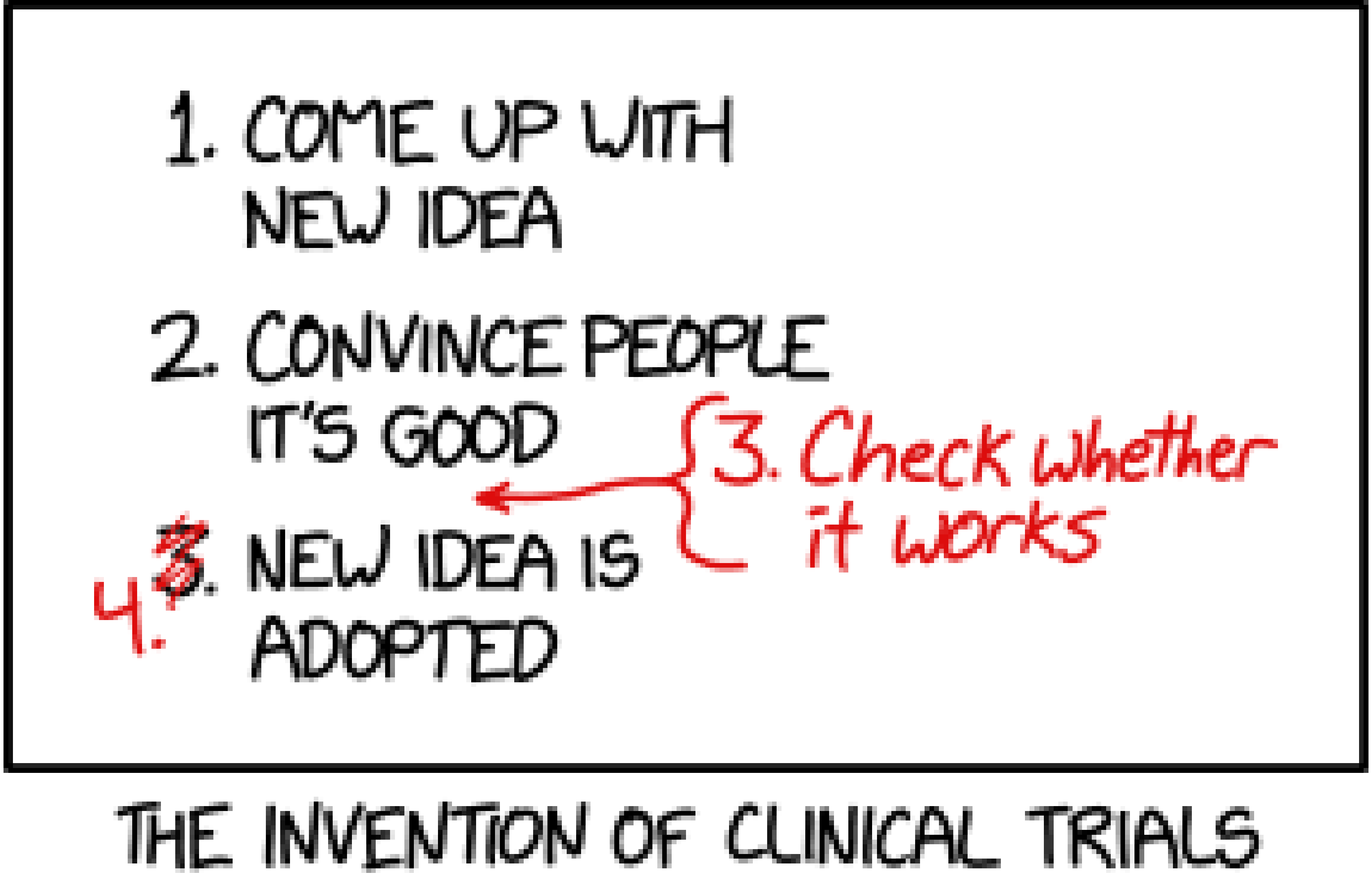
ORIE 5355

Lecture 12: Introduction to Causal Inference and Experimentation

Nikhil Garg

Announcements

Experimentation



[xkcd: Clinical Trials]

