Minimal document example for plot generation using Org Mode with R source code and tikzDevice library

suited for a Doom Emacs distribution.

manuel.fuica.morales@gmail.com

10 September 2021

Contents

1 About

1.1 this tutorial

I'd recommend to read the pdf version of this, or GUI emacs at least, CLI emacs wont show the generated graph.

I write tutorials with the objective of reminding me later how to do the stuff when I forgot, so they are written from a noob-ish POV. They are more like personal notes with a touch of in-case-someone-else-reads-it. It's on the web and it's not private anyway so...

The objective of this file is to be a minimal example of a 'live' document using orgmode capabilities to work with `R`.

I have used as guide mainly the following tutorials:

- `https://orgmode.org/worg/org-contrib/babel/languages/ob-doc-LaTeX.html`
- `https://orgmode.org/worg/org-contrib/babel/languages/ob-doc-R.html`
- `https://orgmode.org/worg/org-tutorials/org-R/org-R.html`
- `https://github.com/erikriverson/org-mode-R-tutorial/blob/master/org-mode-R-tutorial.org`

1.2 `tikzDevice` and `knitr`

- Use:
  1. `file.R` -> `knitr` -> standalone `file.tex` file.
  2. `file.R` -> `tikzDevice` -> `file.tex` file that only contains the plots of the `file.R` program. The plots in the `file.tex` file are written in `tikz`, whereas in the `file.R` the plots are written in, redundantly, `R`.
- both `tikzDevice` and `knitr` are `R` libraries that enable `tex` friendly output of an `file.R`.
- `tikzDevice` enables `R` to create a `file.tex` out of the plots of a `file.R` file.
- `knitr` instead, enables the whole `file.R` file to be exported into a `file.tex` file, including the plots and the rest of the document. You can use it to build `reports`, `beamer` presentations, etc.

1. `R code` -> `tikzDevice` -> file that can be processed by `LaTeX`, but it *only* contains `tikz` code. Requires a separate preamble in order for `LaTeX` to sucessfully generate a `pdf` file from it.

2. `R code` -> `knitr` -> file that can be processed by `LaTeX`, and can be a standalone file which generates a `report` or `beamer` presentation, etc. *Can* be directly proccesed by `LaTeX` and will sucessfully generate a `pdf` file.

2 Software used

2.1 emacs

```
emacs --version
```

```
GNU Emacs 27.1
Copyright (C) 2020 Free Software Foundation, Inc.
GNU Emacs comes with ABSOLUTELY NO WARRANTY.
You may redistribute copies of GNU Emacs
under the terms of the GNU General Public License.
For more information about these matters, see the file named COPYING.
```

2.2 doom

```
~/.emacs.d/bin/doom version
```

```
> Executing 'doom version' with Emacs 27.1 at 2021-09-10 06:37:55
  Doom v3.0.0-alpha (HEAD -> develop 13958c81 2021-07-10 10:30:49 -0400)
```

2.3 org

```
org-version
```

```
9.5
```

2.4 R

```
version
```

```
               -
platform       x86_64-pc-linux-gnu
arch           x86_64
os             linux-gnu
system         x86_64, linux-gnu
status
```

```
major          4
minor          1.1
year           2021
month          08
day            10
svn rev        80725
language       R
version.string R version 4.1.1 (2021-08-10)
nickname       Kick Things
```

3 Emacs configuration

You have to configure your `~/.emacs/` in order for it to work with `R`.

Doom Emacs does a pretty good job. You can read the docs at:

- `https://github.com/hlissner/doom-emacs/blob/develop/modules/lang/ess/README.org`

In short, at time of writing all you have to do is

```
(ess +lsp)               ; emacs speaks statistics
```

in your `~/.doom.d/init.el`. And probably install some stuff via `sudo apt install ...` but that's out of the scope of this document. And remember to `~/.emacs.d/bin/doom sync` after you edit that file.

Once your Emacs is configured to work with `R` ...

4 Simple plot

4.1 dependencies

Your `file.tex` generated by orgmode should contain the `tikz` library.

