Ten things I wish I’d known when I started my PhD

Dean Bodenham

17 November 2016
Overview

A collection of tools and tips to make:

- Developing your statistical software
- Publishing your code
- Writing your thesis and papers

easier, more efficient — and even fun.
Some obvious, some not
Disclaimer

- Just my opinion
- I am not an expert developer
- There are lots of other options
- These tools have helped me
- I use these tools daily
1. Use a Unix-based OS
Two popular distributions:

- Linux (many versions)
- OSX

Everything you see today is free and works with both.
Why use Unix?

• The power of the command line tools:
  ■ grep
  ■ find
  ■ aspell (LaTeX spellcheck)
  ■ wget
  ■ top
  ■ ssh
• These tools can be ‘piped’ together
  ■ e.g. find all files containing word $X$, made after date $Y$
• Don’t know how to use a tool?
  ■ man grep
Useful example

grep --include=*.{Rmd,R} -rnw './' -e "Linux"

./tenthings.Rmd:60:* Linux (many versions)
./tenthings.Rmd:85:grep --include=*.{Rmd,R} -rnw './' -e "Linux"
./tenthings.Rmd:89:grep --include=*.{Rmd,R} -rnw './' -e "Linux"
./tenthings.Rmd:93:## Different flavours of Linux
./tenthings.Rmd:99:## Different flavours of Linux
./tenthings.Rmd:110:- Consider starting with Linux Mint (Ubuntu)
./tenthings.Rmd:169:Linux:
./tenthings.Rmd:203:Linux:
./tenthings.Rmd:387:> - Developed by Linus Torvalds (Linux)
./tenthings.Rmd:436:Linux:
./tenthings.Rmd:494:- `latexdiff` is a Perl script usually pre-installed with Linux/Mac
./tenthings.Rmd:512:- And add path in `~/.bashrc` (Linux) or `~/.bash_profile` (OSX)
Different flavours of Linux

<table>
<thead>
<tr>
<th>Last 12 months</th>
<th>Last 6 months</th>
<th>Last 3 months</th>
<th>Last 1 month</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Mint</td>
<td>2996</td>
<td>1 Mint</td>
<td>2706</td>
</tr>
<tr>
<td>2 Debian</td>
<td>1849</td>
<td>2 Debian</td>
<td>1684</td>
</tr>
<tr>
<td>3 Ubuntu</td>
<td>1587</td>
<td>3 Ubuntu</td>
<td>1378</td>
</tr>
<tr>
<td>4 openSUSE</td>
<td>1281</td>
<td>4 openSUSE</td>
<td>1361</td>
</tr>
<tr>
<td>5 Manjaro</td>
<td>1082</td>
<td>5 elementary</td>
<td>1211</td>
</tr>
<tr>
<td>6 Fedora</td>
<td>1058</td>
<td>6 Manjaro</td>
<td>1004</td>
</tr>
<tr>
<td>7 elementary</td>
<td>887</td>
<td>7 Fedora</td>
<td>966</td>
</tr>
<tr>
<td>8 Zorin</td>
<td>878</td>
<td>8 Zorin</td>
<td>956</td>
</tr>
<tr>
<td>9 CentOS</td>
<td>835</td>
<td>9 CentOS</td>
<td>761</td>
</tr>
<tr>
<td>10 Arch</td>
<td>792</td>
<td>10 deepin</td>
<td>741</td>
</tr>
<tr>
<td>11 deepin</td>
<td>733</td>
<td>11 Arch</td>
<td>707</td>
</tr>
<tr>
<td>12 Mageia</td>
<td>728</td>
<td>12 PCLinuxOS</td>
<td>645</td>
</tr>
<tr>
<td>13 PCLinuxOS</td>
<td>673</td>
<td>13 Antergos</td>
<td>584</td>
</tr>
<tr>
<td>14 Android-x86</td>
<td>652</td>
<td>14 Ubuntu MATE</td>
<td>564</td>
</tr>
<tr>
<td>15 Ubuntu MATE</td>
<td>650</td>
<td>15 Mageia</td>
<td>575</td>
</tr>
<tr>
<td>16 Slackware</td>
<td>554</td>
<td>16 Lite</td>
<td>535</td>
</tr>
<tr>
<td>17 LXLE</td>
<td>548</td>
<td>17 Slackware</td>
<td>512</td>
</tr>
<tr>
<td>18 Lite</td>
<td>543</td>
<td>18 LXLE</td>
<td>504</td>
</tr>
<tr>
<td>19 Antergos</td>
<td>542</td>
<td>19 Android-x86</td>
<td>495</td>
</tr>
<tr>
<td>20 Puppy</td>
<td>538</td>
<td>20 Solus</td>
<td>479</td>
</tr>
</tbody>
</table>

