# A Dirty Dozen: Twelve Common Metric Interpretation Pitfalls in Online Controlled Experiments

Pavel Dmitriev, Somit Gupta, Dong Woo Kim, Garnet Vaz
Analysis and Experimentation Team
Microsoft Corporation
Redmond, WA 98052, USA
{padmitri, sogupta, dok, gavaz}@microsoft.com

## ABSTRACT

Online controlled experiments (e.g., A/B tests) are now regularly used to guide product development and accelerate innovation in software. Product ideas are evaluated as scientific hypotheses, and tested in web sites, mobile applications, desktop applications, services, and operating systems. One of the key challenges for organizations that run controlled experiments is to come up with the right set of metrics [1] [2] [3]. Having good metrics, however, is not enough.

In our experience of running thousands of experiments with many teams across Microsoft, we observed again and again how incorrect interpretations of metric movements may lead to wrong conclusions about the experiment's outcome, which if deployed could hurt the business by millions of dollars. Inspired by Steven Goodman's twelve p-value misconceptions [4], in this paper, we share twelve common metric interpretation pitfalls which we observed repeatedly in our experiments. We illustrate each pitfall with a puzzling example from a real experiment, and describe processes, metric design principles, and guidelines that can be used to detect and avoid the pitfall.

With this paper, we aim to increase the experimenters' awareness of metric interpretation issues, leading to improved quality and trustworthiness of experiment results and better data-driven decisions.

## KEYWORDS

Controlled experiments; Online experiments; A/B testing; Metrics

## 1 INTRODUCTION

Online controlled experiments (e.g., A/B tests) are rapidly becoming the gold standard for evaluating improvements to web sites, mobile applications, desktop applications, services, and operating systems [5]. Big companies such as Amazon, Facebook, Google, LinkedIn, Microsoft invest in in-house experimentation systems, while multiple startups (e.g., Apptimize, LeanPlum, Optimizely, Taplytics) provide A/B testing solutions for smaller sites. At Microsoft, our experimentation system [6] supports experimentation in Bing, MSN, Cortana, Skype, Office, xBox, Edge, Visual Studio, etc. running thousands of experiments a year.

The attractiveness of controlled experiments comes from their ability to establish a causal relationship between the feature being tested and the measured changes in user behavior. Therefore, having the right metrics is critical to successfully executing and evaluating an experiment [1] [2] [3]. Indeed, metrics play a key role throughout the experimentation lifecycle – experiment design, running of the experiment, and overall evaluation of the experiment to make a ship/no-ship decision.

Having the right set of metrics, however, is not enough. There are situations when despite the experiment having been setup and run correctly and a good set of metrics used, the interpretation that the new feature caused the observed statistically significant change in a metric is incorrect. In other cases, the causal relationship holds but the standard interpretation of the metric movement, such as being indicative of positive or negative user experience, is wrong. We call such situations metric interpretation pitfalls. In our experience of working with many teams across Microsoft, we observed again and again how metric interpretation pitfalls lead to wrong or incomplete conclusions about the experiment's outcome.

Perhaps the most common type of metric interpretation pitfall is when the observed metric change is not due to the expected behavior of the new feature, but due to a bug introduced when implementing the feature. While having a rich set of Data Quality metrics may help catch some such issues, often they are domain specific and there is no clear general pattern for detecting and mitigating them. In this paper, we share twelve common metric interpretation pitfalls which we observed repeatedly in different teams at Microsoft, and for which we are able generalize. We illustrate each pitfall with a puzzling example from a real experiment, and describe processes, metric design principles, and guidelines that can be used to detect and avoid the pitfall. Some of these pitfalls were studied before in statistics or other domains, while others, to our

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.
*KDD '17, August 13-17, 2017, Halifax, NS, Canada*
© 2017 Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-4887-4/17/08…$15.00
http://dx.doi.org/10.1145/3097983.3098024

knowledge, were not. Whenever possible we provide references to those related works.

Our contribution in this paper is to increase trustworthiness of online experiments by disseminating the common metric interpretation pitfalls, explaining them, and sharing ways to avoid them in some cases, and detect and mitigate them in others. At Microsoft, it is not uncommon to see experiments that impact annual revenue by millions of dollars, sometimes tens of millions of dollars. An incorrect decision, whether deploying something that appears positive, but is really negative, or deciding not to pursue an idea that appears negative, but is really positive, is detrimental to the business. Online controlled experimentation is a rapidly evolving discipline and best practices are still emerging. Others who run online controlled experiments should be aware of metric interpretation issues, build the proper safeguards, and consider the root causes mentioned here to improve the quality and trustworthiness of their results, leading to better data-driven decisions.

The paper is organized as follows. Section 2 briefly introduces online controlled experiments. Section 3 discusses related work. Section 4 describes a metric taxonomy, a useful way to organize metrics according to their role in experiment analysis. Section 5 is the main part of the paper discussing twelve metric interpretation pitfalls. Section 6 concludes.

## 2 ONLINE CONTROLLED EXPERIMENTS

In the simplest controlled experiment or A/B test users are randomly assigned to one of the two variants: control (A) or treatment (B). Usually control is the existing system, and treatment is the existing system with a new feature X added. User interactions with the system are recorded, and metrics are computed. If the experiment was designed and executed correctly, the only thing consistently different between the two variants is the feature X. External factors such as seasonality, impact of other feature launches, competitor moves, etc. are distributed evenly between control and treatment and therefore do not impact the results of the experiment. Hence any difference in metrics between the two groups must be due to the feature X. This establishes a causal relationship between the change made to the product and changes in user behavior, which is the key reason for widespread use of controlled experiments for evaluating new features in software.

