

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

Ronald (Ryy) Glenn Thomas

9/25/23

## Table of contents

1	Introduction
2	.vimrc
3	.zshrc

1
3
6



Figure 1: vim setup

## 1 Introduction

Start by installing vim (neovim), [R vimtex ultisnips](#)

See post “Setting up a minimal neovim...” for details on installing plugins with Neovim.

[https://www.reddit.com/r/vim/comments/7c7wd9/vim\\_vimtex\\_zathura\\_on\\_macos/](https://www.reddit.com/r/vim/comments/7c7wd9/vim_vimtex_zathura_on_macos/)

<https://stackoverflow.com/questions/40077211/e185-cannot-find-color-scheme>

<https://github.com/morhetz/gruvbox/issues/219>

<https://github.com/junegunn/vim-plug/issues/325>

<https://github.com/dylananaraps/pywal/wiki/Getting-Started>

<https://github.com/dylananaraps/wal.vim>

<https://github.com/dylananaraps/pywal/wiki/Customization>

<https://github.com/lervag/vimtex/issues/1420>  
<https://latextools.readthedocs.io/en/latest/install/>  
<https://mg.readthedocs.io/latexmk.html>  
<https://gist.github.com/LucaCappelletti94/920186303d71c85e66e76ff989ea6b62>  
<https://github.com/lervag/vimtex/issues/1420>  
<https://latextools.readthedocs.io/en/latest/install/>  
<https://github.com/lervag/vimtex/issues/1420>  
<https://github.com/lervag/vimtex/issues/940>  
<https://github.com/lervag/vimtex/issues/663>  
<http://www.math.cmu.edu/~gautam/sj/blog/20140310-zathura-fsearch.html>  
<https://gitter.im/SirVer/ultisnips>  
<https://github.com/SirVer/ultisnips/issues/1107>  
<https://github.com/SirVer/ultisnips/issues/1022>  
<https://github.com/SirVer/ultisnips/issues/850>  
<https://superuser.com/questions/1115159/how-do-i-install-vim-on-osx-with-python-3-support>  
[https://jdhao.github.io/2020/01/05/ultisnips\\_python\\_interpolation/](https://jdhao.github.io/2020/01/05/ultisnips_python_interpolation/)  
[http://witkowskibartosz.com/blog/python\\_snippets\\_in\\_vim\\_with\\_ultisnips.html#.Xnw9gtP7TRY](http://witkowskibartosz.com/blog/python_snippets_in_vim_with_ultisnips.html#.Xnw9gtP7TRY)  
<https://germaniumhq.com/2019/02/07/2019-02-07-Vim-Ultimate-Editing:-UltiSnips/>  
<http://vimcasts.org/episodes/ultisnips-python-interpolation/>  
<https://wraihan.com/posts/vimtex-and-zathura/>

## 2 .vimrc

```
let mapleader = ","
let maplocalleader = " "
" ESSENTIAL plugins
"
call plug#begin()
Plug 'jpalardy/vim-slime'
Plug 'lervag/vimtex'
Plug 'sirver/ultisnips'
"
let g:vimtex_quickfix_mode=0
" let g:UltiSnipsSnippetDirectories = ['~/.vim/UltiSnips']
let g:UltiSnipsJumpForwardTrigger="<tab>"
" let g:UltiSnipsJumpBackwardTrigger="<c-k>"
let g:slime_target = "vimterminal"
let g:slime_no_mappings = 1
let g:slime_vimterminal_config = { "vertical": 1 }
let g:slime_vimterminal_cmd = "R"
let g:slime_cell_delimiter = "```"
"
"plugins for IDE like enhancements
"
Plug 'dense-analysis/ale'
highlight clear ALEErrorSign
highlight clear ALEWarningSign
let g:ale_sign_error = ' '
let g:ale_sign_warning = '.'
let g:ale_completion_enabled = 1
"
Plug 'vifm/vifm.vim'
Plug 'junegunn/fzf', { 'do': { -> fzf#install() } }
Plug 'junegunn/fzf.vim'
Plug 'tpope/vim-unimpaired'
" Plug 'ervandew/supertab'
Plug 'justinmk/vim-sneak'
Plug 'tpope/vim-surround'
Plug 'tpope/vim-repeat'
```

```

Plug 'tpope/vim-commentary'
Plug 'machakann/vim-highlightedyank'
Plug 'vim-airline/vim-airline'
let g:sneak#s_next = 1
let g:sneak#use_ic_scs = 1
let g:airline#extensions#tabline#enabled = 1
let g:airline#extensions#fzf#enabled = 1
" let g:SuperTabDefaultCompletionType = "<C-X><C-O>"
call plug#end()
"
" optional settings
"
colorscheme lightning

set completeopt=menu,menuone,preview,noselect,noinsert
" set completeopt+=longest,menuone,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 termguicolors
set gfn=Monaco:h14
set encoding=utf-8
set lazyredraw
set autochdir
set termguicolors
set number relativenumber
set clipboard=unnamed
set textwidth=72
set colorcolumn=80
set cursorline
set hlsearch
set splitright
set hidden
set incsearch
set noswapfile
set showmatch
set ignorecase
set smartcase

