ORIE 5355

Lecture 14: Experimentation in marketplaces

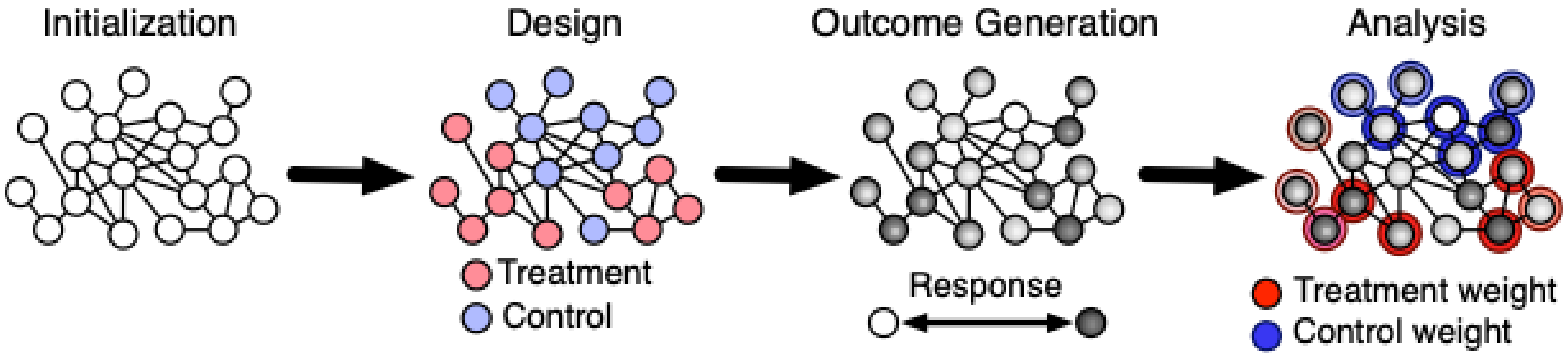
Nikhil Garg

Announcements

https://pollev.com/nikhilgarg713

- Class project is released; Part one due 11/20!
  - Note: 3 tabs in the google doc!
  - Fill out team form as soon as possible (linked in submission instruction tab)
  - Will start posting leaderboard for part 1 by end of week or early next week
    - Will also update github with "autograder" to check for bugs later this week
  - Will discuss project in class on Wednesday
- Remember LLM policy:
  - Allowed to use it for code with attribution; use git commits regularly to document process!
  - *Will NOT be allowed to use it for report.* We will post more instructions for report writing + submission soon.
  - May set up oral conversations with TAs about project strategy as part of grade

Reminder: Network Experimentation



- Initialization: An empirical graph or graph model
- Design: Graph cluster randomization
- Outcome generation: Observe behavior (or observe model)
- Analysis: Discerning effective treatment

Slide credit:
Johan
Ugander,
Stanford

New Zealand assignment

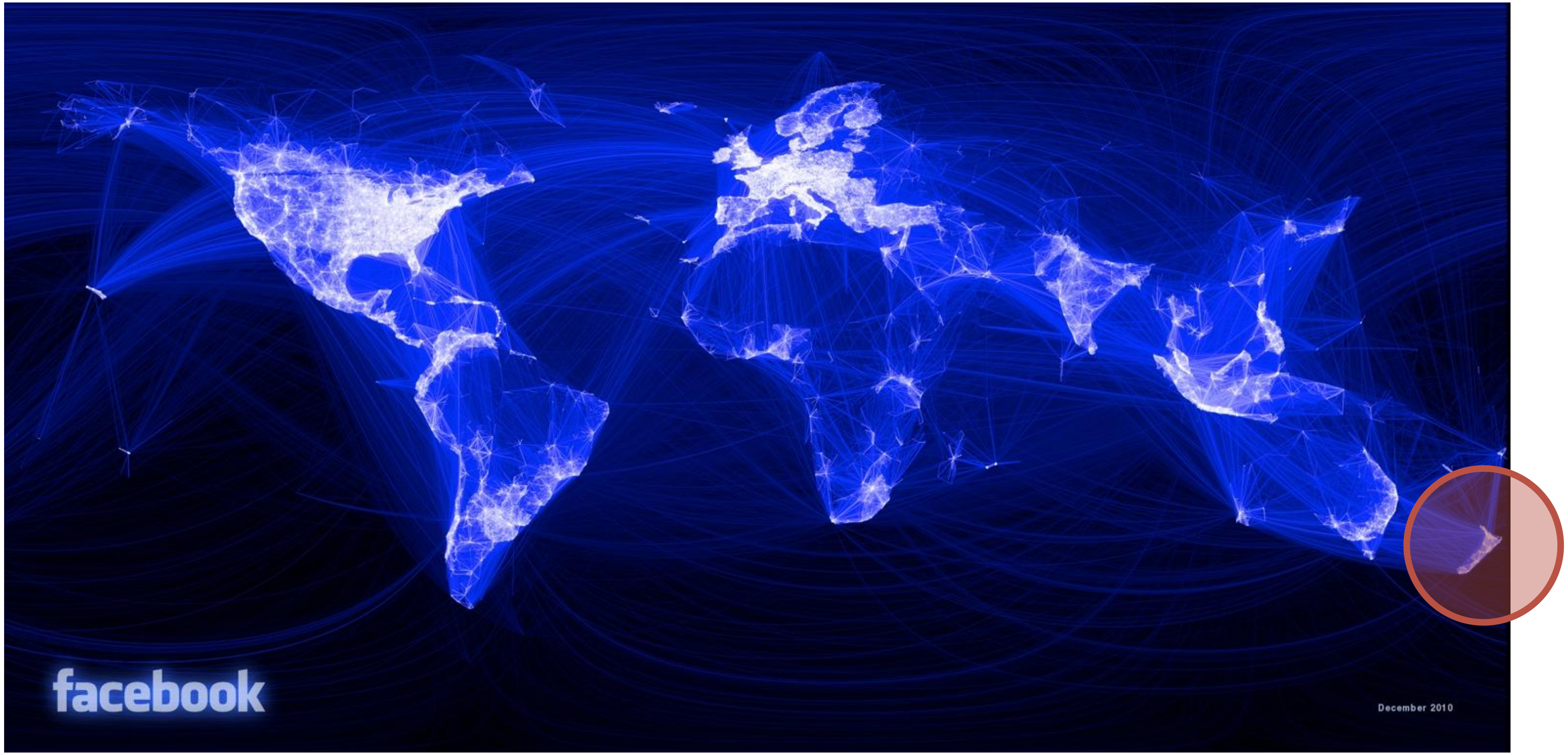


Image credit: Johan Ugander, Stanford

Idea: Pick a region of the graph that is densely connected with each other, but less connected with other parts of the graph. Put treatment in region, control everywhere else

“Graph cluster” randomization

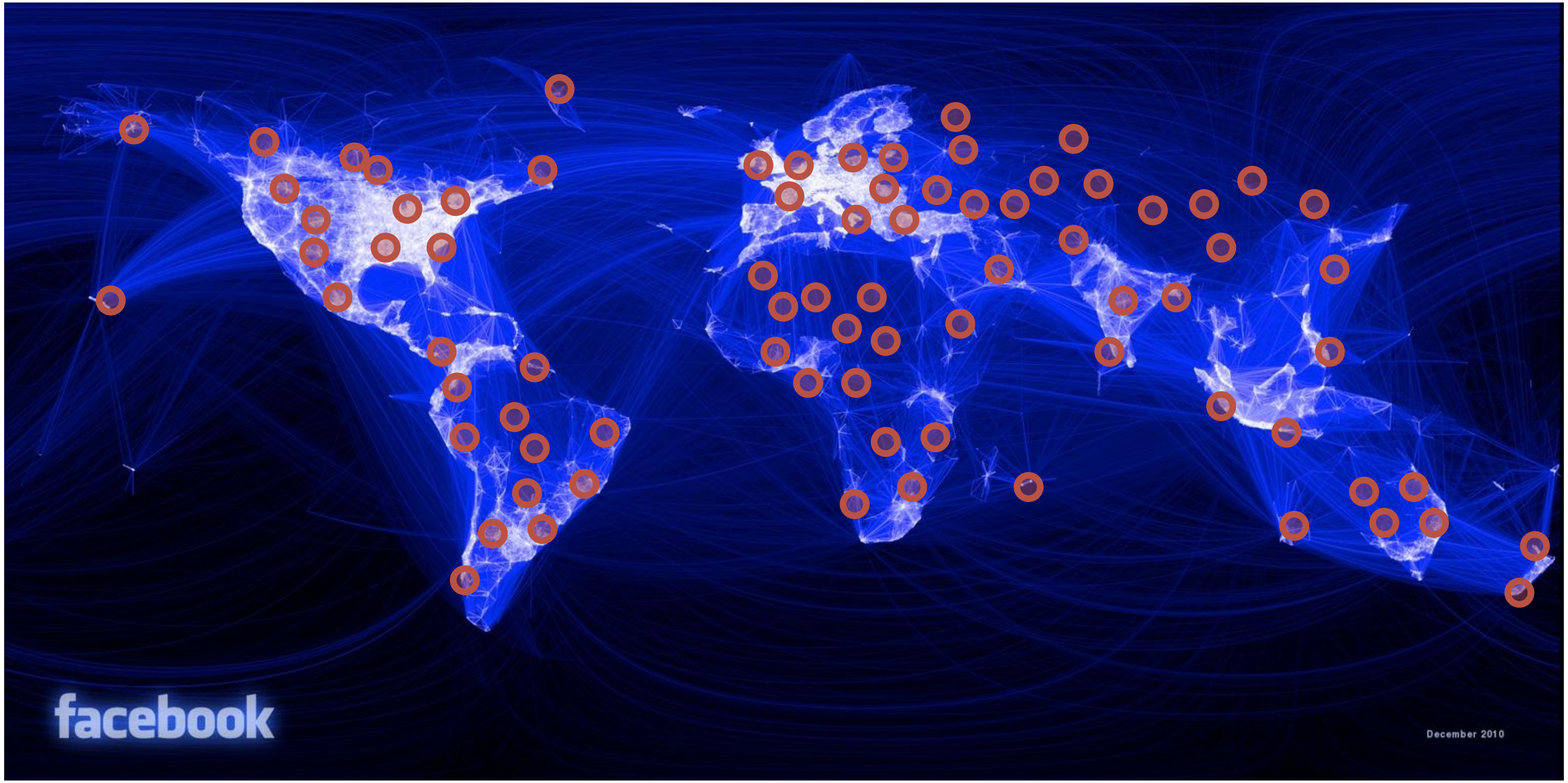
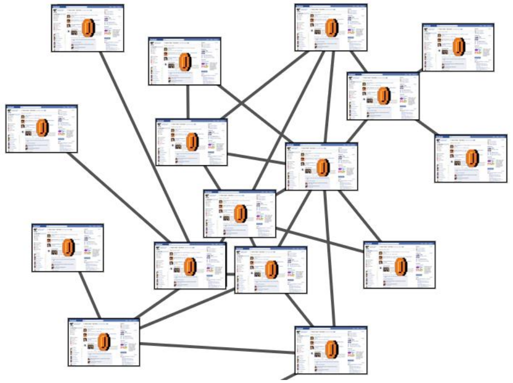


Image credit: Johan Ugander, Stanford

Idea: Algorithmically find many such regions, and then assign half of them treatment, and the other half control

General lesson: “unit” of randomization

- If you randomize at the “individual” level (each individual is its own “unit”), then treatment and control units can interfere with each other
- Solution is often to change the *unit of randomization*: randomize “clusters” instead of individuals
  - Hope: clusters are *close to independent*
  - If independent, experiment is *unbiased*
- Downside: Experiment “variance” goes down with sample size of experiment
  - Before: Sample size is *millions* (of users)
  - Now: Sample size is *hundreds* (of clusters)
- Same bias-variance trade-off we’ve seen before!



