Power Analysis (SOC 412)

Week 5 Lecture 10

Sherrerd Hall 306

J. Nathan Matias

@natematias

civilservant.io

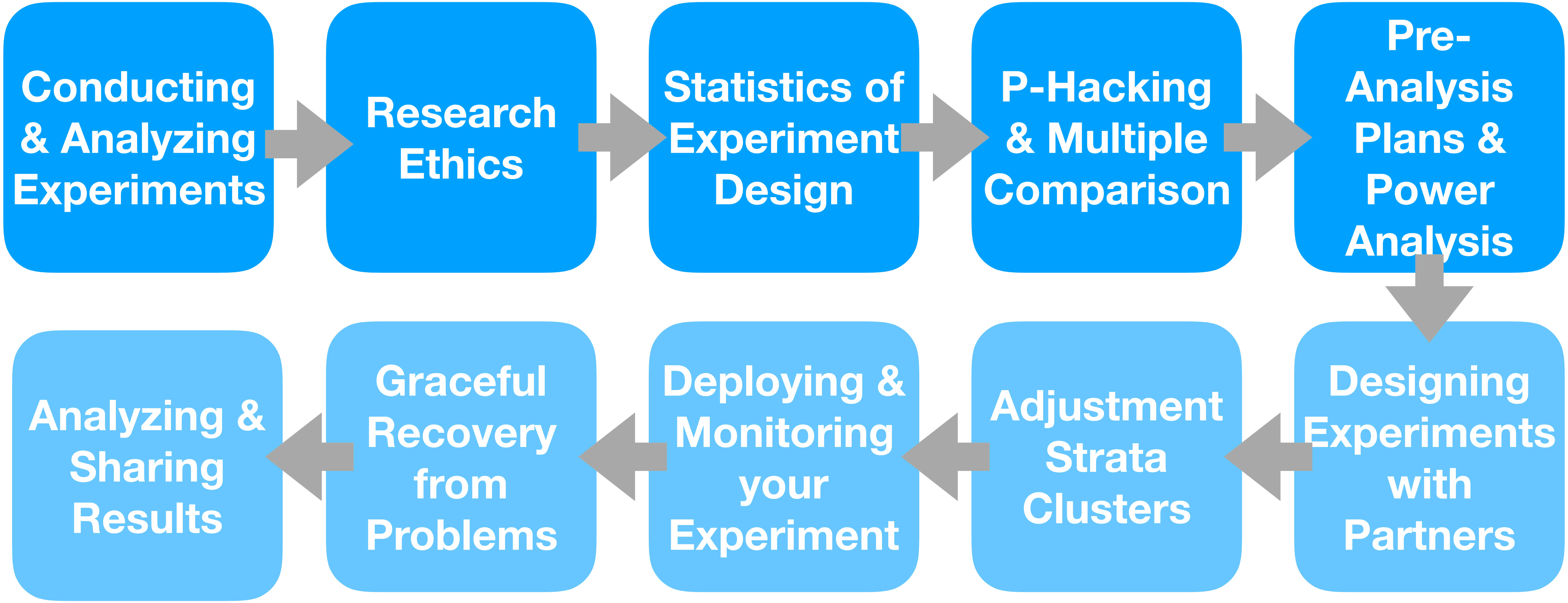
jmatias@princeton.edu

Department of Psychology

PRINCETON UNIVERSITY

CITP

mit media lab



Week 5: Power Analysis and Pre-Analysis Plans

Why Are Experiments So Rare?

It’s hard/expensive to deliver interventions

It’s hard/expensive to collect reliable measurements

They’re hard to design well

	Category/ Phase	Crawl	Walk	Run	Fly
Technical Evolution	**Technical focus of product dev. Activities**	**(1) Logging of signals** **(2) Work on data quality issues** **(3) Manual analysis of experiments** Transitioning from the debugging logs to a format that can be used for data-driven development.	**(1) Setting-up a reliable pipeline** **(2) Creation of simple metrics** Combining signals with analysis units. Four types of metrics are created: debug metrics (largest group), success metrics, guardrail metrics and data quality metrics.	**(1) Learning experiments** **(2) Comprehensive metrics** Creation of comprehensive set of metrics using the knowledge from the learning experiments.	**(1) Standardized process for metric design and evaluation, and OEC improvement**
	Experimentation platform complexity	**No experimentation platform** An initial experiment can be coded manually (ad-hoc).	**Platform is required** 3rd party platform can be used or internally developed. The following two features are required: • **Power Analysis** • **Pre-Experiment A/A testing**	**New platform features** The experimentation platform should be extended with the following features: • **Alerting** • **Control of carry-over effect** • **Experiment iteration support**	**Advanced platform features** The following features are needed: • **Interaction control and detection** • **Near real-time detection and automatic shutdown of harmful experiments** • **Institutional memory**
	Experimentation pervasiveness	**Generating management support** Experimenting with e.g. design options for which it's not a priori clear which one is better. To generate management support to move to the next stage.	**Experiment on individual feature level** Broadening the types of experiments run on a limited set of features (design to performance, from performance to infrastructure experiments)	**Expanding to (1) more features and (2) other products** Experiment on most new features and most products.	**Experiment with every minor change to portfolio** Experiment with any change on all products in the portfolio. Even to e.g. small bug fixes on feature level.

