Introduction

Thomas de Graaff & Daniel Arribas-Bel

September 5, 2014

Introduction

Why this workshop?

- In the *social sciences* few attention to what tools to use (and why they make sense)
- Increasing need/interest for/in openness & transparancy
  - from journals, universities and governments
  - increase in cooperation (over wider distances)
  - access to your own files
  - make yourself more visible
- Better workflows actually increases reproducability and thus
  - productivity
  - contributions to the field at large
  - visibility of errors

What we want (and don't want) with this workshop

- We are mostly interested in the principles behind a good open (scientific) workflow, aware of the facts that
  - there is no final, optimal, set of workflow tools
  - investment is very very costly
- However, being a practical workshop we do
  - work with a specific set of tools (R, markdown, pandoc, git)
  - which is probably most straightforward to use
  - enables us *in this workshop* to make a paper reproducable and open

Workflow

Open?

- Workflow: *Progression of steps (tasks, events, interactions) that comprise a work process, involve two or more persons, and create or add value to the organization's activities* (BusinessDictionary)
- Open workflow: One that enhances *transparency*, *collaboration* and *reproducibility*

Empirical cycle

- Read other papers
- Think of a brilliant idea
- Do:
  1. Collect data
  2. Transform data
  3. Analyze data
  4. Write up results
  5. Present results
  6. Go back to 1. until satisfied
- Send paper to journal and go back once again to i. until referees satisfied
- And... documenting throughout the entire process!!!

Theoretical cycle

- Read other papers
- Think of a brilliant idea
- Do:
  1. State assumptions
  2. Model (simulate)
  3. Analyze model outcome
  4. Write up results
  5. Present results
  6. Go back to i. until satisfied
- Send paper to journal and go back once again to i. until referees satisfied
- And... documenting throughout the entire process!!!

Why bother about a workflow or tools?

- Good scientific practice: *document how you have achieved your results*
- Reproducibility
- Transparency
- Modularity
- Portability (across systems and users)
- Efficiency
- Self-sanity

Why should it be open?

- Open Science
- Reproducibility
- Transparency
- Modularity
- Portability (across systems and users)
- Efficiency

When should I adopt an open workflow?

- The sooner the better
- But think twice about which one (switching is costly)
- Start one step at a time

A journey of a thousand miles begins with a single step

Lao-tzu

Tools for . . .

open workflows

Text editting

- Plain text is **simple**, light, cross-platform, flexible...
- Many **academic** tools are based on plain text (typesetting systems, scripting languages, bibliography managers)
- **Good investment** to learn a rich text editor ("learn once, use for everything"):
  - Efficient typing (command vs. insert modes)
  - Syntax highlighting and indenting
  - Shortcuts, macros and templates
  - Consistent look, feel and behaviour
- Examples: Vim, Emacs, other (TextMate, Sublime text, etc.,...)

Text editing

Vim demo. . .

- Command vs Insert mode
- Syntax highlighting
- LaTeX shortcuts
- Python indenting

Beautiful (and efficient) typesetting

- Documentation of progress, presentation of results (paper or slides) and final products depend on this
- **plain text** + **markup languages** = very powerful
  - Detach content inputting from layout and styling
  - One source, multiple outputs (paper, slides, website. . .)
- Examples: LaTeX, Markdown, Org

Beautiful (and efficient) typesetting

LaTeX and **Beamer**...

- General template
- Sectioning
- Equations (inline, outside)
- Table

Markdown...

Managing lists of papers

- *One reference list to rule them all*
- Create the reference and never worry about proper inserting
  - Bibtex
  - Reference manager
  - Online services (e.g. Mendeley)
- Bibtex demo...

Analyzing data

- **Platforms** for statistical analysis & **scripting languages**
- Examples: Python, R, STATA
- The power of code vs. point-and-click
  - **Flexibility** (Python)
  - Typically **wider range** of methods (STATA)
  - **Extensible** and updated more rapidly (R)
  - **Reproducible** and transparent (remember *exactly* what you did)

Analyzing data

IPython notebook demo...

- Load up data
- Create descriptives
- Scatter plot
- Run a model and simple print
- Print LaTeX output

Saving the workflow

Backup: "You don't need it until you really need it"

- Security copy of all your (valuable) documents
- External drive vs. Cloud solution
- Software to make the process painless or automated
- Many options: TimeMachine, Dropbox, Amazon Glacier...

Saving the workflow

Versioning control: *"How did I get to that table of results?"*

- Save snapshots of a project in an intelligent way
- Allows to trace the *history* of a project/document (very neat example)
- Very well developed for code development
- Examples: DropBox, git, svn...

Putting it all together