From Paper to Program: Challenges of Implementing Permutation Tests

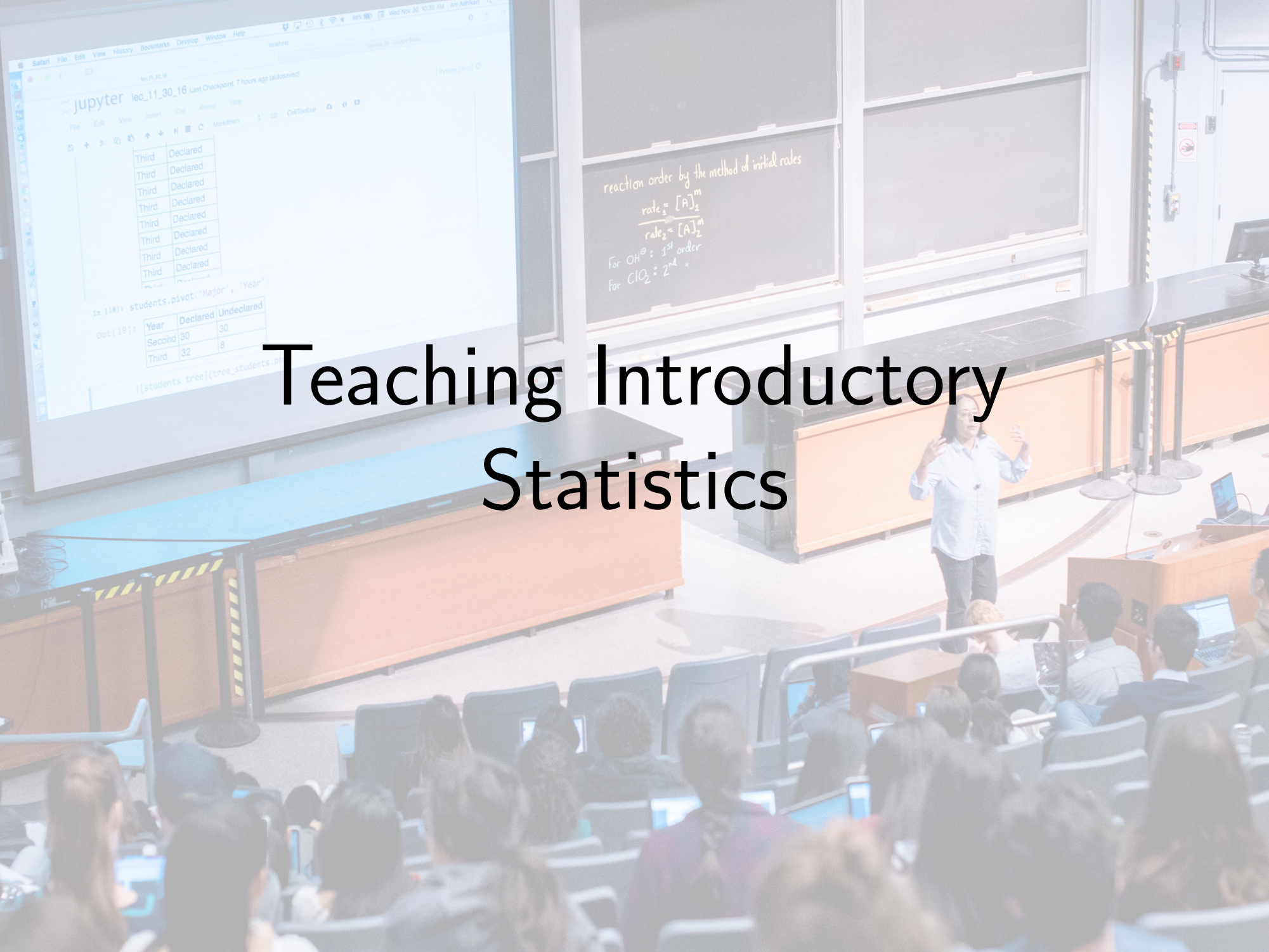
Kellie Ottoboni

Department of Statistics, UC Berkeley
Berkeley Institute for Data Science

June 12, 2018
ISNPS, Salerno

University of California, Berkeley
DEPARTMENT OF STATISTICS

BIDS
BERKELEY INSTITUTE
FOR DATA SCIENCE


Teaching Introductory
Statistics

Introductory Statistics

Important concepts

- Sampling distributions
- p-values
- Confidence intervals

Introductory Statistics

Important concepts

- Sampling distributions
- p-values
- Confidence intervals

Barriers to learning

- Z tests, t tests
- Assumptions
- Formulas

What if we could teach the concepts without the particular details?

