Reproducible and Collaborative Statistical Data Science

Philip B. Stark

Department of Statistics
University of California, Berkeley

Transparency Practices for Empirical Social Science Research
2014 Summer Institute
University of California
Berkeley, CA
5 June 2014

What's with the title?

Why "reproducible" & "collaborative" in the same sentence?

- Same habits, attitudes, principles, and tools facilitate both.
- Reproducibility $\approx$ collaboration w/ people you don't know.
- That includes yourself, 1 week from now.
- Built-in $>$ bolt-on

What does "reproducible" mean?

Always some *ceteris* assumed *paribus*.
But what?

- Heavily overloaded term
- Experiment repeatable in same lab with same procedure?
- Repeatable elsewhere, by others?
- Procedures & other conditions specified adequately?
- Can re-generate figures and tables / re-run code?
- Can see/understand what was done?
- Code & data public?
- Build environment specified adequately?
- Contrast with V&V, replicability, generalizability, etc.

Data science

- Goal: turn data into evidence.
- Science is losing the scientific method: big data and big are computation making things worse.
- Don't trade "trust me" for "show me."
- Were the days of mainframes the good old days?

Focus on *statistical/computational* issues

- What's the underlying experiment?
- What are the raw data? How were they collected/selected?
- How were the raw data processed to get the "data"?
- What analysis was reported to have been done on the "data"?
- Was that analysis the right analysis to do?
- Was that analysis done correctly? Was the implementation numerically stable/sound?
- Were the results reported correctly?
- How many analyses were done before arriving at that the one reported? What were they? What were the results? How was multiplicity treated?
- Were there ad hoc aspects to the analysis? What if different choices were made?
- Can someone else re-use/re-purpose the tools?

Why work reproducibly?

Cornford, 1908. *Microcosmographia Academica*

There is only one argument for doing something; the rest are arguments for doing nothing.

The argument for doing something is that it is the right thing to do.

Another argument

check, reuse, extend, share, collaborate w/ others & your future self. "*Help science stand on your shoulders*: Science should be reproducible. Reproducible research is easy to build upon, is more citeable and more influential. As computational analysis, methods and digital data archival have become the standard in scientific research, it is important that this information is archived, curated, and documented in a way that most Scientific journals do not currently support." `researchcompendia.org`

More Benefits

- Provides (a way to generate) evidence of correctness
- Enables re-use, modification, extension, ...
- Facilitates collaboration
- Exposes methods, which might be interesting and instructive
- Gut feeling that transparency and openness are good
- Claim: Reproducibility is a tool, not a primary goal.
  Might accomplish some of those goals without it, but it's a Very Powerful Tool.

Narrow replicability and reproducibility

- If something only works under *exactly* the same circumstances, shrug.
- If you can push a button and regenerate the figures and tables but you can't confirm what the code does, shrug.

Tools for reproducibility and collaboration

Learn from OSS community

- open source software, open data, open publications
- version control systems, e.g., git
  (not Dropbox, Google Docs)
- data archive/control systems, e.g., git-annex
- documentation, documentation, documentation
- testing, testing, testing
- testing tools, e.g., nose
- issue trackers
- avoid spreadsheets!
  (examples: Rogoff & Reinhart, JP Morgan, Olympics)

Incentives, disincentives, moral hazard

- it's the right thing to do: check, reuse, extend, share, collaborate w/ others & your future self
- stronger evidence of correctness
- greater impact
- greater scientific throughput overall
- no *direct* academic credit
- requires changing one's habits, tools, etc.
- fear of scoops, tipping one's hand, exposure of flaws
- IP issues, data moratoria in "big science," etc.
- *may be* slower to publish a single project
- systemic friction: lack of tools & training
- lack of infrastructure to host runnable code, big data
- lack of support from journals, length limits, etc.
- lack of standards?

Reframing fears

- If I say “trust me” and I’m wrong, I’m untrustworthy. And I’m hindering progress.
- If I say “here’s my work” and I’m wrong, I’m human and honest. And I’m contributing to progress.

When and how?

- Built-in or bolt-on?
- Tools
- Training
- Developing good habits
- Changing academic criteria for promotions:
  How nice that you advertised your work in *Science*, *Nature*, *NEJM*, etc.!
  Where's your actual work? Where's the evidence that it's right? That it's useful to others?