```

```

set gdefault
"
" optional mappings
"

inoremap <F10> <C-x><C-k>
inoremap <F12> <C-x><C-o>
inoremap <silent> <Esc> <Esc>``
"

nnoremap <leader>f :Files<CR>
nnoremap <Leader>' :Marks<CR>
nnoremap <Leader>/ :BLines<CR>
nnoremap <Leader>b :Buffers<CR>
tnoremap <leader>b <C-w>:Buffers<cr>
nnoremap <Leader>r :Rg<CR>
nnoremap <Leader>s :Snippets<CR>
" nnoremap <leader>1 <C-w>:b1<cr>
" nnoremap <leader>2 <C-w>:b2<cr>
" nnoremap <leader>3 <C-w>:b3<cr>
nnoremap <leader><leader> <C-w>w
nnoremap <leader>U <Cmd>call UltiSnips#RefreshSnippets()<CR>
nnoremap <leader>a ggVG
nnoremap <leader>d :EditVimf<cr>
nnoremap <leader>n :ALENext<CR>
nnoremap <leader>t :tab split<cr>
nnoremap <leader>u :UltiSnipsEdit<cr>
nnoremap <leader>v :edit ~/.vimrc<cr>
nnoremap <localleader><leader> <C-u>
nnoremap <localleader><localleader> <C-d>
nnoremap <localleader>m vipgq
nnoremap Q @@
noremap - $
noremap : ;
noremap ; :
"

tnoremap <F1> <C-\><C-n>
" tnoremap <leader>0 <C-w>:ls<cr>:b<localleader>
" tnoremap <leader>1 <C-w>:b1<cr>
" tnoremap <leader>2 <C-w>:b2<cr>

```

```

" tnoremap <leader>3 <C-w>:b3<cr>
tnoremap <leader><leader> <C-w>w
tnoremap ZQ q('no')<C-\><C-n>:q!<CR>
tnoremap ZS style_dir()<CR>
tnoremap ZX exit<CR>
tnoremap ZZ q('no')<C-\><C-n>:q!<CR>
:noremap <C-c><C-c> <Plug>SlimeParagraphSend
:noremap <C-c>v <Plug>SlimeConfig
autocmd FileType r,rmd noremap <localleader>l <Plug>SlimeSendCell<CR>
"

autocmd FileType r,rmd noremap <CR> V<Plug>SlimeRegionSend+
autocmd FileType r,rmd noremap <space><CR> vip<Plug>SlimeRegionSend}
autocmd FileType r,rmd xnoremap <CR> <Plug>SlimeRegionSend+
" maps for R and RMD, space + letter. def as macros. stored in " ~/.viminfo
autocmd FileType r,rmd noremap <space>j @j " j jump to next chunk
autocmd FileType r,rmd noremap <space>h @h " h print head() of object
autocmd FileType r,rmd noremap <space>p @p " p print() object
autocmd FileType r,rmd noremap <space>d @d " d dim() of object
autocmd FileType r,rmd noremap <space>n @n " n names() of object
autocmd FileType r,rmd noremap <space>s @s " s str() of object

```

### 3 .zshrc

```

export ZSH="$HOME/.oh-my-zsh" export EDITOR="vim"
ZSH_THEME="muse" DISABLE_UNTRACKED_FILES_DIRTY="true"
plugins=(zsh-autosuggestions z git pass vi-mode scd common-
aliases you-should-use)

source $ZSH/oh-my-zsh.sh

bindkey -v alias mm='mutt' alias sk='open -a Skim'
alias vc='vim ~/.vimrc' alias vz='vim ~/.zshrc' alias
sz='source ~/.zshrc' alias p2='enscript -C -2 -r -j -
media=Letter' alias p1='enscript -j -media=Letter' alias
yr="yabai -restart-service" alias lt='eza -lrfha -sold' alias
mvim="/Applications/MacVim.app/Contents/bin/mvim"

```

```

alias tp='trash-put -v' alias rm='echo "This is not the com-
mand you are looking for."; false' alias s='scd' alias ZZ='exit'
mma () { /Applications/Mathematica.app/Contents/MacOS/WolframKernel
-script $1 } gz() { git add . git commit -a -m "$1" git push }

cdpath=($HOME/Dropbox/prj          $HOME/Dropbox/sbx
HOME/Dropbox/work)exportTEXINPUTS  ='      .      :
/Users/zennjshr/images : /Users/zenn/shr : ' exportPATH =
"/opt/homebrew/sbin : /opt/homebrew/bin :PATH:$HOME/bin"

export vpc_id="vpc-14814b73" export subnet_id="subnet-
f02c90ab" export ami_id="ami-014d05e6b24240371" export
key_name="power1_app" export instance_type="t2.micro"
export storage_size="30" export ami_id="ami-014d05e6b24240371"
export security_grp='sg-02fcdde25246d9fcd' export static_ip='13.56.106.160'
if type rg &> /dev/null; then export FZF_DEFAULT_COMMAND='rg
-hidden -l ""' export FZF_DEFAULT_OPTS='-m -height
50% -border' fi ZSH_AUTOSUGGEST_HIGHLIGHT_STYLE="fg=011,bg=black,bold,underline"
""

```