Interference in marketplaces

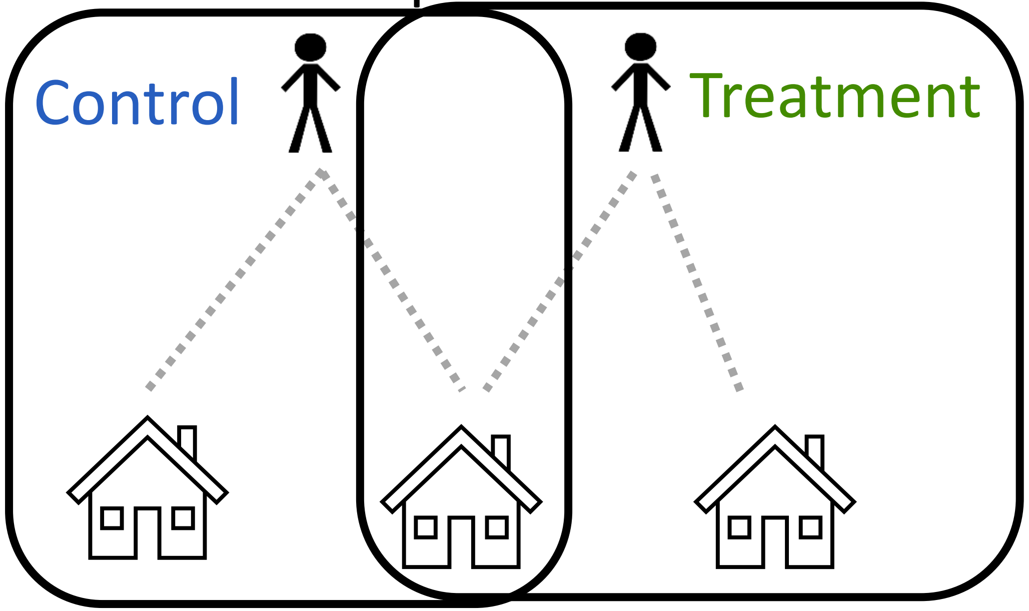
- Interference between treatment and control also arises in marketplaces
- In social networks: Interference because use case is *social* – me getting video messaging doesn't matter if none of my friends get it
- In markets, interference rises from *competition and capacity constraints*
- If I make half the products cheaper, customers will *increase* their purchases of the cheaper items…why?
  - Go from not purchasing at all, to buying the now cheaper item (**new customer**)
  - *Decrease* their purchases of the more expensive items (**cannibalization**)
- *Not* representative of what would happen if I make *all* my products cheaper

  Cannibalization effect would not occur; only attraction of new customers
- Today: experimentation in marketplaces under interference

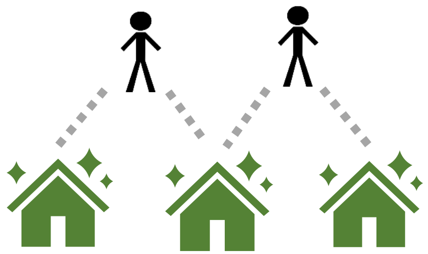
Slide credit:
Hannah Li,
Columbia

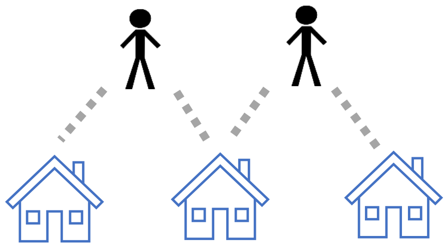
Competition $\Longrightarrow$ *Interference* $\Longrightarrow$ *Bias*

Customer-side experiment



Global Treatment Effect (GTE) = Global Treatment – Global Control

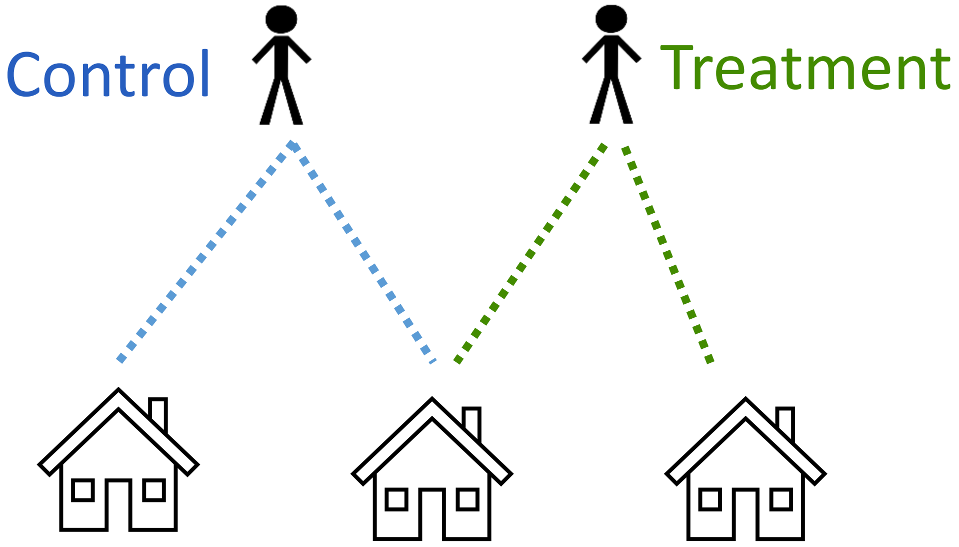




Slide credit:
Hannah Li,
Columbia

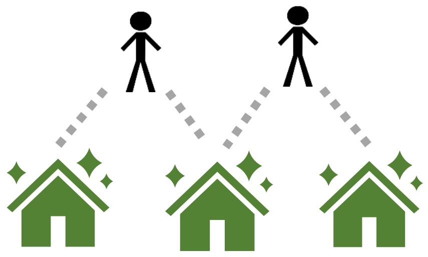
Competition ⟹ Interference ⟹ Bias

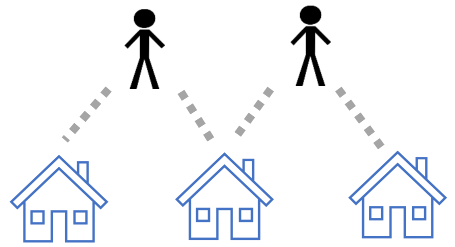
Customer-side experiment



- Suppose feature makes treatment customer more likely to book than control
- Treatment customer books listing
- Reduces supply for control customer
- This instance: overestimate GTE

Global Treatment Effect (GTE) = Global Treatment – Global Control

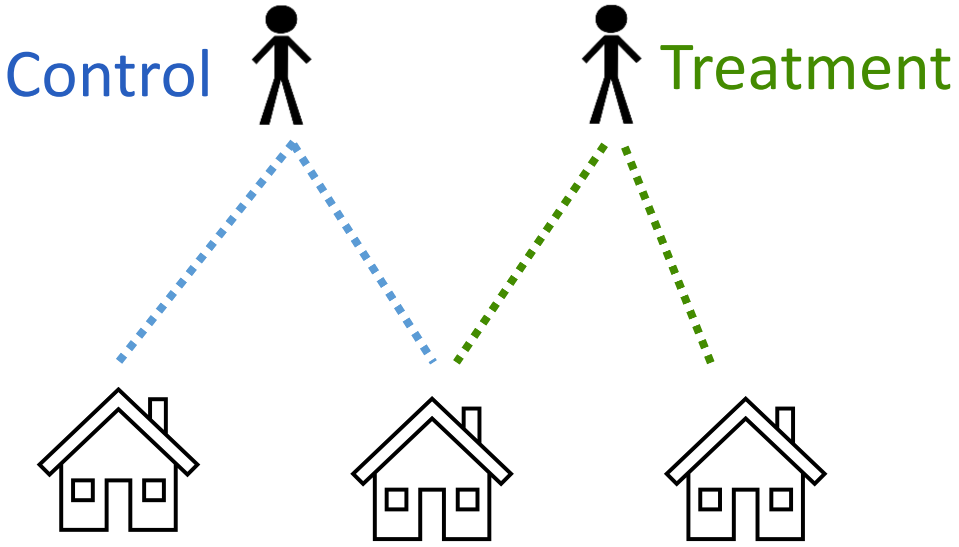




Slide credit:
Hannah Li,
Columbia

Competition ⟹ Interference ⟹ Bias

Customer-side experiment



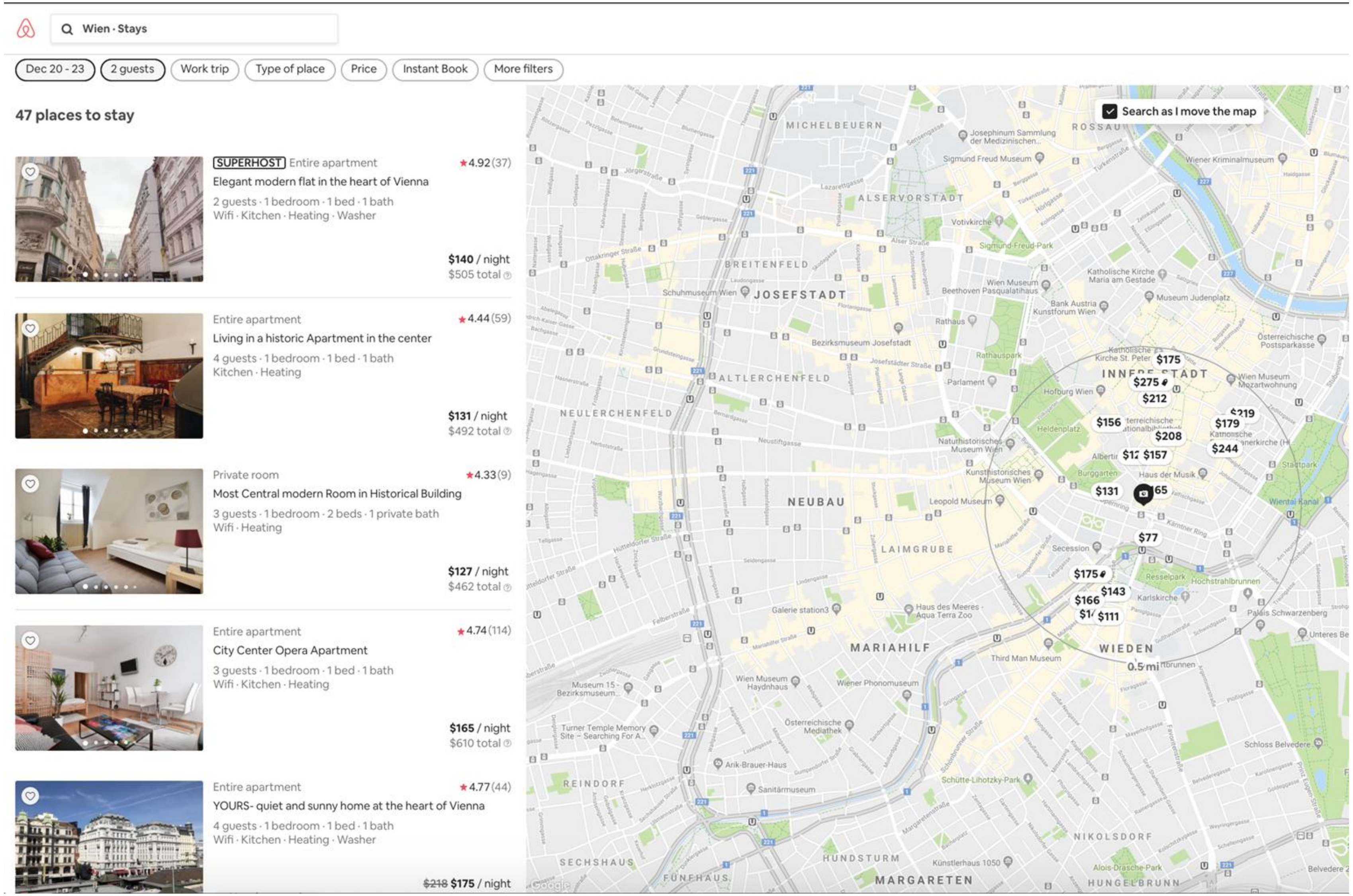
- Suppose feature makes treatment customer more likely to book than control
- Treatment customer books listing
- Reduces supply for control customer
- This instance: overestimate GTE

More generally:

- Change a customer’s booking prob. ⇒ change supply for other customers
- Change a listing’s display ⇒ make other listing relatively more/less attractive

Graph cluster randomization in marketplaces

Example 1: price change experiment on Airbnb



Wien · Stays

Dec 20 - 23 | 2 guests | Work trip | Type of place | Price | Instant Book | More filters

47 places to stay

SUPERHOST Entire apartment ★4.92 (37)
Elegant modern flat in the heart of Vienna
2 guests · 1 bedroom · 1 bed · 1 bath
Wifi · Kitchen · Heating · Washer
$140 / night
$505 total

Entire apartment ★4.44 (59)
Living in a historic Apartment in the center
4 guests · 1 bedroom · 1 bed · 1 bath
Kitchen · Heating
$131 / night
$492 total

Private room ★4.33 (9)
Most Central modern Room in Historical Building
3 guests · 1 bedroom · 2 beds · 1 private bath
Wifi · Heating
$127 / night
$462 total

Entire apartment ★4.74 (114)
City Center Opera Apartment
3 guests · 1 bedroom · 1 bed · 1 bath
Wifi · Kitchen · Heating
$165 / night
$610 total