Tools:

1. Resampling methods
2. Computers

Data Science 8
UC Berkeley

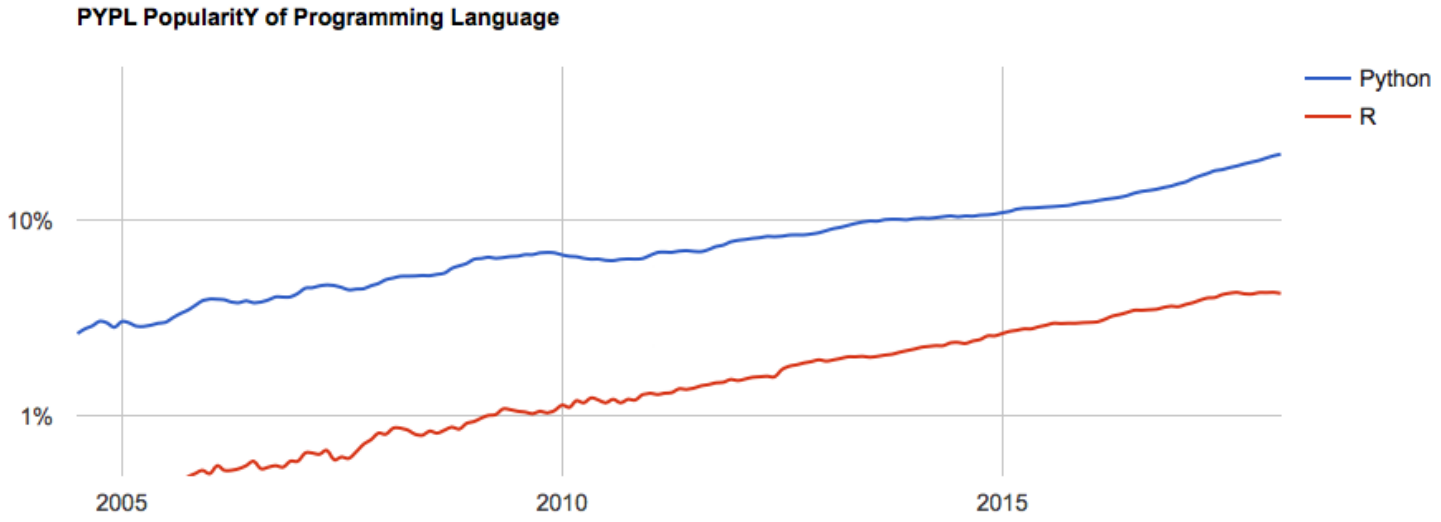
Permutation tests clarify concepts.

- General: it's a procedure, not a formula
- Discrete: counting instead of integration
- Design-based: assumptions come from the data collection

Hesterberg (2015)

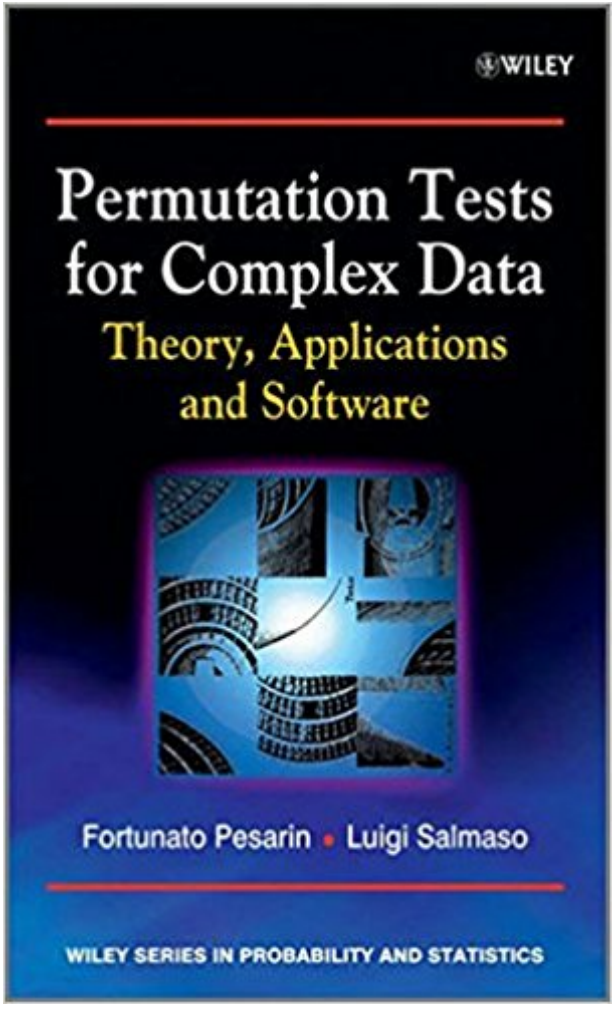
Python is gaining popularity for data analysis.

- General purpose language, “batteries included”
- Libraries for a variety of scientific applications



The book

A companion to Pesarin and Salmaso (2010)

