UNIVERSITY OF CALIFORNIA
Los Angeles

A comparison of tests for online experiments

A thesis submitted in partial satisfaction
of the requirements for the degree
Master of Applied Statistics

by

Alan Dsouza

2022

ABSTRACT OF THE THESIS

A comparison of tests for online experiments

by

Alan Dsouza
Master of Applied Statistics
University of California, Los Angeles, 2022
Professor Hongquan Xu, Chair

Online experiments have grown in popularity but the techniques used to evaluate them have not adapted to the continuous stream of results. The goal of this review is to analyze the limitations of current online experiment tests and evaluate newer techniques that are better suited for continuous assessment. Conducting tests on simulated experiments showed that peeking at results can cause 3 times as many false-positives when using t-test's. The conservative nature of multiple comparison adjustments led to rejecting over 50% of winning ideas. Mixture Sequential Probability Ratio Tests (mSPRT) resulted in few Type-I errors even when monitoring results continuously. Using mSPRT led to 1/6th as many Type-I errors. There are known downsides to mSPRT including implementation complexity and computation costs, but these are likely smaller than the value created from having a more reliable analysis technique.

The thesis of Alan Dsouza is approved.

Maria Cha

Frederic Schoenberg

Chenlu Shi

Hongquan Xu, Committee Chair

University of California, Los Angeles

2022

To my wife, you are the joy of my life. Your love gives me the courage to dream big. Thank you for your endless support.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

Introduction

In recent years, the number of online experiments conducted by consumer technology companies has grown exponentially. Online experiments - also known as A/B tests, split tests, and multivariate tests - have become an essential tool in the product development toolkit. Minimal cost and access to large sample sizes has fueled the growth of "test and learn". Engineers and product managers test even the smallest of features to websites and mobile applications to measure effectiveness. Marketers are using tests to determine which creative resonated with consumers the most. It is worth emphasizing that *learning* is the key goal of experimentation. As such, evaluators of experiments, usually analysts, rely heavily on statistical tools to separate the signal from the noise. The rapid growth in experimentation has not been accompanied by a comparable growth in statisticians, which means that for many of the people conducting these experiments, this will be their introduction to experiment analysis. These first-time practitioners have access to an abundance of techniques, but the exact implementation details are not always clear.

The good news is that there is a mountain of research that has been conducted on experiment analysis and an accompanying amount of academic papers. What is missing is a comparison of various techniques; so, irrespective of which test researchers choose, they are aware of the benefits and pitfalls of the chosen approach. For example, peeking at experiments that are intended to be evaluated using a t-test dramatically increases the odds of Type-I errors; it is unlikely that this is widely known. Beyond declaring if treatment performed better than the baseline experience, practitioners are also interested in why there

might be a difference in performance. For A/B tests, more often than not, analyzing the performance differences between sub-segments of the test population can help make sense of the cause. For example, if a new feature performs better on mobile devices rather than desktops, it might suggest that the mobile version is easier to use, and thus leads to better performance. Part of this review will explore how to properly conduct multiple comparisons to isolate differences in segments of users.

The goal of this review is to compare the techniques most commonly used to analyze experiments. In doing so, some benefits and drawbacks of each method will become apparent. Since we are undertaking the implementation of all of the techniques in Python, the final toolkit will be open-sourced, making it readily accessible to practitioners [Dso].

CHAPTER 2

Data and Exploratory Analysis

Evaluating analysis techniques requires a set of experiments where the exact details within each experiment are known. This makes machine generated data based on industry averages the ideal dataset for this study. Having thought through various possible outcomes (see Simulating Experiment Data) of a test, we can use Python to randomize input parameters and simulate hundreds of experiments, each with thousands of users. Saving the input for each simulated experiment ensures the true parameters are known. Each test is then analyzed using various techniques and the conclusion is compared with the real treatment effect or lack thereof. For example, if 1,000 experiments are simulated and experiment 413 is a case where only one sub-segment has a statistically significant treatment effect, the question is presented: which techniques are able to accurately identify this data?

Even though the data is simulated, parameters must still be provided for the distribution of each of the fields. These parameters are informed by prior knowledge of experiments in the fintech space. The findings of this review are within the constraints of these parameters. Since there are an infinite combination of parameters, it is impossible to model every scenario and thus arrive at which technique is universally the best. The code accompanying this review allows for simulating and testing experiments under a different set of parameters that are more representative of other companies or industries.

2.1 Background

The author's experience with experimentation is based on time spent working in fintech. At a typical fintech company, product managers, designers, and engineers work closely together to develop new features. While analysis and research goes into designing these features, it is still unclear if users will find it useful, which is why the feature is first tested using an online experiment before it is rolled out. The experiment can be rolled out to a portion of users, and these users are then split (usually evenly) between treatment and control. Control is the current state of the application (app), while treatment is users who see the new feature in addition to the regular app. Next, a metric on which treatment versus control can be compared is chosen. At most companies, this will be either conversions (units) or revenue (dollars); something that directly increases business value. For example, a fintech app that generates revenue by referring users to financial products will count every successful referral as a conversion. Once the experiment is live and the primary metric becomes evident, the performance of treatment versus control is measured to see if it is meaningfully better. If it is, the next step is to figure out why it is performing better. In most instances, the feature resonates with a sub-segment of users, which then lifts the aggregate treatment effect. Identifying these segments helps companies learn about which type of features resonate with which audiences. If the treatment performs better overall, it is shipped to all users and becomes the new default.

Since winning features usually generate more revenue for a company, there is an incentive to conclude an experiment as soon as possible. Wait too long to conclude on a highly positive feature and the company loses out on incremental revenue. Similarly, leave a highly negative treatment running for too long and the company loses revenue. So clearly, speed is a valuable feature for measuring A/B tests, but that also means an increased risk of Type-I and Type-II errors. The ideal experiment evaluation technique allows for continuous evaluation, address speed, and minimizes errors. In the next chapter, techniques that could provide the accuracy

of t-tests while allowing for evaluation without having to wait to collect a predetermined sample size are discussed.

2.2 Simulating Experiment Data

Every experiment has a minimum of three components: a randomization unit, the treatment, and a measurable response. For online experiments, the randomization unit is usually a user, treatments are assigned randomly, and the response is usually a business metric that is either binomial (e.g. conversions) or continuous (e.g. revenue).

The process of simulating an experiment starts by selecting how many users are in the experiment using a uniform distribution. Each user is then assigned a sequential identifier (column user_id in the data), whether they are in Control or Treatment using random choice (group), and the date they were first exposed to the treatment (event_date). Whether the user converted or not is decided by sampling from a binomial distribution (convs). If they did convert, the revenue amount is sampled from an exponential distribution (revenue). As mentioned in the background, treatments can have varying impact on different segments of users, so all users are assigned into some predefined set of segments (platform and scoreband).