source: distrowatch.com
Different flavours of Linux

- Mint
- Debian
- openSUSE
- Ubuntu
Starting out?

- Consider starting with Linux Mint (Ubuntu)
- Different frontends:
  - Cinnamon
  - KDE
  - Xfce
Cinnamon, source: linuxmint.com
Try it out

- Live USB - let’s you test it out before installing
  - check wifi
  - check sound
  - check video
  - check filesystem
- Virtual machine: VirtualBox
2. Pick a good editor
Classic choices

- Vim
- Emacs
Vim and Emacs

- Edit text at ‘lightspeed’
- Syntax highlighting
- Autoindentation
- Easily move between languages
Newer options

- Eclipse (diff and auto-version control)
- Rstudio
- Atom
- TextMate
- Nano
- Sublime
Installing Vim

Linux: sudo apt-get install vim-gnome

OSX: brew install vim
3. Use a terminal multiplexer
Terminal multiplexers

- screen (classic)
- tmux (new and growing in popularity)
tmux benefits

• Organise multiple terminals in one screen
• Easily split windows into panes
• Continue server sessions
• Pane shading
• Check out: tmux-resurrect

*also screen
Easy to install

Linux: sudo apt-get install tmux

OSX: brew install tmux

Check it is installed correctly: tmux -V

Check version: tmux -V

(Nov 2016: will install version $\geq 2.1$)

(tmux demo)
4. Use the server
Easily run large-scale simulations

- Your local machine is for development
- Queue a job on the server, and let it run overnight
- This is where knowing:
  - terminal commands
  - Vim/Emacs
  - screen/tmux

will pay off **ALOT**.
5. Create R packages, not scripts
R packages

• Binds your scripts together
• Forces you to write documentation
  ▪ But the payoff is **worth it**
  ▪ (ever come back to code after a couple of months...?)
• Easily integrate **tests** (more later)
• Getting started: devtools and roxygen2 packages
• Check out Hadley Wickham’s online book
• Example later
6. R or C++? Use both: Rcpp
Bring the the speed of C++ to R

- R is often criticised for being slow
  - bad at for loops
- C++ is fast, but lacks many of R’s functions
- Rcpp is an R packages that links R and C++
A short experiment

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Speed-up using Rcpp</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fibonacci</td>
<td>~300 times</td>
</tr>
<tr>
<td>Pisum</td>
<td>~100 times</td>
</tr>
<tr>
<td>sum (naive)</td>
<td>~3500 times</td>
</tr>
<tr>
<td>sum (internal)</td>
<td>~0.12 times</td>
</tr>
</tbody>
</table>

- Ideas for benchmarks from julialang.org
- Using microbenchmark package
- To reproduce these results, see this GitHub repo
- Check out the Rcpp gallery
7. Write unit tests for your code
Test your code

- **Unit tests** are pieces of code that test your functions
- Make sure your code *works* as it should
- Avoid breaking your (previously working) code
- Forces you to break your code down into logical chunks
- Can be useful for debugging
- Longer in the short term, but **shorter in the long term**
- Check out the testthat R package
  - Makes testing quick and easy
Very quick

devtools::test()

(demo)
Start testing

- Hadley Wickham’s tutorial
- R journal article

Note: Similar framework in Python
8. ggplot2
Beautiful and powerful plotting with ggplot2

- Number one comment about R: great plots
- ggplot2 another Hadley Wickham package
  - along with devtools, testthat and many others
- http://docs.ggplot2.org/current/
source: http://www.r-graph-gallery.com/
source: http://www.r-graph-gallery.com/
source: http://docs.ggplot2.org/current/facet_grid.html
Check out:

http://www.r-graph-gallery.com/

to see what is possible (even with base R)
9. Git
Git: version control

- Ever had code (R, C++, LaTeX...) ‘stop working’?
- Git allows you to easily create checkpoints and versions
- Developed by Linus Torvalds (Linux)
- Git is FAST
- Allows you to work offline
- Other version control systems: Subversion (SVN), Mercurial
Git: basic commands