Obfuscation, trust, and reproducibility

- CERN Large Hadron Collider (LHC) ATLAS and CMS: Both use COLLIE for confidence limits, code proprietary to the team.
  Nobody outside the team will ever have access to the raw data.
- (Elsewhere) surprising rationalizations for not working reproducibly, e.g., "it means more if someone reproduces my work from scratch than if they can follow what I did."

Personal failure stories

Multitaper spectrum estimation for time series with gaps: lost *C* source for MEX files; old MEX files not compatible with some systems.

> *Unfortunately I was not able to find my code for multitapering. I am pretty sure I saved them after I finished my thesis, along with all the documentation, but it seems like I lost them through one of the many computer moves and backups since. I located my floppy (!) disks with my thesis text and figures but not the actual code.*

Poisson tests of declustered catalogs: current version of code does not run.

Mending my ways: Auditing Danish Elections

- Joint work with Carsten Schuermann, ITU DK
- Risk-limiting audit of Danish portion of EU Parliamentary election and Danish national referendum on patent court
- Use nonparametric sequential test of hypothesis that outcomes are wrong
- Risk limit 0.1% (99.9% confidence that outcome is right)
- $\approx$4.6 million ballots, 98 jurisdictions, 1396 polling places
- SRS of 1903 ballots from EU race, 60 from referendum

1. first risk-limiting audit conducted at 99.9% confidence (the highest previously was 90%)
2. first risk-limiting audit of a parliamentary election
3. first risk-limiting audit of a national contest
4. first risk-limiting audit that crossed jurisdictional boundaries
5. first risk-limiting audit outside the U.S.A.
6. first risk-limiting audit of a hand-counted election
7. first risk-limiting audit to use sort-and-stack as a commitment to ballot interpretation
8. smallest margin ever audited with a risk-limiting audit (0.34%)
9. largest contests ever audited with a risk-limiting audit (2.3 million ballots in each contest, 4.6 million total)
10. largest sample ever audited in a ballot-level risk-limiting audit ($>$1900 individual ballots)

Towards reproducible social science

- Verified underlying theorems and checked formulae; currently in peer review
- Coded all algorithms twice, once in ML and once in Python
- ML provably correct; written (partly) using pair programming
- Tested both implementations independently
- Compared output to validate
- Some crucial pieces also in HTML5/Javascript, on the web
- Entire analysis is in an IPython notebook *and* an ML program
- Data are official election results; some web scraping
- All code and data in a git repo
- Photo-documentation of part of the process, including generating seed with dice

2014 Danish EU Parliamentary Election

Party	Votes	% valid votes	seats
A. Socialdemokratiet	435,245	19.1%	3
B. Radikale Venstre	148,949	6.5%	1
C. Det Konservative Folkeparti	208,262	9.1%	1
F. SF - Socialistisk Folkeparti	249,305	11.0%	1
I. Liberal Alliance	65,480	2.9%	0
N. Folkebevaegelsen mod EU	183,724	8.1%	1
O. Dansk Folkeparti	605,889	26.6%	4
V. Venstre, Danmarks Liberale Parti	379,840	16.7%	2
total valid ballots	2,276,694		
blank ballots	47,594		
other invalid ballots	7,929		
total invalid ballots	55,523		
Total ballots	2,332,217		
Eligible voters	4,141,329		
Turnout	56.32 %		

`http://www.dst.dk/valg/Valg1475795/valgopg/valgopgHL.htm` (last accessed 29 May 2014)

21-4 Danish Unified Patent Court membership referendum

yes	1,386,881	62.5%
no	833,023	37.5%
valid votes	2,219,904	
blank ballots	77,722	
other invalid votes	6,157	
total invalid votes	83,879	
total ballots	2,303,783	
eligible voters	4,124,696	
turnout	55.85%	

`http://www.dst.dk/valg/Valg1475796/valgopg/valgopgHL.htm`
(last accessed 29 May 2014)

Initial sample size

Contest information

Ballots cast in all contests: 2303783 Smallest margin (votes): 553,858. Diluted margin: 24.04%.