To evaluate the various analysis techniques, their effectiveness against the outcomes commonly observed in online experiments are evaluated. The two most common outcomes are that the treatment either has a statistically significant improvement in the primary metric of interest, or the treatment has no effect or a negative effect. Another common outcome is that the treatment might have a meaningfully positive impact, but only in a segment of users. It is unrealistic to expect the treatment has the exact same impact across all users, which is why evaluating experiments by segments is a common practice. Lastly, the experiment could have a novelty effect where the treatment is positive initially but fades to zero over time.

2.3 Exploratory Data Analysis

The dataset is a collection of 1,000 simulated experiments. The number of users in each experiment is randomly selected from a uniform distribution. Each user is included only once in the experiment, irrespective of how often they use the feature. Figure 2.1 shows the distribution of users across all the experiments. Part of this review includes evaluating the performance of tests across various sample sizes.

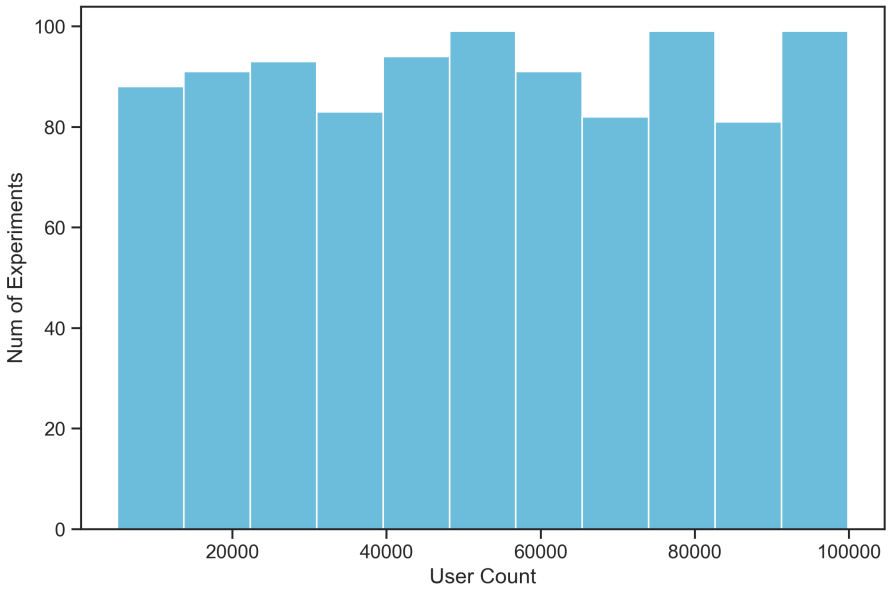

Figure 2.1: Experiment count by number of users

Each experiment is made up of users who have certain user level attributes and actions that are of interest. A sample of the data within each experiment is shown in Table 2.1. Each user has a unique identifier, allowing for group size calculation, which is then used in statistical tests. Next, the user is randomly assigned a group; this is the treatment effect. This could be a single treatment, or multiple in the case of multivariate experiment. For these experiments, 50% of users were randomly assigned a treatment called "variant1". Those not assigned the treatment were tagged as "control".

Users were then randomly assigned attributes such as the platform they are using to

Table 2.1: Sample of experiment result dataframe

user_id	group	event_date	platform	scoreband	convs	revenue
8	variant1	2022-03-12	iOS	nearprime	1	54.56905
9	control	2022-03-19	iOS	subprime	0	0.00000
10	variant1	2022-03-02	iOS	subprime	0	0.00000
11	control	2022-03-17	Mweb	prime	1	129.64000
12	variant1	2022-03-31	Android	nearprime	0	0.00000

interact with the product and their credit score band. Figure 2.2 and Figure 2.3 show the distribution of users across various attributes. The split amongst segments is not uniform, and is instead distributed based on prior observations within fintech companies. When randomizing the platform variable, it is assumed that 55% of users use iOS applications, 25% Android, 15% a mobile web browser, and 5% a desktop. For scoreband, it is assumed that 40% of users are subprime (score less than 600), 30% are near-prime (score 600-720), and 30% are prime (score above 720). Across companies and industries, practitioners are likely to use many attributes, each with different distributions, when assessing experiment results. Attributes, or dimensions, are relevant because in many instances the treatment effect can vary across segments of users. Some experiments have been simulated to have a lift within one of these segments. Three multiple comparison techniques are evaluated to understand how effective they are at identifying treatment effect within segments.

Each of the simulated experiments is based on a scenario detailed in section 2.2. Figure 2.4 shows the distribution of number of experiments by true lift (as randomly selected from a normal distribution), grouped by scenario. Experiments with no real treatment effect, referred to as neutral, have a true effect of 0, but could still result in false-positive conclusions.

An important aspect of A/B testing is the value of knowing if the treatment is effective as soon as possible. This inevitably leads to peeking (as discussed in Problem with peeking) at performance and drawing conclusions. Doing so invalidates many statistics principles

