Experiments and Causality

This course introduces students to experimentation in data science. The course pays particular attention to forming causal questions, and to the designing experiments that can provide answers to these questions. Weekly discussion are guided by this *Preparing for Live Session* document.

Schedule

Week	Topics	Async Reading	Sync Reading	Assignment Due
1	Experimentation	FE 1, NYT	Feynman, Suburbs, Shoes, Predict or Cause	None
2	Apples to Apples	FE 2; Lewis & Reiley (p. 1-2.5, §1; §2A-B)	Poor Economics, Ch. 1, 3, 6; Lakatos (O): Rubin, sections 1 & 2	Essay 1, PS 0
3	Quantifying Uncertainty	FE 3.0, 3.1, 3.4	Blackwell, Lewis and Rao 1, 3.1, 3.2	PS 1
4	Blocking and Clustering	FE 3.6.1, 3.6.2, 4.4, 4.5	(O): Cluster Estimator, BlockTools, When to Cluster	Three Project Ideas
5	Covariates and Regression	MM 1, FE 4.1-3, MM 2, MHE p. 16-24	Opower (O): FE Appendix B (p. 453), rerandomization	Two Page Description

6	Regression; Multi-factor Experiments	MM 6.1, MM 95-97, FE 9.3.3, 9.4	Montgomery Sections 1, 3.0, 3.1, 3.2, 3.5, 4.2, Skim 5	PS 2
7	HTE	FE 9, Multiple Comparisons, and Demo	Goodson (O): JLR 1, 2, 3.1, 4.3, Etsy	–
8	Noncompliance	FE 5	G&G 2005; TD, Ch 7; TD, Ch 9	PS 3
9	Spillover	FE 8 and lyft and (O) uber	Miguel and Kremer; Blake and Cohey 2, 3	Project Check-In
10	Causality from Observation?	MM 3.1, 4.1, 5.1	Incinerators, Glynn, Dee (O): Glassberg Sands, Lalive, Rubin, Section 3	–
11	Problems, Diagnostics and the Long View	FE 11.3	DiNardo and Pischke, Simonsohn (O): Robinson	PS 4, Pilot Data
12	Attrition, Mediation, Generalizabilty	FE 7, 10, Bates 2017	Alcott and Rogers	
13	Creative Experiments	FE 12, (O): Ny Mag,	Broockman Irregularities, Hughes et al. (O): Uber Platform	PS 5

		Science, FE 13		
14	Final Thoughts		Freedman	Presentation
15	–	(O): Retracted LaCour, (tl;dr), Podcast (audio))		Final Paper

Description

This course begins with a discussion of the issues with causal inference based on observational data. We recognize that many of the decisions that we care about, whether they be business related or theoretically motivated, are *essentially* causal in nature.

The center of the course builds out an understanding of the mechanics of estimating a causal quantity. We present two major inferential paradigms, one new and one you are likely familiar with. We first present randomization inference as a unifying, intuitive inferential paradigm. We then demonstrate how this paradigm sits in complement to the classical frequentist inferential paradigm. These concepts in hand, we turn focus to the design of experiments and place particular focus both answering the question that we set out to answer, and achieving maximally powered experiments through design.

The tail of the course pursues two parallel tracks. In the first, students form a research question that requires a causal answer and design and implement the experiment that best answers this question. At the same time, new content presented in the course focuses on the practical stumbling blocks in running an experiment and the tests to detect these stumbling blocks.

We hope that each student who completes the course will:

- Become skeptical about claims of causality. When faced with a piece of research on observational data, you should be able to tell stories that illustrate possible flaws in the conclusions.
- Understand why experimentation (generating one's own data by doing deliberate interventions) solves the basic causal-inference problem. You should be able to describe several examples of successful experiments and what makes you feel confident about their results.
- Appreciate the difference between laboratory experiments and field experiments.
- Appreciate how information systems and websites can be designed to make experimentation easy in the modern online
- Understand how to quantify uncertainty, using confidence intervals and statistical power calculations.
- Understand why control groups and placebos are both important.
- Design, implement, and analyze your own field experiment.
- Appreciate a few examples of what can go wrong in experiments. Examples include administrative glitches that undo random assignment, inability to fully control the treatment (and failure to take this inability into account), and spillovers between subjects.

Computing is conducted primarily in R.

If you are looking to work on something over the break between semesters, we recommend that students spend a little time familiarizing themselves with \`data.table\` which is the data manipulation idiom that we will be using in the course.

- [Here] is a lecture on the topic created by Grant McDermott at the University of Oregon.
- There is also a course, created by the package authors at Data Camp. I recommend that you **do not** take this course. The leadership at Data Camp was credibly accused of sexual harrassment, and as is described [here] actively worked to avoid accountability. As an example, Rstudio has walked away from collaborating and teaching using Data Camp. The course exists, and Data Camp has removed the harrasser from leadership; we leave it to you to evaluate giving mind-share to the company, but don't provide a link.

Compute Environment

There are several options for how to build a compute environment for this course.

