What's Mine is Yours, What's Yours is Mine

Simplifying Significance Testing With Big Data

Karan Matnani
Last Mile Tech, Amazon
kmatnani@amazon.com

Valerie Liptak
Last Mile Tech, Amazon
liptav@amazon.com

George Forman
Last Mile Tech, Amazon
ghforman@amazon.com

ABSTRACT

At Amazon Last Mile, we deliver over 3.5 Billion [1] packages every year, making us one of the largest delivery companies in the world. At this scale, even small changes can have a big business impact. In general, business impact is assessed using controlled experimentation. A standard approach to evaluating whether controlled experiments resulted in a significant change has been to use a t-test. However, despite our scale the law of large numbers fails to produce a normal distribution, and the t-test fails up to 99% of the time. In addition to exhibiting non-normal distributions our application has restrictions on the granularity of control and treatment group splits and also suffers from geospatial correlation which causes treatment effects to be applied across both control and treatment groups at finer granularities (e.g., delivery of packages to multiple homes by the same delivery agent in one stop; a building falling on different routes on two different days depending on other stops on that day). This introduces a tradeoff between separability of effects through coarse granularity and detection of smaller treatment effects with fine granularity. In this paper we solve the t-test dilemma using a resampling test at scale, and further leverage this test to create a scalable, repeatable methodology for randomization split granularity choice under these constraints. We produce a sensitivity optimized randomization strategy using a data driven approach that has been applied successfully within multiple real experiments at Amazon Last Mile Tech and is generalizable to any experiment.

1 INTRODUCTION

At the scale of billions of packages delivered annually, small changes can have a substantial effect on customer experience. Principled experimentation drives business in the right direction. At Amazon Last Mile, changes affecting customers are rolled out based on the results of a controlled experiment. Examples of experiments are: Changing the experience of the mobile application used by delivery agents, updates to routing and navigation, or using new algorithms to pick the places they drop off the package to.

To quantify the effect of an experiment, the subjects of the experiment are split into Control (C) and Treatment (T) groups, and the goal is to make these splits as fair as possible to have unbiased, robust experimentation. The quality of the split is measured on three factors. Bias: whether there is a difference in the target variable between the groups before applying the treatment. Power: the ability to detect small effects of the applied treatment. Mixing: the amount of control instances that experience treatment effects, and vice versa. Ideally, we want to maximize power while minimizing bias and mixing.

Making mistakes in experimentation is costly because of rollbacks, developer and scientist time spent in deep dives, and the lost opportunity cost. Raising the bar with resampling tests adds value with informed decision making, and rework prevention. With this context, we present the motivation, then the experiment design, followed by the experiment results, and finally a real world application with ideas for more applications. Our contribution includes presenting a case for the permutation test, and showing how it can be made scalable in a real-world scenario.

2 MOTIVATION: T-TEST SIGNIFICANCE TESTS FAIL AT SCALE

2.1 What makes a good Randomization Strategy

In the design of this methodology, we decided to measure the quality of the split using three types of metrics.

(1) Unbiasedness: There will always be some differences between Control and Treatment, but we don't want to declare they are statistically significant unless they are from the effect of the application of our treatment.
(2) Power: the ability to detect small effects of the applied treatment.
(3) Mixing: the amount of spill of control into treatment and vice versa.

Ideally, there would be negligible pre-treatment differences in the distributions of a business metric like delivery time across control and treatment groups. Minute lifts or drops post treatment would be detectable. There would be no spilling of control into treatment or vice versa.

2.2 Why not use a t-test?

The t-test is a parametric statistical test for determining whether two samples were drawn from different underlying distributions [4]. It is a standard default approach for most experimentation. As a parametric test it has several underlying assumptions which are violated frequently in practice.