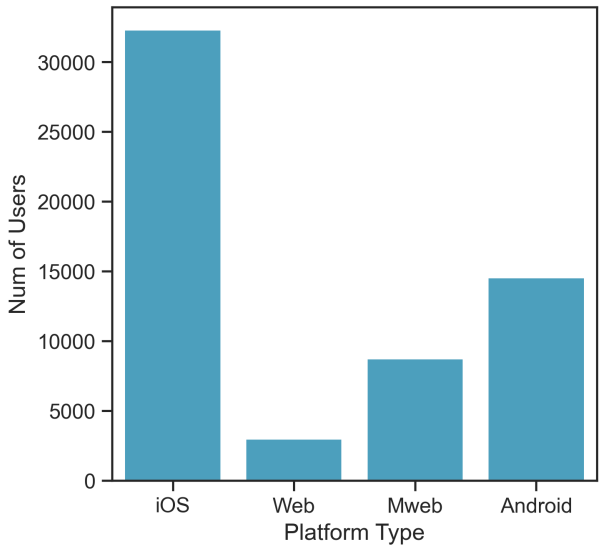

Figure 2.2: Platform distribution

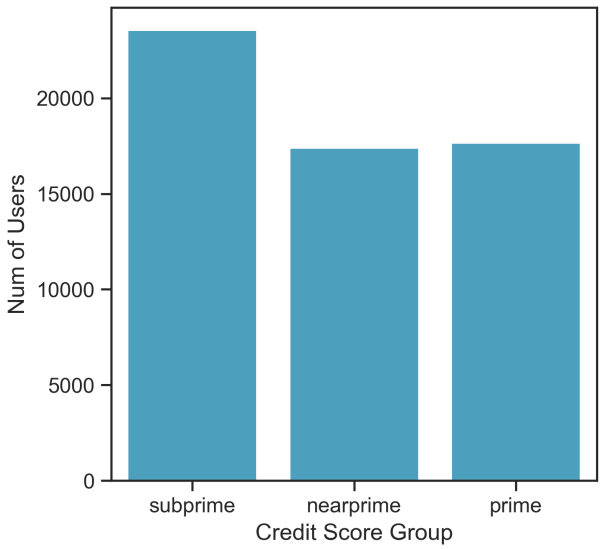

Figure 2.3: Scoreband distribution

and can lead to incorrect conclusions because of increased Type-I errors. Figure 2.5 is the conversion rates for one experiment. At first glance, especially when looking at the first few days of data, there is no clear impact from the treatment. In actuality, in this experiment, the treatment has a 1.9% lift over control. Since peeking invalidates the statistical basis for t-tests, we will evaluate a Bayesian approach and mixture sequential probability ratio test, both of which allow for continuous monitoring and measurement.

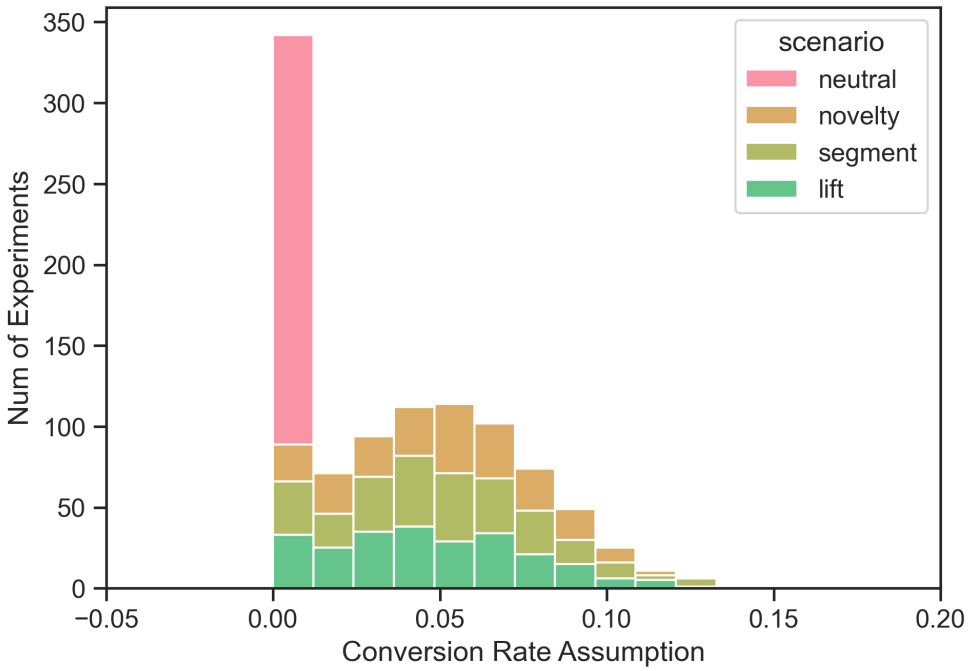

Figure 2.4: Treatment effect distribution by scenario

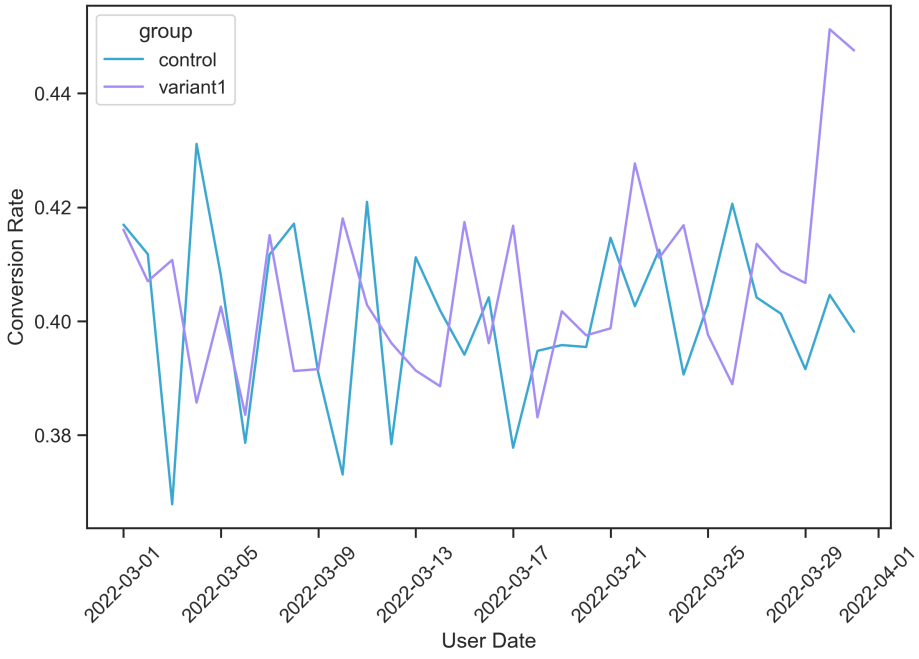

Figure 2.5: Treatment effect trended over time

CHAPTER 3

Tests for Binomial Metrics

3.1 T-Test

As online experiments gained popularity in the early 2000's, there was a need to develop tests to measure if the observed treatment effect was real or simply random noise. Several articles on the subject of t-test efficacy were assessed, but there is no clear explanation for why the t-test became the test of choice. Even today, a google search for "ab test significance calculator" yields hundreds of results offering t-test based calculators. The popularity of a frequentist approach is not surprising; the test needs few inputs and is easy to compute. When dealing with a binomial metric, if the sample size of each group and the number of group members with a successful event (e.g. conversion) are both available, the treatment effect is calculable, as well as whether it is significant or not and the confidence interval for the treatment effect. The formulas are given below.

$$control\ conversion\ rate = p_c, variant\ conversion\ rate = p_v$$

$$lift = \theta = \frac{p_v}{p_c} - 1$$

$$\sigma_c = \sqrt{\frac{p_c * (1 - p_c)}{sample_c}}, \sigma_v = \sqrt{\frac{p_v * (1 - p_v)}{sample_v}}$$

$$\sigma_{\text{Diff}} = \sqrt{\sigma_c^2 + \sigma_v^2}$$

Now that the variance of the individual independent samples as well as the variance of their difference has been elucidated, the T statistic value is calculated and used to find the p-value, where

$$T\ statistic = \frac{p_v - p_c}{\sigma_{\text{Diff}}}$$

The last step is to then find the confidence interval for θ.

