Organizing research projects with an efficient open-source tool (emacs org-mode)

Feiming Chen

October 23, 2018

Outline

Topic

Abstract

I will introduce an amazing computing environment (*emacs org-mode*) in which you can work day in and day out. Whether you are a statistician or data scientist or algorithm/software developer, you will benefit from this elegant open-source tool for efficient and robust reproducible research. I will outline my favorite set-up (Virtual Machine, Linux, Emacs, LaTeX, CDLaTeX, R, ESS) and demonstrate its potential in statistical analysis and algorithm development. This presentation itself is also written in the emacs org-mode (exported to PDF via LaTeX/Beamer).

What is Emacs Org-mode?

- *An amazing open-source productivity tool!*
- Write and Organize research notes or projects for private use or publication (PDF, HTML, or Customized)

Example (A .org text file)

```
* Topic 1
** Slide Title
*** Block 1
    Text
*** Block 2
    Text
* Topic 2
  Text
```

Example (The same file folded)

```
* Topic 1 ...
* Topic 2 ...
```

Pros and Cons

Competing Apps/IDE's: Microsoft Office, R Studio, R Markdown, Visual Studio, Javadoc, Overleaf/ShareLaTeX, LyX, etc.

Cons

- Not good for collaborative use (unlike Microsoft Office).
- Steep learning curve: Emacs, keyboard shortcuts, ...

Pros and Cons

Competing Apps/IDE's: Microsoft Office, R Studio, R Markdown, Visual Studio, Javadoc, Overleaf/ShareLaTeX, LyX, etc.

Cons

- Not good for collaborative use (unlike Microsoft Office).
- Steep learning curve: Emacs, keyboard shortcuts, ...

Pros

- Writing efficently (e.g. focus on one section at a time).
- Organizing efficiently (e.g. reduce clutter and have the big picture of your composition)
- Blending multiple languages (human, computer, mathematics, ...) in one document (Reproducible Research, Literate Programming)

Topic

About Emacs

- Emacs is an old text-based writing tool (vs. GUI-based).
- Extensive use of keyboard shortcuts for efficient writing.
- Cannot cut and paste with non-text objects (graphics, spreadsheets, etc.)
- Learning takes time but ultimately rewarding
- Community support (google: "emacs" + question)
- Org Mode is pre-installed.

About Emacs Org-mode

Organizing information

- Show outline and Hide lower-level details by folding sections/subsections.
- Type `*` or `**` to start a (sub)section (hint: set variable `org-hide-leading-stars` to `t`; see more)
- Manage all related projects in one file, instead of using folders/sub-folders.

About Emacs Org-mode

Organizing information

- Show outline and Hide lower-level details by folding sections/subsections.
- Type `*` or `**` to start a (sub)section (hint: set variable `org-hide-leading-stars` to `t`; see more)
- Manage all related projects in one file, instead of using folders/sub-folders.

Coding in *multiple* languages and *reproducing* results.

- Examples: `R`, `C++`, `LaTeX`, `python`, `sh`, `perl`, `sql`.
- Code blocks can pass information in between.
- Supports Noweb literate programming style: Name a code block and use that name in other code (analogous to *macro*)

An example of writing source code in Emacs Org-mode

- Type <s then <TAB> to insert a code block (see Org-mode Easy templates); then type language identifier (e.g. R, C).

An example of writing source code in Emacs Org-mode

- Type `<s` then `<TAB>` to insert a code block (see Org-mode Easy templates); then type language identifier (e.g. `R`, `C`).
- Type `C-c '` to edit the code in a dedicated buffer with language-specific features (syntax highlighting, language help, etc.).

An example of writing source code in Emacs Org-mode

- Type <s then <TAB> to insert a code block (see Org-mode Easy templates); then type language identifier (e.g. R, C).
- Type C-c ' to edit the code in a dedicated buffer with language-specific features (syntax highlighting, language help, etc.).
- Type C-c C-c to evaluate the code block.

An example of writing source code in Emacs Org-mode

- Type `<s` then `<TAB>` to insert a code block (see Org-mode Easy templates); then type language identifier (e.g. `R`, `C`).
- Type `C-c '` to edit the code in a dedicated buffer with language-specific features (syntax highlighting, language help, etc.).
- Type `C-c C-c` to evaluate the code block.
  - The result can be *hashed* (so *NO* computation will be done again if the code remains the same) with `:cache yes` property.