This is represented by `\usepackage{tikz}` somewhere in the preamble — before the `\begin{document}` line of the exported `file.tex` —. You accomplish this by putting

```
#+LATEX_HEADER: \usepackage{tikz}
```

somewhere early in the document.

Also, at least in my case, my `R` distribution, installed via command line using the typical `sudo apt install r-cran-...` did not come with the `tikzDevice` library pre-installed — I'm a complete noob with R so I do not know if that's default —. Credits to this thread for the help:

- `https://lightonphiri.org/blog/r-graphical-representation-installing-tikzdevice-pa`

4.2 codeblocks

From within a shell — ahem, `tmux`, ahem — you have to

```
$ R
> install.packages('tikzDevice')
```

then follow the prompts.

> I mean, you can do that from within an orgmode code source block, but that led to some issues that are not worth solving for the document's purpose.

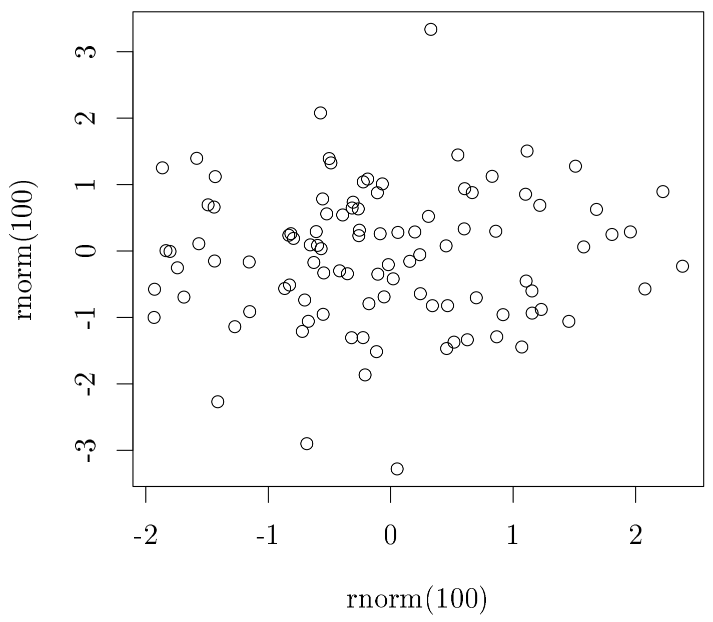
Once the `tikzDevice` library is installed, you can proceed to evaluate the following code blocks — using `C-c C-c` — :

- Actually load the library

```
library("tikzDevice")
```

- And the magic part

```
tikz(console=TRUE, width=4, height=4)
plot(rnorm(100), rnorm(100))
dummy <- dev.off()
```



4.3 org code of the above

```
...
...
...
Once the =tikzDevice= library is installed, you can proceed to evaluate
the following code blocks --- using =C-c C-c= --- :

- Actually load the library
#+begin_src R :session :exports code :results silent
  library("tikzDevice")
#+end_src

- And the magic part

#+name: test_plot
#+begin_src R :session :exports both :results output latex :file test.png
```

```
  tikz(console=TRUE, width=4, height=4)
  plot(rnorm(100), rnorm(100))
  dummy <- dev.off()
#+end_src

#+RESULTS: test_plot
#+begin_export latex
% Created by tikzDevice version 0.12.3.1 on 2021-09-10 04:53:27
% !TEX encoding = UTF-8 Unicode
\relax
\begin{tikzpicture}[x=1pt,y=1pt]
\definecolor{fillColor}{RGB}{255,255,255}
\path[use as bounding box,fill=fillColor,fill opacity=0.00] (0,0) rectangle (289.08,28
...
...
...
```

4.4 exporting to pdf

After all that coding all you have to do is export the `file.org` to a `file.tex` which is then converted to a `file.pdf` via `pdflatex` or another `latex/tex` engine.

You accomplish that pipeline via the `org-export-dispatch` — you can `M-x org-export-dispatch` — and select the `[l] Export to LaTeX` option and then `[o] As PDF file and open`.

If you want some of the functionality of a jupyter notebook within emacs there you go. Still jupyter has the widgets. But I think this is pretty cool too.

Happy hacking.

manuel.fuica.morales@gmail.com

10 September 2021