$$\theta\ upper\ bound = \theta + (T\ critical * \sigma_{\text{Diff}})$$

$$\theta\ lower\ bound = \theta - (T\ critical * \sigma_{\text{Diff}})$$

Even the mighty t-test has limitations, especially in the context of online experiments. The first limitation is a misunderstanding of what p-values represent. When conducting experiments, companies are interested in knowing whether the treatment (new feature) is better as well as the precise value created from the treatment. Once the treatment effect, a point estimate, is measured, analysts and business stakeholders then interpret the p-value being the probability of the measured treatment effect being accurate. However, p-value is defined as the probability - under the assumption of no effect or no difference (null hypothesis)

- of obtaining a result equal to or more extreme than what was actually observed [Dah08]. This true meaning of p-value does not address what businesses need. The options are to educate everyone in calculating and reviewing test results or to use alternative methods that directly address what A/B testers are looking for. This limitation of t-tests is echoed by industry expert, Chris Stucchio [Stu15]. The other limitations are discussed in the Problem with peeking and Multiple Comparisons sections.

3.1.1 Problem with peeking

Peeking, or reviewing the results before the predetermined sample size has been collected, is another pervasive issue with the t-test approach [JKP17]. In a business setting, speed is money. Being able to quickly identify winning ideas and improve on them can have a meaningful impact on revenue. Conversely, delaying decisions can cause bottlenecks, slowing down other ideas from being tested. Combine this value of speed with the ease of continually monitoring experiments and many situations present where experimenters check p-values daily and call the experiment a success as soon as it reaches significance, thus undoing all the work that went into articulating a hypothesis, establishing a minimum detectable effect, and then calculating the required sample size.

Intuitively, the best way to think of a t-test is that it takes into account a baseline rate and the minimum detectable effect that is expected from the treatment and then recommends a sample size. This sample size ensures that there is enough data to plot the two distributions and control for a maximum of 5% Type-I error. Thus, peeking, or checking before the sample size is collected, dramatically increases the false-positive rate. Figure 3.1 shows a sample experiment where there is no real lift, yet the p-value is statistically significant for a brief period.

The ideal solution is not for researchers to pretend like they do not have access to real-time metrics and should just wait until the sample size is collected. Instead, techniques that embrace continuous evaluation should be considered. The error rate caused by peeking is

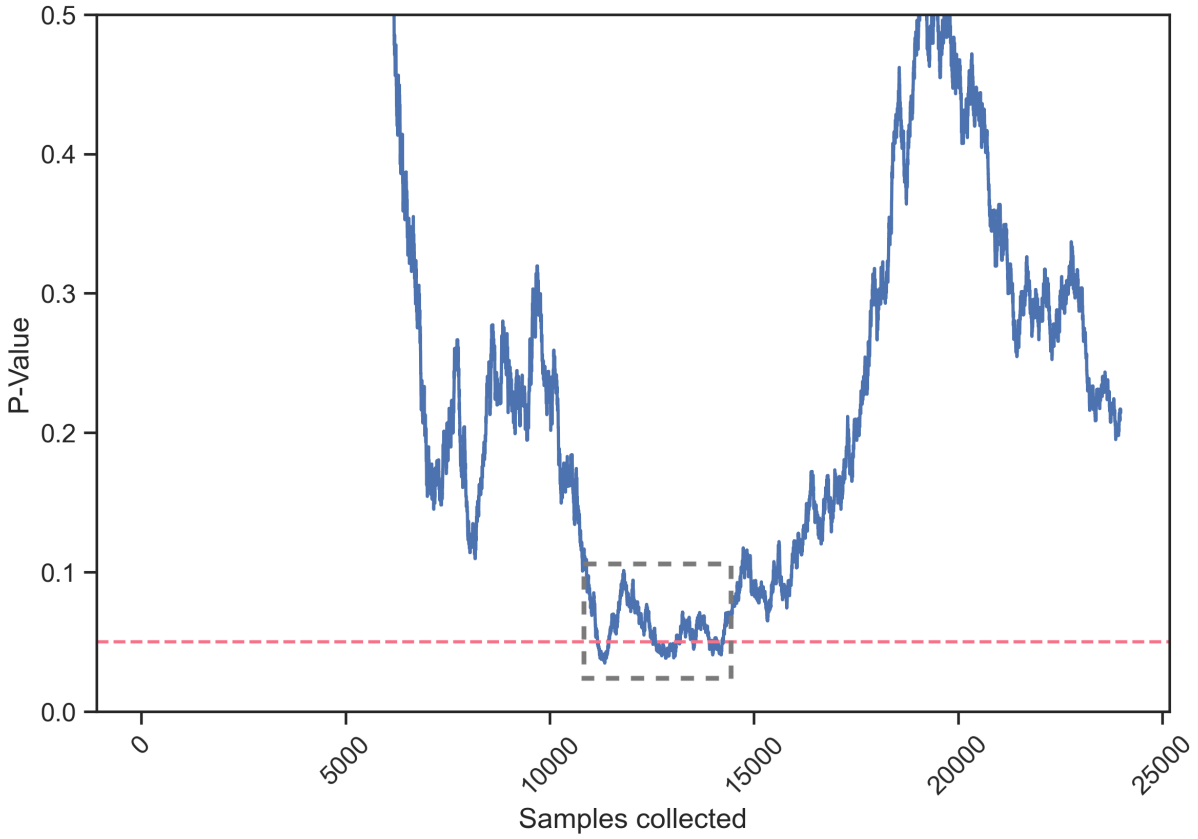

Figure 3.1: P-value by sample size for an experiment with no lift shows a window of sample size where the p-value was significant

discussed in Results.

3.2 Multiple Comparisons

Multiple comparisons, or multiple testing, is a method in which multiple parameters and/or subsets of data are analyzed for the same experiment. Multiple comparisons are a necessity when conducting A/B experiments because, in addition to knowing if the treatment is better or not, companies want to learn why it is better. The best way to identify the cause is by analyzing multiple events such as clicks, applications, conversions, and revenue. It is very common to find that certain features resonate differently amongst various segments of users.

The more items that are compared within an experiment, the more likely it is to find some that appear statistically different; this is known as the family-wise error rate (FWER). So researchers need to find ways to balance uncovering insights from the data with keeping error tolerance α within a reasonable bound. Three techniques that control for FWER are discussed in the following subsections. The performance of each are evaluated in the Results section.

A detailed summary of the three techniques is shown in Figure 3.2. The alpha reference line shows the maximum amount of tolerance for each test, which leads to an increase in Type-I errors. Benjamini-Hochberg splits the area, leading to a balance between Type-I and Type-II errors. Holm-Bonferroni and Bonferroni are even more conservative.

