Experimentation in a Ridesharing Marketplace

Nicholas Chamandy · Follow
Published in Lyft Engineering
9 min read · Sep 2, 2016

Part 1 of 3: Interference Across a Network

Technology companies strive to make data-driven product decisions — and Lyft is no exception. Because of that, online experimentation, or A/B testing, has become ubiquitous. The way it's bandied about, you'd be excused for thinking that online experimentation is a completely solved problem. In this post, we'll illustrate why that's far from the case for systems — like a ridesharing marketplace — that evolve according to network dynamics. As we'll see, naively partitioning users into treatment and control groups can bias the effect estimates you care about.

> *To consult the [data scientist] after an experiment is finished is often merely to ask [her] to conduct a post mortem examination. [She] can perhaps say what the experiment died of.*
>
> — paraphrasing R. A. Fisher, 1938.

What Fisher was getting at, simply put, is that experiment design matters. So much so that a careless experiment design can sometimes result in data that's useless for answering the question of interest.

Example: Subsidized Prime Time

Imagine that the entire Lyft network is encapsulated by the tiny square region illustrated below. When users A and B open the Lyft app (at approximately the same time), there is only one driver available nearby. We call such a scenario **undersupply**.

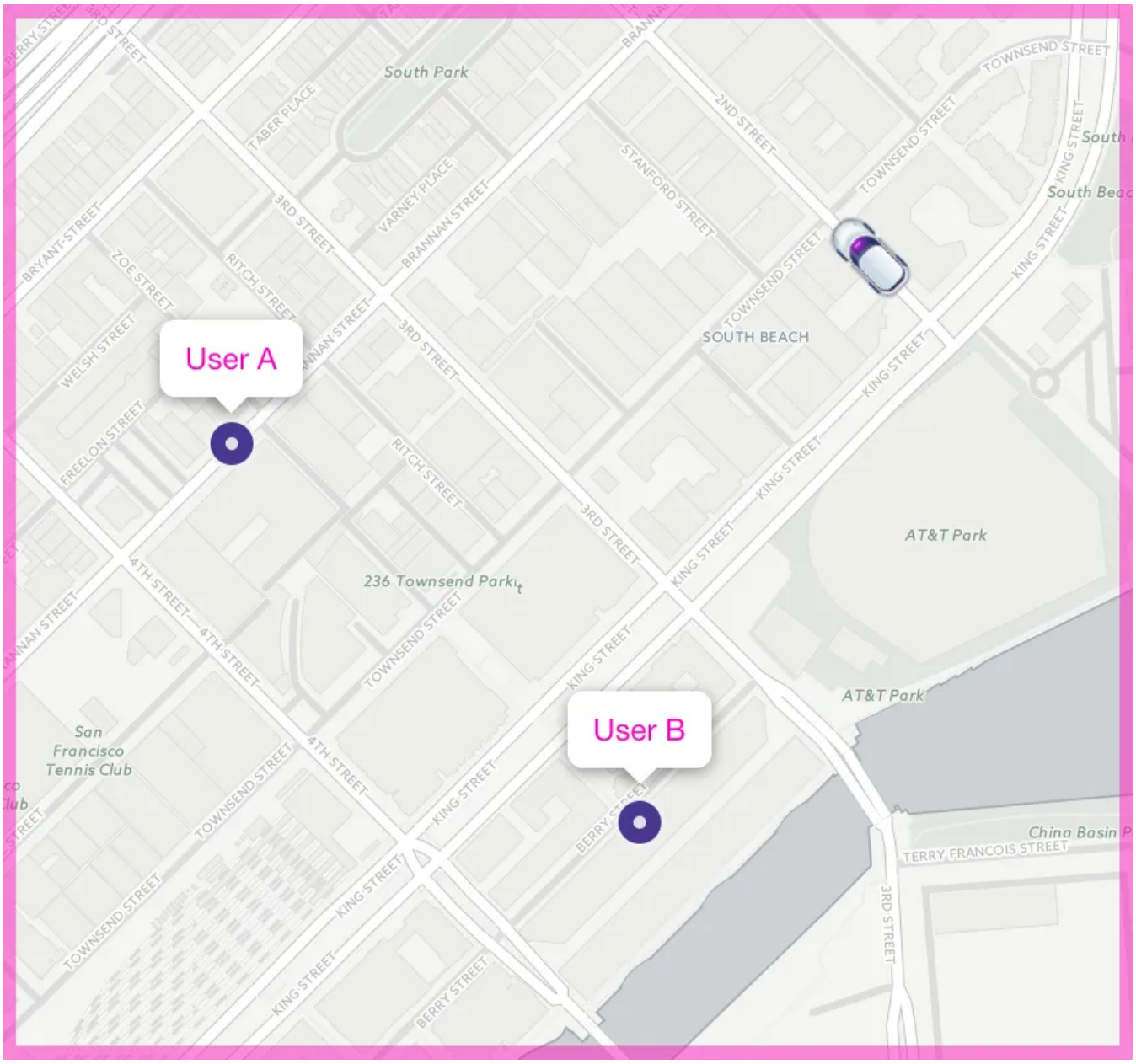


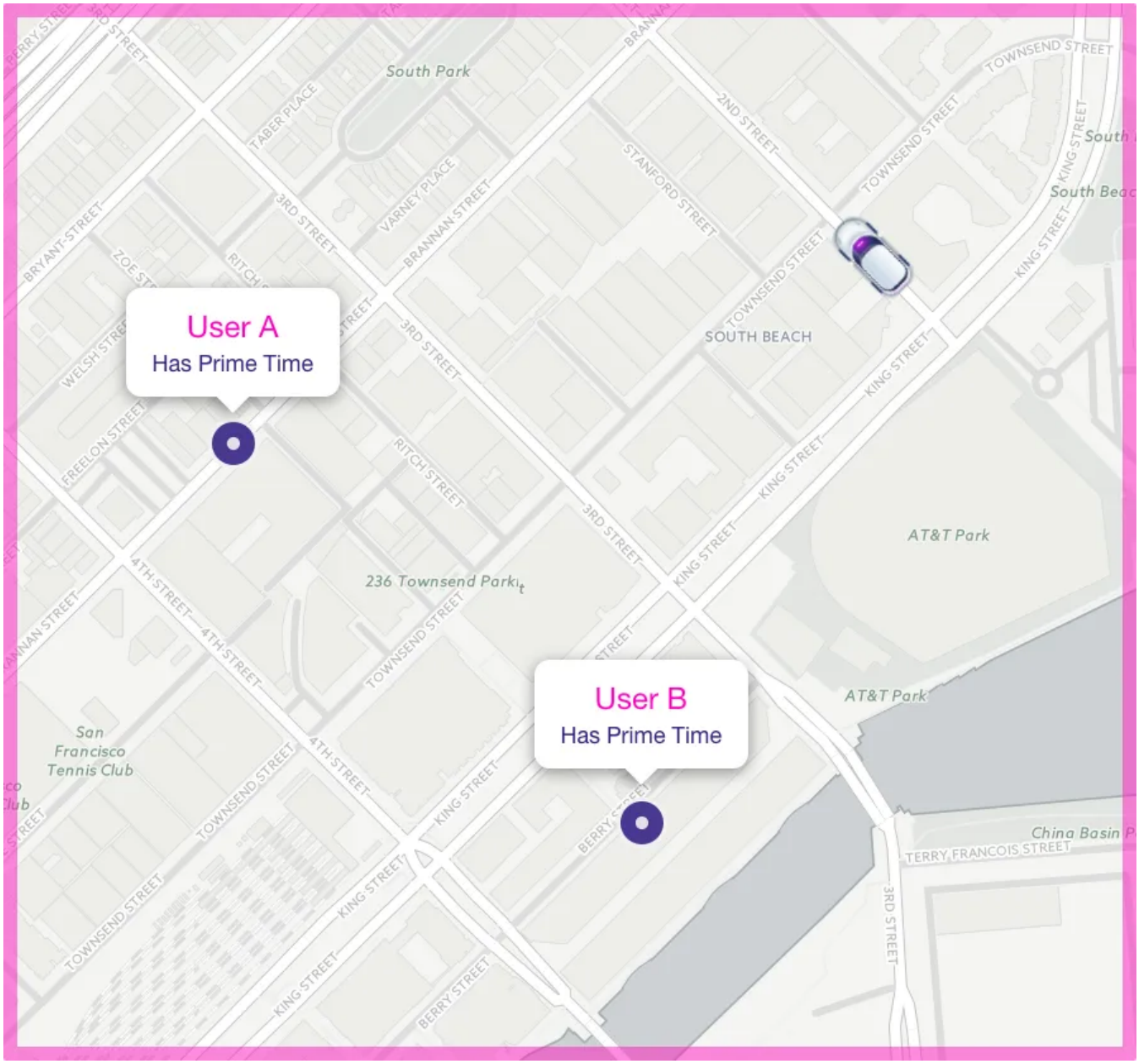
Figure 1. A typical undersupply scenario: two passengers and a single available driver. In this example, both passengers might experience Prime Time pricing.