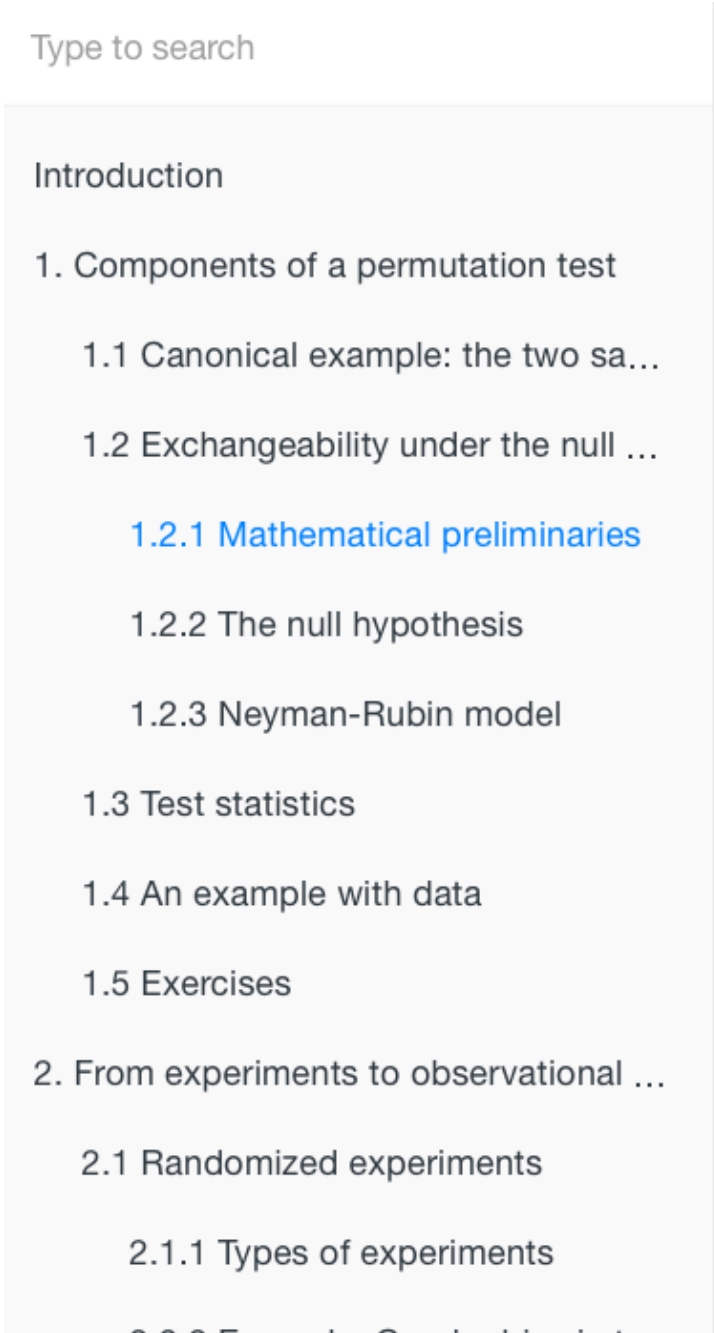


- Target audience: late undergraduate to early PhD or domain researchers with some statistics background
- Conceptual: how to determine an appropriate permutation test
- Carefully consider experimental design and assumptions
- Introduce concepts about causal inference