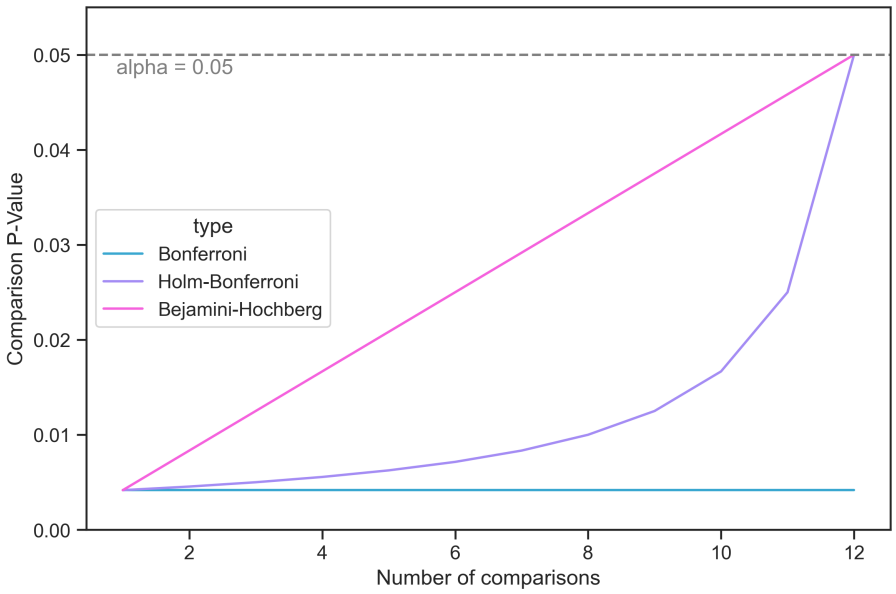

Figure 3.2: Adjusted p-value difference between Bonferroni, Holm-Bonferroni, and Benjamini-Hochberg assuming 12 comparisons

3.2.1 Bonferroni Correction

The simplest multiple comparison adjustment is the Bonferroni Correction technique [Bon36] named for famed Italian mathematician Carlo Emilio Bonferroni. This method recommends that for an experiment with m comparisons and a desired α to use an $adjusted\,\alpha = \frac{\alpha}{m}$. This adjustment is built on Boole's inequality which says that for any finite or countable set of events, the probability that at least one of the events happens is no greater than the sum of the probabilities of the individual events [Boo47]. It is easy to see that this approach is very conservative. Assuming the researcher desires 12 comparisons across various metrics and dimensions, that suggests an adjusted alpha of just 0.0041. This threshold is very difficult to achieve, even when there is a real treatment effect. So the adjustment will definitely reduce Type-I errors, but likely also increases Type-II errors. The benefit of this approach is simplicity while its drawback is its strictness.

3.2.2 Holm-Bonferroni Adjustment

The Holm-Bonferroni method attempts to keep total Type-I errors under the value of α while still minimizing Type-II errors.

The process starts with calculating the p-value for each of the m comparisons, giving us $P_1, P_2, ..., P_m$. These are sorted from lowest-to-highest. For a desired α, using the sorted p-values and starting at 1, calculate

$$P_k < \frac{\alpha}{m + 1 - k}$$

If true, reject H_k and continue to the next value. When the result is false, stop the loop and fail to reject all the other comparisons. This approach is as strict as Bonferroni on the smallest p-value, but relaxes the constraint slightly for each subsequent comparison. Due to its lower Type-II errors than the Bonferroni correction, the Holm-Bonferroni technique should result in better overall accuracy.

3.2.3 Benjamini-Hochberg Adjustment

Rather than focus on FWER, Benjamini-Hochberg proposed the False Discovery Rate (FDR) metric [BH95]. Where FWER is concerned about getting one or more false positives in an experiment, FDR is the proportion of discoveries that are false. The latter is a more balanced approach to Type-I and Type-II errors. It accepts more false-positive risk (which is not very harmful in a software experimentation setting, unlike healthcare, where the stakes are higher), while reducing false-negatives, which could be more costly to companies since they would reject potentially lucrative ideas.

The beginning of the process is very similar to Holm-Bonferroni. Calculate the p-value for each of the m comparisons, giving us $P_1, P_2, ..., P_m$. These are sorted from lowest-to-highest. For a desired α, using the sorted p-values and starting at 1, calculate

$$P_k \leq \frac{k}{m} * \alpha$$

If true, reject H_k and continue to the next value. When the result is false, stop the loop and fail to reject all the other comparisons. As seen in Figure 3.2, Benjamini-Hochberg has a linear adjustment with more comparisons, while Holm-Bonferroni has a parabolic shape, making it more conservative.

3.3 Bayesian Approach

Increasingly, Bayesian techniques are gaining popularity for A/B testing. By adapting to new information and adjusting assumptions based on the observed distribution, a Bayesian approach does not need to wait to collect some predetermined sample size, and thus reduces Type-I errors that we discussed in the Problem with peeking section. Another benefit is better interpretability. P-values represent the probability of seeing a result at least as extreme as the observed effect, when in reality researchers want to know the probability of treatment being better than control. The intuitive reasoning for using a Bayesian approach is best

explained by Stucchio, 2015 [Stu15]. In its simplest form, it involves changing your opinion as more evidence is collected. For an A/B test, each new sample should inform the claim that treatment is better or not. Let λ be our treatment parameter of interest (which for this review is conversion rate).

$$P(\lambda|evidence) = \frac{P(evidence|\lambda)P(\lambda)}{P(evidence)}$$

However, using a Bayesian approach does have a couple of drawbacks. First, it requires the knowledge of a prior distribution. Secondly, summing up the probability is essentially equal to finding the cumulative distribution using an integral which can get mathematically complicated and computationally expensive. Thankfully, it can be shown that the Beta distribution does a good job of approximating the prior distribution of a binomial parameter, turning the integral into a less intimidating closed form equation. Detailed steps are shared by Miller [Mil], arriving at the final equation shown below.

$$if\ \alpha = number\ of\ successes, \beta = number\ of\ failures$$

$$p_A \sim Beta(\alpha_A, \beta_A)$$

$$p_B \sim Beta(\alpha_B, \beta_B)$$

$$Pr(p_B > p_A) = \sum_{i=0}^{\alpha_B - 1} \frac{B(\alpha_A + i, \beta_A + \beta_B)}{(\beta_B + i)B(1 + i, \beta_B)B(\alpha_A, \beta_A)}$$

3.4 mixture Sequential Probability Ratio Test (mSPRT)