In such cases, Lyft sometimes applies Prime Time pricing in order to maintain driver availability. Prime Time is a price surcharge expressed in percentage terms, and can take on different values depending on the extent of undersupply (+25%, +50%, etc). For simplicity, in this example we assume that Prime Time is binary — there either is Prime Time (at some fixed value) or there is not. We also assume that the supply effects of Prime Time happen at a slower timescale than the demand effects, and therefore we can ignore them. In other words, that passengers react more quickly to Prime Time than drivers do.

Suppose that we want to estimate the effect of Lyft **subsidizing** Prime Time — i.e. paying Prime Time on behalf of the passenger, without ever even displaying it to the

passenger. We'll use a green marker in subsequent pictures to denote a passenger who got the subsidy.

A fun metaphor here is that of two parallel universes. We are interested in the difference between the **factual** universe, where users get Prime Time when there is undersupply, and a **counterfactual** one, where Lyft subsidizes Prime Time. These two universes are illustrated in the picture below. Note that without any intervention, we would only observe the top universe, which we call the global control. The global treatment, on the other hand, corresponds to treating all passengers with the Prime Time subsidy.



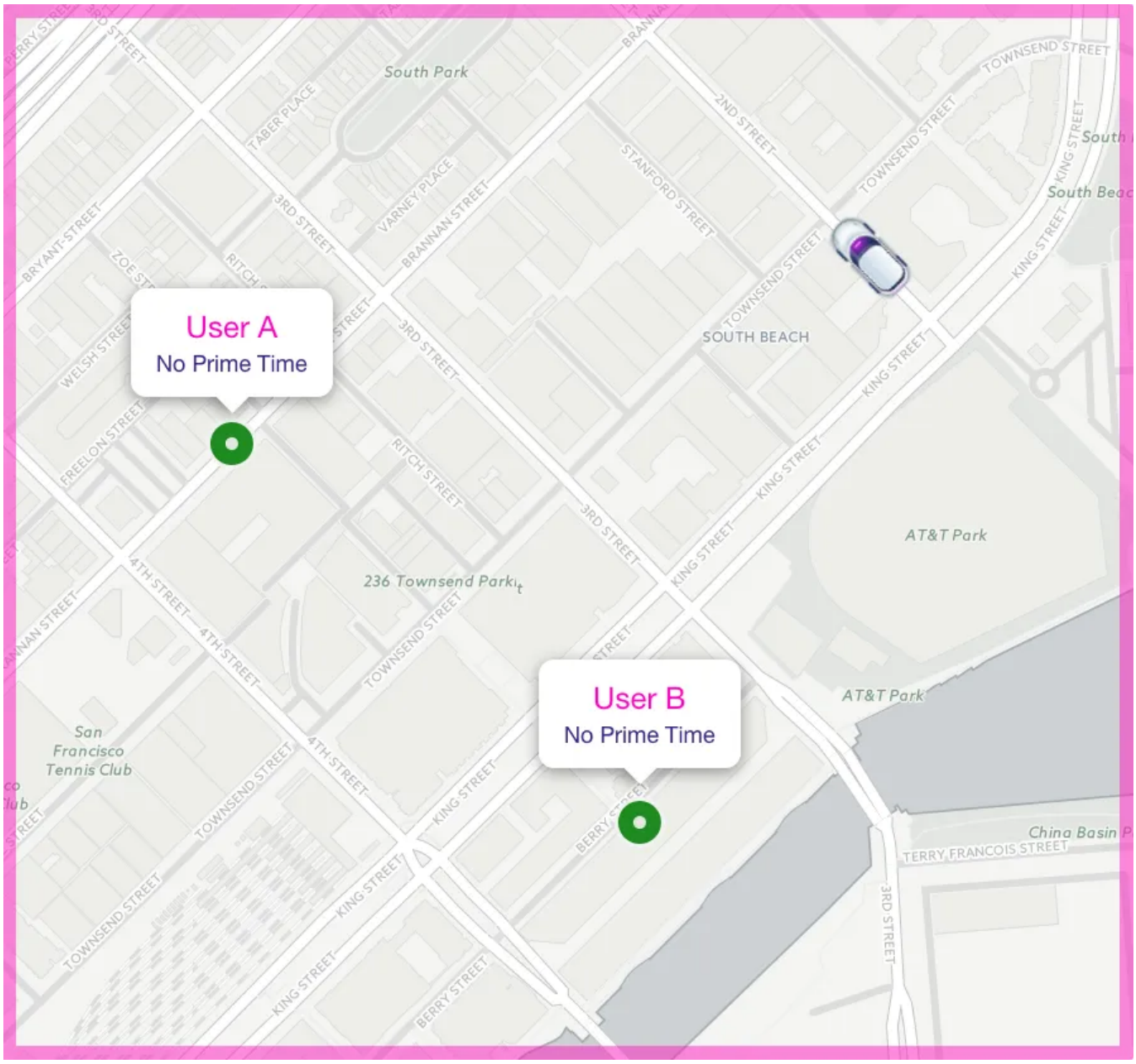


Figure 2. Factual and counterfactual universes. In real life (top box), both users experience Prime Time. This scenario is also known as the global control. In the bottom picture, both users get a Prime Time subsidy — the global treatment. We cannot observe both of these parallel realities, and would observe the global control without some intervention.