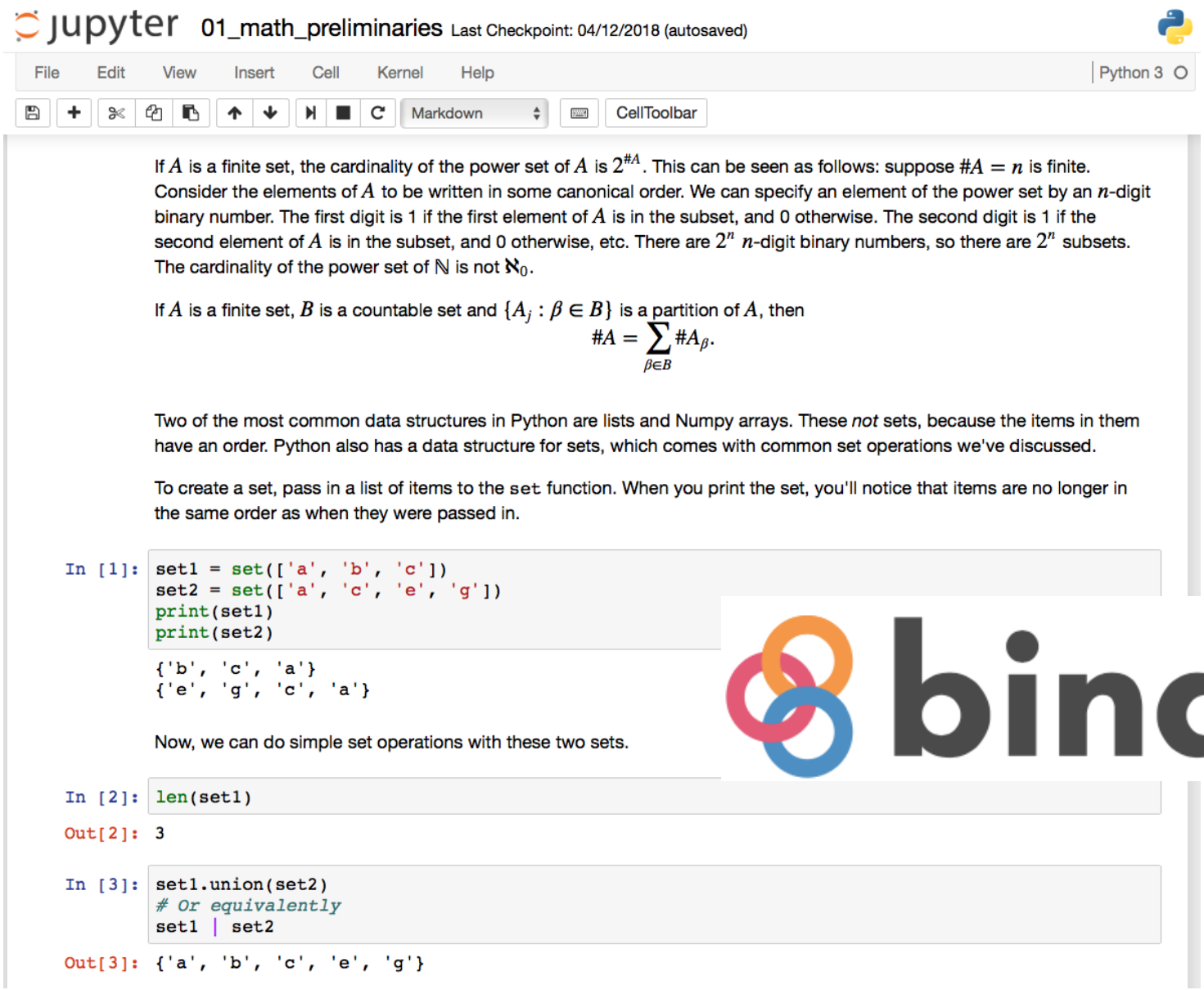
Three chapters

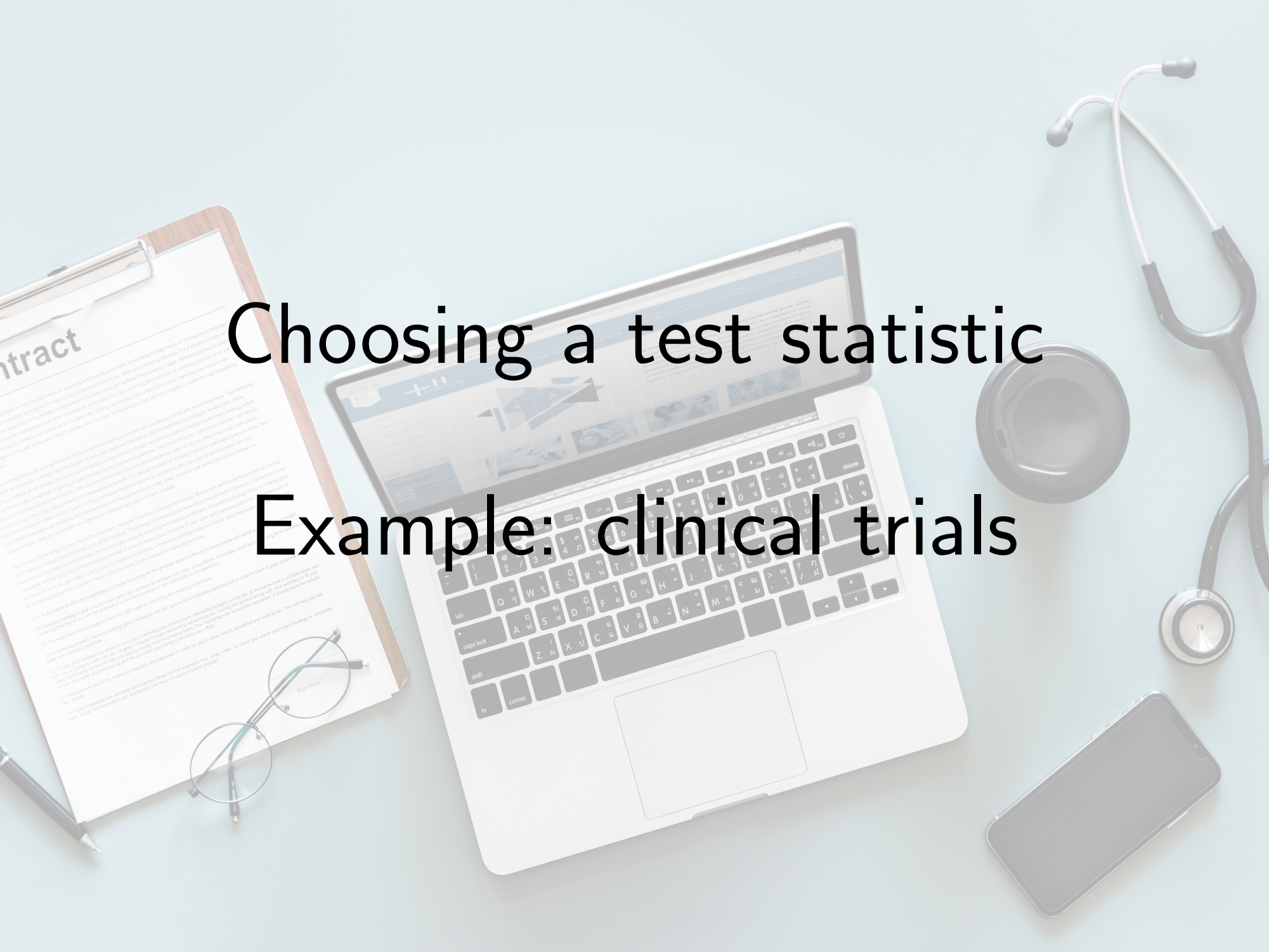
1. Components of a permutation test
   The null invariance and the test statistic
2. From experiments to observational studies
   Randomization, confounding, the "implied experiment"
3. Computation for permutation tests
   Sampling algorithms and pseudorandom numbers

The book will be open-source online...



with interactive Python code.