1 3 23

Susan Athey
@Susan_Athey

Replying to @causalinf

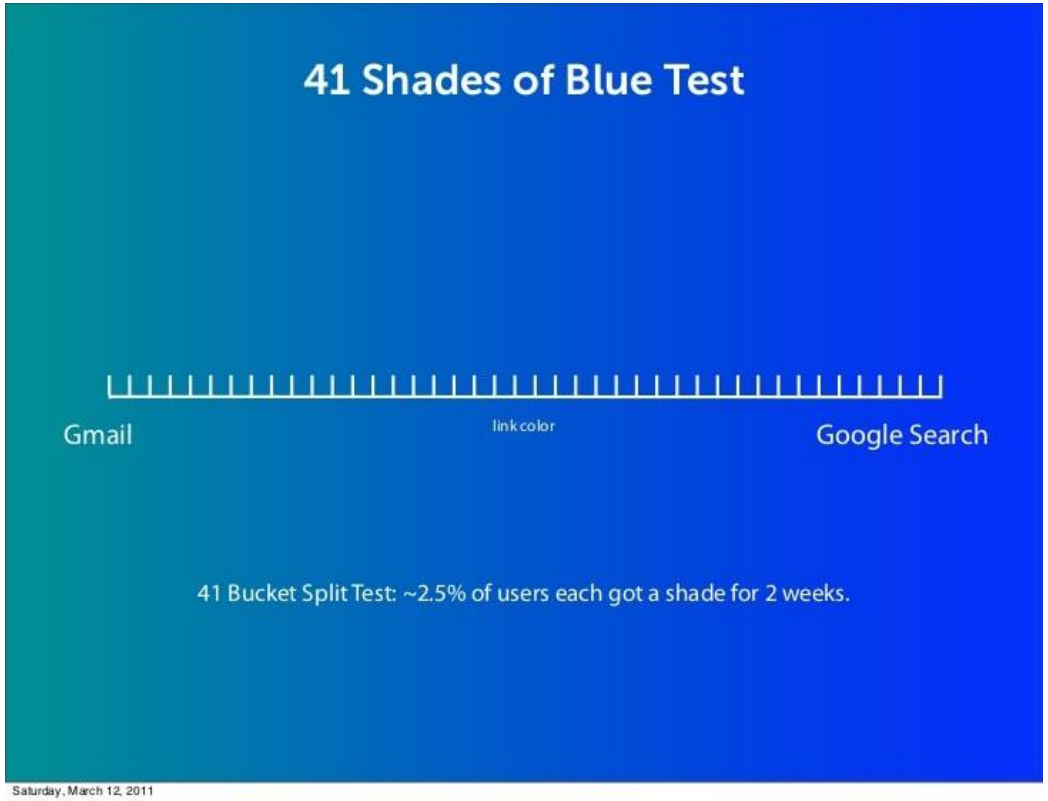
One point we make is that, perhaps surprisingly, the most commonly used empirical skills for economists in tech firms are those from causal inference/empirical applied micro/ experimental design, and people trained with those skills add a lot of value to tech firms.

10/16/18, 10:59 PM

19 Retweets 79 Likes

Companies use experiments everywhere

- Google/Microsoft/AirBnb/Uber/etc have hundreds or thousands of experiments live at any given time
- Everything from user interfaces to pricing and recommendation algorithms to headlines on news websites are tested
- Google infamously tested 41 shades of blue for the color of links in search results pages



[metrics-driven-design-by-joshua-porter-5-728.jpg (728×561) (slidesharecdn.com)]

Module overview

- Basics of A/B testing
  - Why experimentation?
  - Common mistakes in running and analyzing tests
- A/B testing in social networks and marketplaces
  - Interference between “test” and “control”
  - Experiments over time and space
  - Adaptive experimentation
- Guest lecture TBA
- Other topics in causal inference and experimentation
  - Causal inference with observational data
  - Experimentation culture in companies; making decisions with many experiments over time

Why experimentation?