- You have the option of using a one-click available UCB Datahub [<–link that syncs course content to datahub].

    - If you do not want to re-sync content, or manage the syncing yourself (via a shell) you can navigate to the Datahub directly: You can get to it here.
    - This is a minimal instance – you're capped at 1GB memory, but it is a really nice way to work on async coding without having to start any machinery of your own. You should be able to knit, save, and edit as you like.
    - The courses' upstream repository is *entirely* segmented from your copy of this, so feel free to make any changes that you want. Note, however, that this also means that changes you make in the datahub will *not* be present on your own fork of the repository. In other words, things that happen in the datahub, stay in the data hub.
- You can alternatively use this Docker image on your machine, or any other machine that has a docker engine. (This image builds from a canonical Rocker image).
    - This short tutorial provided by ROpenSciLabs is just enough to get you going and dangerous.
- Finally, if you're brave, or you know the history of your computer, you can install locally.

Books

We use two books in this course, and read a third book in the second week. We recommend that you buy a paper copy of the two textbooks (we've chosen textbooks that have a fair price), and would understand if you digitally read the third book. Support a local bookstore if you can; but, we've included a link to Amazon for those who cannot.

- *Field Experiments: Design and Analysis* is the core textbook for the course. It is available on Amazon for $40 [link] and is necessary to succeed in the course.
- *Mastering Metrics* is the secondary textbook for the course. It is available at Amazon for $20 [link].
- *Poor Economics* is the third book for the course. It is available for purchase on Amazon for $15 [link], and from the UC Library digitally [link].
- *More than Good Intentions* was previously used in the course. For folks with an interest in questions of development, it is an interesting read. It is available at Amazon for $10, new, or $3 used [link]. But, you could also read this digitally.

Articles

- We have made all the articles we read in the couse available in the repository. However, it is a *great* practice to get used to establishing a VPN to gain access to

all the journal articles that are available through the library subscription service. Instructions for connecting are available on the UCB library website. Journal access is one of the greatest benefits to belonging to a University, we suggest you use it.
- David has made a great resource that has suggestions for further reading. You can access it in this living google doc.

Office Hours (all times Pacific)

Day	Time	Instructor
Monday	4:00-5:00	David
Monday	6:30-7:30	Tim
Tuesday	5:30-6:30	Micah
Tuesday	6:30-7:30	Tim
Wednesday	3:00-4:00	Scott
Wednesday	4:00-5:00	David
Thursday	5:30-6:30	Micah
Thursday	3:00-4:00	Scott
(Friday before PS)	4:00-5:00	Alex
(Saturday after PS)	9:00-10:00 a	Alex

- In weeks where we have problem sets due, we will hold extra office hours on the Friday before the weekend. As well, when you are working through your project

design, the instructors will schedule individual one-on-one conversations as necessary with student groups.
- On Saturdays after we turn in problem sets, we will hold extra office hours to review the work that you've done and the feedback that you've received. For obvious reasons, you can only attend these Saturday OH if you have submitted your via PR.

Grading and Scoring

- Problem Sets (45%, 9% each) A series of problem sets, mostly drawn from FE, many requiring programming or analysis in R.
    - We encourage you to work together on problem sets, because great learning can come out of helping each other get unstuck. We ask that each person independently prepare his or her own problem-set writeup, to demonstrate that you have thought through the ideas and calculations and can explain them on your own. This includes making sure you run any code yourself and can explain how it works. Collaboration is encouraged, but mere copying will be treated as academic dishonesty.
    - At this point, the course has lived for a number of semesters, and we have shared solution sets each semester. We note in particular that struggling with the problems is a key part of the learning in this course. Copying from past solutions constitutes academic dishonesty and will be punished as such; you should know that we have included language in the solutions that will make it clear when something has been merely copied rather than understood.
- Essays (20%, 10% each) You will write two essays in the course. The first essay is an individual assignment. The second essay, 3 research designs for your final project, will be drafted and submitted as a group.
- Class Experiment (30%) In teams of 3-5 students, carry out an experiment that measures a causal effect of interest. See the `./finalProject/` folder for much more information
- Async Concept checks (5%) Throughout the course, we have included concept checks, hikes, and yogas. These are our measure of preparedness of the async content.
- Late Policy: You're busy and things come up – kids get sick, parents stop by unannounced, managers ask you to reformat your TPS reports, you learn that your 261 project has accumulated $50,000 in compute costs – we get it. You've got five (5) days to turn things in late without penalty, without explanation, and without notice. We'll count at the end of the semester. After you use those 5,

each additional day (or part thereof) comes at the cost of 10% on the assignment. That is, 1% off your end-of-semester total grade. Here’s the other twist though – we need to provide solutions back to your classmates who have completed their work. So, no individual assignment can come in more than 5 days late; any assignment that does will score a zero. If you see ahead of time that you’re going to have a conflict – a major release, a vacation, etc. – talk with your instructor to work out an alternative. We’ll work with you, but the more notice, the better.