POLI 229: Social Science Replication

Gareth Nellis

Winter 2019

E-mail: gnellis@ucsd.edu
Office Hours: By appointment
Office: SSB 350

Web: garethnellis.github.io
Class Hours: Th 09:00–11:50
Class Room: SSB 104

Course description

The purpose of this class is to learn how to do cutting-edge empirical research in the social sciences by replicating others' work. For each class, we will have a non-UCSD faculty member visit campus. A week before their visit, they will circulate a working paper—typically one drawing on an experimental or quasi-experimental design—along with the data and code needed to reproduce the results. Small teams of graduate students will reanalyze the data, and propose further robustness tests and extensions. A presentation and group discussion of these findings, with input from the paper's author, will last about two hours. Classes will start with short (50 minute) lectures covering various topics in experimental design and analysis. This is an opportunity for graduate students to engage with scholars from leading departments, take a deep dive into work in progress, and to pick up new tools and best practices.

Prerequisites

The class is aimed at students in the second, third, and fourth years of the PhD program in political science. A strong understanding of probability, regression, and causal inference is required. Prior knowledge of `R` is highly recommended.

Course requirements and grading policy

1. **Full replication (x1, as part of a pair; 40%).** This is the principal assignment and you should work on it very carefully. Here are some pointers.

- You should produce three outputs:
    a. Replication plan. Once you've read the visitor's working paper, write a replication plan outlining in words the additional tests you'd like to do. Finalize this document *before you do the tests themselves*, and don't make changes after you start analyzing. Given time constraints, I would limit this to 2–3 pages. It's more important to spend time with the data.

   b. A replication report. This can be as long as you wish. It should walk the reader through your replication and its findings. You will hand this over to the paper's author for their reference. Be clear in your report which replication tests you registered (in the replication plan) and which are exploratory. Note that non-registered tests are absolutely fine; you just need to mark them.
   c. An in-class presentation of your replication. This should last at least one hour, but the timing is very flexible. You should take any and all questions along the way. We may also pause the presentation for wider group discussion.
- For the replication itself:
  - Verify the original analysis runs, and reproduce the primary analyses.
  - Check data cleaning steps as far as possible.
  - Identify the key features of the research design, as well as the main estimands and estimators, and simulate it using `DeclareDesign`. Comment on the design diagnosis in your report and presentation.
  - If the paper's analysis was pre-registered, read the PAP(s) and see to what extent the plan was followed. Note there are often very good reasons for deviating from a PAP. Do you agree with the deviations? Are they consequential for the paper's results and conclusions?
  - Check robustness: alternative specifications, estimators, codings of key measures, etc.
  - Extend the analysis, perhaps using new data (e.g. other outcomes, heterogeneous treatment effects).
  - Think hard about the relationship between the theory advanced in the paper and the evidence given. Do the empirics support the proposed explanation?
- Be collegial, constructive, humble, and measured at all times. Highlight a paper's strengths and not just any weaknesses. You may disagree with an author's analysis or interpretation. You might find errors. Communicate these points tactfully. We all want to work in an environment where openness is rewarded and colleagues help each other out.
- Practice the presentation. Divide up the material so each person presents for roughly the same amount of time.
- You should collaborate on all three replication documents using RMarkdown and Github (see below).
- Feel free to get in touch with the author directly—for clarifications or to request extra materials (e.g. survey instruments or cleaning code).
- Arrange to meet with me on the Monday or Tuesday before class to go over your draft presentation.

2. **Weekly memos (x7, individually; 20%).** For seven of the ten weeks you should write a 2-page (single-spaced) memo responding *in advance of class* to the visitor's paper for that week. You don't need to write a memo in the week you are on the replication team. You must write a memo for the first class (Jan 10). Otherwise, you can decide how to space the memos, according to your schedule.

The memo should have two parts:

a. A design declaration. Use `DeclareDesign` to run simulations on a mock up of the paper's research design. Focus only on the key estimand, unless you have a particular interest in other quantites being estimated. Print out a table summarizing the diagnosis and comment on it.
b. A mini peer review. Write at least one full page responding to other aspects of the paper.

What are the paper's contributions? What are its shortcomings? Where is the paper unclear? How might it be revised or extended? Focus on the empirics, but feel free to comment on the theoretical logic and the theory/evidence match too. *Wherever possible, support your points with reference to the data.* This means you should engage with the data even in weeks when you are not on the replication team.

The memos will ensure we have plenty to talk about in class. Upload your memo to the shared Github repo by 6pm on Wednesday—the day before the class in which the paper is to be discussed. Make time to read others' memos on Wednesday night or Thursday morning.

3. **Pre-analysis plan with simulated data & analysis (x1, individually; 20%).** This is due on the Friday of finals week. The assignment is to write a pre-analysis plan for an empirical study you wish to conduct, perhaps as part of your dissertation work. The planned study may be observational or experimental. The PAP should be at least 10 pages in length. It must include an analysis using simulated data.