Basics of causal inference

Ways to make decisions

- HiPPO – highest paid person's opinion
  - Most charismatic person's opinion
  - Consensus opinion
  - Majority opinion
- A structured 'logic-based' decision-making process
- Using past data to guess at what the effect of a product launch will be

All of these have their place, but sometimes they're not enough

Confounding: the challenge with observational data

- Suppose you’re a data scientist at the Department of Parks and Recreation
- You have data on which trees fell last year
- You want to answer, “Did we do preventative maintenance on the right trees?”
- You look at the data, and surprisingly find…

  …Trees that you did preventative maintenance on were *more likely* to fail than trees on which no work was performed!
- What happened?

Confounding, continued

- Now, you're a data scientist at a subscription-based company (for example, Netflix)
- You know that your company has been running a promotion: it identifies people who a model predicts are likely to fail, and then it sends them a coupon for a discount
- You crunch the data, and find…

  …That a higher percentage of the people who were sent a coupon quit, than the percentage of people who were not sent a coupon and quit.
- What happened?

Confounding, continued…

- Such correlations are everywhere
  - Daily death rates are higher in the hospital than they are outside of it
  - People who received ads to quit smoking last year are more likely to be smoking today, than people who didn’t receive such ads
- What’s going on?
  - Maintenance (likely) doesn’t cause tree failure
  - Hospitals don’t (usually) cause death
  - Coupons (likely, usually) don’t cause someone to quit a web-service
- Correlation doesn’t equal causation

Confounding: Correlation doesn’t equal causation

- In each case, we don’t know if our “intervention” *caused* the bad event to happen.
- More likely explanation: past decision-makers did a *good* job at identifying who needed help
  - Did maintenance on trees actually on verge of failing
  - Sent coupons to people actually more likely to quit
  - Sent actually sick people to the hospital
- …and the treatment helped, but wasn’t perfect
  - Prevented some trees from failing, but not all of them
  - Prevented some from quitting, but not all

Challenge with observational data

- The past data doesn’t (easily) tell us the counter-factual: “what would have happened if I *didn’t* do maintenance on the tree”
  - Also called the “potential outcome”
- There are (many) observational data analysis techniques to try to measure this counter-factual
  - The Nobel Prize in Economics in 2021 was awarded for developing them

…but, they’re hard to do

…and even harder to convince people that you’ve done them correctly

- In many systems, you can run experiments!

Why experiments help

- You want to answer: “would this customer have quit if I didn’t send them a coupon.”
- Unfortunately, you can’t BOTH (a) send a customer a coupon, AND (b) NOT send that same customer a coupon
- But you can: take two (otherwise identical) customers and send only one of them a coupon (but choose which one uniformly at randomly)
  - Do this for enough customers (send half a coupon), and then measure the fraction of people in each group that quit
- Randomization *breaks* the confounding (self-selection effect)