- Quick and easy to get started:
  - `git init`
  - `git add .`
  - `git commit`
- Recovering versions is also easy
  - Three different methods, depending on situation
Git for backups

• Ever worried about backing up scripts?
• Get a free GitHub or Bitbucket account
  ▪ upload Git repo’s with: git push -u origin master
• Lost your code or want to sync on a different computer?
  ▪ git clone
  ▪ git pull
Git demo
(demo)
Using Git

- Use Git for version control with:
  - code scripts
  - LaTeX scripts
- Do not use Git with:
  - data files
- Can use it to save images, but be careful
- Create a .gitignore file
Installing Git

Linux: sudo apt-get install git
OSX: brew install git
Check it works: git --version
Learning Git

- There is a learning curve
  - but only a few commands really necessary
- Introduction by Atlassian
- Learn Git branching
Write meaningful commit messages

source: https://xkcd.com/1296/
10. git-latexdiff
Check differences between tex files

- Originally, latexdiff works with two files as input
- git-latexdiff works with versions of the same file with Git

(demo)
1 Showcasing latexdiff

latexdiff is a great tool that is very useful and works on the command line. Try it out!

Euler’s identity (incorrect):

\[ e^{2i\pi} + 3 = 0 \]  \hfill (1)
1 Showcasing latexdiff

latexdiff is an awesome tool that is very useful. Try it out - it also works with Git!

Euler’s identity:

\[ e^{i\pi} + 1 = 0 \] (1)
1 Showcasing latexdiff

latexdiff is a great tool that is very useful and works on the command line. Try it out - it also works with Git!

Euler's identity (incorrect):

\[ e^{2i\pi} + 31 = 0 \]  

(1) Differences with latexdiff
Installing \texttt{latexdiff}

- \texttt{latexdiff} is a Perl script usually pre-installed with Linux/Mac
- To check: \texttt{latexdiff -V}
- Troubleshooting: \texttt{see this page}
Installing git-latexdiff

```bash
mkdir -p ~/build/
cd ~/build/
git clone https://gitlab.com/git-latexdiff/git-latexdiff.git
cd ~/build/git-latexdiff
sudo make install  # ignore the error
```

- And add path in `~/.bashrc` (Linux) or `~/.bash_profile` (OSX)
- See the [git-latexdiff website](https://gitlab.com/git-latexdiff/)
- Full installation instructions: [here](https://gitlab.com/git-latexdiff/)
- Full credit to:
  - [Frederik Tilmann](https://gitlab.com/git-latexdiff/), creator of latexdiff
  - [Matthieu Moy](https://gitlab.com/git-latexdiff/), creator of git-latexdiff
Easy Command:

- Compare current version and two versions back:
  
git latexdiff HEAD~2 --main main.tex

- Detailed post: here
A couple of other thoughts
11. Reproducible results and figures
Reproducible results

• Save every experiment with key results
  ■ May need to run again/tweak at a later stage
  ■ Combine with Git
• All results should appear by running **one command**
• Write a README for each experiment
• Scientific integrity
Reproducible figures

- Less obvious, but just as important
- Use a two-step approach:
  - One script to generate the data
  - Another script to generate the figure from the data
12. Work through a seminal book in your area
Become an expert

- While the courses are good, they are really about **breadth**
- Ask your supervisor to suggest a book for you to work through
  - Give yourself a year (per book)
- Nothing comes to mind? How about Feller?
Contents
Outline

1. Use a Unix-based OS
2. Pick a good editor
3. Use a terminal multiplexer
4. Use the server
5. Create R packages, not only scripts
6. R or C++? Use both: Rcpp
7. Write unit tests for your code
8. For nice plots, try `ggplot2`
9. Use Git
10. `git-latexdiff`
11. Reproducible work
12. Pick a book
Thank you!
Slides prepared with:

- rmarkdown
- reveal.js
- pdf2svg
Questions?

Slides available at: goo.gl/Myz9Hu