4. **Participation (20%).** Full participation is expected. Getting a top participation grade won't be easy! It will very much depend on the quality and quantity of your contributions to group discussion.

Visitor schedule

- Jan 10: Nate Jensen (UT Austin)
- Jan 17: Rob Blair (Brown)
- Jan 24: Dawn Teele (UPenn)
- Jan 31: Alex Coppock (Yale)
- Feb 7: Pia Raffler (Harvard)
- Feb 14: Jennifer Bussell (UC Berkeley)
- Feb 21: *TO BE CONFIRMED*
- Feb 28: Saad Gulzar (Stanford)
- Mar 7: Josh Kertzer (Harvard)
- Mar 14: Susan Hyde (UC Berkeley)

Familiarize yourself with the visitors' research agendas.

Depending on travel schedules, there will be a dinner, lunch, or coffee organized for each speaker and several students. I will be in touch by email about arrangements. If you'd like to meet visitors one on one, please let me know. This will usually be possible.

Readings

You must read every visitor's paper in depth before class, even if you aren't writing a memo that week. Classes also feature one or two assigned readings on various topics having to do with research design, analysis, and measurement. These will be addressed in brief lectures at the start of each class. I recommend purchasing this book:

- Gerber, Alan S and Donald P Green (2012). *Field Experiments: Design, Analysis, and Interpretation*. W.W. Norton. .

Software tools

The class tries to instill excellent workflow habits, and to teach tools that will make your research easier to conduct and more transparent. For this reason, the software and presentational requirements will be quite strictly enforced. All class work should be done using an R-RMarkdown-Git integration via RStudio and Github. We will also use Dropbox for some tasks. The software is free.

R & RMarkdown

Replications, weekly memos, and pre-analysis plans should be written and coded in single, continuous RMarkdown (Rmd) documents. You should knit to html or pdf for written submissions, and to beamer or Xaringan for slide presentations.[1] Everything should be "one-click" replicable, meaning all text and analysis output compiles from a raw Rmd with a single click of "Knit" in RStudio.

To get started in RMarkdown, review this guide.[2]

Some tips on coding in `R`:

- Code should be tightly ordered and extensively annotated. Good code is at least as important as good writing. It makes potentially embarrassing mistakes easier to catch.
- Scrupulously follow the Don't-Repeat-Yourself (DRY) principle. Look for vectorized solutions. For repetitive operations—even simple ones—write your own functions and apply them. One piece of advice is to eschew for-loops when first learning `R`. This forces you to write more idiomatic code and quickly pick up time-saving tricks.
- I'm a big fan of the `tidyverse` suite of packages, especially `dplyr` and `purrr`. `dplyr` is unbeatable for data manipulation—much more concise, consistent, and intuitive than base equivalents. `purrr` irons out many of the unpredictabilities that characterize the base `apply` functions, although it's not yet widely used.
- To make figures, use `ggplot2`.
- To make tables, use `kable` (with `kableExtra`), `stargazer`, and `broom`.
- I suggest working with pipes (`%>%` and `%<>%`). These cut down on redundant syntax and keep the environment de-cluttered.
- If you can't figure something out, don't give up and go with a quick fix (e.g. copying and pasting code), for which you know there's probably a better solution. Google is your friend. Stackoverflow is you best friend and you should learn how to post questions there for times when you're really stuck.[3]
- Good intros to `R` include *R for Data Science* and *R Inferno*.

I'll read your code carefully and may ask you to rewrite parts that can be improved. Don't feel at all worried or intimidated. Reach out for help to me and fellow students when you're struggling.

[1] Xaringan is html-based and thus allows you to build dynamic content like Shiny apps into your slides.
[2] For more complex documents, including book projects, I recommend Bookdown by Yihui Xie.
[3] Be sure to use MWEs.

Git(hub)

Once you have RStudio up and running, the next thing to do is to get it to work with the version control system, Git. This was invented by and for software developers, but it is invaluable for social scientists too. It keeps a record of all committed changes, allows for easy reversion, and simplifies sharing and collaboration.

A few steps are needed to link up Git and RStudio. You need to install Git on your computer. Next, make a Github account. Then configure RStudio to recognize that account. This guide from the University of Zurich walks you through the steps.

Dropbox

Authors will send data and code at least one week before their visit. This be stored in a Dropbox folder to be shared with class members. Always pull data directly from the online Dropbox folder. (This is good practice and very helpful for one-click replicability.) To do this, you will (a) need the file url, which you can get by right-clicking on the relevant file in Dropbox.com, and (b) need to change the last four letters of that url from "dl=0" to "raw=1". For example:

```
# import dummy data direct from db
  dummy_data <- read_csv(
    url("https://www.dropbox.com/s/5wmcdhzc1kbdecj/20181222-dummy-data.csv?raw=1")
    )
```

DeclareDesign

We will be getting to know `DeclareDesign`, a new `R` package that makes it straightforward to run Monte Carlo simulations of complex research designs, and thus to assess their properties (e.g. power, bias, RMSE). Follow this guide to get started, and play with the vignettes in the design library.