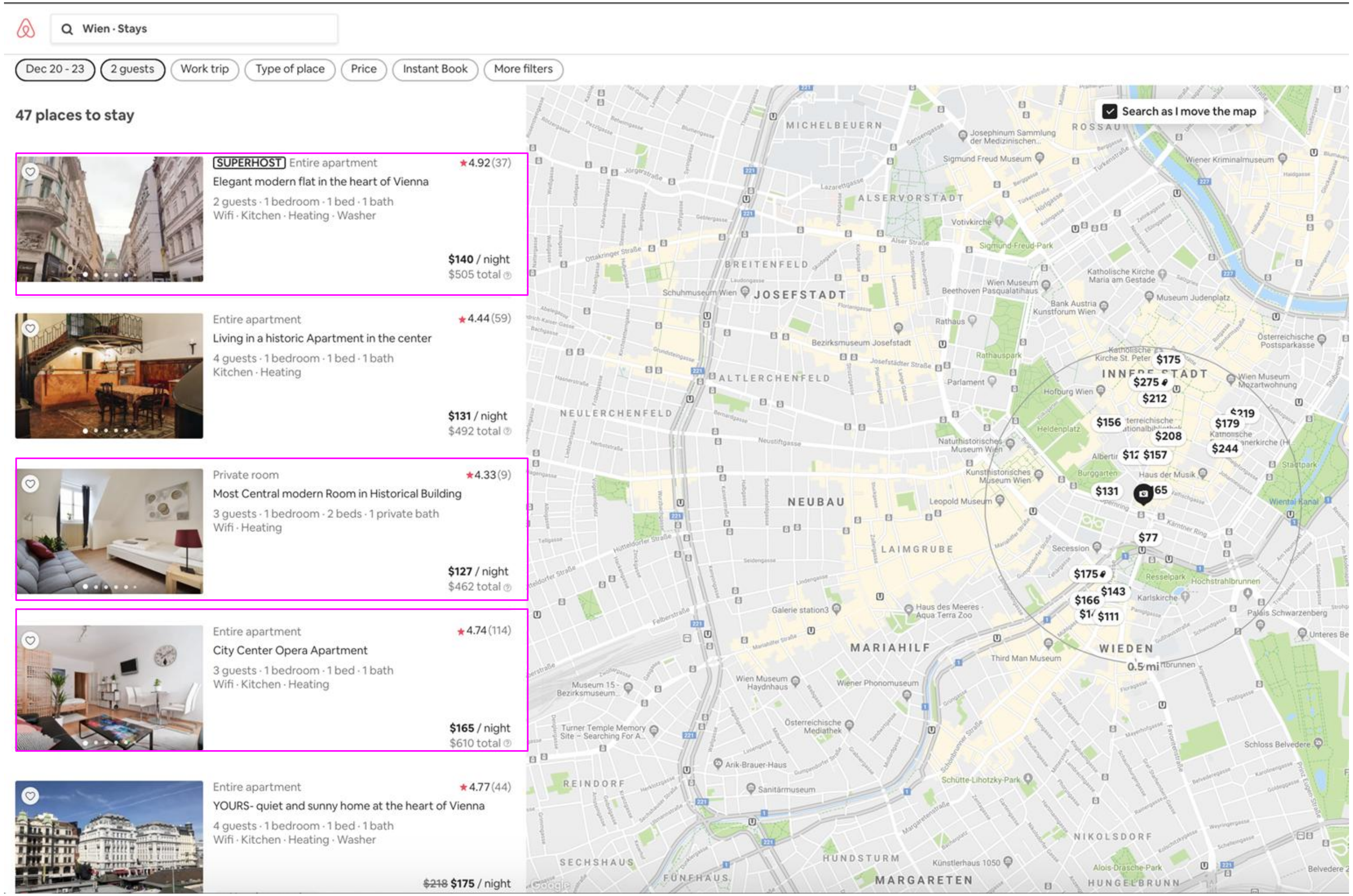
Entire apartment ★4.77 (44)
YOURS- quiet and sunny home at the heart of Vienna
4 guests · 1 bedroom · 1 bed · 1 bath
Wifi · Kitchen · Heating · Washer
~~$218~~ $175 / night

☑ Search as I move the map

Slide credit:
Dave Holtz,
UC Berkeley

Example 1: price change experiment on Airbnb

If lower fees on half of the listings, bookings for those listings ↑ 3% ☺



Wien · Stays

Dec 20 - 23 | 2 guests | Work trip | Type of place | Price | Instant Book | More filters

47 places to stay

SUPERHOST Entire apartment ★4.92 (37)
Elegant modern flat in the heart of Vienna
2 guests · 1 bedroom · 1 bed · 1 bath
Wifi · Kitchen · Heating · Washer
$140 / night
$505 total

Entire apartment ★4.44 (59)
Living in a historic Apartment in the center
4 guests · 1 bedroom · 1 bed · 1 bath
Kitchen · Heating
$131 / night
$492 total

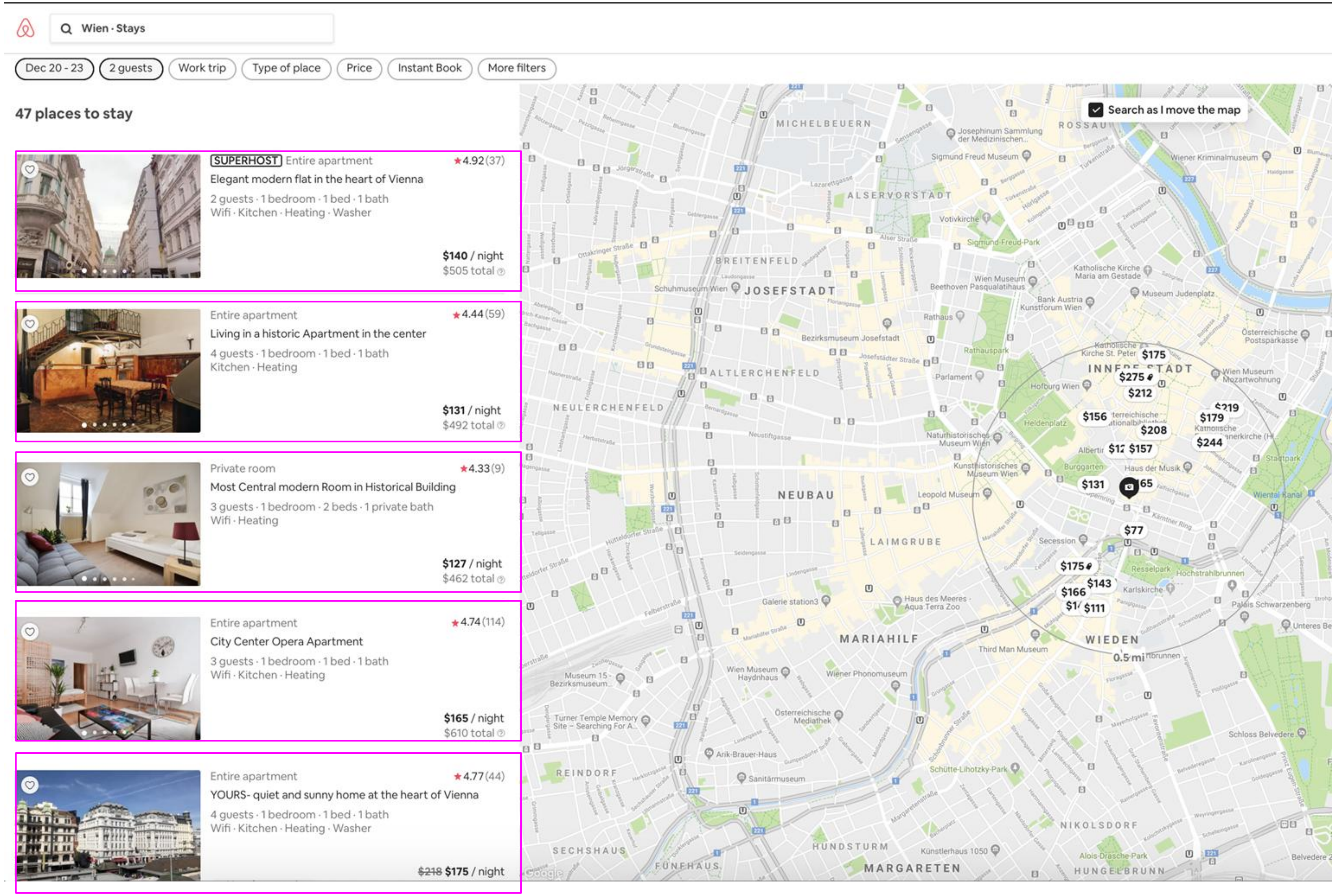
Private room ★4.33 (9)
Most Central modern Room in Historical Building
3 guests · 1 bedroom · 2 beds · 1 private bath
Wifi · Heating
$127 / night
$462 total

Entire apartment ★4.74 (114)
City Center Opera Apartment
3 guests · 1 bedroom · 1 bed · 1 bath
Wifi · Kitchen · Heating
$165 / night
$610 total

Entire apartment ★4.77 (44)
YOURS- quiet and sunny home at the heart of Vienna
4 guests · 1 bedroom · 1 bed · 1 bath
Wifi · Kitchen · Heating · Washer
~~$218~~ $175 / night

Slide credit: Dave Holtz, UC Berkeley

Example 1: price change experiment on Airbnb



Wien · Stays

Dec 20 - 23 | 2 guests | Work trip | Type of place | Price | Instant Book | More filters

47 places to stay

SUPERHOST Entire apartment ★4.92 (37)
Elegant modern flat in the heart of Vienna
2 guests · 1 bedroom · 1 bed · 1 bath
Wifi · Kitchen · Heating · Washer
$140 / night
$505 total

Entire apartment ★4.44 (59)
Living in a historic Apartment in the center
4 guests · 1 bedroom · 1 bed · 1 bath
Kitchen · Heating
$131 / night
$492 total

Private room ★4.33 (9)
Most Central modern Room in Historical Building
3 guests · 1 bedroom · 2 beds · 1 private bath
Wifi · Heating
$127 / night
$462 total

Entire apartment ★4.74 (114)
City Center Opera Apartment
3 guests · 1 bed · 1 bath
Wifi · Kitchen · Heating
$165 / night
$610 total

Entire apartment ★4.77 (44)
YOURS- quiet and sunny home at the heart of Vienna
4 guests · 1 bedroom · 1 bed · 1 bath
Wifi · Kitchen · Heating · Washer
~~$218~~ $175 / night