- **Assumption of normal distribution** The t-test uses summary statistics of the two samples to fit a known distribution to each sample. Then these two distributions are compared for overlap to determine the probability that the two samples could have been drawn from the same underlying distribution. Standard t-test approaches fit to a Gaussian (normal) distribution. However, the actual data may not be normally distributed. For example, Figure 1 shows that the target variable in our experiment exhibits a heavy-tailed distribution, and we observe that the matching Gaussian does not imitate the true distribution well. Many practitioners have asserted that due to the central limit theorem the non-normal distribution will approximate a Gaussian as the number of samples increases [7]. However, this is only true if you perform certain clever

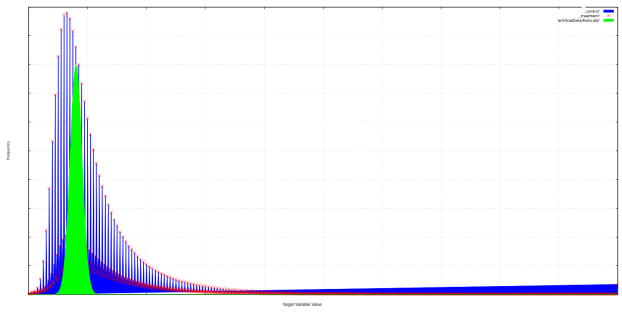

Figure 1: Gaussian super-imposed on observed delivery time distribution. Both control (blue) and treatment (red) overlap and have a heavy-tailed distribution. By comparison the Gaussian (green), or normal distribution with the same mean and standard deviation is much different. The standard t-test approximates our heavy-tailed distribution with this Gaussian distribution.

transformations of the data. Sampling more from a heavy-tailed distribution will not produce data that is normally distributed.

- **Assumption of independence** The t-test assumes independence of data points. This assumption is frequently violated in real world scenarios. In our experiment, delivery time per package is not independent for mixed stops because both control and treatment addresses may be delivered to in the same stop so the attribution of treatment will get mixed into both buckets, clearly violating the independence assumption. Additionally, delivery time in prior stops could affect the delivery time of subsequent stops.
- **Assumption of specific knowledge: Which t-test will you pick?** If the end user is not a scientist, they would find it difficult to pick the right type of t-test. Expecting users to have specific knowledge reduces adoption, so this method doesn't scale. For example, in the Python Scikit-learn implementation [5], the t-test requires making decisions on whether tests are:
  (1) One-sample or two-sample.
  (2) One-sided or two-sided.
  (3) Paired or unpaired (for two-sample tests).
  (4) Homoscedastic (equal variance assumption) or heteroscedastic (for two sample tests).
  (5) Fixed significance level (boolean-valued) or returning p-values.

To test how the t-test works with our Amazon delivery data we ran an A/A experiment that tested 1,000 different potential C/T splits to see if there was an a-priori significant difference between the two groups (no treatment was applied so the two groups should be the same). At the end of our simulation we found that the t-test almost always declared significance when it should not have. The column "% Significant Runs based on t-test ($p < 0.01$)" in Table 1 shows how often it declared significance with $p < 0.01$. If the test had worked, it would be around 1%, as seen in Table 2. Note that this could actually mislead us to think that there was a significant impact when there was no change. This shows that a t-test is not reliable when its assumptions are violated.

Table 1: t−test based significance on real delivery data. Actual figures replaced with order of magnitude.

Randomization Unit (C/T)	Order of Magnitude of # Distinct Groups	% Significant Runs (p<0.01)
Delivery Station	Hundreds	99.90%
Postal Code	Thousands	97.80%
Street	Millions	53%
Buildings	Tens of Millions	79.50%

2.3 A better method: Permutation Test (a Resampling Test)

2.3.1 Permutation Test. Statistical significance tests are intended to determine whether two sample datasets were drawn from different underlying distributions. The t-test is a parametric test that does this through making assumptions on the sample distribution to figure out the probability that the two samples were drawn from the same underlying distribution. In fact there is another nonparametric approach we can take which calculates these distributions directly from the data and estimates their overlap directly. This is an established methodology known as a *permutation test*, with various formulations including bootstrapping and Monte Carlo tests [2].

To understand how permutation tests work we first assume that the null hypothesis is true [3]. In that case the C and T assignment of a given measurement is interchangeable because the treatment had no effect. Therefore we can directly calculate the distribution of differences between C and T distributions under the null hypothesis by randomly sampling C and T groups from the dataset and calculating the difference between the summary statistics of that assignment. We can then compare the true C/T difference to the differences under the null hypothesis to determine if the true C/T test was significantly different. Pseudocode for the approach follows.