Choosing a test statistic

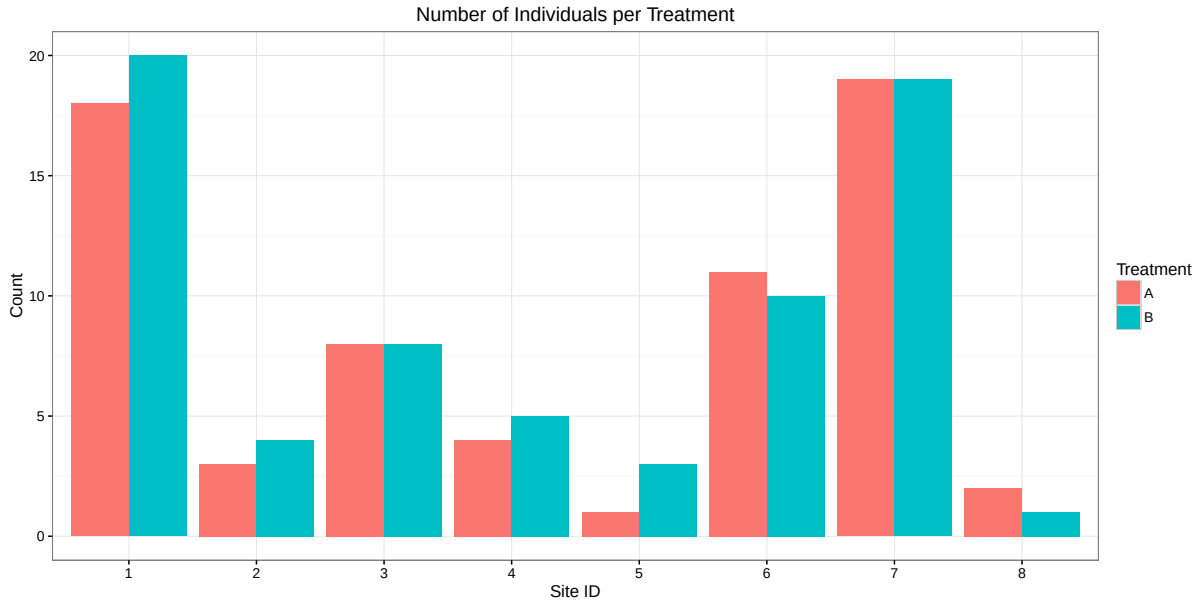
Example: clinical trials

Clinical trial data

Randomized experiment to treat GERD

- 136 patients
- 8 sites
- 7 clinical endpoints, baseline X and follow-up Y measures

Did treatment A reduce symptoms compared to treatment B?

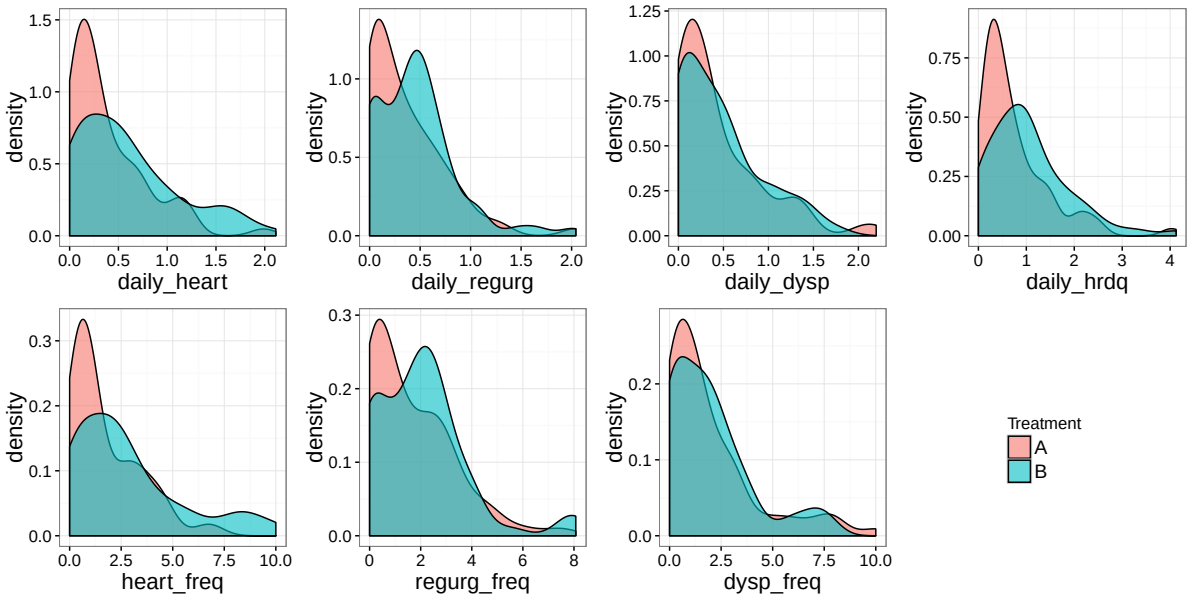


Ottoboni, Lewis, and Salmaso (2018)

The permutation test

Test the **sharp null hypothesis** that treatment has no effect:

$$H_0 : Y_{ij}(A) = Y_{ij}(B), \forall \text{individuals } i \text{ and sites } j$$