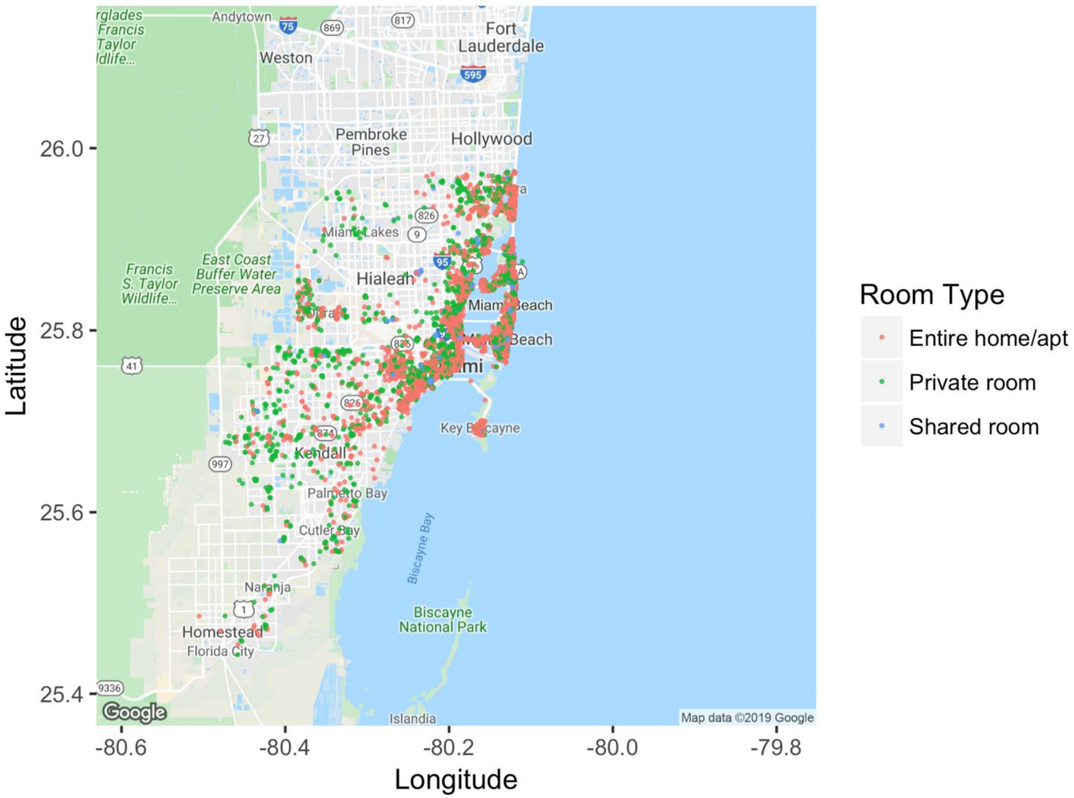
☑ Search as I move the map

If lower fees on all the listings, Overall bookings flat 🙁

Slide credit: Dave Holtz, UC Berkeley

Slide credit:
Dave Holtz,
UC Berkeley

Approach 1: transform the marketplace into a network



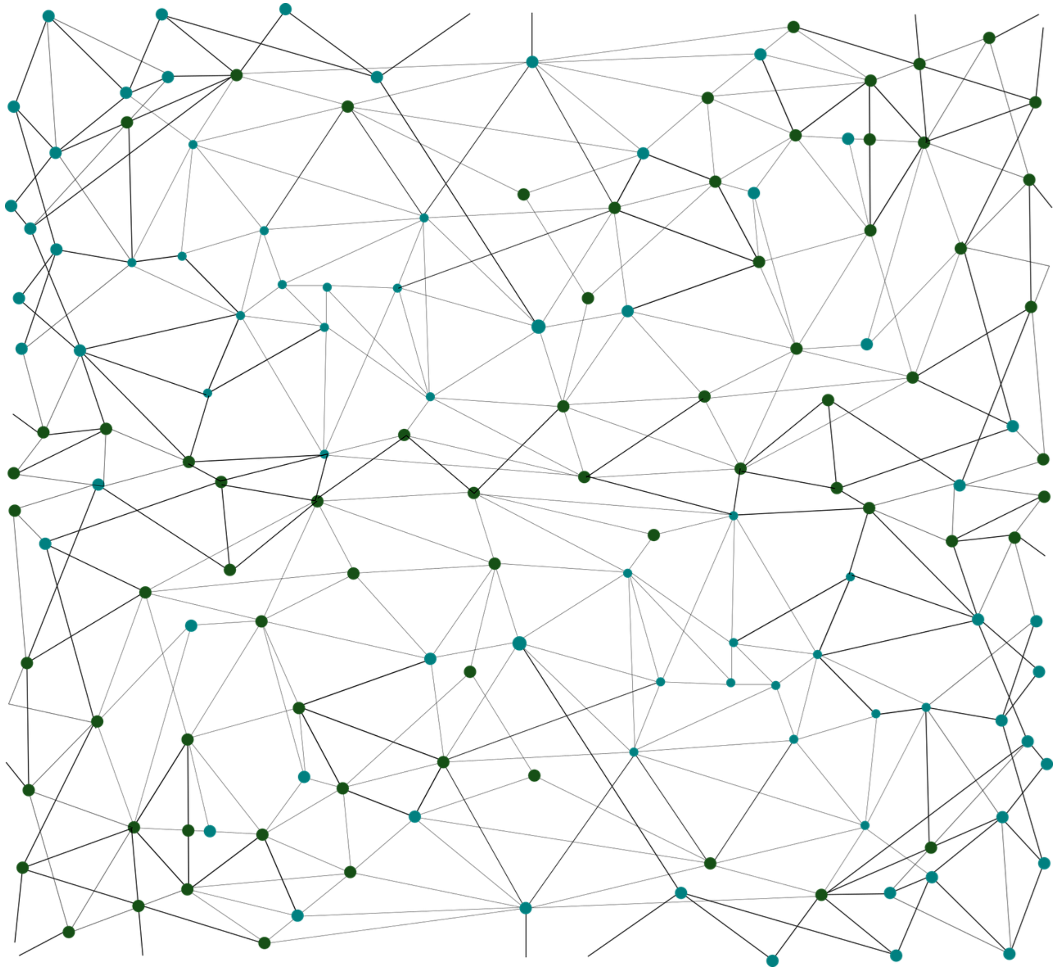


Image source

Network experiment designs + analysis techniques

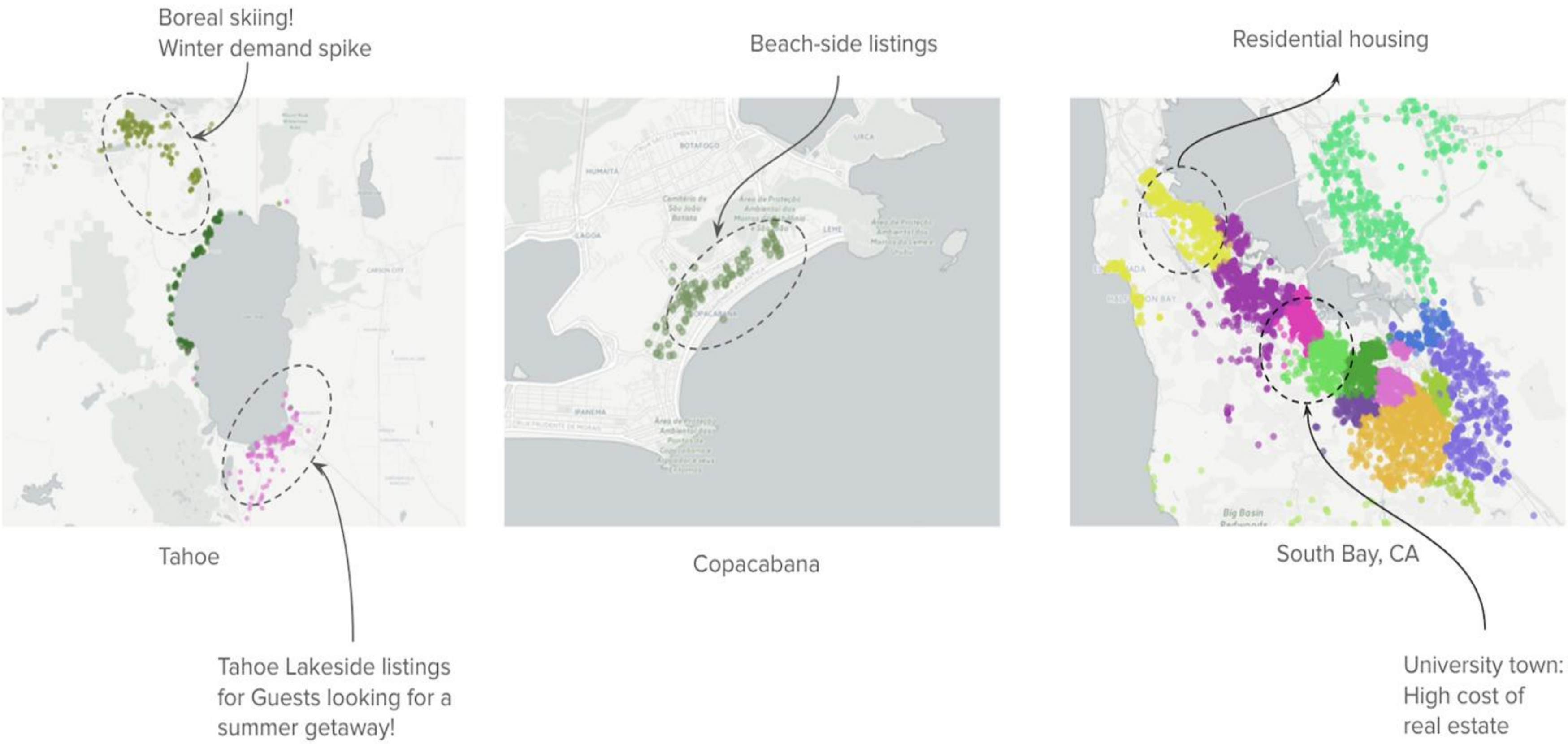
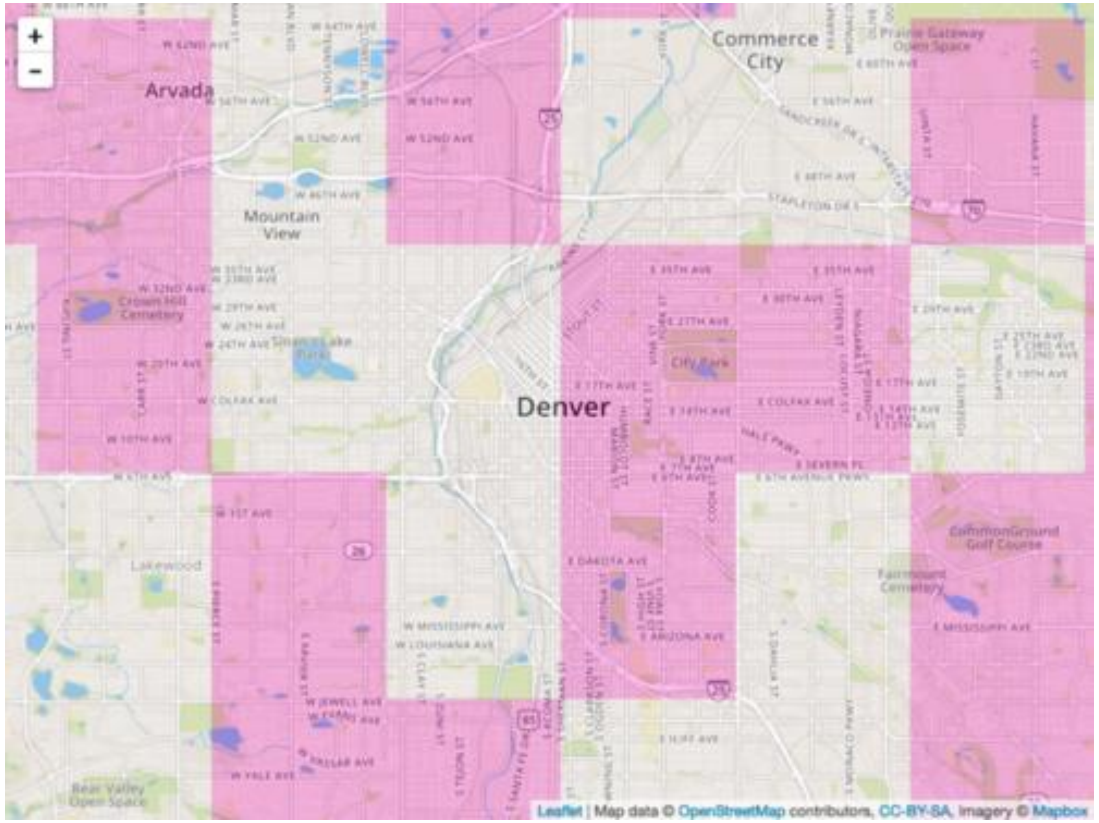
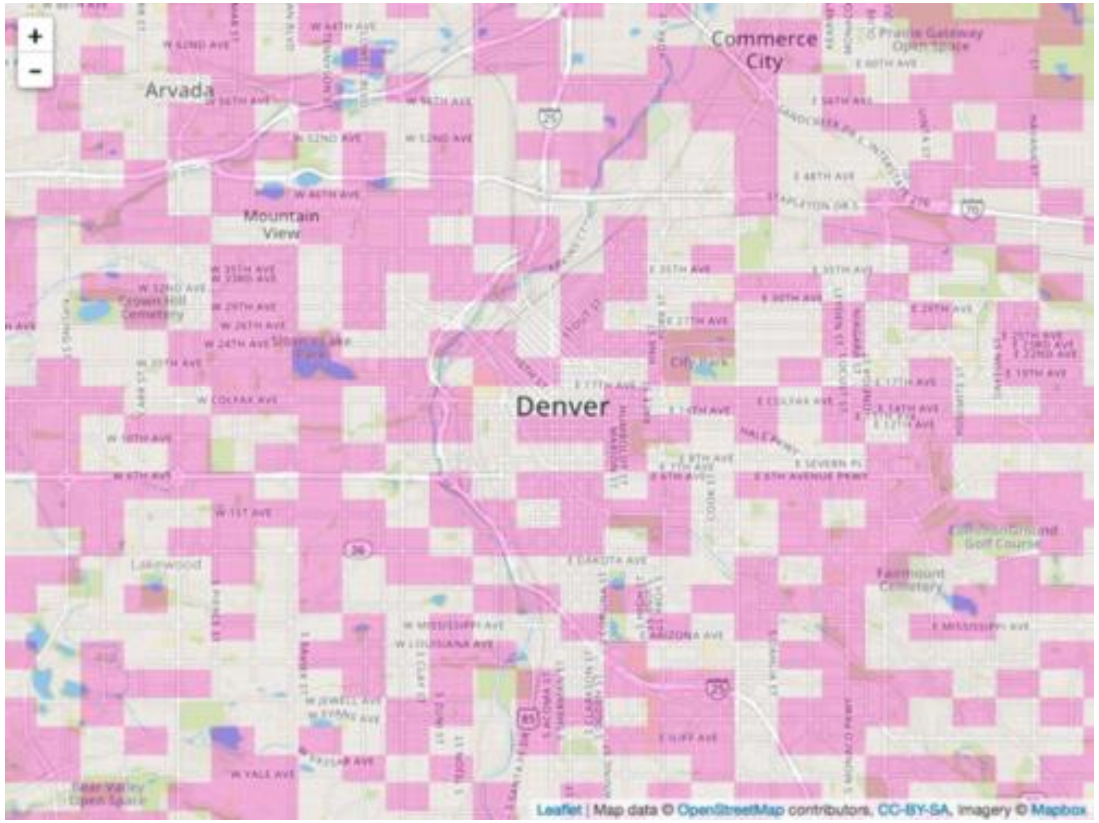