Given Data D with original assignment C_{true} and T_{true}, summary statistic S, p-value p, and desired number of resampled permutations R
for i *in* R **do**
  Select C_i and T_i from D;
  Add $S(C_i) - S(T_i)$ to null distribution N
end
Calculate the rank r of $S(C_{true}) - S(T_{true})$ in N. If $r/100 < p/2$ or $r/100 > p/2$ (for a two-sided test) then the treatment had a significant effect.

Algorithm 1: Algorithm for Permutation Test

2.3.2 Scalability. While this test is simple in theory, historically it was not frequently used due to the expense of performing the permutations. However due to advances in computational capabilities we have implemented the permutation test in a distributed library using Dask [6] and have successfully used it in multi-GB datasets involving millions of measurements.

One step described in the process of selecting a randomization strategy is the choice of key to split on. We index the data by granularity key to promote fast joins using Dask's data frames. A Dask DataFrame is a large parallel DataFrame composed of many smaller Pandas DataFrames, split along the index. Batching the

Table 2: Permutation test based significance on real delivery data.

Randomization Unit (C/T)	Order of Magnitude of # Distinct Groups	% Significant Runs (p<0.01)
Delivery Station	Hundreds	1.0%
Postal Code	Thousands	1.0%
Street	Millions	1.0%
Buildings	Tens of Millions	1.0%

data into multiple mini-batches, which is a factor of the number of available processor cores, ensures complete utilization of processing power, accelerating computation. Data that has been processed, gets dropped leaving behind a summary generated by a summarization function, like mean. Arbitrary summarization functions are enabled through an abstract class. The code is heavily parallelized but can run in a few minutes on a laptop on a dataset with millions of rows.

2.3.3 Implementation. We randomly split all data points into C/T buckets allocating 50% to each bucket. We used each group's mean delivery time as our summary statistic. We permuted the treatment assignments of our addresses 1,000 times. We assume that with 1,000 permutations we can effectively approximate the true distribution that we would obtain from testing every possible permutation of the data (which would take a long time since there are an order of tens of millions of events in our data).

As we can see in Table 2 the permutation test performs as expected (by definition), unlike the parametric t-test whose assumptions are violated by our data.

3 EXPERIMENT DESIGN

3.1 Picking a randomization strategy

Controlled experiments, where the experiment set is split between a control and treatment set followed by comparing the effect of treatment on the target variables, is the gold standard for experimentation. A critical decision for randomization between C and T is the choice of key that is used for splitting. In general, the advice has been to make that key as granular as possible for proper randomization (e.g., every delivery, every page load for ads, every session for a customer visiting a website), keeping in mind customer experience and system limitations. In the next section, we describe a methodology for evaluating different keys to provide a way to select the randomization key in order to obtain a fair split.

Previously, in order to find a random split, we would choose a level of granularity that seemed intuitively correct, and then test for pre-existing differences. If at that point there was a pre-existing difference, then we would re-roll the dice and generate a new split, and iterate on this process till a fair split was found. So for example, if we are planning an experiment in week 27 we would use week 26 data and repeatedly select a random C/T set until one was found where there is no statistically significant difference between the C and T. Imagine we iterated 6 times before finding this split. If we used week 25 data, would we have found that this split was fair? If we needed to choose 6 times before finding a good split it is improbable that this splitting methodology produces splits whose fairness quality is stable. What does that mean for week 27, our experimental period? We need a better methodology for choosing a granularity that is likely to produce fair splits from the start.

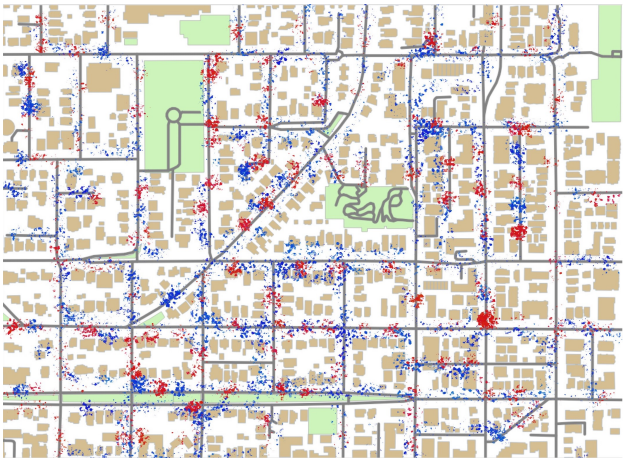

Figure 2: A random split of a part of a city visualized with hypothetical (not actual) delivery events colored by Building Ids, split C/T groups. Every colored pixel represents a delivery event. The red group is control and the blue is treatment. As we can see it is possible that the same delivery stop could contain buildings in both C and T (mixing).