Other benefits of experimentation

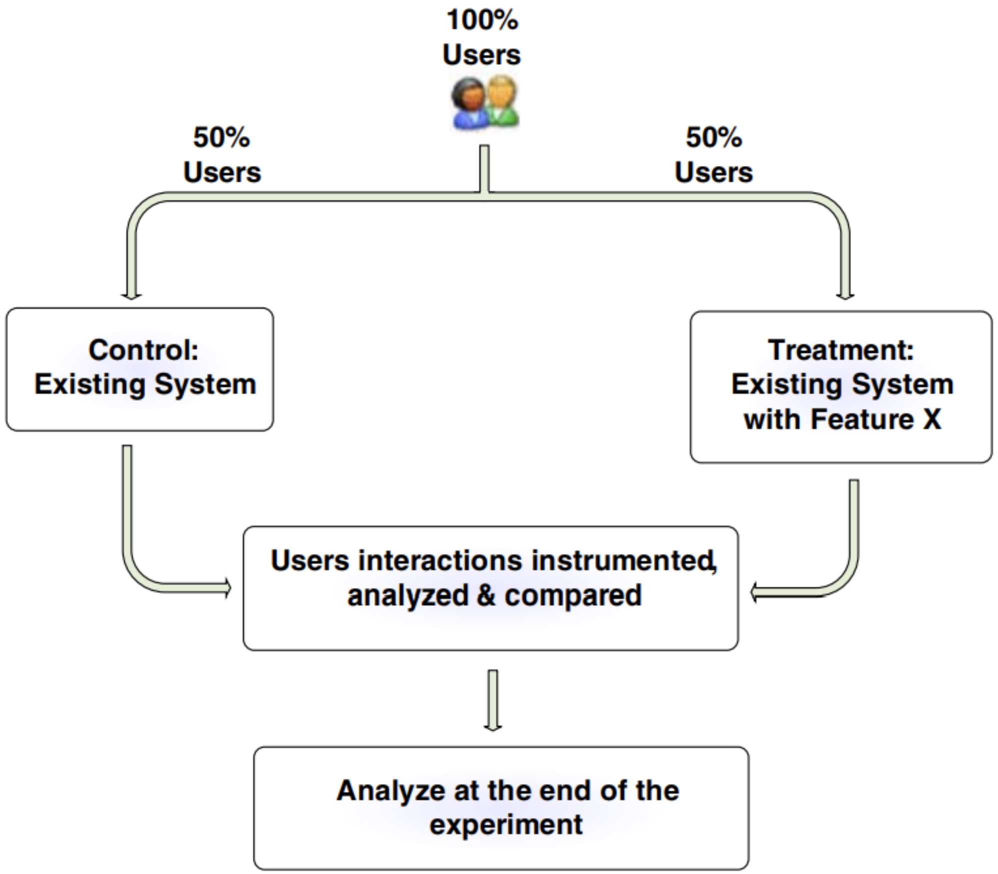
- You don't need to convince people that selection-bias didn't happen – you randomized in a way to make sure it didn't*
- At their most basic*, they're easy to run and analyze – don't need fancy statistics
- Often in new systems, you have no past data to even try to make your decision on
  - No one has used the new feature you want to decide whether to launch

* Often not true in people-centric systems, we'll discuss these in detail starting next week

Introduction to A/B testing

Basics of basic A/B testing

- Have an idea for a system change
- Give X% of your users the changed system, everyone else the old system
- Decide the *metric* you care about
- Check if your system improved the metric
- Launch your product if good things happened



[Source: Controlled experiments on the web: survey and practical guide]

An example

- Suppose you’re a click-baity news organization and have two headlines that you want to test
- Metric: % of people who click on the headline
- Give half the people who land on your website one headline, the other half the other headline
- Wait a day, and measure the % of people who clicked on each headline
- Run a statistical test to see if the difference between the % clicked is significant*
- Choose the better headline, and use that going forward

Some math

- Suppose we have Treatment $(X = 1)$ and Control $(X = 0)$

  Call them "arms" (treatment arm and control arm)
- Binary outcome $Y \in \{0, 1\}$
- Ground truth outcomes for treatment (Y_1) and control (Y_0)

  Y_0 : Fraction of people in control group who have outcome $Y = 1$

  True Average treatment effect: $Y_1 - Y_0$
- We give each arm to N people each; get sample measurements $\hat{Y}_1$ and $\hat{Y}_0$