Boreal skiing!
Winter demand spike
Beach-side listings
Residential housing
Tahoe
Copacabana
South Bay, CA
Tahoe Lakeside listings for Guests looking for a summer getaway!
University town: High cost of real estate

Image credit: Dave Holtz, UC Berkeley

- Now, listings are connected if they tend to be *substitutes*
- Much more complicated to learn the network structure
- Once have network structure, use cluster randomization techniques
- Challenge: “graph” might be too interconnected

Spatial randomization in ride-hailing





[Experimentation in a Ridesharing Marketplace | by Nicholas Chamandy | Lyft Engineering]()

Beyond spatial (and graph cluster) randomization: experimenting over time

Switchbacks

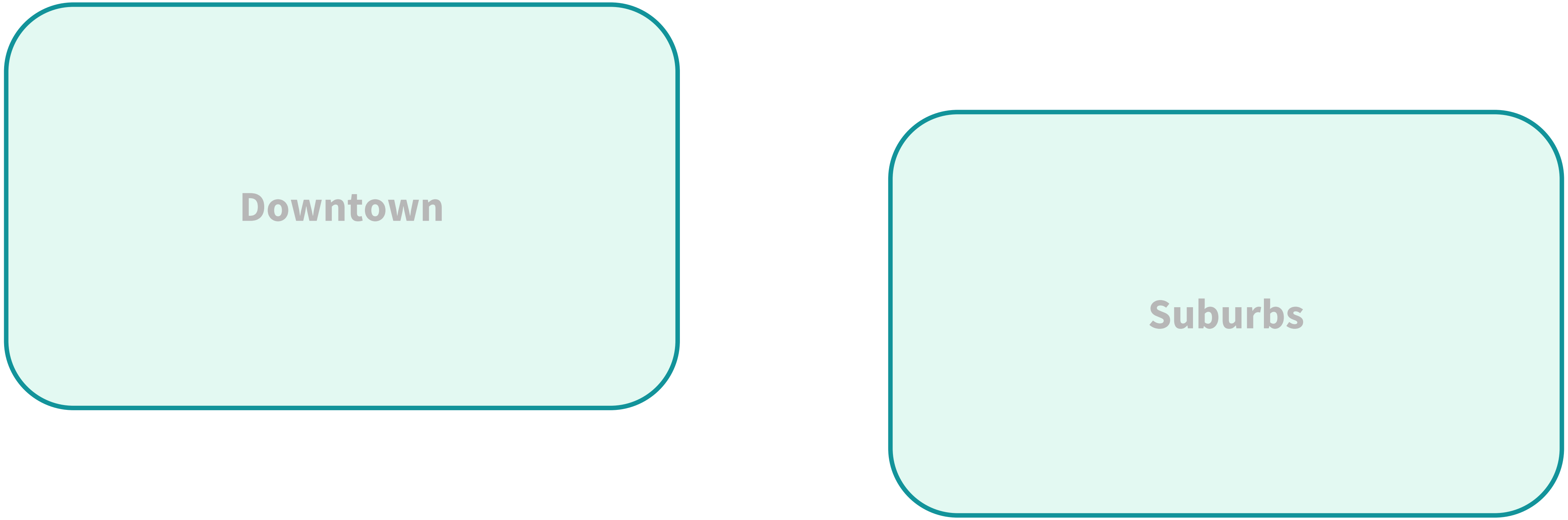
Why is cluster randomization not enough?

- Often difficult to define the clusters
- There legitimately might not be enough “clusters” that don’t interfere with one another
  - In AirBnB, rentals near Disney Land (in Los Angeles) might compete with rentals near Disney World (in Orlando)
  - In ride-hailing, a driver in a suburb could be instead choose to drive in the city

Driver Positioning Example

Slide credit:
Uber MX team

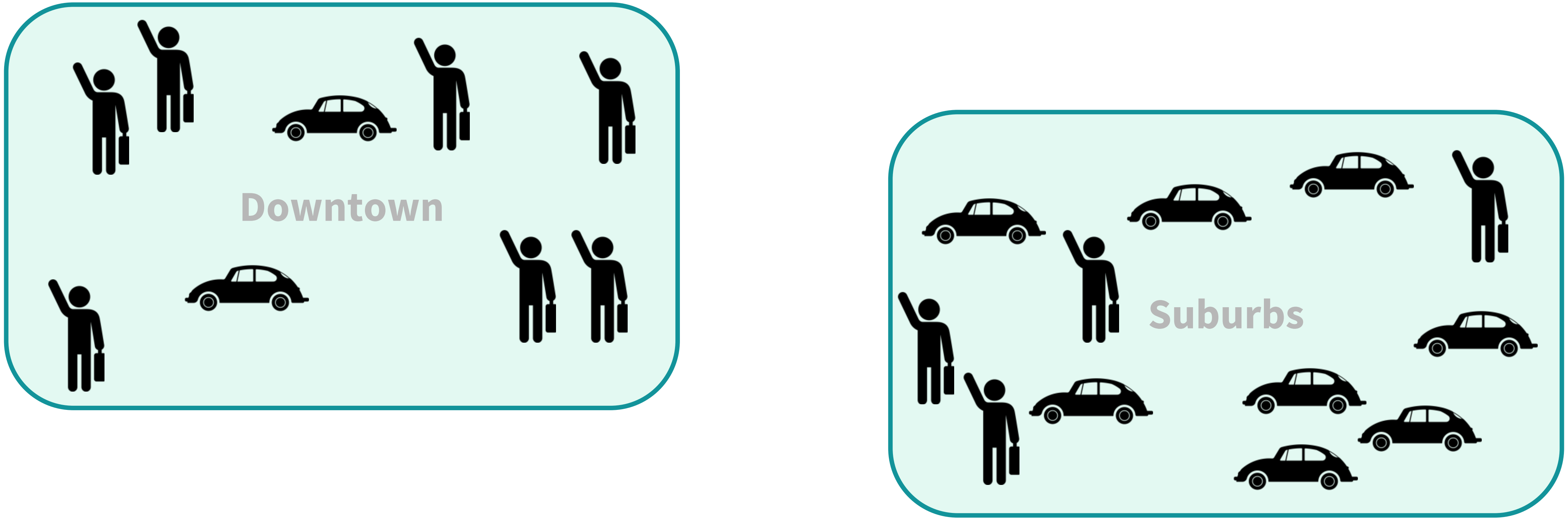
Suppose our city has two geos: downtown and the suburbs



Driver Positioning Example

Slide credit:
Uber MX team

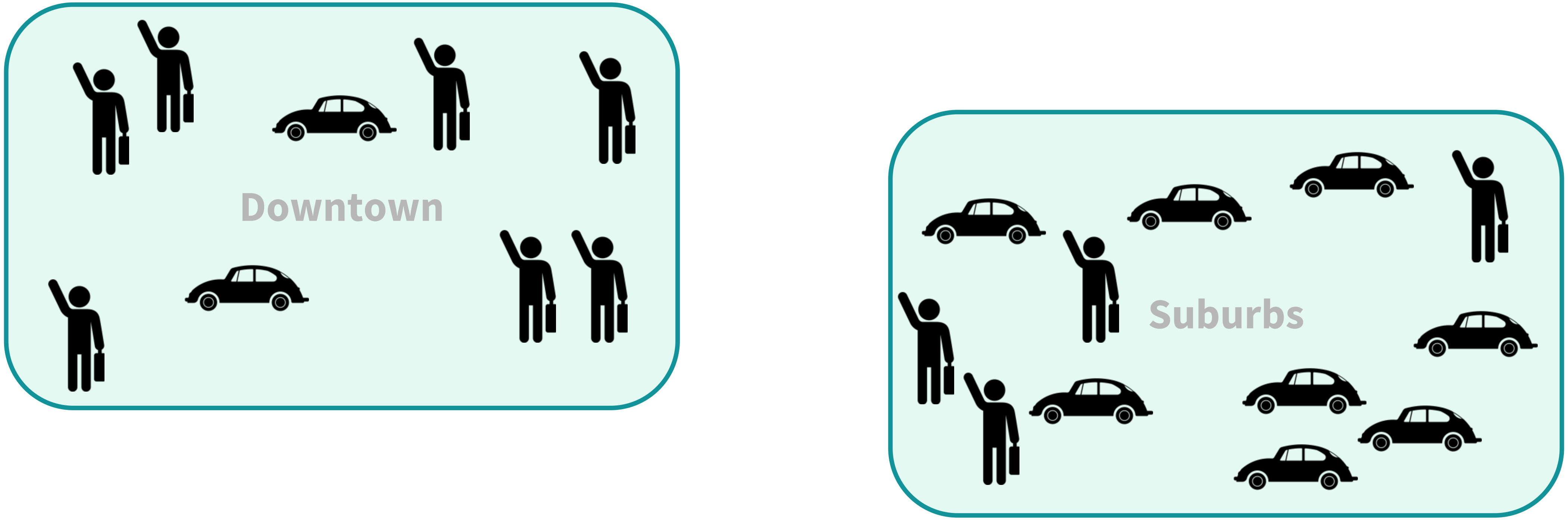
We notice that we are chronically undersupplied in downtown and oversupplied in the suburbs. Uber is concerned that this adversely impacts driver earnings.



Slide credit:
Uber MX team

Driver Positioning Example

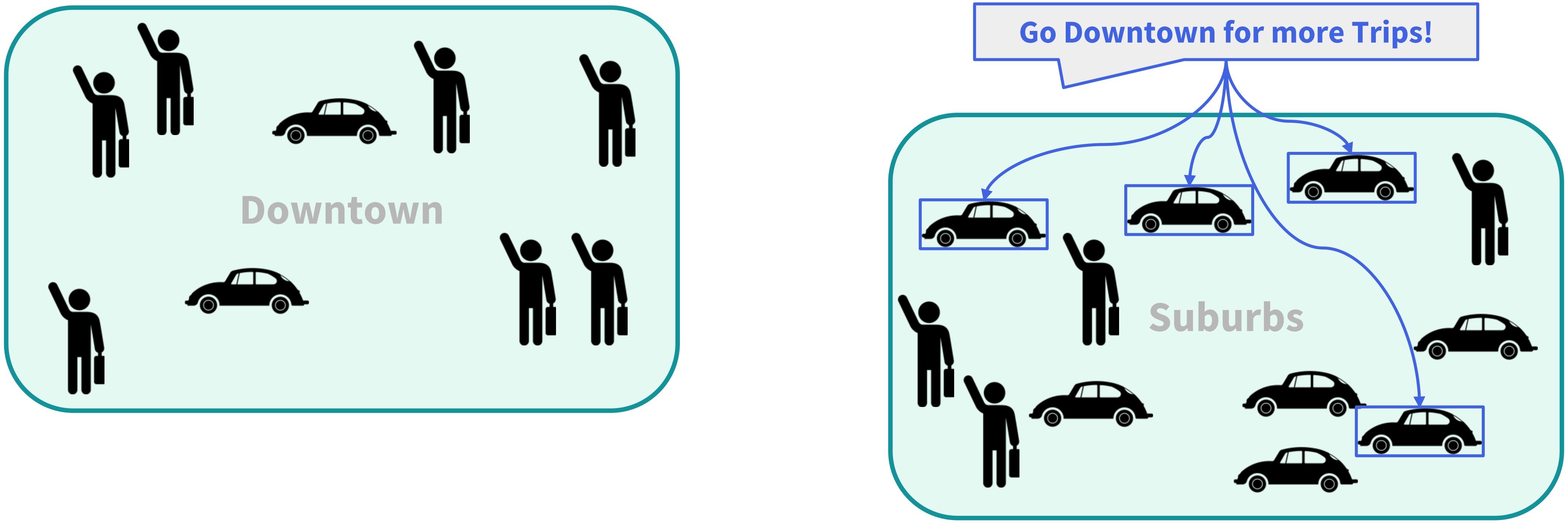
Tech builds a product that dynamically identifies over- and under-supplied areas and sends repositioning recommendations to drivers in over-supplied areas.