Using the permutation test methodology described above we apply this to the different potential splitting keys available to us for an experiment in changing the delivery locations vended for package delivery. We judge the efficacy of the result based on the criteria described in section 2.1.

3.2 Choice of key

We had a choice of randomizing by the following keys in decreasing order of granularity:

- **Building**: The full normalized string address. eg: 425 106th Ave NE, Anytown, State 12345. It is a grouping context, typically referring to units in a building or complex. Figure 2 shows delivery events colored by Building Id, with a random C/T split. Every pixel represents a hypothetical delivery event. When colored by control and treatment, the figure shows how a split would look. Assume the red group is control, and the blue group is treatment.
- **Streets**: The name of the street. eg: 106th Ave NE. Figure 3 shows delivery events colored by Street, with a random C/T split. Every pixel represents a hypothetical delivery event. When colored by control and treatment, the figure shows how a split would look. Assume the red group is control, and the blue group is treatment.
- **Postal Codes**: The zip code. eg: 98004. Imagine a city split by zip codes with half of them in control and the other half in treatment.
- **Delivery Stations**: The site where packages are received, sorted and prepared for delivery. This is the origin for all deliveries planned for a particular route taken for package delivery. eg: DS-xxx.

Our key metric for this experiment is *Delivery Time*. Delivery time is defined as the time taken to deliver packages at a particular stop/address. It includes time to park at the stop, find packages in the vehicle, walking to and from the delivery point, handing over the packages. Pointing them to the right location to drop

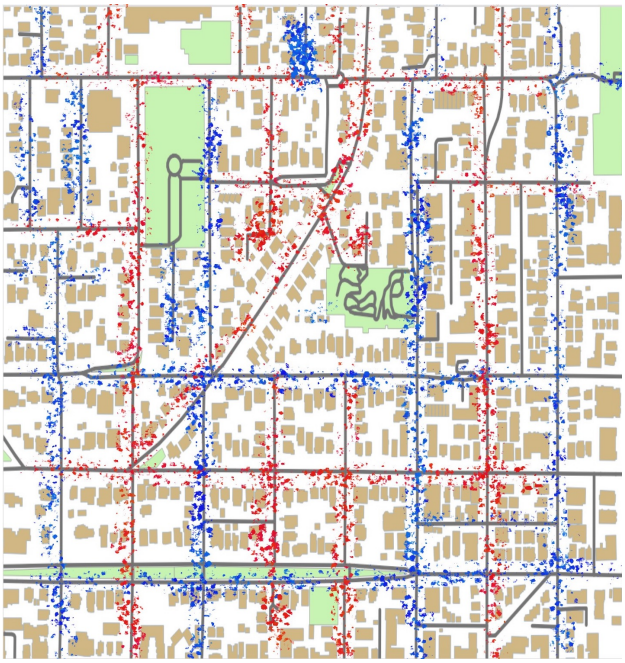

Figure 3: A random split of a part of a city visualized with hypothetical (not actual) delivery events colored by Street Name, split into C/T groups. Every colored pixel represents a delivery event. The red group is control and the blue is treatment. We see it is less likely that a stop would contain both C and T deliveries (mixing).

off the package is optimized by Delivery Point models on which this strategy was first applied.

When addresses are split between C/T for each of these keys we need a metric to measure the desirability of this split. Among these strategies, there are two general risks that make experiments ineffective.