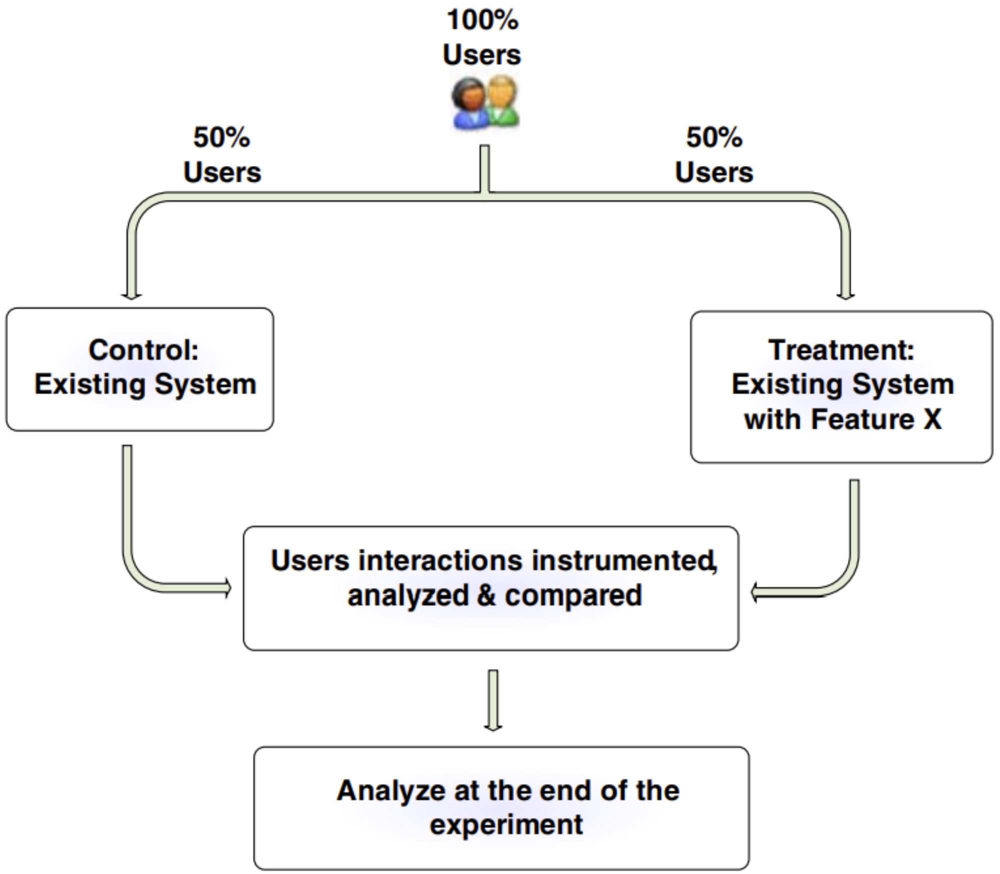
$$\hat{Y}_1 = \frac{\#\, \mathrm{I}[Y = 1 | X = 1]}{N}$$

- Average treatment effect estimate: $\hat{Y}_1 - \hat{Y}_0$
- Run a *hypothesis* test to see if the difference is *significant*
  - "Standard": difference is statistically significant if $\mathrm{p_{value}} < \alpha = 0.05$

    (wrong for decision-making)
  - [statsmodels.stats.proportion.proportions_ztest — statsmodels]
- Good post: [A/B testing: A step-by-step guide in Python | by Renato Fillinich | Towards Data Science]

Easy, right?

- Have an idea for a system change
- Give X% of your users the changed system, everyone else the old system
- Decide the *metric* you care about
- Check if your system changed anything
- Launch your product if good things happened



[Source: Controlled experiments on the web: survey and practical guide]

Key challenges in basic A/B testing

- What is the objective? How do you measure it? (*Can* you measure it?)
- What % of the users do you give the treatment to?
  - For how long?
  - What if you have thousands of potential treatments?

  You don’t want to waste time testing when one is obviously better
- How do you analyze the results?
- What is the bar for launching the product?