Naive, but correct, statistic: difference in mean outcomes

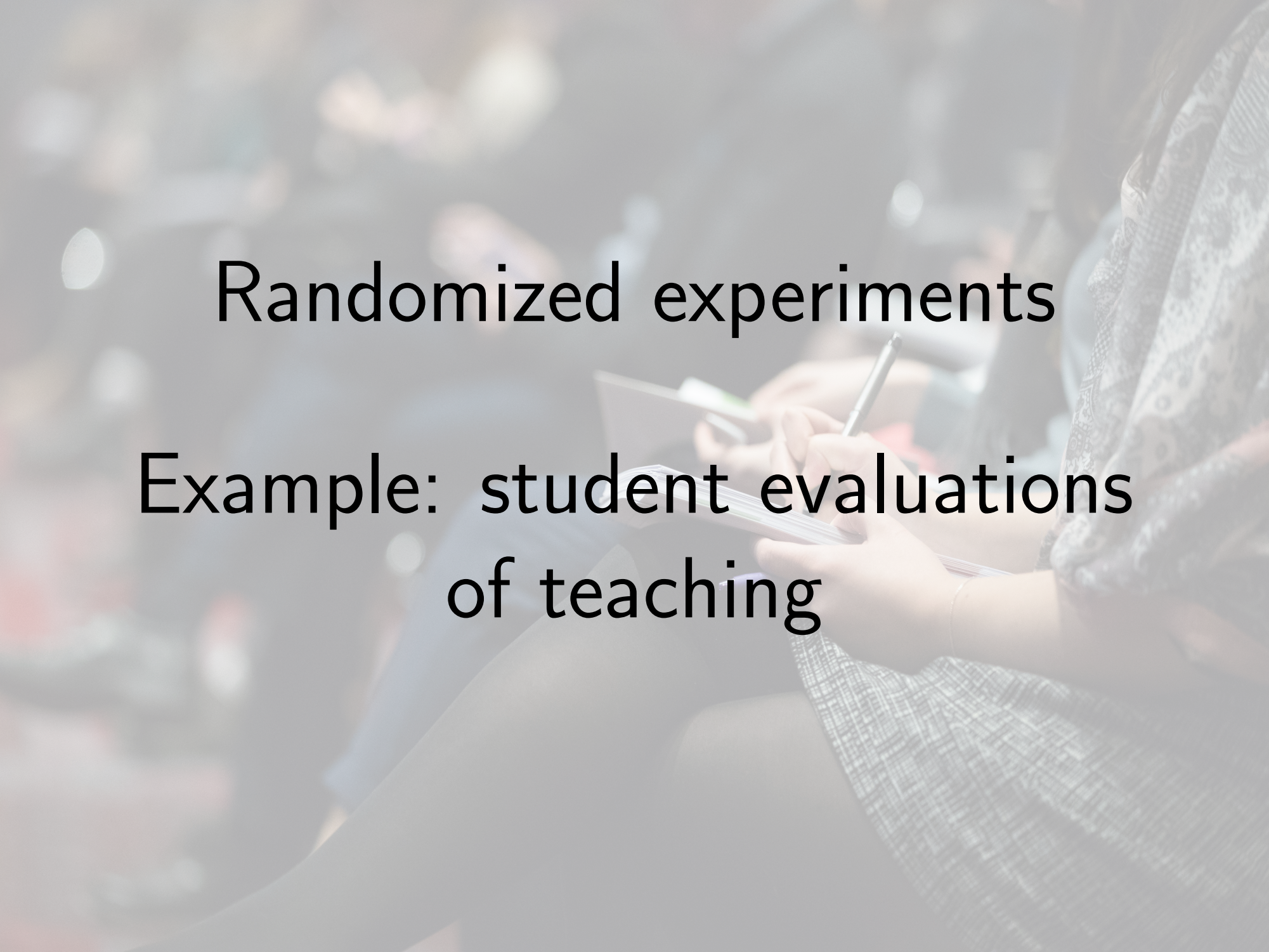
$$\overline{Y(A)} - \overline{Y(B)}$$

Controlling for covariates

- Difference scores: $\left(\overline{Y(A)} - \overline{X(A)}\right) - \left(\overline{Y(B)} - \overline{X(B)}\right)$
- t statistic from a linear model

$$Y_{ij} = \alpha_j + \beta X_{ij} + \gamma Z_{ij} + \varepsilon_{ij}$$

- Two ways to find a permutation distribution:
  - Permute Z
  - Permute estimates $\hat{\varepsilon}$ from reduced model (*Freedman and Lane (1983)*)



Randomized experiments

Example: student evaluations of teaching

Teaching evaluations

Are SET a valid measure of teaching effectiveness?

Data from MacNell et al. (2014)

- Students were randomized to 4 online sections of a course.
- In two sections, the instructors swapped identities.
- Was the instructor who identified as female rated lower on average?

Boring, Ottoboni, and Stark (2016)

Neyman-Rubin model, generalized

$$\text{Rating } r_{ijk} = \text{ SET given by student } i \text{ to instructor } j \text{ when they appear to have gender } k$$

If gender doesn't matter, $r_{ij\text{male}} = r_{ij\text{female}}$.