- **Mixing** In GeoSpatial experiments effects are frequently geospatially correlated, so finer granularity results in a larger portion of data with mixed control/treatment effects. When the unit of randomization is fine grained, say at the address level, then neighboring houses could be in different splits, one in C, and the other in T. In this scenario, when a delivery agent stops the van to drop off a package for one of these houses, they also drop off a package at the neighboring house in the same stop thus leaking the benefit of the vehicle stop time for the treatment house, into the control house and vice versa. Intuitively, this should be a frequently occurring problem when the unit of randomization is very fine grained. In Figures 2 and 3, you can look at the overlap between the red and blue pixels and notice that in the building based split, the overlap is a lot more frequent, whereas for streets, the overlap is mostly on street intersections, much fewer in number than buildings.
- **Biased Selection** To avoid the mixing problem, we consider a coarse granularity like postal codes. The only time mixing would happen would be for the rare cases where a stop is on the edge between multiple postal codes. However, in selecting postal codes, we introduce biases in our business metrics like Delivery Time. For example, in Seattle the neighborhood of Queen Anne is more spread out than the densely populated neighborhood of South Lake Union, which means that on a metric like delivery time, we would expect widely variant results even before we apply our treatment, thus polluting the experiment.

Firstly we want to choose a randomization strategy which does not have a high probability of significant differences in A/A testing on historical data. In addition we want to choose a randomization strategy that minimizes the impact of mixing on our detection of effects that are significant to our application. So in our case we want to choose a split that is able to detect important changes in spite of the mixing and bias effects. To estimate this we simulate this by applying the minimum important change to real historical delivery data and then apply the permutation test to see whether the change was detected as statistically significant.

Data To make this concrete, we used actual delivery stops across all of the US. Every row in this joined dataset corresponds to a package delivery event to an address. We aggregated all deliveries made to an address at the same entry time by the same delivery agent, into one delivery time data point. Building ids were used to determine house number and street name of the addresses. We then picked one strategy at a time, and repeated the process for each of them, producing cumulative metrics for every day of data.

Control Treatment Split Streaming this data, we randomly allocated every address into the control or treatment group with a 50/50 split based on the strategy being evaluated. So if we were splitting by postal codes, we allocated, say, every address within 98101 into C and for 98108 into T. The allocation of an address to a group varied per simulation. To account for the variance in this bucketing method, we ran 1000 simulations of allocating an address to a C/T group. Therefore, the same address may be in the control group in one simulation and in treatment in another. This is an integral part of using a permutation test, where a large sample of possible permutations of allocation are considered.

Delivery Time Calculation To calculate delivery time for a delivery event, we took the total vehicle stop time in seconds, divided that by the number of packages delivered at that stop. If a stop contained addresses that were mixed between control and treatment, we made a note of it, and split the delivery time in the appropriate ratio (if out of 5 packages 3 were for a control address then we split the time as 60:40 between C/T).

Distribution of target Variable We worked assumption free about the distribution on the target random variable across both splits.

4 EXPERIMENT RESULTS

After running 1000 simulations over tens of Millions of shipments and calculating the Mean Delivery Time over each of the permuted C/T splits, we analyzed results based on the criteria of a good randomization strategy as enlisted in section 2.1.

4.1 Power & Mixing

Walking through Table 3, the first column is the randomization strategy in increasing order of granularity. As there are lot more houses than postal codes, the number of distinct groups increases (finer granularity) as we go downwards. Since there was no applied difference between Control and Treatment, any difference in their means is entirely due to chance as seen by the tiny difference noted in the Difference (C-T) column . But there can be some

Table 3: Experiment Results Table. Using the permutation test, none of the candidate randomization units have a high probability of a significant difference in A/A testing on historical data. Note that in the finest granularity we can have a great deal of mixed C/T effects. However, in the two finer granularities, we can also detect smaller differences between C and T.

Randomization Unit C/T	Mean difference (C - T)	95% Confidence Interval	% Stops with any mixing	Avg. Mixing per stop
Delivery Station.	0.002	[-1.93, 1.94]	0.00%	0.00%
Postal Code	0.006	[-1.01, 1.03]	0.02%	0.00%
Street	0.000	[-0.13, 0.13]	0.78%	0.30%
Building	0.002	[-0.15, 0.15]	7.03%	3.10%

variation due to random assignments to different groups, and the 95th percent confidence interval in the next column shows how wide this difference can be. Thus we can only declare statistically significant differences that are larger than this interval. However, differences smaller than this interval may be significant to the business, but we may not be able to detect that they had happened if we pick the wrong strategy. This is a distribution resampling method [2], which does not make any assumptions on the shape of the distributions.