An example of writing source code in Emacs Org-mode

- Type `<s` then `<TAB>` to insert a code block (see Org-mode Easy templates); then type language identifier (e.g. `R`, `C`).
- Type `C-c '` to edit the code in a dedicated buffer with language-specific features (syntax highlighting, language help, etc.).
- Type `C-c C-c` to evaluate the code block.
  - The result can be *hashed* (so *NO* computation will be done again if the code remains the same) with `:cache yes` property.
  - Both the code block and result block can be folded for clear view.

An example of writing source code in Emacs Org-mode

- Type `<s` then `<TAB>` to insert a code block (see Org-mode Easy templates); then type language identifier (e.g. `R`, `C`).
- Type `C-c '` to edit the code in a dedicated buffer with language-specific features (syntax highlighting, language help, etc.).
- Type `C-c C-c` to evaluate the code block.
  - The result can be *hashed* (so *NO* computation will be done again if the code remains the same) with `:cache yes` property.
  - Both the code block and result block can be folded for clear view.
- Type `C-c C-v f` to export the code blocks to a code file (e.g. `*.R`) by *tangling*.

Example Code Blocks

```
* Example
  We may write code in =R= like
  #+begin_src R
    setwd("~/job/2018-Projects/GASP/src")
    cars %T>% plot %>% summary
  #+end_src

  #+RESULTS[ba3b...]:
  | Min.    : 4.0 | Min.    :  2 |
  | 1st Qu.:12.0 | 1st Qu.: 26 |
  | Median :15.0 | Median : 36 |
  | Mean    :15.4 | Mean    : 43 |
  | 3rd Qu.:19.0 | 3rd Qu.: 56 |
  | Max.    :25.0 | Max.    :120 |

  We may put the plot on the document by:

  #+begin_src R :file plot.png :results output graphics
    cars %>% plot(main = "Cars")
  #+end_src

  #+RESULTS[5d5c...]:
  file:plot.png
```

Org Mode is ideal for *Problem Solving* through the *Divide and Conquer* strategy

- Create and organize your information.

Org Mode is ideal for *Problem Solving* through the *Divide and Conquer* strategy

- Create and organize your information.
- Focus on the content creation first, worry about document creation (format/style) later.

Org Mode is ideal for *Problem Solving* through the *Divide and Conquer* strategy

- Create and organize your information.
- Focus on the content creation first, worry about document creation (format/style) later.
- *Focus/Divide and Conquer*: Write a skeleton for your project/problem, then use *folding* and *code buffer* to *focus* on a specific problem/topic/code at a time.

Org Mode is ideal for *Problem Solving* through the *Divide and Conquer* strategy

- Create and organize your information.
- Focus on the content creation first, worry about document creation (format/style) later.
- *Focus/Divide and Conquer*: Write a skeleton for your project/problem, then use *folding* and *code buffer* to *focus* on a specific problem/topic/code at a time.
- *Writing Algorithm*: Org-mode facilitates hierarchical organization, functional programming, unit testing, data analysis, etc.

Topic

Installation Guide

Ask IT permission for just one software: Virtual Machine. (e.g. VMware Player or VirtualBox)

Installation Guide

Ask IT permission for just one software: Virtual Machine. (e.g. VMware Player or VirtualBox)

Why VM is convenient?

- A virtual machine (VM) is an emulation of a computer system.
- I use VM to install a Linux system and all the open-source goodies.
- All the software installed in VM do *not* require IT permission.

Intall others yourself: Linux, Emacs, LaTeX, CDLaTeX, R, ESS, ...

I use Linux distribution: Fedora Scientific Spin (many tools pre-installed like `Emacs`, `R`, `LaTeX`)

Intall others yourself: Linux, Emacs, LaTeX, CDLaTeX, R, ESS, ...

I use Linux distribution: Fedora Scientific Spin (many tools pre-installed like `Emacs`, `R`, `LaTeX`)

I recommend these Emacs packages

- Emacs Speaks Statistics (ESS) (speed up `R` programming)

```
$ sudo yum install emacs-ess
```

- Emacs Auto-Complete extension (predictive writing)

```
$ sudo yum install emacs-auto-complete
```

- AUCTeX (typing Math)

```
$ sudo yum install emacs-auctex
```

Topic

Productivity Hints

- In Linux, swap Ctrl and Alt key (as Ctrl key is used heavily) through "Keyboard Layout" or `xmodmap`.
- Leverage Emacs, Emacs Auto-Complete (predictive writing); ESS (helps `R` programming); CDLaTeX (speed up writing `LaTeX`).