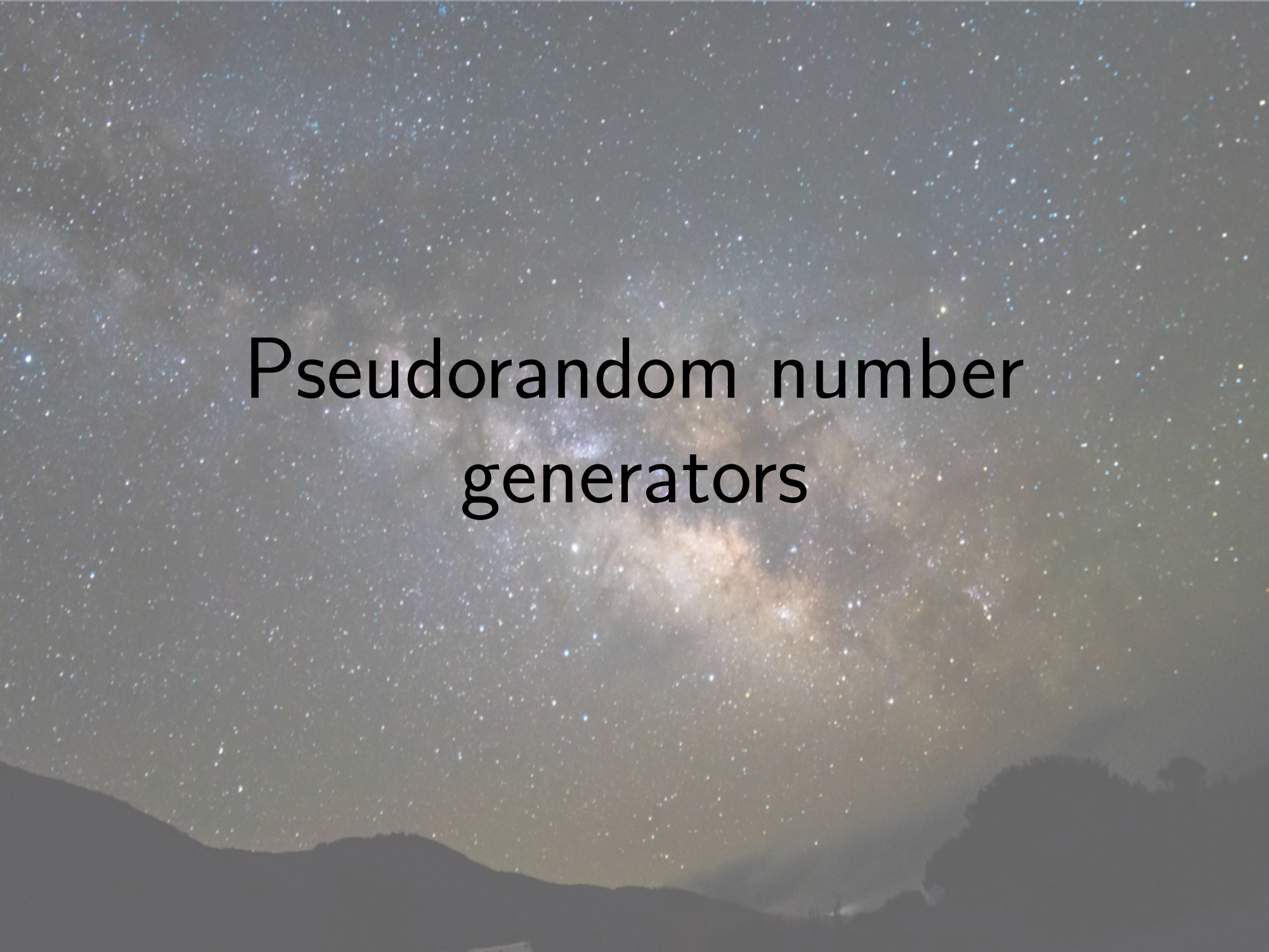
Stratified two-sample test

- For each instructor, permute perceived gender assignments
- Use difference in mean ratings for female-identified minus male-identified