Contest 1. Contest name: 2014 Denmark EU Patent Referendum

Winners: 1

Reported votes:

Candidate 1 Name: yes Votes: 1386881

Candidate 2 Name: no Votes: 833023

Add candidate to contest 1 | Remove last candidate from contest 1

Add contest | Remove last contest

Audit parameters

Risk limit: 0.1%

Expected rates of differences (as decimal numbers):

Overstatements. 1-vote: 0 2-vote: 0

Understatements. 1-vote: 0 2-vote: 0

Starting size

☑ Round up 1-vote differences. ☐ Round up 2-vote differences. Calculate size 60.

```
(* elect M = M'¬
¬
   Invariant:¬
   Let i be the row with the highest non elected coefficient.¬
   M' = M where line i is altered, ¬
   where the new "blue" eleement is the first hopeful element in xi.¬
*)¬
¬
fun elect M = ¬
    let ¬
      val xs = enum 0 (map (fn (_, (x, _) :: _) => x) M)¬
      val (j, y) = findmax xs¬
    in¬
      elect' j M ¬
    end¬
and elect' 0 ((_, (r, d) :: R) :: Ps) = ((SOME ((r, d)), R) :: Ps)¬
  | elect' n (P :: Ps) = P :: (elect' (n-1) Ps)¬
¬
(* dhondt M n = M' ¬
 ¬
   Invariant :  M' is teh result of electing n candidates in M¬
*)¬
fun dhondt M 0 = M¬
  | dhondt M n = dhondt (elect M) (n-1) ¬
¬
(* Computing inequalities a la Philips talk.  *)¬
¬
```

IP[y]: Notebook danmark14EU Last Checkpoint: May 28 07:06 (unsaved changes)

File Edit View Insert Cell Kernel Help

In [2]:
```
%matplotlib inline
import math
import numpy as np
import scipy
from scipy.stats import binom
import pandas as pd
import matplotlib.pyplot as plt
#
def dHondt(partyTotals, seats, divisors):
    '''
    allocate <seats> seats to parties according to <partyTotals> votes,
    using D'Hondt proportional allocation with <weights> divisors
    '''
    pseudoCandidates = np.array([partyTotals,]*seats, ).T/divisors.astype(float)
    sortedPC = np.sort(np.ravel(pseudoCandidates))
    lastSeated = sortedPC[-seats]
    theSeated = np.where(pseudoCandidates >= lastSeated)
    partySeats = np.bincount(theSeated[0], minlength=len(partyTotals)) # number of seats for each party
    inx = np.nonzero(partySeats)[0]  # only those with at least one seat
    seated = zip(inx, partyTotals[inx], divisors[partySeats[inx]-1])
                                        # parties with at least one seat,
                                        # number of votes that party got,
                                        # and divisor for last seated in the party,
    theNotSeated = np.where(pseudoCandidates < lastSeated)
    partyNotSeats = np.bincount(theNotSeated[0], minlength=len(partyTotals)) # number of non-seats for each party
    inx = np.nonzero(partyNotSeats)[0]
    notSeated = zip(inx, partyTotals[inx], divisors[partySeats[inx]])
                                        # parties with at least one unseated,
                                        # number of votes that party got,
                                        # and divisor for the first non-seated in the party
    if (lastSeated == sortedPC[-(seats+1)]):
        raise InputError("Tied contest for the last seat!")
    else:
        return partySeats, seated, notSeated, lastSeated, pseudoCandidates
```

Software environments for reproducible/collaborative research & teaching

- Teaching, research labs, multi-PI and multi-institute collaborations: anything with 2 or more computers.
- In computational courses, can take two weeks to get everyone "on the same page" w/ software, VMs, etc.
  OS matters, versions matter, build environments matter, ...
- Work done by one PhD student is rarely usable by the advisor or the next PhD student—much less by the rest of the world. Claerbout's experience.
- BCE: reproducible recipe to (re)create software environment that fosters reproducible work.
- See also `http://datasciencetoolbox.org/`

BCE first-cut ingredients

- a version of linux, perhaps Ubuntu
- docker
- lxc
- git, a git gui, gitlabhq, git-annex assistant
- Python + IPython + Numpy + Scipy + Matplotlib + Pandas + Cython + other libraries
- R and various libraries for statistics and machine learning
- mySQL, MariaDB, SQLite
- LaTeX, BibTeX + AMS, Beamer, & other styles
- some stack for distributed computing
- test suites for all the software

Teaching reproducible and collaborative computational research

- Hard to teach an old dog new tricks.
- Solution: Work with puppies.
- Statistics 157, fall 2013:
  Reproducible and Collaborative Statistical Data Science
- Project: improved earthquake forecasts for Southern CA
- Sought to replicate and extend work of PhD student
- Syllabus includes introduction to virtual machines, GitHub, IPython, SCEC data
- `http://youtu.be/Bq71Pqdukeo`, `Git_That_Data.mp4`
- How can we work reproducible practices into the whole curriculum?