Productivity Hints

- In Linux, swap Ctrl and Alt key (as Ctrl key is used heavily) through "Keyboard Layout" or `xmodmap`.
- Leverage Emacs, Emacs Auto-Complete (predictive writing); ESS (helps `R` programming); CDLaTeX (speed up writing `LaTeX`).

Example (Using CDLaTeX for fast insertion of LaTeX templates)

- Math Symbols: type `‘a` inserts: `\alpha`
- Math Modifier: `’m` ⇒ `\mbox{}`
- Math Template: `fr<TAB>` ⇒ `\frac{}{}`
- Environment: `equ<TAB>` ⇒ Equation Environment.
- Type `C-c {` creates a LaTeX environment template.

References

- Emacs Homepage
- Org-mode Homepage
- A youtube tutorial video: Getting Started With Org Mode
- Article: A Multi-Language Computing Environment for Literate Programming and Reproducible Research
- Article: Active Documents with Org-Mode
- My online post: How to Use Emacs Org-Babel Mode to Write Literate Programming Document in R Language
- Making this presentation with Org-mode: Beamer export

Extra: Making a beamer presentation

```
#+TITLE: Presentation Title
#+AUTHOR: Your Name
#+OPTIONS: H:2 num:t toc:t
#+STARTUP: beamer
#+BEAMER_THEME: Frankfurt

* Introduction (Section)
** History (Slide Title)
   A story.
*** Example 1 (Example Block)
*** Example 2
    - Item 1
    - Item 2
    - Item 3
```

Extra: This presentation in Folded View.

```
* HEADER                                          :noexport:ARCHIVE:...
* Introduction
  * Abstract...
  * What is Emacs Org-mode?...
    * A .org text file                                    :B_example:BMCOL:...
    * The same file folded                                :B_example:BMCOL:...
  * Pros and Cons...
    * Cons...
    * Pros...
* Emacs and Org-mode
  * About Emacs...
  * About Emacs Org-mode
    * Organizing information...
    * Coding in /multiple/ languages and /reproducing/ results....
  * An example of writing source code in Emacs Org-mode...
  * Example Code Blocks...
  * Org Mode is ideal for /Problem Solving/ through the /Divide and Conquer/ strategy...
* How to Install?
  * Installation Guide...
    * Why VM is convenient?...
  * Intall others yourself: Linux, Emacs, LaTeX, CDLaTeX, R, ESS, ......
    * I recommend these Emacs packages...
* A Few More Things
  * Productivity Hints...
    * Using CDLaTeX for fast insertion of LaTeX templates          :B_example:...
  * References...
  * Extra: Making a beamer presentation :B_example:...
  * Extra: This presentation in Folded View....
  * Summary...
```

Summary

- Emacs Org-mode is a great research tool for writing notes, articles, programs, data analysis, etc. in *one place* in a *reproducible* manner.

Summary

- Emacs Org-mode is a great research tool for writing notes, articles, programs, data analysis, etc. in *one place* in a *reproducible* manner.
- The raw file is a *TEXT* file: readable, light, portable. It can be exported to beautiful reports, slides, web files, etc.

Summary

- Emacs Org-mode is a great research tool for writing notes, articles, programs, data analysis, etc. in *one place* in a *reproducible* manner.
- The raw file is a *TEXT* file: readable, light, portable. It can be exported to beautiful reports, slides, web files, etc.
- Good for private, non-collaborative use.

Summary

- Emacs Org-mode is a great research tool for writing notes, articles, programs, data analysis, etc. in *one place* in a *reproducible* manner.
- The raw file is a *TEXT* file: readable, light, portable. It can be exported to beautiful reports, slides, web files, etc.
- Good for private, non-collaborative use.
- Facilitates: *Problem Solving*, *Literate Programming*, and *Reproducible Research*.

Summary

- Emacs Org-mode is a great research tool for writing notes, articles, programs, data analysis, etc. in *one place* in a *reproducible* manner.
- The raw file is a *TEXT* file: readable, light, portable. It can be exported to beautiful reports, slides, web files, etc.
- Good for private, non-collaborative use.
- Facilitates: *Problem Solving*, *Literate Programming*, and *Reproducible Research*.
- *Hard Question for You*: Time is the most precious thing. Should I invest considerable time to learn this (relatively) obscure tool to achieve some elegance in organizing information?