From the granularity of split, we note: because the last 2 rows have many more distinct groups, they are sensitive enough to detect smaller differences, performing better on the sensitivity metric. If we randomize by Street, we could detect an improvement or deterioration of greater than 0.13 seconds.

If there were too much mixing (e.g. 50% of stops), we wouldn't see any difference in the delivery times for C vs. T, limiting our statistical power to detect a real difference. As expected, there was no mixing at the Delivery Station level, while it increased with increased granularity. The 7.03% mixing at Building id makes sense because often delivery agents deliver to adjacent buildings in one stop, and 7.03% of them seem to overlap between C/T in our dataset.

Notice that although the Building id has the finest granularity, it does not yield the most sensitive confidence interval in Table 3. This is because 7.03% of the stops experienced mixing of control and treatment groups.

4.2 Statistical Significance, and the effects of Mixing

We wanted to confirm that we would be able to detect actual differences regardless of mixing. Let's say the treatment reduces delivery time, but because of mixing, that reduction also applies to control addresses in the same stop. To test this pre-experiment, we pretended that treatment resulted in the smallest important improvement in delivery time. We did this by artificially reducing the delivery time in the treatment set after the split. In order to maximize the effect of mixing, we also extended that improvement to any control address within the same stop (if it was a mixed C/T stop). Then we ran the permutation test as normal to see if we could detect this improvement. We hoped to find that, even under large amounts of simulated mixing:

(1) We can still detect a sizable difference between the T & C averages.
(2) We find a significant difference under random divisions of T & C (exceeds the 95% confidence interval).

We found that we could detect this difference with Street and Building ids, but couldn't detect it with Postal Code and Delivery Station splits. This ruled out the first two options, because even if an experiment worked well, choosing these units wouldn't give us the confidence that it did work. Between Street and Building id, we noted that Street had the highest sensitivity, least mixing and could clearly detect a change with a small confidence interval. As a result, we choose Street for this dataset from the US.

5 APPLICATION

5.1 Last Mile at Amazon

The first version of this analysis was done to validate if we can come up with a good methodology that is statistically correct, repeatable, and applicable on available data. At Last Mile, having found a useful application with Delivery Time estimates, this is now being applied to other business metrics that indicate customer satisfaction.

We have used this approach for making decisions about experiments in the US and a number of other countries, rapidly with the help of this tool to measure the effectiveness of those experiments. The next sub-section highlights one example of the application of this method to a recent successful experiment. We can now expand this method to be applied to any experiment and find the smallest important difference of significance, and confidently apply it. This approach has been used in various geospatial experiments at Amazon and is extensible to almost any other metric and use case due to its nonparametric nature.

5.2 Application to Production Experiments

While A/A tests on historical data are helpful, the only way to truly tell if a treatment helped the customer is to launch it and do a controlled test to estimate the treatment effect.

Table 5 shows the results of an experiment launch for a large country. Control and Treatment have an even 50/50 split. The first column is the business metric we care about. Here, we only mention the type to preserve business confidentiality. The next column is the difference in Control and Treatment means for each of these metrics. The rank is the rank of the difference in the known range. The "Is Significant?" column shows whether or not the effects observed are statistically significant. As we can see the nonparametric permutation test has been successfully applied to various metrics with different qualities including continuous metrics, binary metrics, and rates/percentages–without changes in formulation.

5.3 Generalizability of the Idea

The permutation test and granularity selection approaches we propose here are general and readily expand to other domains and data types. As a nonparametric test it is straightforward to apply and simplifies the experimental pipeline so non-scientists

Table 4: Effects of Mixing. Given a simulated important change, the change was detectable in two out of four options. While the change was detected as significant, we can see that the greater level of mixing in the Building group impairs our ability to detect fine changes as compared to using the street level of granularity.

Randomization Unit C/T	% Stops with any mixing	95% Confidence Interval of C - T difference in seconds	Real C – T difference in seconds	Improvement Detected?
Delivery Station	0.00%	[-1.93, 2.47]	0.883	**No**
Postal Code	0.02%	[-1.03, 1.06]	0.937	**No**
Street	0.78%	[-0.13, 0.13]	1.204	**Yes**
Building	7.03%	[-0.15, 0.15]	0.957	**Yes**

Table 5: Sample 50% dial up results: Shows statistically significant change in metric #1, #2 and #4. Also shows metric #3 was not improved significantly. Note that even though some of these changes seem small they were being detected as significant by a t-test. Though all of these metrics have different types and some are rare events we are able to run the nonparametric permutation test on all of them to correctly detect significance.

