Statistical Programming and Open Science Methods

Introduction, Development Environment and Project Organization

Joachim Gassen
Humboldt-Universität zu Berlin

September 02, 2022

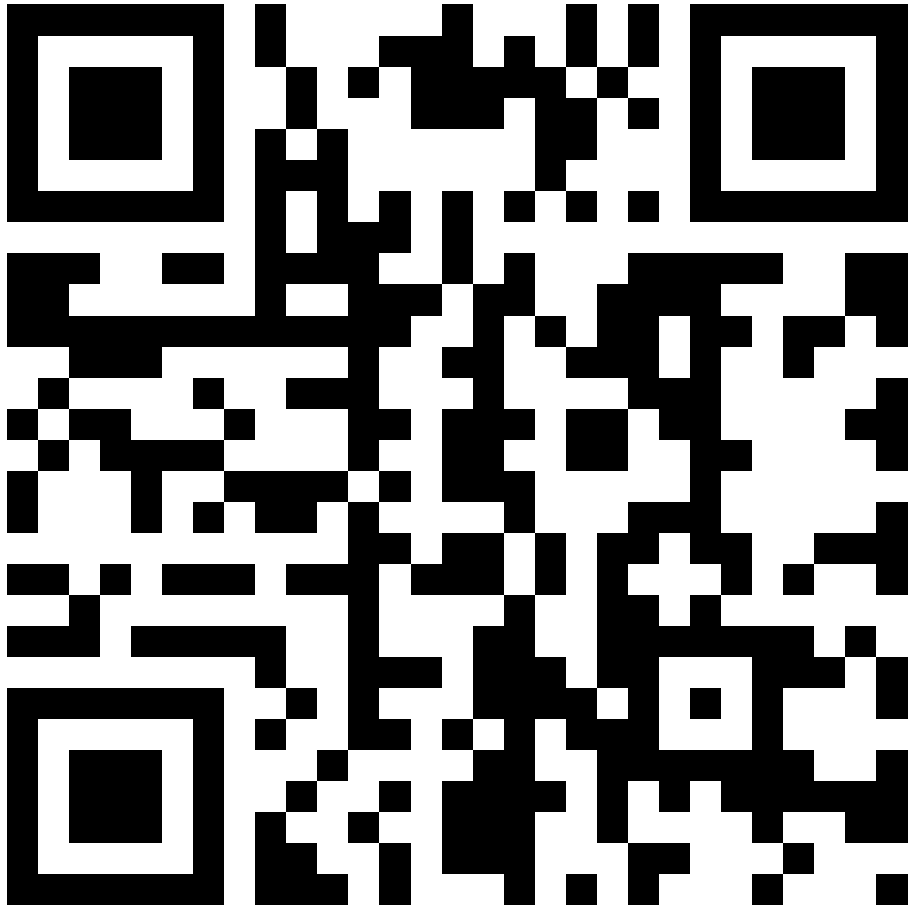
SFB/Transregio 266
ACCOUNTING FOR TRANSPARENCY

Introduction

Time table October 10

When?	What?
09:00	Welcome and Introduction
09:30	The development environment and project organization
10:30	Coffee
11:00	Using Git and Github
12:30	Lunch
14:00	Statistical programming languages: An overview
15:30	Coffee
16:00	Functional versus object-oriented programming
19:30	Pizza at Due Forni, Schönhauser Allee 12

Instead of an introduction round . . .