Suppose that the metric of interest is the total number of rides completed on average (or in expectation). We would like to know how this number changes between the two parallel universes. Let's assume a simple probability model to make this example easy. Specifically:

- When there is no Prime Time, a passenger who opens the app and sees a driver available always requests a ride
- When there is Prime Time, the same passenger has a 50% chance of requesting a ride

- Neither drivers nor passengers ever cancel — every request leads to a completed ride

Under the global control scenario, the average number of rides taken by passengers A and B is 0.75. To see why, assume without loss of generality that passenger A opens the app a few seconds before passenger B. Half the time, user A will take the ride despite seeing Prime Time, and the number of rides is 1. A quarter of the time, User A will choose not to request, and User B will take the single ride instead. Otherwise, both users decline the Prime Time, and n0 rides are taken. By symmetry, the same is true if B opens first. The expectation is therefore

$$\begin{aligned}
\mathbb{E}[\text{rides}] &= \mathbb{E}[\text{rides}|A \text{ opens first}]\,\mathbb{P}(A \text{ opens first}) + \mathbb{E}[\text{rides}|B \text{ opens first}]\,\mathbb{P}(B \text{ opens first}) \\
&= \mathbb{E}[\text{rides}|A \text{ opens first}] \times 0.5 + \mathbb{E}[\text{rides}|B \text{ opens first}] \times 0.5 \\
&= \mathbb{E}[\text{rides}|A \text{ opens first}] \\
&= 1 \times \mathbb{P}(A \text{ requests}) + 1 \times \mathbb{P}(A \text{ declines and } B \text{ requests}) + 0 \times \mathbb{P}(A \text{ and } B \text{ decline}) \\
&= 1 \times 0.5 + 1 \times 0.5 \times 0.5 + 0 \\
&= 0.75.
\end{aligned}$$