As discussed in the Problem with peeking section, since Type-I error inflation is a pervasive problem, researchers have been exploring alternative techniques to measure treatment effects. One of the earliest solutions to continuous evaluation was the Sequential Probability Ratio Test (SPRT), proposed by Abraham Wald in 1945 [Wal45]. This approach calculates the test statistic after each sample is collected, and a ratio statistic is computed. This statistic

is then compared to a tolerance threshold, resulting in a decision to either keep the test running to collect more samples, or accept the treatment, or accept control.

$$H_0 : p = p_0, H_1 : p = p_1$$
$$\Lambda_k := \prod_{i=1}^{k} \frac{p_1(X_i)}{p_0(X_i)}, k = 1, 2, ...$$
$$S_i = S_{i-1} + \log \Lambda_i$$

After each S_i is calculated, we can compare it to our tolerance threshold to see if the result is significant, and if it is, the test can be stopped. In notation, this means,

$$If\ a < S_i < b : keep\ running$$
$$else\ if\ S_i >= b : Accept\ H_1$$
$$else\ if\ S_i <= a : Accept\ H_0$$

Where a and b depend on the desired Type-I and Type-II error tolerance. The most commonly used Type-I and Type-II error probabilities are $\alpha = 0.05$ and $\beta = 0.2$

With the rise of online experiments, the sequential technique was revisited by several industry experts working on tools that are used by many companies [AM17]. One newer adaptation of SPRT is the mixture Sequential Probability Ratio Test first proposed by Herbert Robbins in 1970 [Rob70]. It has since been adapted for modern A/B testing by Johari, Pekelis, and Walsh in partnership with Optimizely, one of the largest A/B testing platforms[JPW19]. The proofs from the aforementioned papers (seen below) discussed and implemented in C++ and R by Stenberg in 2019 [Ste19]. Using this implementation, the approach was replicated in Python in order to compare the performance of mSPRT against other approaches.

$$If\ \theta = treatment\ effect,\ then\ likelihood\ ratio\ \Lambda = \frac{f_{\theta_1}(x_n)}{f_{\theta_0}(x_n)}$$

$$Let\ \pi(\theta) > 0\ denote\ mixture(prior)\ distribution$$

$$\tilde{\Lambda}_n = \int_{\theta\in\Theta} \Lambda_n \pi(\theta) d\theta = \int_{\theta\in\Theta} \prod_{i=1}^{n} \frac{f_\theta(x_i)}{f_{\theta_0}(x_i)} \pi(\theta) d\theta$$

$$\mathbb{P}_{\theta_0}\left[\tilde{\Lambda}_n > \frac{1}{b}, n \geq 1\right] \leq b\ for\ any\ b > 0$$

$$inf\left[n : \tilde{\Lambda}_n < \alpha^{-1}\right]\ becomes\ stopping\ rule$$

$$p - value\ as\ p_n = min\left[1, min(\tilde{\Lambda}_t^{-1} : t \leq n)\right]$$

$$\mathbb{P}_{\theta_0}\left[p_n \leq \alpha\right] \leq \alpha\ for\ any\ n$$

Notice that only α needs to be specified, which for the purpose of this review $\alpha = 0.05$. Since Type-II error probability β is not specified, the test can run for as long as needed. Alternatively, a sample size threshold can be established after which the test is stopped.

CHAPTER 4

Results

4.1 T-Test

Before reviewing the results, it is worth restating the various scenarios. Lift is where there is a real treatment effect. Neutral is where there is no effect. Novelty is a situation where there is an initial treatment effect but quickly fades away. For evaluation purposes, Novelty should be considered the same as Neutral. The last scenario is a Segment lift, in which case there is a real lift within a segment of users that should be detected.

Table 4.1 shows t-test results when administered after all samples are collected. Its accuracy lives up to expectations and accurately identifies 78.6% of Lift experiments as having a statistically significant difference in performance. The 21.4% false negative rate, or Type-II errors, are also aligned with the expectation of 80% power. These statistics are a good reminder that a significant difference may remain undetectable for as many as 1 in 5 ideas. In the Neutral scenario, the t-test claimed significance only 4.3% of the time. This is expected due to the use of an α of 0.05 for our analysis. In the novelty scenario, an 18.6% false positive rate is seen. This is not entirely surprising since there is a real lift in the beginning, which points to the complexity of this scenario. Lastly, in the segment scenario, the overall t-test rejects 61.2% of experiments. This is a particularly bad outcome because it might lead to the rejection of good ideas that resonate with a subset of users. The Segment scenario is revisited in the review of multiple comparisons results. Following is a discussion of these same metrics while simulating peeking, where the results are checked every 3 days.

Table 4.1: T-Test accuracy results

Scenario	Num of experiments	T-Test Significant	Type-I Error	Type-II Error
Lift	242	190	-	21.5%
Neutral	253	11	4.3%	-
Novelty	237	44	18.6%	-
Segment	268	104	-	61.2%

4.1.1 Peeking

As expected, Peeking, which in this case means checking the result every 3 days, increases the chances of finding a winning effect; thus, in the true lift scenario, leads to fewer Type-II errors (18.6% vs 21.4%). The main issue however, is the drastic increase in Type-I errors in the Neutral scenario; 14.2% vs 4.3% - which is 3.3 times higher than the t-test! 14.2% is also significantly higher than our alpha threshold. These results highlight the issue with using t-tests in a continuous evaluation setting. Researchers should be motivated to find better techniques for evaluating experiments. Peeking does especially poorly in the Novelty scenario with a 55.7% Type-I error rate. This would lead researchers to celebrate over 50% of ideas as winners yet fail to observe an improvement in business metrics. In the Segment scenario, peeking's generosity leads to fewer Type-II errors.

Table 4.2: T-Test accuracy when peeking every 3 days

Scenario	Num of experiments	T-Test Significant	Type-I Error	Type-II Error
Lift	242	197	-	18.6%
Neutral	253	36	14.2%	-
Novelty	237	132	55.7%	-
Segment	268	137	-	48.9%

4.2 Multiple Comparisons

As discussed in the Multiple Comparisons section, identifying user segments where the treatment resonated is valuable information that helps inform future investments. The permutation of 4 Platforms and 3 Scorebands means a total of 12 distinct segments were evaluated. Unfortunately, as seen in Table 4.3, none of the three techniques evaluated did a particularly good job of identifying a segment level effect. Of the 268 simulated experiments, Bonferroni correction correctly identified the segment for 126 experiments. Holm-Bonferroni performed identically and also identified 126 experiments. Benjamini-Hochberg, which we know is the most forgiving approach, performed marginally better and identified 128 experiments. This translates to a Type-II error rate of over 52%. For a business, wrongly walking away from 52% of winning ideas is extremely costly.