https://trr266.wiwi.hu-berlin.de/shiny/sposm_survey/

Let's have a look at your reponse

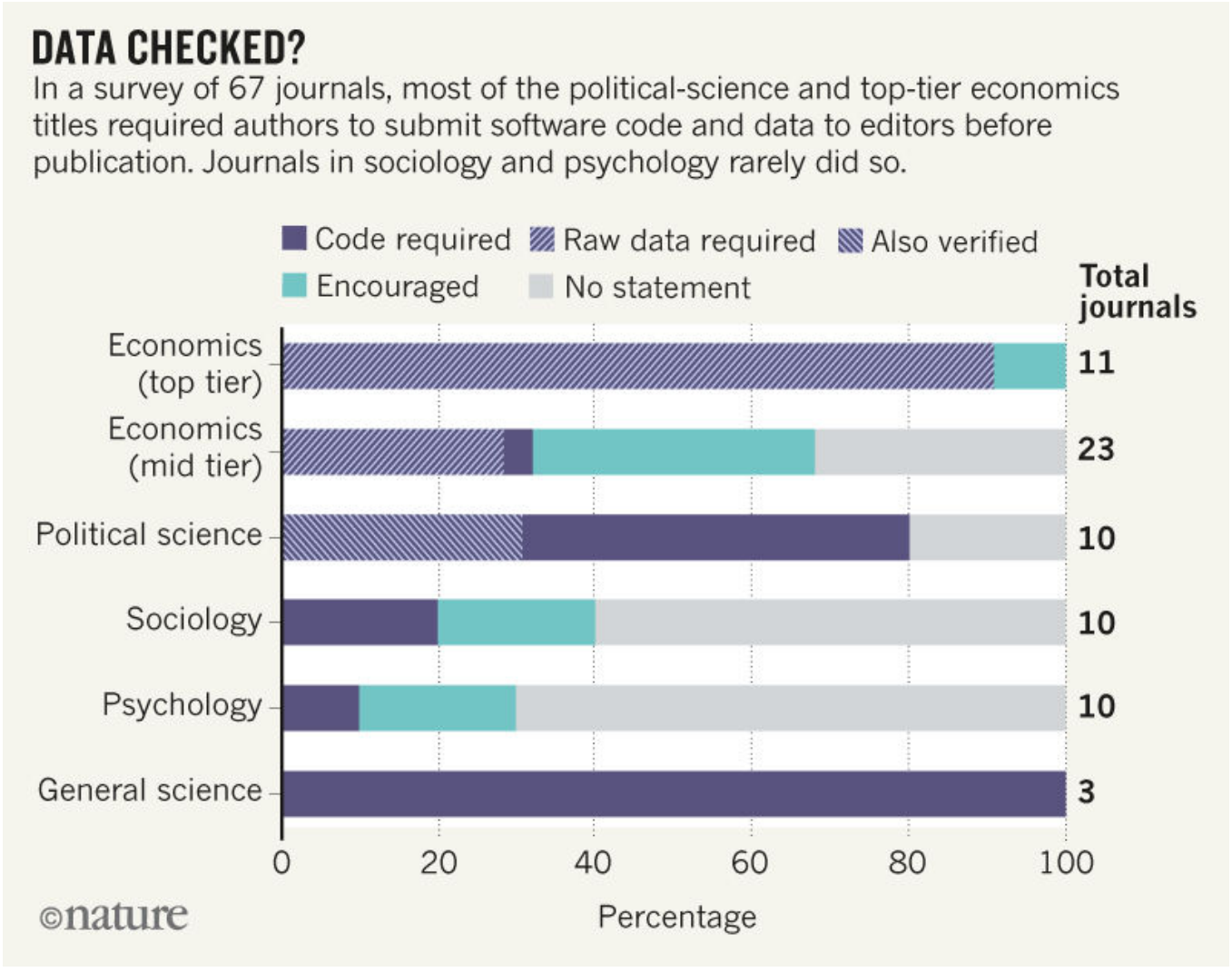
https://trr266.wiwi.hu-berlin.de/shiny/sposm_survey_eval/

Open Science

Open Science is the practice of science in such a way that others can collaborate and contribute, where research data, lab notes and other research processes are freely available, under terms that enable reuse, redistribution and reproduction of the research and its underlying data and methods.