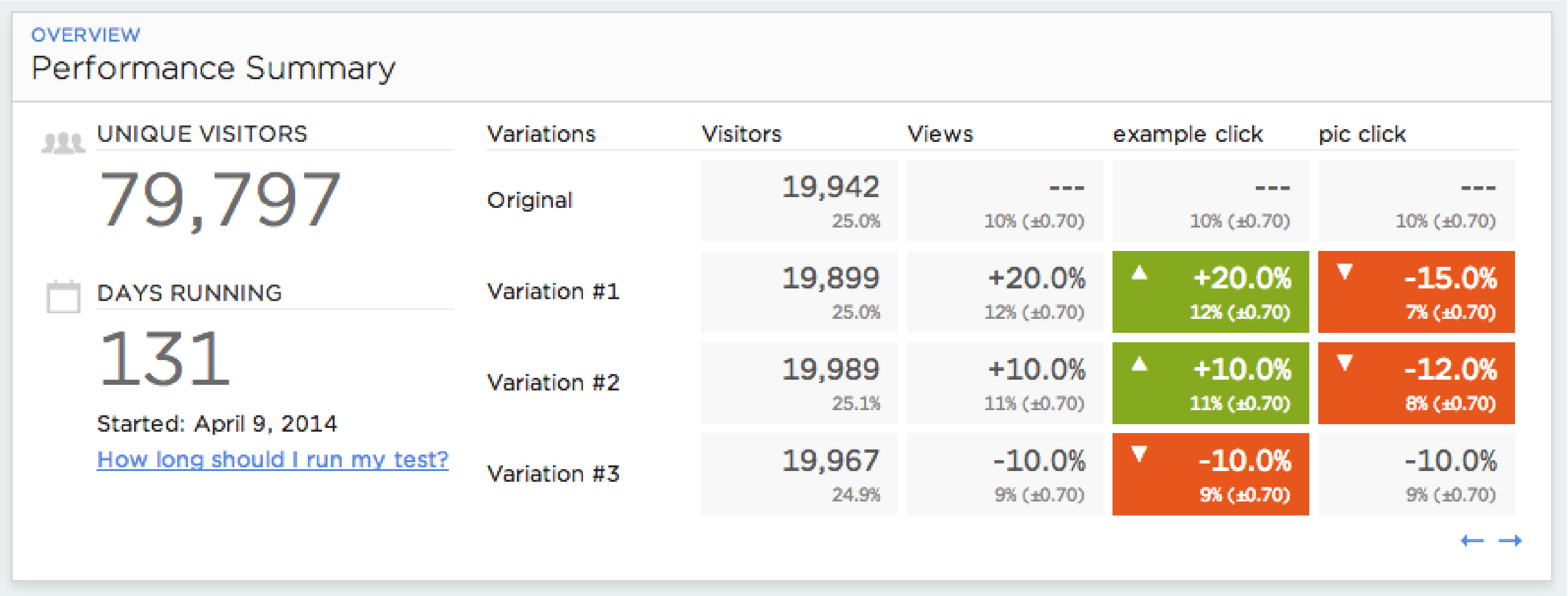
  How much better does a new feature have to be in order to launch?
- Next time: what if you have *interference* between treatment and control (standard in online marketplaces)

Peeking: a common mistake in running A/B tests in online marketplaces

Experiment Dashboards

In modern internet experiments, it’s easy to see experimental results *while they are happening*

Sample results dashboard:



OVERVIEW

Performance Summary

UNIQUE VISITORS

79,797

DAYS RUNNING

131

Started: April 9, 2014

How long should I run my test?

Variations	Visitors	Views	example click	pic click
Original	19,942 25.0%	--- 10% (±0.70)	--- 10% (±0.70)	--- 10% (±0.70)
Variation #1	19,899 25.0%	+20.0% 12% (±0.70)	+20.0% 12% (±0.70)	-15.0% 7% (±0.70)
Variation #2	19,989 25.1%	+10.0% 11% (±0.70)	+10.0% 11% (±0.70)	-12.0% 8% (±0.70)
Variation #3	19,967 24.9%	-10.0% 9% (±0.70)	-10.0% 9% (±0.70)	-10.0% 9% (±0.70)

[Image credit: Ramesh Johari (Stanford; also Optimizely at time of presentation)]