Table 4.3: Multiple comparison successful identification of segment with treatment effect

Num of experiments	Bonf	Holm-Bonf	Ben-Hoch
268	126	126	128

Looking further into factors that might have impacted accuracy, Benjamini-Hochbergs results were plotted against the true lift within the segment (Figure 4.1) and the sample size of the experiments (Figure 4.2). As expected, the accuracy is better when the lift is larger and/or when the sample size is larger. In these simulated instances, a lift above 6% or sample above 80,000 users leads to fewer Type-II errors. In the future, for experiments that do not meet these thresholds, it is important to rely more on observed data rather than reject ideas entirely due to lack of statistical significance. Companies can undertake a similar exercise of simulating experiments based on their baseline rate, number of users, and expected treatment effect to understand the baseline needed for multiple comparison techniques to be able to detect an effect.

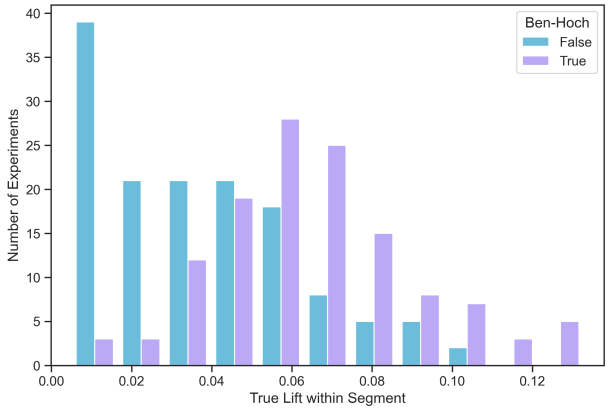

Figure 4.1: Benjamini-Hochberg accuracy by lift magnitude

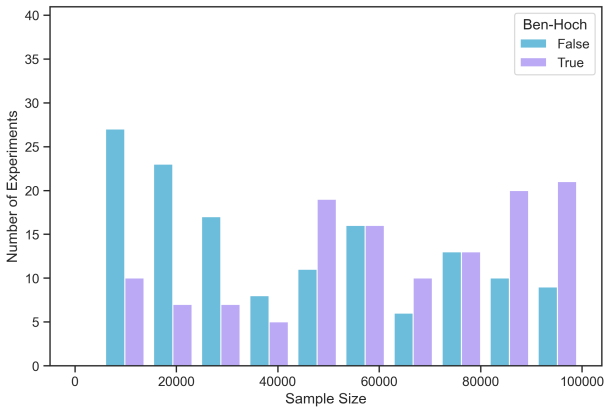

Figure 4.2: Benjamini-Hochberg accuracy by sample size

Table 4.4: Bayesian test accuracy results

Scenario	Num of experiments	Bayesian Significant	Type-I Error	Type-II Error
lift	242	190	-	21.5%
neutral	253	10	4.0%	-
novelty	237	44	18.6%	-
segment	268	104	-	61.2%

4.3 Bayesian

The performance of the Bayesian test, as seen in Table 4.4, is near identical to the t-test. It demonstrates a marginally lower Type-I error rate of 4%, while all other metrics are identical. Even though the approach is dramatically different, the results are the same as the t-test.

Even when peeking, the Bayesian approach has similar results to a traditional t-test as seen in Table 4.5. This underwhelming performance could be due to the simplifying assumption of using a Beta distribution rather than incorporating real observed priors into the model. Adding priors increases the complexity since researchers would have to retrieve historical data for all the metrics they are testing, which might be infeasible. The added complexity makes the Bayesian approach less appealing to those without the time, skills,

Table 4.5: Peeking impact to Bayesian test accuracy

Scenario	Num of exp	Bayesian Peeking Sig	Type-I Error	Type-II Error
Lift	242	196	-	19.0%
Neutral	253	36	14.2%	-
Novelty	237	131	55.3%	-
Segment	268	137	-	48.9%

and resources to build an automated framework. The bottom line, however, is that since these results are similar to a basic t-test, it does not deliver much added value when using the Beta distribution approach.

4.4 mSPRT

mSPRT falls under the broader Bayesian umbrella but has a very different implementation, which is why there are very different results across the board. The Type-II error rate at 35.5% is higher than all other approaches, but results show significantly lower Type-I errors even in the Novelty scenario. In general, since mSPRT is designed for real-time evaluation, it is the most effective approach when measured on minimizing Type-I errors. On inspecting the Type-II errors it was found that the test ran out of sample, so the insignificant conclusion is more a factor of running out of sample size. If tests were to be left running when needed, fewer Type-II errors are expected. In instances where mSPRT correctly identified a treatment effect, it did so on average with just 30% of the sample in the experiment. The ability to conclude experiments up to 70% sooner is incredibly valuable for businesses. This speed to result combined with continuous evaluation make mSPRT a compelling front-runner for measuring A/B tests.

It is worth mentioning that a more practical cost of the mSPRT approach is its compute cost. Since it is an iterative approach that calculates a statistic after each observation, it is a computationally heavy process. While a t-test can be conducted in less than 1 second,

Table 4.6: mixture Sequential Probability Test accuracy results

Scenario	Num of experiments	mSPRT Significant	Type-I Error	Type-II Error
Lift	242	156	-	35.5%
Neutral	253	6	2.4%	-
Novelty	237	15	6.3%	-
Segment	268	50	-	81.3%

mSPRT calculations ranged from 1 to 15 seconds, and scaled with the number of users in the experiment. It would be impractical to calculate the metric after every observation, so practitioners need to decide a reasonable frequency to check results. Daily is a reasonable frequency.

Diving deeper into the accuracy for the mSPRT approach, the test results have been plotted by sample size and true lift in Figure 4.3. The errors seem to occur below a 4% lift. When the actual lift is small, a larger sample size does not seem to be improving the test accuracy. In very large sample size scenarios, 50,000 users and above, the errors seem to happen when the lift is below 2%. This is reassuring, since it points to the possibility of the higher Type-II error being reduced with a larger sample size.