Fabijan, A., Dmitriev, P., Olsson, H. H., & Bosch, J. (2017, May). The evolution of continuous experimentation in software product development: from data to a data-driven organization at scale. In *Proceedings of the 39th International Conference on Software Engineering* (pp. 770-780). IEEE Press.

	Category/ Phase	Crawl	Walk	Run	Fly
Organizational Evolution	**Engineering team self-sufficiency**	**Limited understanding** External Data Scientist knowledge is needed in order to set-up, execute and analyse a controlled experiment.	**Creation and set-up of experiments** Creating the experiment (instrumentation, A/A testing, assigning traffic) is managed by the local Experiment Owners. Data scientists responsible for the platform supervise Experiment Owners and correct errors.	**Creation and execution of experiments** Includes monitoring for bad experiments, making ramp-up and shut-down decisions, designing and deploying experiment-specific metrics.	**Creation, execution and analyses of experiments** Scorecards showing the experiment results are intuitive for interpretation and conclusion making.
	Experimentation team organization	**Standalone** Fully centralized data science team. In product teams, however, no or very little data science skills. The standalone team needs to train the local product teams on experimentation. We introduce the role of Experiment Owner (EO).	**Embedded** Data science team that implemented the platform supports different product teams and their Experiment Owners. Product teams do not have their own data scientists that would analyse experiments independently.	**Partnership** Product teams hire their own data scientists that create a strong unity with business. Learning between the teams is limited to their communication.	**Partnership** Small data science teams in each of the product teams. Learnings from experiments are shared automatically across organization via the institutional memory features.
Business Evolution	**Overall Evaluation Criteria (OEC)**	OEC is **defined** for the first set of experiments with a few key signals that will help ground expectations and evaluation of the experiment results.	**OEC evolves** from a few key signals to a structured set of metrics consisting of Success, Guardrail and Data Quality metrics. Debug metrics are not a part of OEC.	OEC is **tailored** with the findings from the learning experiments. Single metric as a weighted combination of others is desired.	OEC is **stable**, only periodic changes allowed (e.g. 1 per year). It is also used for setting the performance goals for teams within the organization.

Figure 5. The “Experimentation Evolution Model”.

Fabijan, A., Dmitriev, P., Olsson, H. H., & Bosch, J. (2017, May). The evolution of continuous experimentation in software product development: from data to a data-driven organization at scale. In *Proceedings of the 39th International Conference on Software Engineering* (pp. 770-780). IEEE Press.

What we will cover today

Power Analysis

Pre-Registration

Reviewing Concepts about Experiments

Population: the group you're sampling from

Intervention (treatment)**:** the thing you plan to test

Unit of observation: the units you will be observing

Treatment unit: the units you will be treating

Arm: each condition that people will be assigned to

Reviewing Concepts about Experiments

Potential outcomes: the value of the outcome variable under each arm

Estimand: the “true effect” in the population of your experiment.

Assignment: the process (random) of assigning units to conditions

Reveal: the process of observing the outcomes

Estimator: the method for estimating the estimand

Potential Outcomes (ATE = 2.15)

ID (Units)	CONTROL	TREATMENT	Effect
1	0	1	1.592
2	6	8	2.486
3	3	4	1.599
4	0	2	2.179
5	1	2	1.531
6	1	3	2.507
7	3	5	2.282
8	6	8	2.283
9	9	11	2.992
10	9	11	2.088
11	0	1	1.041
12	0	3	3.261

Estimand

Reviewing Concepts about Experiments

Potential outcomes: the value of the outcome variable under ***each*** arm

Estimand: the “true effect” in the population of your experiment.

Assignment: the process (random) of assigning units to conditions

Reveal: assigning and observing outcomes

Estimator: the method for estimating the estimand

Reviewing Concepts about Experiments

Power Analysis: estimating the sample size needed to conduct an experiment

- Example: https://egap.shinyapps.io/power-app/

Experiment Diagnosis: simulating and diagnosing all aspects of the study design

- Example: http://declaredesign.org/

Declaring & Diagnosing a Design in R

Example code at:

https://github.com/natematias/SOC412/blob/master/lecture-code/Lecture%2010%20-%20Power%20Analysis.ipynb

Longer example at:

https://github.com/natematias/poweranalysis-onlinebehavior/blob/master/Choosing-Sample-and-Estimators.ipynb