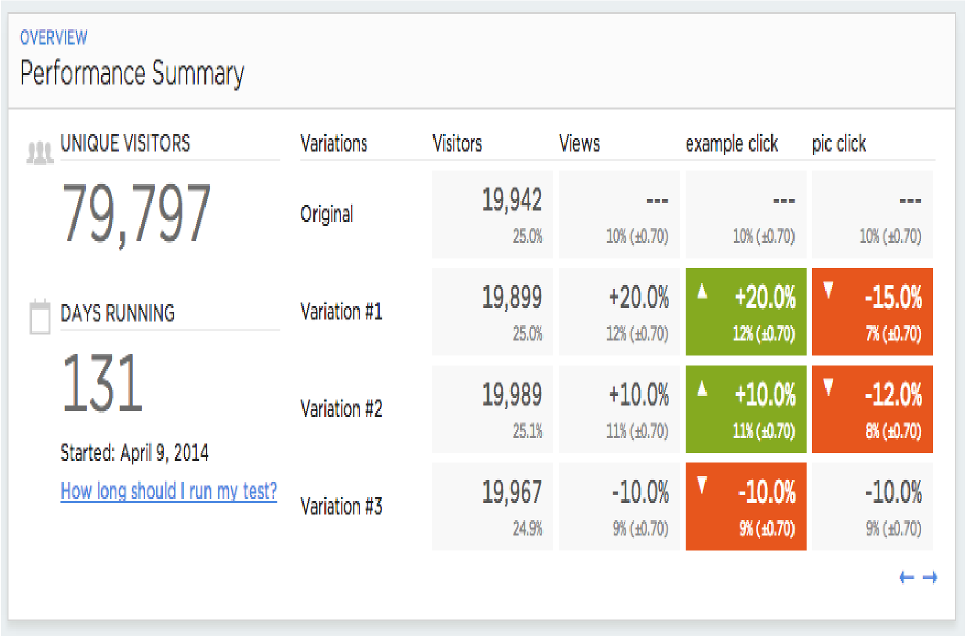
Peeking

In modern online setting, the approach I described above is wasteful

So you continuously monitor (stare at) the results dashboard.

> You rely on the dashboard to tell you when your results are significant.

- As soon as results are significant, you end the test and declare victory
- This is called adaptive sample size testing:
  - You adjust the test length in real-time, based on the data coming in.
  - If difference $\widehat{Y_1} - \widehat{Y}_0$ is *huge*, end the experiment early



[Slide credit: Ramesh Johari (Stanford; also Optimizely at time of presentation)]

Effect of peeking

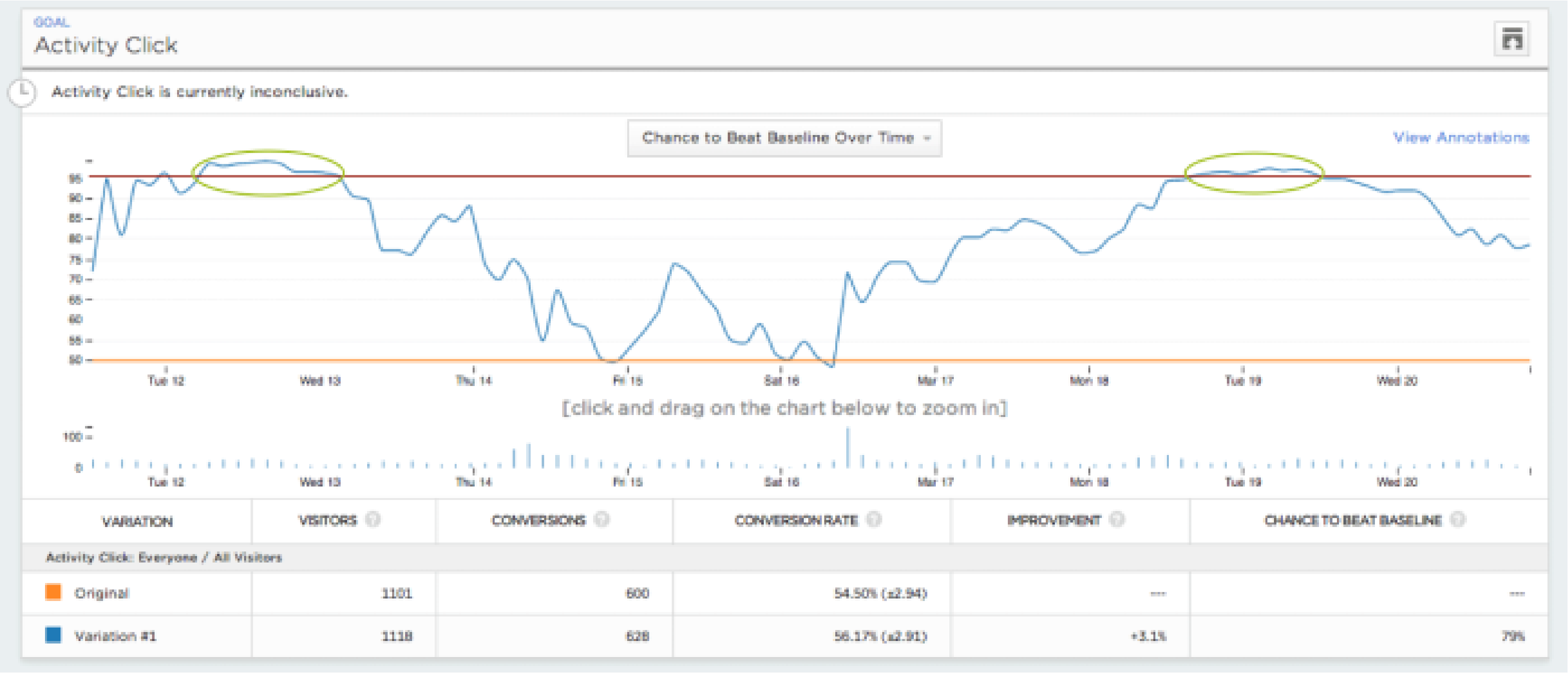
- Suppose 100 different individuals run A/A tests (same arm is treatment and control, so you know that $Y_1 - Y_0 = 0$)
- Each continuously monitors the dashboard, and waits for a significant result, i.e., p-value < 5% (up to a maximum of 10,000 visitors).
- *How many find a significant result and stop early?*