Academic honesty

You are expected to do your own work, and to properly attribute ideas, quotations, and sources. Please consult the university's website on academic integrity.

Disabilities policy

Students with disabilities should please inform me of any accommodations you may need.

Email policy

I will reply to emails within two business days.

Acknowledgements

Huge thanks to Claire Adida and Lauren Prather for advice and help with initial planning; to UCSD's International Institute for funding; Katia Todorova for organizing speakers' travel; Macartan Humphreys for first coming up with the idea for the class format; and above all to the visitors for so generously sharing their work and giving up their time.

Class Schedule

Thursday, 2019-01-10—Nate Jensen (University of Texas, Austin)

Lecture topics: Pre-analysis plans; `DeclareDesign`

Reading:

- Feynman, Richard P (1974). "Cargo cult science". In: *Engineering and Science* 37.7, pp. 10–13. Link.
- Blair, Graeme, Jasper Cooper, Alexander Coppock and Macartan Humphreys (2018). "Declaring and diagnosing research designs". In: *Unpublished manuscript, UCLA*. Link.

Thursday, 2019-01-17—Rob Blair (Brown University)

Lecture topics: Randomization & blocking; randomization inference

Reading:

- Alex Coppock. "Randomization Inference with ri2." Vignette.
- Young, Alwyn (2018). "Channeling fisher: Randomization tests and the statistical insignificance of seemingly significant experimental results". In: *Quarterly Journal of Economics*. Link.
- Gerber, Alan S and Donald P Green (2012). *Field Experiments: Design, Analysis, and Interpretation*. W.W. Norton. Chapter 3. *This should mostly be refresher.*

Thursday, 2019-01-24—Dawn Teele (University of Pennsylvania)

Lecture topics: Covariate adjustment; specification curve analysis

Reading:

- Bloniarz, Adam, Hanzhong Liu, Cun-Hui Zhang, Jasjeet S Sekhon and Bin Yu (2016). "Lasso adjustments of treatment effect estimates in randomized experiments". In: *Proceedings of the National Academy of Sciences* 113.27, pp. 7383–7390. Link.
- Simonsohn, Uri, Joseph P Simmons and Leif D Nelson (2015). "Specification curve: Descriptive and inferential statistics on all reasonable specifications". In: *Working paper*. Link.

Thursday, 2019-01-31—Alex Coppock (Yale University)

Lecture topics: Measurement/construct validity—indexing; survey methods for sensitive topics

Reading:

- Rosenfeld, Bryn, Kosuke Imai and Jacob N Shapiro (2016). "An empirical validation study of popular survey methodologies for sensitive questions". In: *American Journal of Political Science* 60.3, pp. 783–802. Link.

Thursday, 2019-02-07—Pia Raffler (Harvard University)

Lecture topics: Power analysis

- Glennerster, Rachel and Kudzai Takavarasha (2013). *Running Randomized Evaluations: A Practical Guide*. Princeton University Press. Chapter 6.

Thursday, 2019-02-14—Jennifer Bussell (University of California, Berkeley)

Lecture topics: Qualitative research & transparency

- Louise Corti. "Show me the data: research reproducibility in qualitative research." Read blog post and peruse links.

Thursday, 2019-02-21—Visitor TBC

Lecture topics: Multiple hypothesis testing; attrition

Reading:

- John H. McDonald. *Handbook of Biological Statistics*. "Multiple comparisons." Link.
- Gerber, Alan S and Donald P Green (2012). *Field Experiments: Design, Analysis, and Interpretation*. W.W. Norton. Chapter 7.

Thursday, 2019-02-28—Saad Gulzar (Stanford University)

Lecture topics: Directed acyclic graphs (DAGs); mediation

Reading:

- Gerber, Alan S and Donald P Green (2012). *Field Experiments: Design, Analysis, and Interpretation*. W.W. Norton. Chapter 10.

Thursday, 2019-03-07—Josh Kertzer (Harvard University)

Lecture topics: One-sided non-compliance

Reading:

- Gerber, Alan S and Donald P Green (2012). *Field Experiments: Design, Analysis, and Interpretation*. W.W. Norton. Chapter 5.

Thursday, 2019-03-14—Susan Hyde (University of California, Berkeley)

Lecture topics: Two-sided non-compliance

Reading:

- Gerber, Alan S and Donald P Green (2012). *Field Experiments: Design, Analysis, and Interpretation*. W.W. Norton. Chapter 6.