Business Metric	Difference T-C	Rank	Is Significant?	Normal Range (95%)
Metric 1: a continuous number	0.023	98.2	YES	[-0.0212:0.0223]
Metric 2: a binary metric	(0.01032)	0.0	YES	[-0.0057:0.0058]
Metric 3: a percentage metric	0.000005	74.3	NO	[-0.0012:0.0013]
Metric 4: a percentage metric	2.31%	100	YES	[-0.0035:0.0038]

can do experiments easily and count on the robustness of the statistical results.

5.3.1 Outliers with Outsize Impact. In general, this method can be used for distributions that are heavy-tailed or contain significant outliers, or (especially) where independence assumptions are violated frequently enough for t-tests to fail (which can be determined by A/A tests on random splits prior to the experiment). As an example, a common issue in retail is where a popular item can disproportionately drive outcome metrics like number of sales or clicks. The same concept applies to many other applications. Because the permutation test will randomly assign this popular item to C or T over multiple permutations it will account for the fact that a large portion of the expected difference between C and T is due to only one item, resulting in a more accurate assessment of whether the treatment difference is significant or simply the result of one popular item leading the metrics astray.

5.3.2 Non-Normal Distributions. Because the permutation test is nonparametric we have no assumptions on the underlying qualities of the distribution. While we have illustrated a heavy-tailed distribution here we have also successfully applied this test to other types of distributions, including binary/bimodal and standard normal distributions. This means we do not have to worry about characterizing our underlying distribution before applying a significance testing methodology, something which simplifies the experimental process considerably.

6 CONCLUSION

Careful and correct experimentation is key to making the right decisions for systems and organizations. While our computational tools have become more powerful our significance analysis has generally stagnated with the t-test. As we have shown, appropriately applying the t-test is a nontrivial problem (especially if sophisticated scientific expertise is not available), and the t-test can be shockingly wrong when its assumptions are violated. Our results show that it is time for us to re-evaluate the use of nonparametric methods like the permutation test that were previously computationally intractable for most big data use cases. This is particularly important as we work to simplify the process of experimentation and open experimental tools to a larger audience.

In addition to showing how this nonparametric test can be used on a variety of different metrics and use cases in a big data setting, we also show how it can be used to inform other experimental choices such as the choice of split granularity. We provide a decision framework for these types of experimental choices and explore the use of this framework in practice.

ACKNOWLEDGEMENTS

We would like to thank our managers Amber Roy Chowdhury for doing a thorough review of our paper and suggesting edits, Sanjay Kumar and Umar Farooq, for their support and guidance.

REFERENCES

[1] 2020 Don Davis | May 26, 2020 Georg Richter | May 20, 2020 Bloomberg News | Mar 30, 2020 Harry Drajpuch | Mar 12, and 2020 Bloomberg News | May 20. 2020. Amazon is the fourth-largest US delivery service and growing fast. https://www.digitalcommerce360.com/2020/05/26/amazon-is-the-fourth%E2%80%91largest-us-delivery-service-and-growing-fast/

[2] BT Efron and RJ Tibshirani. 1994. An Introduction to the Bootstrap. New York, NY: Chapman & HallHall. *CRC Monographs on Statistics & Applied Probability* (1994).

[3] Ronald Aylmer Fisher. 1936. Design of experiments. *Br Med J* 1, 3923 (1936), 554–554.

[4] B. Guo and Y. Yuan. 2017. A comparative review of methods for comparing means using partially paired data. *Statistical Methods in Medical Research* 26 (2017), 1323 – 1340.

[5] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. *Journal of machine learning research* 12, Oct (2011), 2825–2830.

[6] Matthew Rocklin. 2015. Dask: Parallel Computation with Blocked algorithms and Task Scheduling. In *Proceedings of the 14th Python in Science Conference*, Kathryn Huff and James Bergstra (Eds.). 130 – 136.

[7] Eugene Seneta et al. 2013. A tricentenary history of the law of large numbers. *Bernoulli* 19, 4 (2013), 1088–1121.