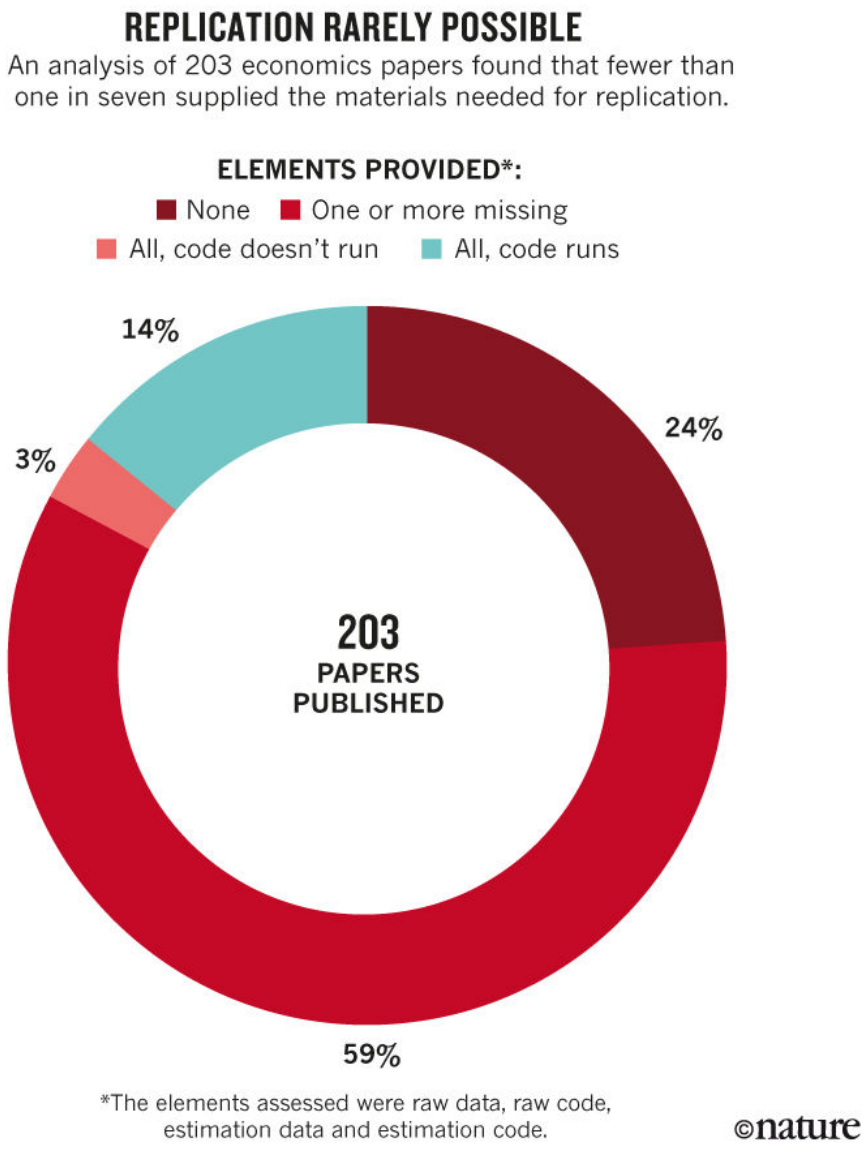
— FOSTER, https://www.fosteropenscience.eu/

Data and code repositories are on the rise . . .



Gertler, Galiani and Romero (Nature, 2018)

. . . but yet fail to guarantee reproducible results



Gertler, Galiani and Romero (Nature, 2018)

Why is that? One reason

Those of you who use Stata for research projects will be familiar with code that starts like this

```
clear all
set more off
cd  "/Users/joachim/Dropbox/icas/analysis/"
use "stata_dsets/verbal_and_survey_data.dta", clear
drop if !verbal_exp_sample
drop if !re & !fr
drop if subject_code > 11
* ...
```

What is wrong with this code?

