x C-o`. and a pop-up menu with possible completions with appear.

```
---
title: "Penguins data analysis"
author: "R.G. Thomas"
date: "`r Sys.Date()`"
output:
  pdf_document:
    keep_tex: true
    includes:
header-includes:
    - \usepackage{lipsum, fancyhdr, titling, currfile}
    - \usepackage[export]{adjustbox}
    - \pagestyle{fancy}
    - \pretitle{
    - \begin{flushright} \includegraphics[width=3cm,valign=c]{sudoku.pdf}
```