Slide credit:
Uber MX team

Driver Positioning Example

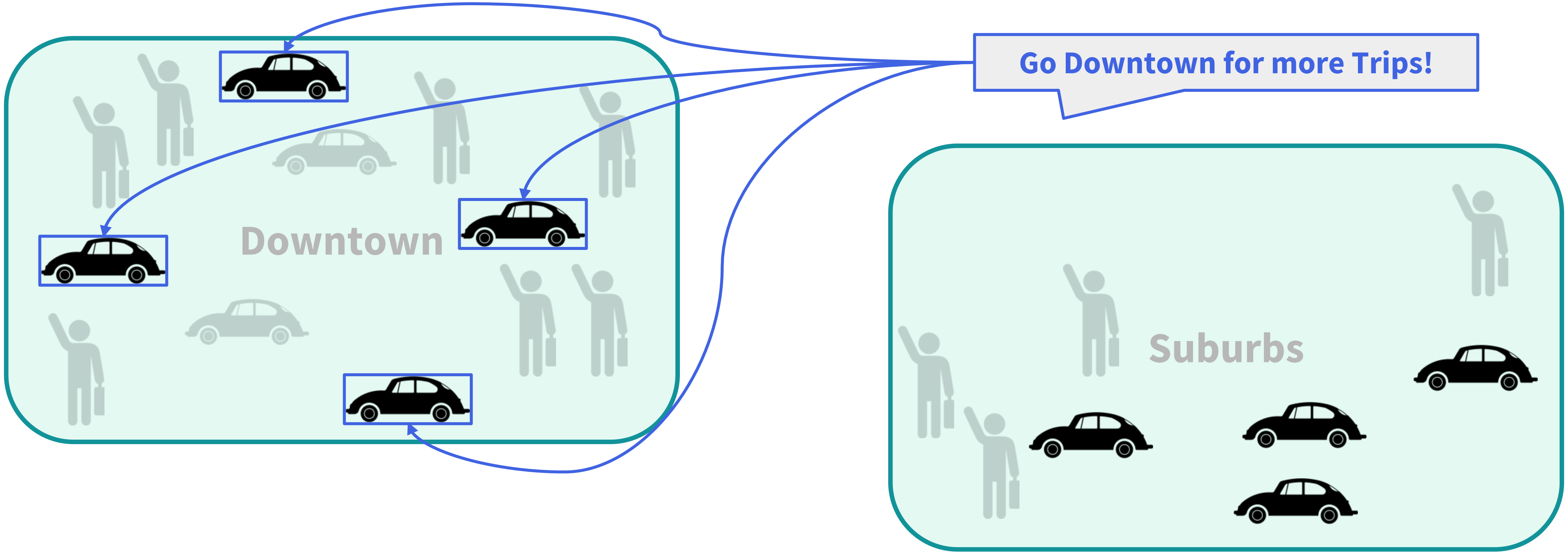
To test this, Uber runs a driver A/B experiment where 50% of drivers in the Suburbs are asked to relocate to Downtown. (The other 50% do not get recommendations.)



Driver Positioning Example

Slide credit:
Uber MX team

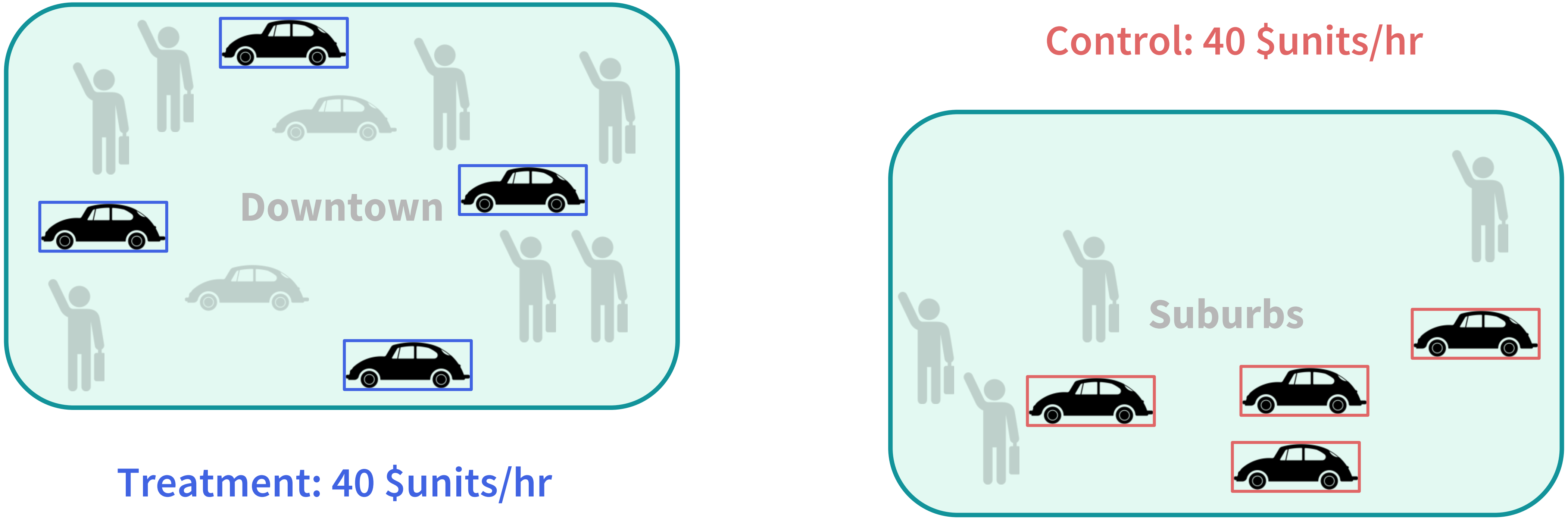
Suppose the drivers follow the recommendation and relocate



Slide credit:
Uber MX team

Driver Positioning Example

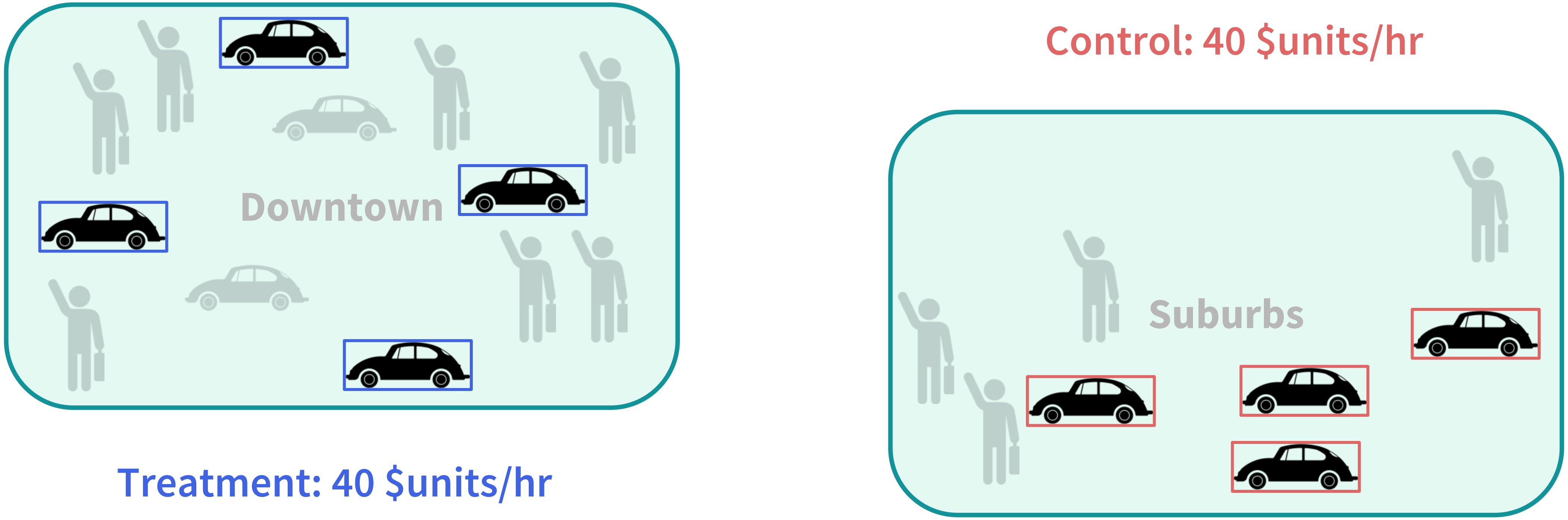
Suppose we find that drivers who got the repositioning message (and relocated) had the same earnings per hour as drivers who didn't get the message!



Slide credit:
Uber MX team

Driver Positioning Example

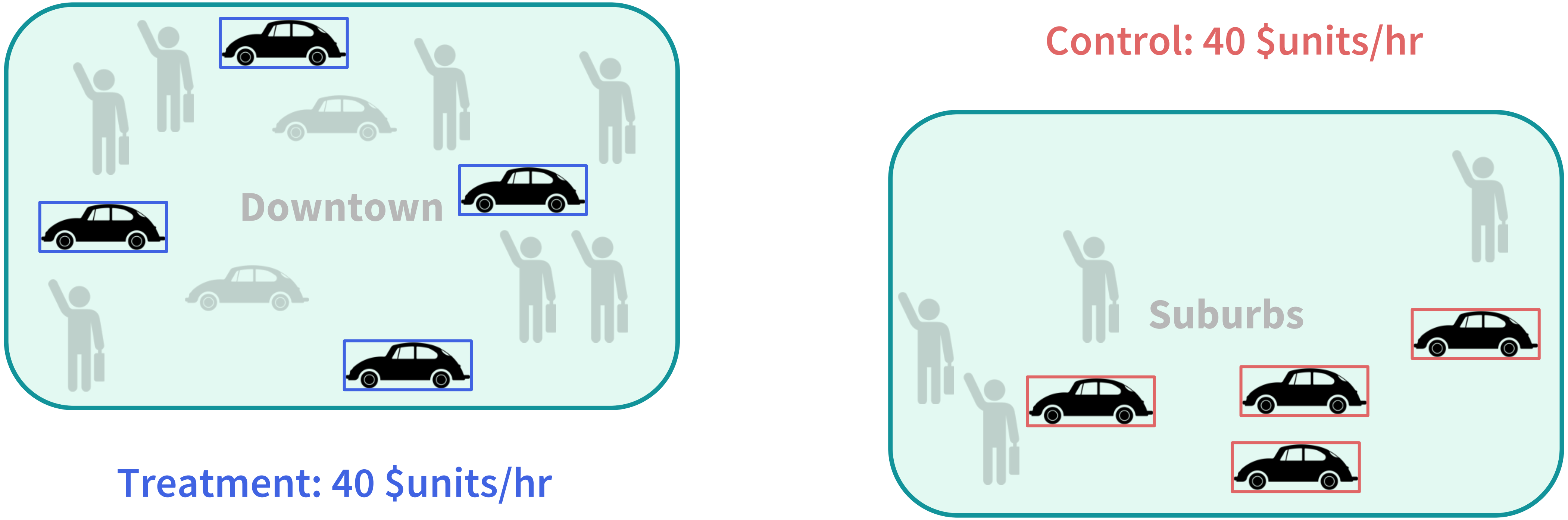
On the basis of this A/B earnings comparison, we might conclude that this product did **nothing** to raise driver earnings.