  Remember, $\alpha = 0.05$ means that if there is no true difference ($Y_1 - Y_0 = 0$), then 5% of the time you will falsely declare that $\hat{Y}_1 - \hat{Y}_0 \neq 0$ in a statistically significant way (false positive)
- Answer: **Over HALF!** find a significant result if they peek
- In A/B testing, "peeking" can dramatically inflate false positives.

[Slide credit: Ramesh Johari (Stanford; also Optimizely at time of presentation)]

What went wrong?

A sample run of an *A/A* test (graph is of p-values over time)



If you wait long enough, there is a high chance of an eventually inconclusive result looking "significant" along the way!

[Slide credit: Ramesh Johari (Stanford; also Optimizely at time of presentation)]

Peeking: what to do about it

You have two options

- Don’t peek: set a sample size N before the experiment starts, and don’t end early no matter how large the effect is
  - Easy to do the statistics as taught above; no danger of inflating false positives
  - Could be wasteful: what if the effect is clearly huge?

    Even medical trials have a procedure to end early if a drug clearly fantastic
- Peek, but do fancy statistics to make sure your p-values are valid
  - This is the approach Optimizely implemented on their dashboards
  - If you’re at a big company with an established experimental culture, they probably have a dashboard that does this

Other challenges

Key challenges in basic A/B testing

- What is the objective? How do you measure it? (*Can* you measure it?)
- What % of the users do you give the treatment to?
  - For how long?
  - What if you have thousands of potential treatments?

  You don't want to waste time testing when one is obviously better
- How do you analyze the results?
- What is the bar for launching the product?

  How much better does a new feature have to be in order to launch?
- Next time: what if you have *interference* between treatment and control (standard in online marketplaces)

Technical details we're not covering

- Power analyses: how do you decide how long to run your experiment?
- Various statistical tests to analyze outcomes
  - What if you had non-binary outcomes (or even continuous outcome)
  - What if you had *heterogeneous* treatment effects (different groups of people respond differently to the treatment)
  - How to "peek" at your results without messing up the statistical tests
- How to run and analyze *adaptive* experiments
  - If you have many arms, how to adapt sample sizes to arms over time