Basic things to consider when doing open science

- Can one build an environment that allows reproducing my analysis with reasonable resources?
- If I am using commercially licensed Software: Is this software essential for my analysis? (think about Stata)
- Is the data that I use publicly obtainable and have I documented where to get it from?
- If the data is not publicly available: Is it essential for my analysis?
- Is the code that I use relying on some idiosyncrasies of my development environment (paths, support software)?
- Is my code readable?
- Have I tried to replicate my analysis in different environments?

Setting up your development environment

Things that you need

- An operating system (ideally not commercially licensed)
- A shell
- Interpreter and/or compiler for your favorite programming languages
- An editor

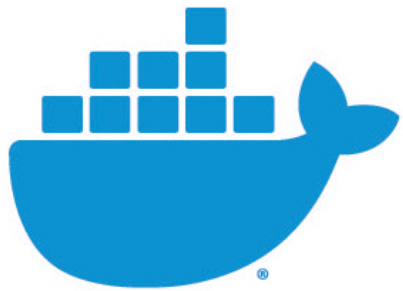
Things that are (very) nice to have

- An integrated development environment (IDE) containing an editor, debugger and additional build tools
- A latex environment for producing nicely formatted output
- A version control system

Let's start from scratch

- Install a linux-type OS
- Install R/RStudio and Python
- How does one do that if you like your old computing environment?
- Buy a new computer???
- Not really.

An IDE Container using Docker



- Take a look at this file: https://github.com/joachim-gassen/sposm/blob/master/docker/Dockerfile

Project Organization

Disclaimer

Some of the following is borrowed from:

Gentzkow, Matthew and Code and Jesse M. Shapiro (2014):
Data for the Social Sciences: A Practitioner's Guide,
http://web.stanford.edu/~gentzkow/research/CodeAndData.pdf

(a very good read)

Run every project like a collaborative project

- You are always collaborating (if only with your future self)
- Make sure that the structure of your project becomes visible
- If projects become complex a `README_FIRST` document can help
- Identify the important outputs of your project (a paper?, presentations?) and specify the dependencies to generate them

Naming conventions

> There are 2 hard problems in computer science: cache invalidation, naming things, and off-by-1 errors. — Leon Bambrick

- Naming is about consistency. Find your style and stick to it
- Two main concepts: `CamelCase` and `snake_case` ...
- ... and a special place in hell for people who use spaces in file or variable names (as it tends to break the tool chain)
- Strive to make names revealing, pronounceable and searchable
- Name functions and code files as verbs or verb phrases
- Taste based: datasets singular versus plural (most people say singular)

Automation

"Wie het gemak niet zoekt, is lui" ("Those who are not looking for convenience are lazy") — Dutch proverb

- Automate everything that can be automated
- Write and maintain a script that produces all output
- Use a directory structure that separates input from output
- Do not store output permanently

Dependencies

- Dependencies describe how outputs depend on inputs
- For example:
  - Raw data depends on running scraping routine `scrape.py`
  - Clean data depends on running `cleanup_data.py` with the scraped data
  - The sample depends on running `merge_data.R` on clean data
  - `tables.tex` depends on running `analysis.R` on sample
  - `figures.eps` depends on running `create_visuals.R` on sample
  - The paper depends on running `texi2pdf` on `paper_text.tex`, `tables.tex` and `figures.eps`
- The easiest way to maintain and document dependencies is to use GNU Make

Makefile example (TABS are needed for indentation)

```
data.csv: scrape.py
	python scrape.py
clean_data.csv: clean_up_data.py
	python clean_up_data.py
sample.RDS: clean_data.csv
	Rscript merge_data.R
tables.tex: sample.RDS
	Rscript analysis.R
figures.eps: sample.RDS
	Rscript create_visuals.R
paper.pdf: paper_text.tex tables.tex figures.eps
	texi2pdf paper_text.tex

paper: paper.pdf
all: paper
clean:
	rm *.csv *.RDS *.eps tables.tex *.pdf
```

The sad news...

Makefiles and project management do not do all the work for us:

```
$ make paper
make: *** No rule to make target ‘paper_text.tex’,
needed by ‘paper.pdf’.  Stop.
```