Driver Positioning Example

Slide credit:
Uber MX team

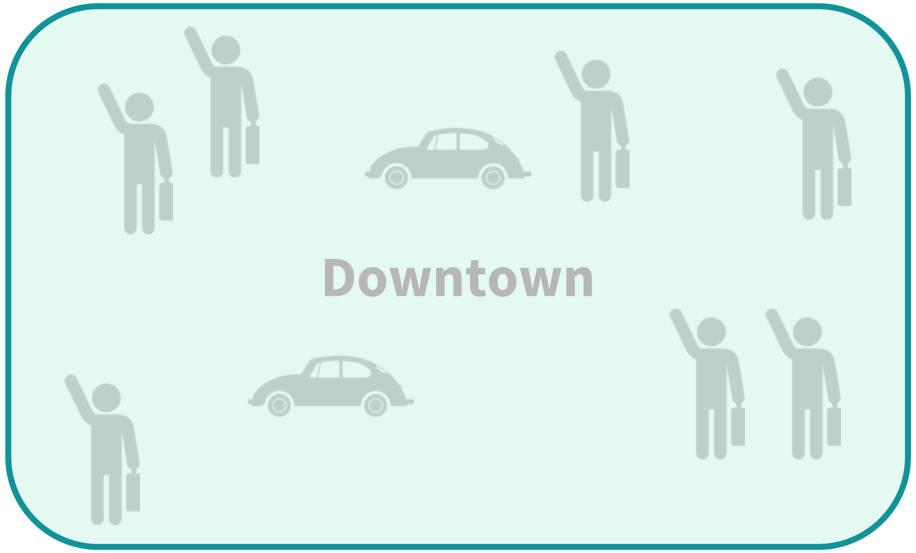
The mistake here is that by moving drivers out of the Suburbs, we increased the earnings opportunities of the Control drivers. Control was **contaminated**.



Driver Positioning Example

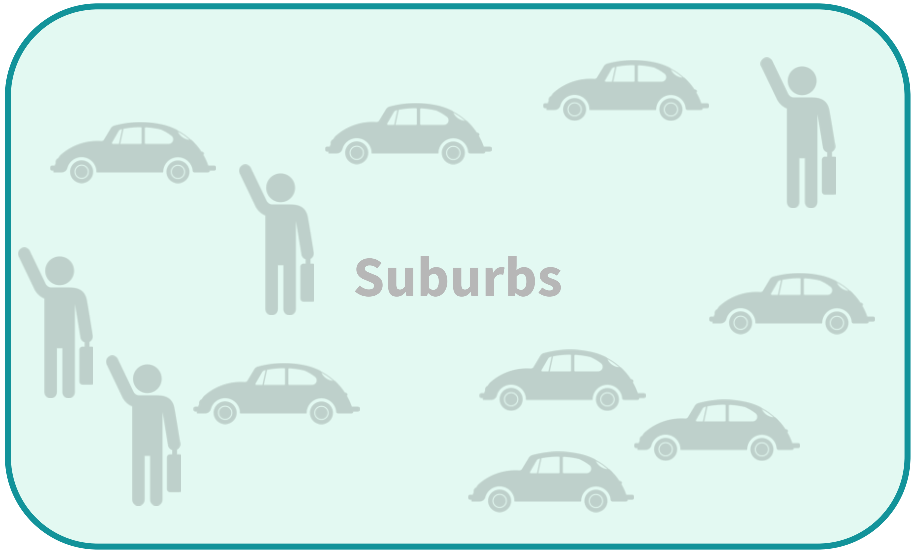
Slide credit:
Uber MX team

Counterfactually, *had we not sent the repositioning messages*, we might have seen the following driver earnings:



Counterfactual
Downtown: 40 $units/hr

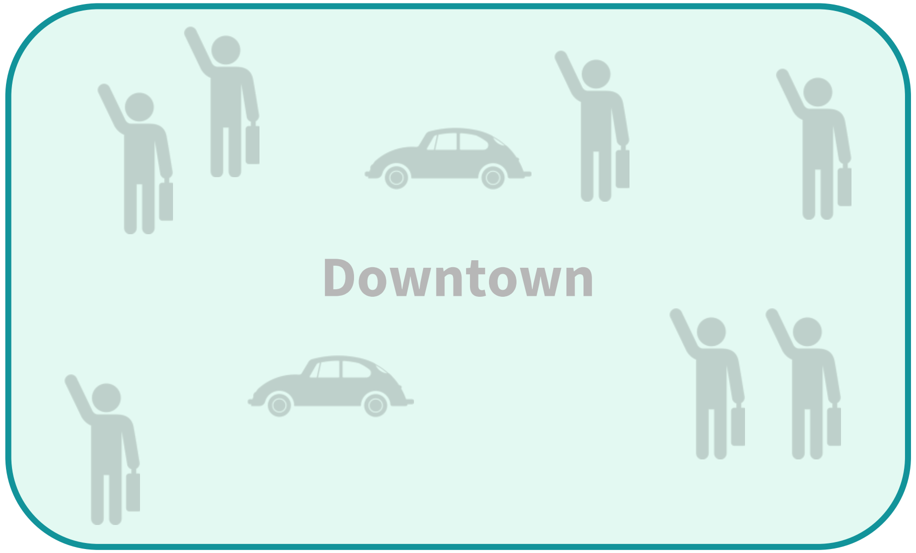
Counterfactual
Suburbs: 30 $units/hr



Driver Positioning Example

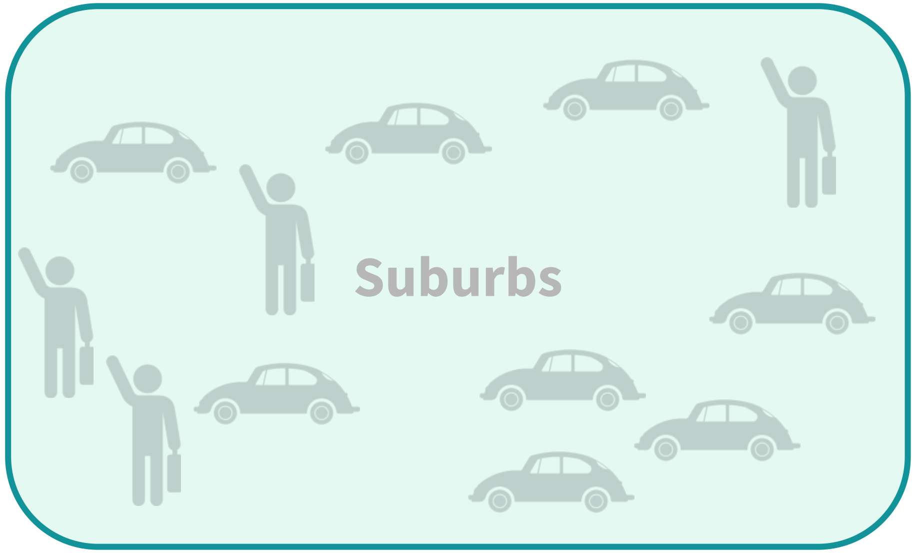
Slide credit:
Uber MX team

So in fact, the supply repositioning product increased earnings by **10 $units/hr** for both the treatment *and* the control group!



Counterfactual
Downtown: 40 $units/hr

Counterfactual
Suburbs: 30 $units/hr



Why is cluster randomization not enough?

- Often difficult to define the clusters
- There legitimately might not be enough “clusters” that don’t interfere with one another
  - In AirBnB, rentals near Disney Land (in Los Angeles) might compete with rentals near Disney World (in Orlando)
  - In Uber, a driver in a suburb could be instead choose to drive in the city
- What happened?
  - Giving the treatment to (some) drivers in the suburbs *decreased* competition for other drivers in the suburb, and *increased* competition for drivers in downtown
  - Both driver-level A/B testing and graph-cluster randomization would learn biased estimates
- We’d have to cluster at the city-level to prevent such interference
  - Still might not be enough: drivers commute from Sacramento to SF to work

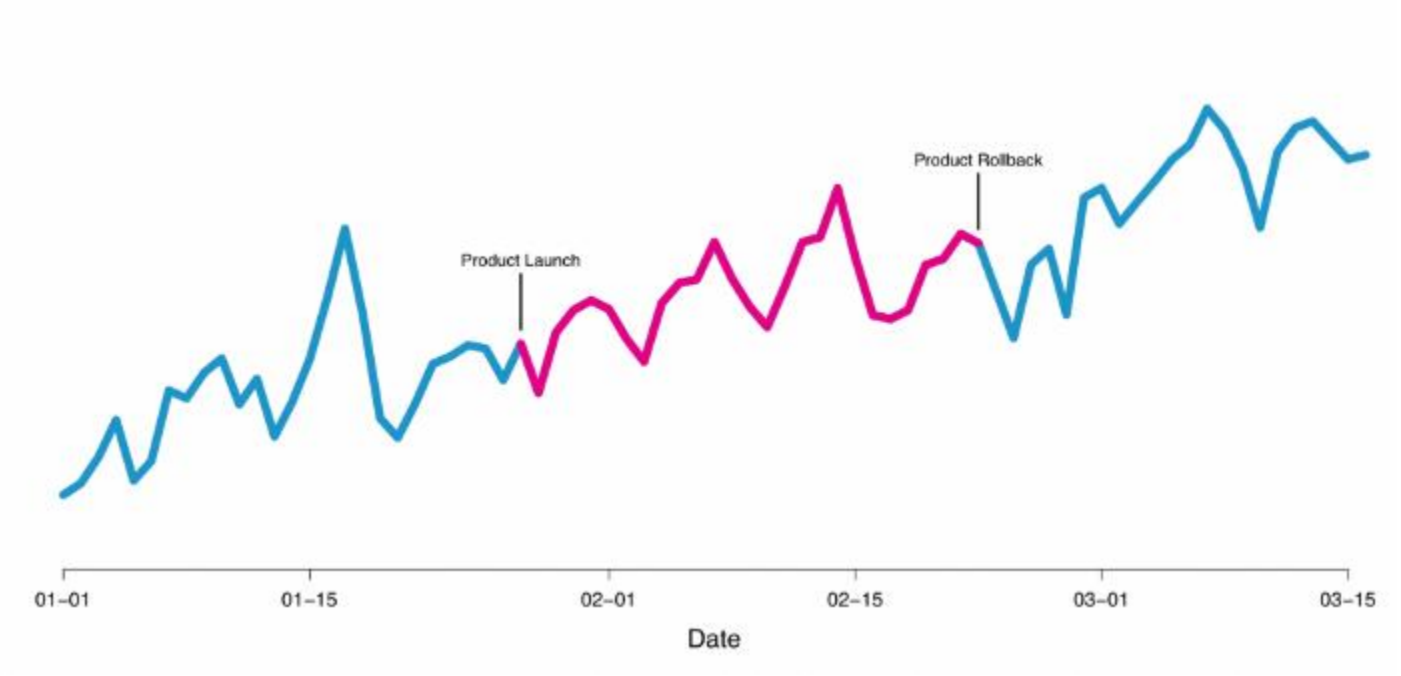
A solution: what about time?

- So far, we've thought about partitioning user clusters (often geographically correlated), or literally partitioning space (New Zealand; listings in Palo Alto)
- This is problematic when there isn't enough unique space clusters
- Time to the rescue! Allocate the *same* set of users (same city, same region of space…) to treatment or control, at different times
- Most naïve: allocate entire city to control up to time T, and then entire city to treatment after that, to time $2T$
    - Compare your metric from the control and treatment periods

Challenge with naïve solution: time-varying marketplace

"The outside world often has a much larger effect on metrics than product changes do" – AirBnb, (Jan Overgoor) [Experiments at Airbnb | by AirbnbEng | The Airbnb Tech Blog | Medium]

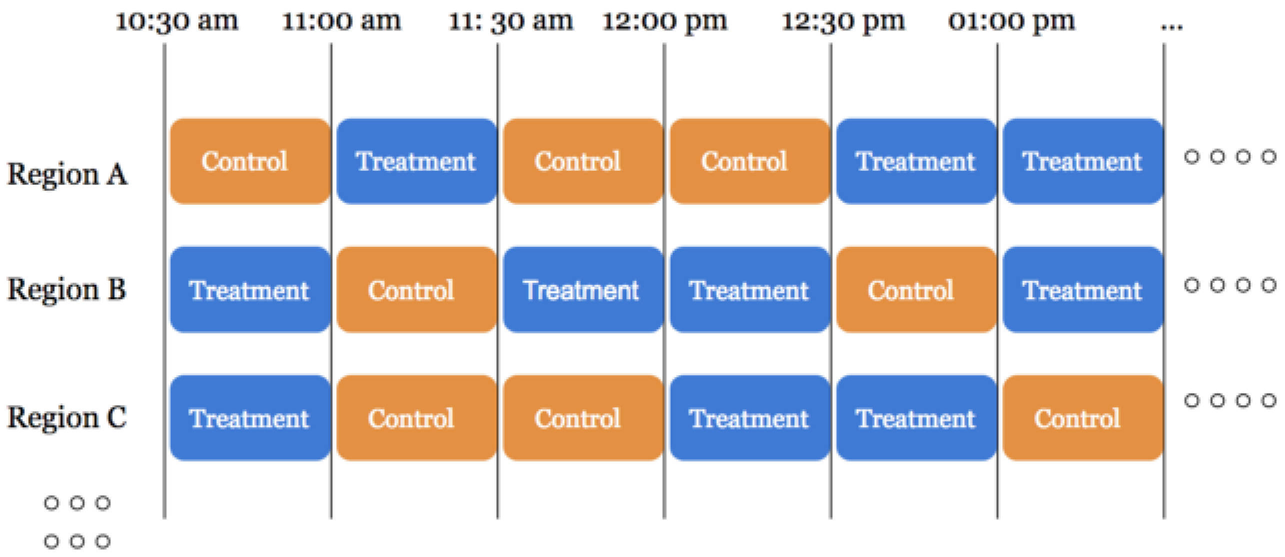
If you compare the control period (earlier), to the experiment period (later), are changes because of the product or because of underlying marketwide changes, like seasonality?