Under the global treatment, neither passenger sees Prime Time and the situation is much simpler. The first user to open the app automatically takes a ride, and the second is out of luck. Since a single ride is always taken, 1 is also the expectation. Comparing these two universes, we see that the global average treatment effect, i.e. the ground truth we would like to estimate, is given by

$$\begin{aligned}
\text{Treatment Effect} &= \left(\frac{\mathbb{E}[\text{rides in the global treatment}]}{\mathbb{E}[\text{rides in the global control}]} - 1\right) \times 100\% \\
&= +33\tfrac{1}{3}\%
\end{aligned}$$

Subsidizing Prime Time results in a 1/3 increase in rides in our simple model. This treatment effect is of course unobservable in real life. So we must find some way to estimate it.

Randomizing passengers

The standard way to A/B test an intervention like subsidized Prime Time is to pseudo-randomly assign users (in this case passengers) to either the treatment or control group, for instance by hashing their user IDs into buckets. In our example, the average result of such a randomization is that one user sees Prime Time while

the other doesn't — as illustrated by the following picture. This scenario corresponds to yet a third (mutually exclusive) parallel universe!

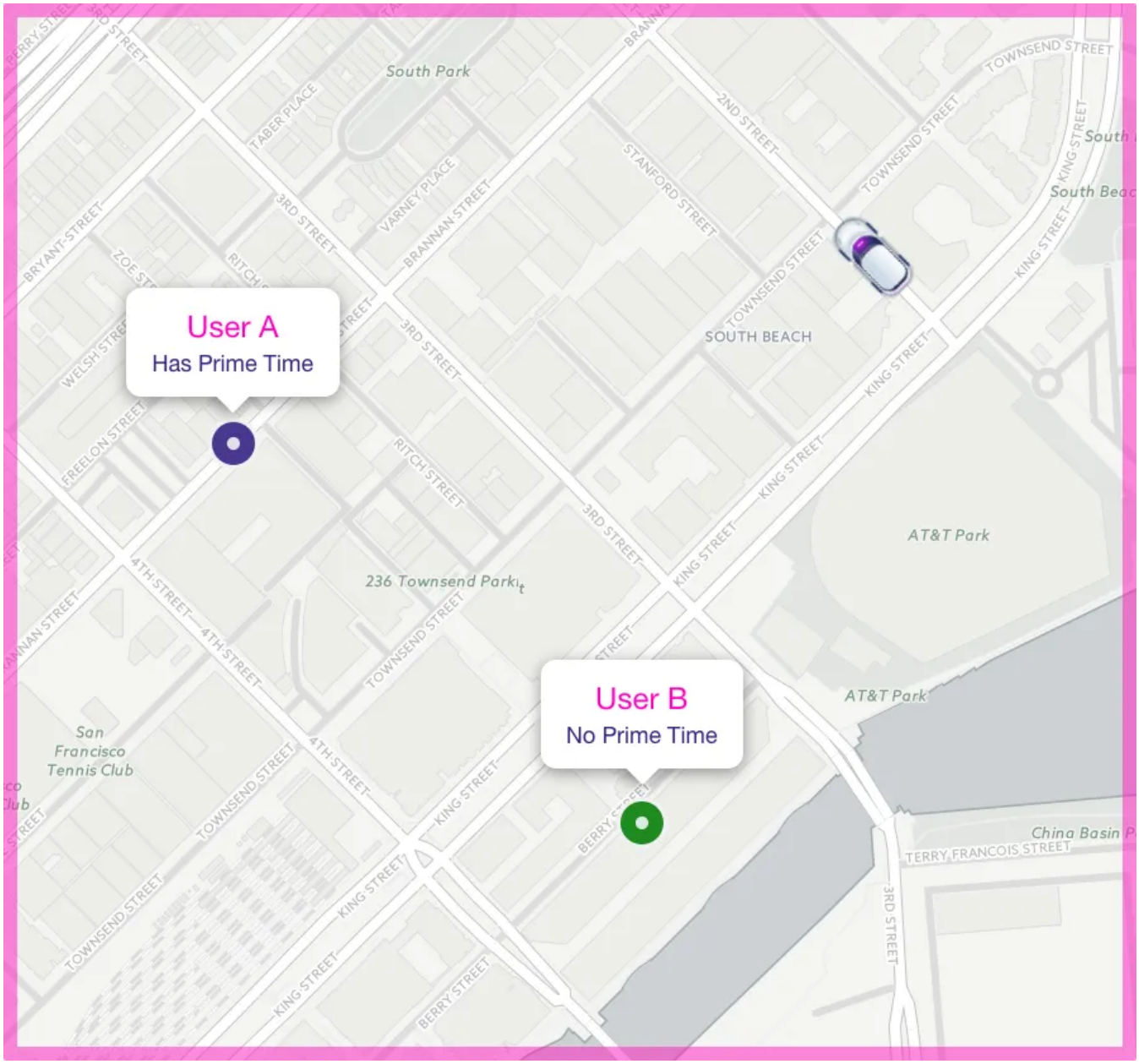


Figure 3. One realization of the random-user experiment. In this example, User A ended up in the control (got Prime Time), and User B in the treatment (Prime Time subsidy).

In order to estimate the effect of the treatment on a metric of interest for a random-user experiment like this one, one typically does the following:

1. Estimate the global control metric value by restricting to users in the control group
2. Estimate the global treatment metric value by restricting to users in the treatment group
3. Compute the relative difference between the estimates from 1 and 2