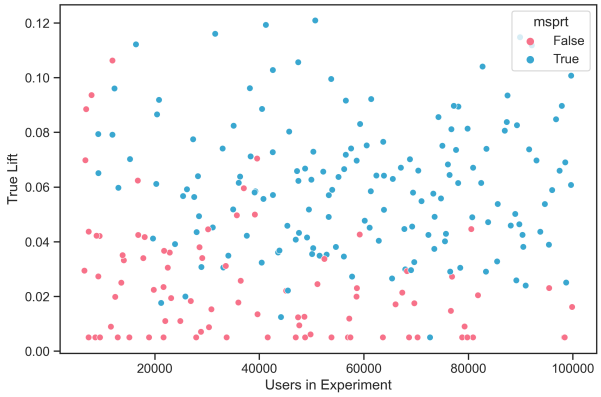

Figure 4.3: mSPRT accuracy plotted by experiment sample size and true lift

4.5 True Lift Versus Point Estimate

The goal of running an online experiment is knowing "*how much* better is treatment than control?". This topic is orthogonal to the purpose of this review, but is still worth covering. The challenge with precise measurement is that the data only affords us the sample mean of treatment effect and not the population mean. As such, it is critical to remember that a point estimate is unlikely to be the exact number of the impact of the new feature. Since the data for 242 experiments where there is a true lift is available, the difference between the true lift (the input the data was sampled on) and the observed lift was compared in Figure 4.4. It is seen that while the median is 0, for over 20% of experiments (top and bottom 10 percentile), the difference between sample and population lift is greater than 1%. This might not impact the business outcome, but it is something practitioners should be aware of and account for. The best-practice should be to always share the lift as an interval rather than a precise estimate. This is intellectually honest as well as practically beneficial, since the business can decide if they are happy with the outcome even at the lower end of the confidence interval.

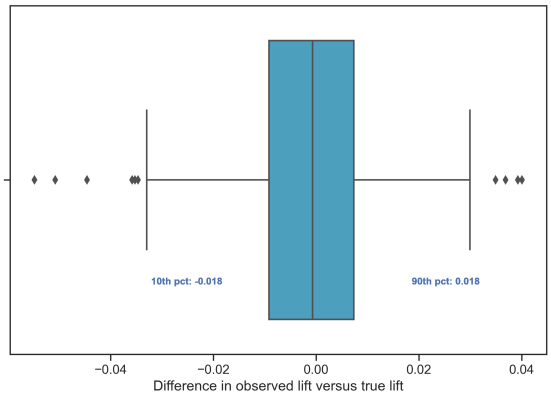

Figure 4.4: Boxplot of the difference between observed lift and true lift shows that for over 20% of experiments, the difference is larger than 1%

CHAPTER 5

Conclusion

T-tests deliver exactly what they promise: consistently using a 95% confidence level with 80% power will deliver 5% Type-I errors and 20% Type-II errors. The issue, however, is the improper use of these tests. It was shown how peeking dramatically increases Type-I errors yet most researchers are blind to this and continue to operate under the 5% assumption. It is not their fault. Online A/B testing made large scale experimentation with real-time results ubiquitous, yet none of the analysis tools have kept up with the rapid evolution of hypothesis testing. While a Bayesian approach was evaluated, it was not found to provide much benefit beyond the basic t-test. This might have looked different had pre-experiment data been incorporated to reduce variance, which is assesable in the future. Unless a researchers has the discipline to calculate the required sample size before starting an experiment and then wait until that sample is collected before evaluating the results, the t-test is not the appropriate tool for analyzing online experiments.

With businesses valuing speed to decision, the mSPRT delivers better performance for analyzing A/B tests, which is why leading tools have adopted it[JPW19]. When evaluating results frequently, it limits Type-I errors by being conservative. In Table 4.6, mSPRT has a Type-I error rate that is 1/6th that of the t-test in the Neutral scenario. As the sample size increases, the performance converges to a standard t-test. This balance of speed without compromising the accuracy of findings makes it an excellent analysis tool.

Adopting mSPRT would also allow its use in multiple comparisons. This is great because none of the techniques evaluated in Multiple Comparisons provided the ability to detect an

effect unless it was very large. Potentially walking away from winning ideas 50% of the time because of statistics is unacceptable. This overly conservative nature of multiple comparison adjustments make them largely unusable in a business setting. While the mSPRT was not evaluated for multiple comparisons, the execution should be identical to how it was conducted for the overall test. The downside here would be computation cost, which is discussed below.

There are likely hurdles preventing broad adoption of mSPRT. The first is implementation complexity. For a t-test, only a handful of summary statistics are necessary to use one of the countless online calculators to find the p-value and confidence intervals. Meanwhile, the mSPRT requires access to arrays with each observation, and then the ability to implement the formulas mentioned in mixture Sequential Probability Ratio Test (mSPRT). The lack of readily available tools, be it online or Python/R packages, makes broad adoption especially challenging. The work by Stenberg is a step in the right direction[Ste19]. For those that solve the implementation complexity, compute time remains an issue. mSPRT, due to its iterative calculation after each observation, takes an order of magnitude more time to compute than a t-test. This additional compute time slows down workflows and adds real costs in the form of server time. Additionally, the compute time scales as more users are added to the experiment. Regardless, the benefits far outweigh the costs, so mSPRT is still the recommended approach.

REFERENCES

[AM17] Vineet Abhishek and Shie Mannor. "A nonparametric sequential test for online randomized experiments." *Arxiv*, 2017.

[BH95] Yoav Benjamini and Yosef Hochberg. "Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing." *Journal of the Royal Statistical Society: Series B*, **57 (1)**, 1995.

[Bon36] C.E. Bonferroni. "Teoria statistica delle classi e calcolo delle probabilità." *Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commerciali di Firenze*, 1936.

[Boo47] George Boole. "The Mathematical Analysis of Logic." *CreateSpace Independent Publishing Platform, 2015*, 1847.

[Dah08] T. Dahiru. "P - value, a true test of statistical significance? A cautionary note." *Ann Ib Postgrad Med.*, **6(1):21-26**, 2008.

[Dso] Alan Dsouza. "Product Analytics Tools." `https://github.com/dsouz/product-analytics-tools`.

[JKP17] Ramesh Johari, Pete Koomen, Leonid Pekelis, and David Walsh. "Peeking at A/B Tests: Why It Matters, and What to Do about It." *Association for Computing Machinery*, 2017.

[JPW19] Ramesh Johari, Leo Pekelis, and David J. Walsh. "Always Valid Inference: Bringing Sequential Analysis to A/B Testing." *Arxiv*, 2019.

[Mil] Evan Miller. "Bayesian A/B testing." `https://www.evanmiller.org/bayesian-ab-testing.html`.

[Rob70] Herbert Robbins. "Statistical Methods Related to the Law of the Iterated Logarithm." *The Annals of Mathematical Statistics*, **41(5)**, 1970.

[Ste19] Erik Stenberg. "Sequential A/B Testing Using Pre-Experiment Data." *Uppsala Universitet*, 2019.

[Stu15] Chris Stucchio. "Bayesian A/B testing at VWO." *Visual Web Optimizer*, 2015.

[Wal45] Abraham Wald. "Sequential Tests of Statistical Hypotheses." *The Annals of Mathematical Statistics*, **16 (2) 117 - 186**, 1945.