Switchbacks

- For each region (city, graph cluster, neighborhood, etc), simply *switch back and forth* on whether that region is assigned to treatment or control
- For each unit of space-time, randomly assign it treatment or control
- Hope: that different units of space-time don't interfere with one another

  Then, analyze like you do a simple A/B test or graph cluster randomization test
- Sometimes interference still happens; need to deal with that in analysis



	10:30 am	11:00 am	11: 30 am	12:00 pm	12:30 pm	01:00 pm	...
Region A	Control	Treatment	Control	Control	Treatment	Treatment	○○○○
Region B	Treatment	Control	Treatment	Treatment	Control	Treatment	○○○○
Region C	Treatment	Control	Control	Treatment	Treatment	Control	○○○○
○○○ ○○○							

Image credit: Switchback Tests and Randomized Experimentation Under Network Effects at DoorDash | by DoorDash | Medium (David Kastelman, Data Scientist & Raghav Ramesh, Machine Learning Engineer)

Experimentation summary so far

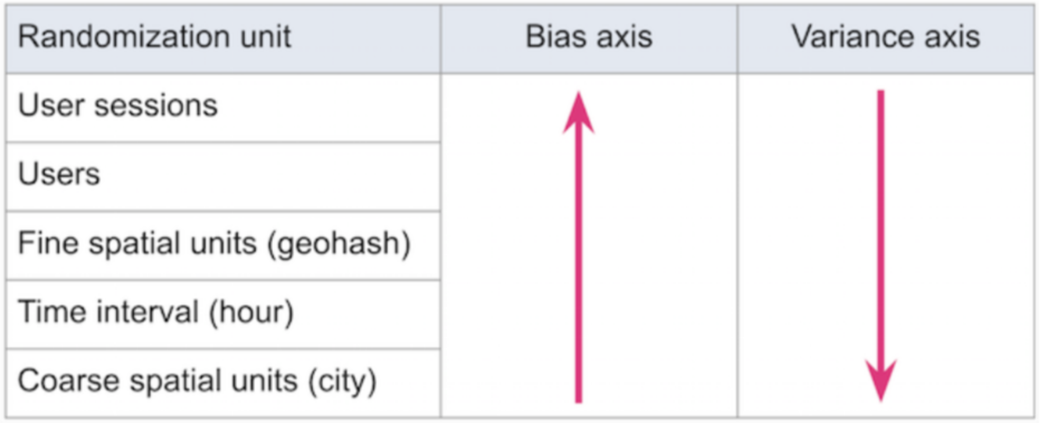
- Several different experimental designs
    - Classic, individual level A/B testing
    - Graph cluster randomization
        - More generally, *spatial* randomization
    - Switchbacks: randomization over time

Reminder 1: Bias-variance trade-off

- Bias-variance trade-off:
  - Smaller clusters (units) => more likely to interfere => more *bias*
  - Bigger clusters (units) => fewer clusters (units) => more *variance*
- What does each mean?

  Variance: If you run multiple experiments, each gives you a different answer

  Bias: If you run multiple experiments: each gives you the same wrong answer



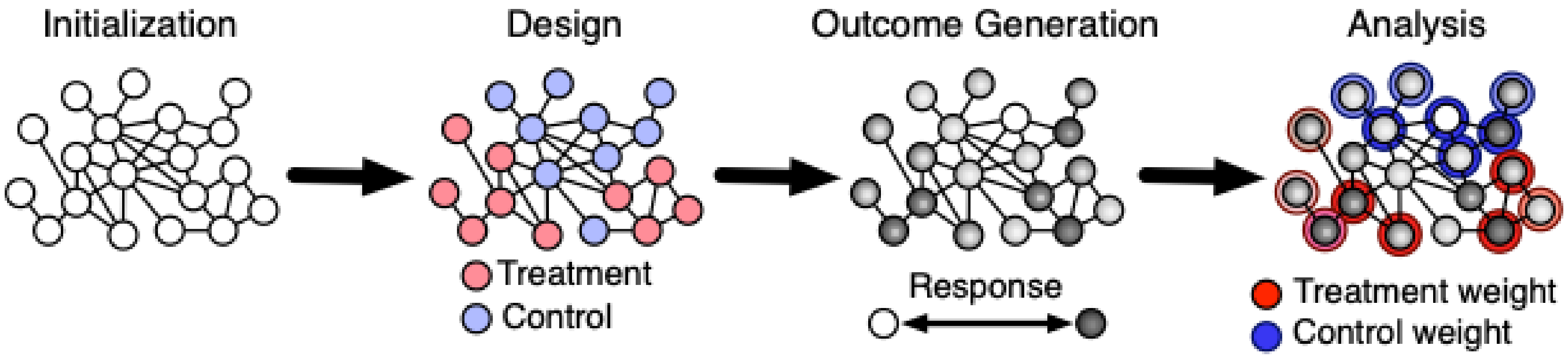
Randomization unit	Bias axis	Variance axis
User sessions	↑	↓
Users		
Fine spatial units (geohash)		
Time interval (hour)		
Coarse spatial units (city)		

Table 1. Different choices of experimental units correspond to different points on the bias-variance tradeoff spectrum. In the context of network experiments, bias comes from interference effects; variance comes from decreasing unit set cardinality, and from between-unit heterogeneity.

Experimentation in a Ridesharing Marketplace | by Nicholas Chamandy | Lyft Engineering

Reminder 2: Design & Analysis

Two parts of running a good experiment: design and analysis



Design: Who gets assigned to treatment, who gets assigned to control

Analysis: Given the assignments and metrics for each unit, how do we calculate the Global Treatment Effect?

We have focused on design: **good design** simplifies analysis, **bad design** makes analysis impossible

Experimentation culture

Classical power analyses

- In a past statistics class, you might have learned “power analysis”
  - If the “true effect” is at least as big as X, then an experiment on N samples will *reject the null hypothesis* at least Z% of the time.
  - If the true effect is 0 (null hypothesis is true), the experiment will falsely reject it no more than α% of the time.
  - Given X, Z%, and α%, easy to calculate fixed sample size N
  - You run an experiment, with N samples
- This reflects a “scientific” approach to experiments: an experiment that rejects false hypotheses and accepts true hypotheses
- This is ***a wrong approach*** in practice

  You don’t care about doing good science

Problems with classical approach

Your goal: you want to quickly launch amazing products (large $Y_1 - Y_0$)

It's ok to not launch an "ok" product that would reject the null (small $Y_1 - Y_0$)

Also ok to sometimes launch "useless" products (zero $Y_1 - Y_0$)

Never want to launch a product that hurts your metrics (very negative $Y_1 - Y_0$)

Your advantage: You have *many* possible products to launch/experiments to run; limitation is sample size

Classical approach:

- Sample size N optimized to find small effects (small X $= Y_1 - Y_0$)
- Wastes samples & time that could be spent on other experiments

“Discovery-driven” experimentation

Insight: You have *many* experiments. If one product looks mediocre early in the experiment, just move on

Run an experiment *just* long enough to determine if it’s an amazing product (large $Y_1 - Y_0$) or if it’s a dud

- “Peek”, but smartly this time, based on $\widehat{Y_1} - \widehat{Y}_0$
- Upper threshold $\mathrm{u}(n)$ to stop experiment and *declare victory;* decrease with more samples n
- *Lower* threshold $\ell(\mathrm{n})$ to stop experiment and *declare loss;* increases with more samples n

Result

- Bad science: you’ll often reject small, positive products
- But you’ll find amazing products as quickly as possible

More generally, *adaptive experimentation* when have many different arms of a treatment (for example, 41 shades of blue); remove poor arms quickly, focus on best ones

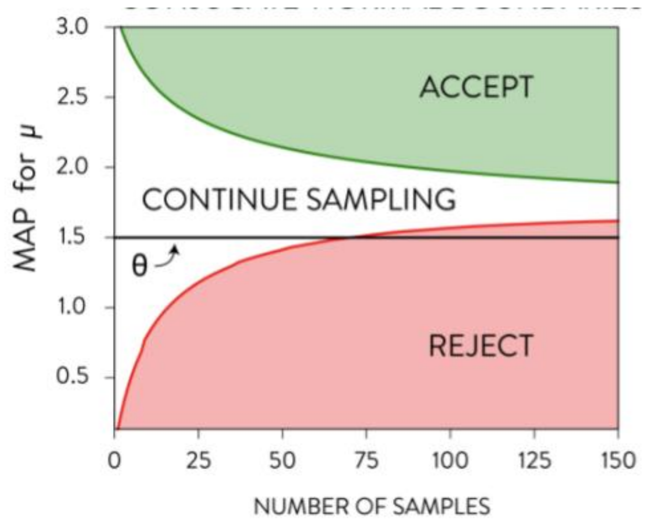


Image credit: Large scale experimentation | Stitch Fix Technology – Multithreaded, Sven Schmit, Brian Coffey

Paper: “Optimal Testing in the Experiment-rich Regime” Sven Schmit, Virag Shah, Ramesh Johari

Simulation

Build a “simulator” for how your market performs

Lyft blog post: have drivers drive around in simulator, matched with riders using their algorithms

- Can simulate how different matching algorithms perform
- Also can simulate pricing algorithms
    - Need assumptions on how individual riders will respond (big assumption)
    - Under these assumptions, can learn market-wide effects of algorithm

      i.e., simulate interference patterns, if we know “first-order” effects of product
- Also can simulate different *experimentation* methods
    - Know the ground truth, simulate what different protocols would find
    - For example: in homework we simulated different experiment designs using the same historical data from an A/B test

Universal holdout

Downsides of standard approaches:

- Test one product at a time
- Usually enroll as few users as possible (don't want to waste sample size)
- Experiments are usually short → Don't observe long-term metrics

What if you want to know, "What is the total effect on everything I launched last quarter on customer retention?"

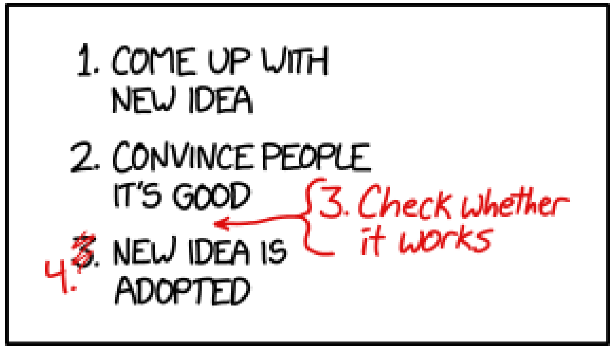
Solution: Universal holdout

- Each quarter (or month or year…), hold out same set of users from *every product* you launch that quarter
- End of quarter, compare metrics for that group to all other users; re-enroll a new set of universal holdout for next quarter

Universal Holdout Groups at Disney Streaming | by Tian Yang | disney-streaming | Oct, 2021 | Medium

General pipeline of launching a product

- Come up with idea, iterate on design
- Code it up, and evaluate on simulator
- Test in real experiment in one city/market
- If that goes well over time, roll out in multiple markets
- Continue rolling out in more and more markets
- Eventually, will have rolled out everywhere


1. COME UP WITH NEW IDEA
2. CONVINCE PEOPLE IT'S GOOD
3. Check whether it works
~~3.~~ 4. NEW IDEA IS ADOPTED

THE INVENTION OF CLINICAL TRIALS

xkcd: Clinical Trials

Experimentation summary so far

- Several different experimental designs
  - Classic, individual level A/B testing
  - Graph cluster randomization
    - More generally, *spatial* randomization
  - Switchbacks: randomization over time
- These experimental techniques are not workable sometimes
  - Product is “public-facing” – hard to roll back
  - Interference really network/city wide, so spatial randomization less effective
  - Sensitive change, so can’t launch in many cities at once
  - It takes a long time for effect to occur
- Future lecture, time permitting: “synthetic control”

  Launch in just a few cities. Then, create a model for how that city would have behaved without the treatment, based on other how control cities actually behaved.