Why have a toolkit for open science workflows?

"Scratch" image from PhD Comics, 3/12/2014
Demand for more open science is increasing

Why scientists must share their research code

‘Reproducibility editor’ Victoria Stodden explains the growing movement to make code and data available to others.

Monya Baker

13 September 2016

From Nature News

Open Science Policy Platform

Group that advises the Commission on how to develop open science policy. Meeting reports, member details and background

From https://ec.europa.eu/research/openscience
Current available R packages/workflows

- rrtools
- ProjectTemplate
- makeProject
- devtools, usethis directly
...Just another competing package?

Image from: https://xkcd.com/927/
... Probably. But, there are still things missing or not addressed

- Fairly complicated
- Assume complex analyses
- Don't automate enough
- Not targeted to "casual coders"
  - e.g. most health researchers
- Are not opinionated enough
  - "Here are options and a general workflow, you decide the rest"
What is needed and why?

- Lots of support and guidance
  - Many (health) researchers *don't know* about open science
  - ...Or *how to do it*
- Opinionated on what services and tools to use
  - Many tools and "moving parts"... it's *overwhelming*
- Automation of most tasks, rather than through documentation
  - *Default* for open science should be *easy*
Enter the prodigenr package
Main function...

Generate project directory, setup_project():

```
 ├── R
 │   ├── README.md
 │   └── fetch_data.R
 │       └── setup.R
 ├── data
 │   └── README.md
 ├── doc
 │   └── README.md
 ├── .Rbuildignore
 ├── .gitignore
 ├── DESCRIPTION
 └── ProjectName.Rproj
     └── README.md
```
similar structure to R packages

Advantages:

- Established infrastructure
- Well developed through devtools, usethis
- Handles range in project difficulty
  - (simple to complex analyses)
Some of the other functions:

Add scientific product, e.g. poster, create_poster():

```
├── R
│   ├── README.md
│   │   ├── fetch_data.R
│   │   └── setup.R
│   └── data
│       └── README.md
├── doc
│   ├── README.md
│   └── poster.Rmd
├── .Rbuildignore
├── .gitignore
├── DESCRIPTION
└── ProjectName.Rproj
    └── README.md
```
Other functions:

- Add author(s) to project metadata (=DESCRIPTION file):
  - `add_first_author()`
  - `add_coauthor()`
- Include some common "tools":
  - `include_mit_license()` for code
  - `include_strobe()` for health research reporting
Current stage of development

- Participating in Mozilla Open Project Leader Training
- Participated in Mozilla Global Sprint
  - Gained some feedback
- Soon submit next version to CRAN
Next steps and future plans

- Focus prodigenr on project generation
- Tag/version bump after e.g. manuscript submission
- Reproducibility tools (e.g. travis, docker)
- Other functionality to new package, rostools
- Teaching material and more docs
How does this help?

- Open science is evolving
  - This automates and bundles together common tools
  - (of course, strongly opinionated on which tools)
- Natural extension to the devtools/usethis framework
- Researchers need easy tools
  - *This one of the first steps* toward that
Reason for this talk…

- Looking for feedback, thoughts, comments
- But mostly… seeking potential contributors/collaborators!

Contact info
- GitHub: @lwjohnst86
- Email: lwjohnst@ph.au.dk
- Slides: https://github.com/lwjohnst86/erum2018

Licensed under a CC-BY 4.0 International License.