Let's see what happens when we apply this logic to our simple example. Remember that each user has a 50% chance of opening the app first. Let's first consider user B, who happens to be in the treatment group (subsidized Prime Time). In our simple model, she is guaranteed to request and complete exactly one ride if she opens the app first. On the other hand, if she opens the app second, she will complete one ride if and only if user A decided not to request. That event also happens with a 50% probability, so *given* that user A opens the app first, user B expects to take half a ride. Combining all this, the expected number of rides for user B is

$$\begin{aligned}\mathbb{E}[\text{rides by } B] &= \mathbb{E}[\text{rides by } B \mid B \text{ opens 1st}] \times \mathbb{P}(B \text{ opens 1st}) \\ &\quad + \mathbb{E}[\text{rides by } B \mid B \text{ opens 2nd}] \times \mathbb{P}(B \text{ opens 2nd}) \\ &= 1 \times 0.5 + 0.5 \times 0.5 \\ &= 0.75.\end{aligned}$$

The situation for user A is even easier. User A cannot take a ride if user B opens the app first — so the expected value is 0 in that case. And we know that user A, who sees Prime Time, will request a ride half of the time given driver availability. So the expected number of rides completed by A is

$$\begin{aligned}\mathbb{E}[\text{rides by } A] &= \mathbb{E}[\text{rides by } A \mid A \text{ opens 1st}] \times \mathbb{P}(A \text{ opens 1st}) \\ &\quad + \mathbb{E}[\text{rides by } A \mid A \text{ opens 2nd}] \times \mathbb{P}(A \text{ opens 2nd}) \\ &= 0.5 \times 0.5 + 0 \times 0.5 \\ &= 0.25.\end{aligned}$$

Now let's compute an estimate of the percent change in our metric due to the Prime Time subsidy.

$$\begin{aligned}\text{Treatment Effect Estimate} &= \left(\frac{\mathbb{E}[\text{rides by } B]}{\mathbb{E}[\text{rides by } A]} - 1\right) \times 100\% \\ &= +200\%\end{aligned}$$

Obviously, this is much bigger than the ground truth effect size of 33% that we calculated above — we overestimated the effect of the Prime Time subsidy by a factor of 6! Admittedly, two users is not very many, so you might think that this fictional A/B test is suffering from small sample size problems. Surely, a user cannot actually take 0.25 rides. But imagine that the real Lyft network is composed of

copies of this 2-passenger sandbox, all evolving independently over time, replenishing drivers and passengers at a constant rate. We can construct a much larger scale example, with many such boxes, where all of the above calculations still hold.