Results

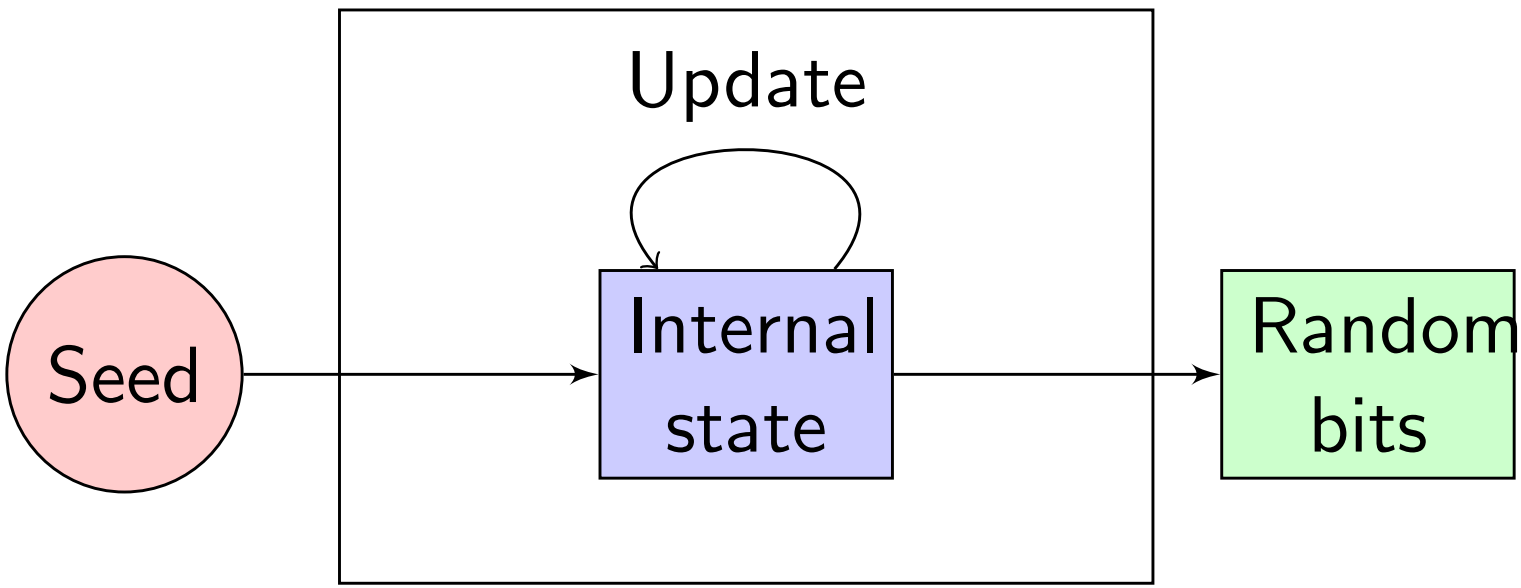
In all categories, the male-identified instructor was rated higher.

Characteristic	**M-F**	**perm** P	**t-test** P
Overall	0.47	0.12	0.128
Caring	0.52	0.10	0.071
Consistent	0.47	0.21	0.045
Enthusiastic	0.57	0.06	0.112
Fair	0.76	0.01	0.188
Feedback	0.47	0.16	0.054
Helpful	0.46	0.17	0.049
Knowledgeable	0.35	0.29	0.038
Praise	0.67	0.01	0.153
Professional	0.61	0.07	0.124
Prompt	0.80	0.01	0.191
Respectful	0.61	0.06	0.124
Responsive	0.22	0.48	0.013


Pseudorandom number generators

What is a PRNG?

- A user-supplied **seed value** used to set the internal state
- A function that maps the **internal state to pseudorandom bits**
- A function that **updates the internal state**



Pigeonhole principle in action

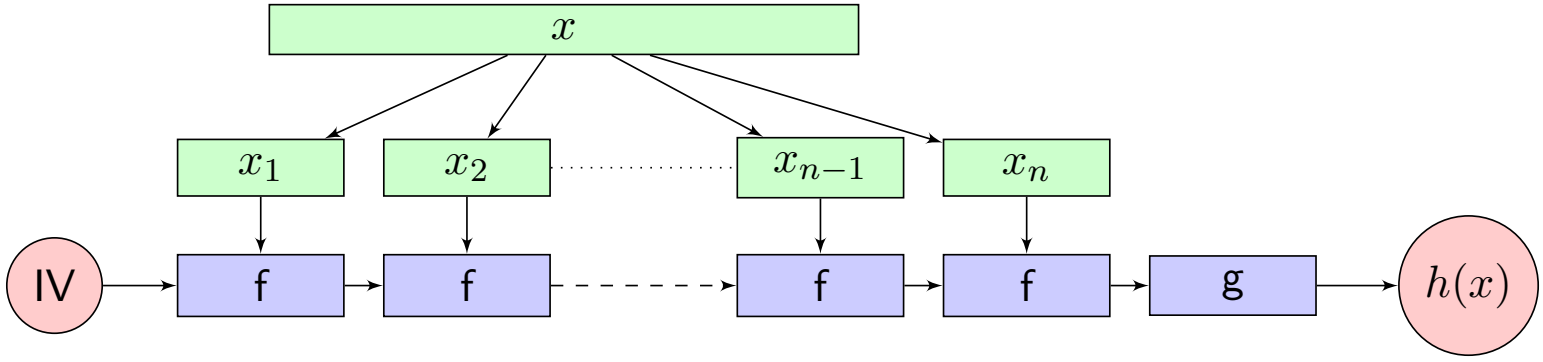
Theorem (Too few pigeons)

If $\binom{n}{k}$ is greater than the size of a PRNG's state space, then the PRNG cannot possibly generate all samples of size k from a population of n.

PRNG	# Internal states	# Possibilities	Proportion of attainable possibilities
32-bit linear congruential generators	4 billion	Samples of 10 out of 50 items ≈ 10 billion	0.4
Mersenne Twister	$\approx 2 \times 10^{6010}$	Permutations of 2084 items $\approx 3 \times 10^{6013}$	0.0001

Cryptographically secure PRNGs

Hash functions take in a message x of arbitrary length and return a value $h(x)$ of fixed size (e.g. 256 bits)



Cryptographic hash functions:

- computationally infeasible to invert
- difficult to find two inputs that map to the same output
- small input changes produce large, unpredictable changes to output
- resulting bits are uniformly distributed

From paper to program

- Teach the concepts of hypothesis testing using permutation tests
- Open-source online book with interactive Python examples
- Packages on GitHub: `permute`, `cryptorandom`, `pscore_match`