Statistical interference

What happened in the above example is due to a statistical phenomenon known as **interference** (not to be confused with inference). To properly define it, we first have to introduce the notion of a **potential outcome**. The idea behind potential outcomes is simple: every **experimental unit** (e.g. user) walks around with two pieces of paper, one in each back pocket. On one of these papers is written that subject's inevitable outcome should she happen to be assigned to the control group. On the other, her outcome given assignment to the treatment. Together, the two pieces of paper are a unit's potential outcomes — the set of things that could potentially happen to her if she participates in the experiment. Typically, these outcomes are considered fixed and deterministic — the only thing that is random is the unit's assignment to an experiment group.

A key assumption of causal inference is that what's written on those two pieces of paper is unaffected by the experimental assignment that the unit happens to get, *and by the assignments of every other unit in the experiment*. Interference occurs when the group assignment of unit A changes any of the potential outcomes of unit B. This is precisely what we saw in the toy example above, with the outcome of interest being whether or not a ride is completed. When user A's Prime Time is subsidized, user B is less likely to be able to complete a ride (regardless of whether or not user B's Prime Time is also subsidized).

In medical statistics, the notion of interference arose in the study of vaccines for infectious diseases. The effectiveness of a vaccine on one subject's outcomes depends on how many others in his social circle also received the immunization. In other words, one subject's treatment can offer protective benefit to other, possibly untreated subjects. The result is that the measured difference between treated and untreated subjects (the benefit attributed to the vaccine) will shrink. Above, user A's Prime Time was "protective" for user B's propensity to successfully complete a Lyft ride — which in this case led to an *exaggeration* of the true effect size. In general, interference bias can occur in either direction.

Lyft is not the only technology company trying to mitigate statistical interference in A/B testing. Researchers from Google and eBay have observed the same phenomenon in applications where advertisers or users interact within online auctions. Coarser randomization, say at the auction level, can help (but not completely) mitigate the bias. The eBay example is particularly germane to our toy example as the authors characterize interference bias in relation to supply and demand elasticity. The interference problem also occurs in experiments for social networks, where a user's response to treatment may contaminate adjacent nodes in the graph. Some progress has been made on this problem for relatively static networks, with graph clustering playing a central role. Complicating things in our world is the fact that the Lyft network is both two-sided (passengers and drivers) and has a graph structure which is incredibly dynamic. Thus interference is difficult to model explicitly.

Alternative experiment designs

Randomizing users is certainly not the only way to construct online experiments in a ridesharing marketplace. One can alternatively randomize app sessions, spatial units ranging from square blocks to entire cities, or even time intervals. The coarser these experimental units, the stronger the protection against interference bias in your effect estimates. However, the cost is increased variance in your estimators, because coarse units are naturally less numerous than fine units (variance scales as one over the sample size) — and sometimes just as heterogeneous. This cost can be substantial. Nevertheless, alternating time intervals between global control and global treatment configurations was a successful strategy for the Lyft Marketplace team in the early days of experimentation. The table below positions these various randomization schemes on the continuum of bias-variance tradeoffs.

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.

To rigorously quantify these tradeoffs, however, requires a careful simulation study. Before we can embark on that adventure, we need to describe the elaborate simulation framework designed and built by the Lyft Data Science team. Luckily, that is precisely the subject of our next installment of this blog post, ***Part 2: Simulating a ridesharing marketplace***. Stay tuned!

> *Interested in experiment design, marketplace optimization, or Data Science in general? Lyft is hiring! Drop me an email at chamandy@lyft.com.*

Lyft Data Science

Follow

Written by Nicholas Chamandy

634 Followers · Writer for Lyft Engineering

Scientific Director at Lyft

More from Nicholas Chamandy and Lyft Engineering