## 3 RELATED WORK

Controlled experiments are an active research area, fueled by the growing importance of online experimentation in the software industry. Research has been focused on topics such as scalability of experimentation systems [7], rules of thumb and lessons learned from running controlled experiments in practical settings [8] [9], new statistical methods to improve metric sensitivity [10], projections of results from a short-term experiment to the long term [11] [12]. These works provide good context for our paper and, in a similar style to this work, share lessons and learnings from running online controlled experiments in practice. These works, however, do not address the practical problem of metric interpretation.

The importance of metrics as a mechanism to encourage the right behavior has been recognized and studied in the management domain since at least 1956 [13]. It is articulated using statements such as "What gets measured, gets managed" [14], "What you measure is what you get" [15], and "You are what you measure" [16]. In recent years, several works were published focusing specifically on defining metrics for online controlled experiments [1] [2] [3]. In [1] and [2], several principles for designing good Overall Evaluation Criteria (OEC) metrics are outlined. In [3] a data-driven approach is described for evaluating different OEC metrics. These works are related to this paper in that some interpretability pitfalls we describe are due to poor metric design. We, however, discuss a broader set of pitfalls, most of which apply to metrics designed according to the best practices, and affect all metrics rather than just the OEC metrics.

Few works have focused directly on the topic of interpretability of metrics in an online controlled experiment. Three such works are [17] [18] [19]. These works discuss two issues with the null hypothesis testing framework: early stopping of experiments, and p-value misinterpretation, and propose solutions. As part of the discussion below, we present a practical view on these issues, share examples, and discuss the solutions proposed in [17] [18] [19] in the context of other approaches to avoiding these pitfalls. In addition, we also share many pitfalls which can occur even after the issues with null hypothesis testing framework have been addressed.

## 4 METRIC TAXONOMY

Most teams at Microsoft compute hundreds of metrics to analyze the results of an experiment. While only a handful of these metrics are used to make a ship decision, we need the rest of the metrics to make sure we are making the correct decision. In this section we describe a metric taxonomy that we have found useful for both guiding the metric design process and interpreting experiment results. The taxonomy assigns clear roles to different types of metrics and specifies their usage in evaluating the experiment outcome.

**Data Quality Metrics.** The first question we answer when looking at the outcome of an experiment is whether we can trust that the experiment was configured and run correctly such that we can trust the experiment results. It is not uncommon to have telemetry inconsistencies in the new feature, bugs in its implementation, or other issues. The purpose of Data Quality metrics is to alert of such issues.

One metric that has proven effective in spotting data quality issues in an experiment is the ratio of the number of users in the treatment to the number of user in the control. A fundamental requirement in A/B testing is that treatment and control samples are drawn at random from the same population. This means that the number of users in the samples should satisfy the expected ratio. If the actual ratio is different than the expected (a chi-square test can be used), it means something is wrong with the sampling process. We call this situation Sample Ratio Mismatch (SRM). Most of the time SRM implies a severe selection bias, enough to render the experiment results invalid [20] [21].

We give more examples of Data Quality metrics later in the paper. Having a comprehensive set of Data Quality metrics is key to detecting many pitfalls described below.

**Overall Evaluation Criteria (OEC) Metrics.** After checking the Data Quality metrics, next we want to know the outcome of the experiment. Was the treatment successful, and what was its impact? The set of metrics we look at to answer this question are called the OEC (Overall Evaluation Criteria) metrics [9] [22].

The OEC metrics are usually leading metrics that can be measured during the short duration of an experiment, but at the same time are indicative of long term business value and user satisfaction. It is ideal to have just a single metric as the OEC for a product. It can be a composite metric which is a combination of a few other metrics. Designing a good OEC is not easy, and is a topic of active research [1] [2] [3]. A discussion of an OEC for a search engine can be found in [9].

**Guardrail Metrics.** In addition to OEC metrics, we have found that there is a set of metrics which are not clearly indicative of success of the feature being tested, but which we do not want to significantly harm when making a ship decision. We call these metrics Guardrail metrics. For instance, on a web site like Bing or MSN, Page Load Time (PLT) is usually a guardrail metric. Small degradations in guardrail metrics can be expected – any additional feature on a site usually causes a slight increase in PLT due to, for example, a slightly larger HTML size. Large degradations, however, are generally not allowed.

**Local Feature and Diagnostic Metrics.** Local feature metrics measure the usage and functionality of individual features of a product. Examples of feature metrics are metrics measuring click-through Rate (CTR) on individual elements on a web page, or metrics measuring the fraction of users transitioned from one stage of a purchasing funnel to another. These metrics often serve as Diagnostic metrics for the OEC, helping understand where the OEC movement (or lack of it) is coming from.

While improvement in a local metric is usually a good thing, it is often accompanied by a degradation in another related local metric. For example, as discussed in [8], it is very hard to increase Overall CTR on a search engine page, but it is very easy to increase CTR on some element of the page at the expense of other elements. There can also be cases where large unexpected improvement in a local metric is due to undesirable side effects of the treatment. Therefore, movements in local metrics need to be interpreted carefully.

## 5 METRIC INTERPRETATION PITFALLS

In this section, we share twelve common metric interpretation pitfalls we observed when running experiments at Microsoft. All the numbers reported below are statistically significant at p-value 0.05 level or lower, unless noted otherwise.

### 5.1 Metric Sample Ratio Mismatch

We ran an experiment on the MSN.com homepage where, when users clicked on a link, in treatment the destination page opened in a new browser tab while in control it opened in the same tab. The results showed 8.32% increase in Page Load Time (PLT) of the msn.com homepage. How could this one line JavaScript change cause such a large performance degradation? Was there a bug introduced during this change?

The key to uncovering the mystery was to carefully examine the definition of the metric:

$$PLT(variant) = \frac{\sum_{homepage\ loads\ p} PLT(p)}{\sum_{homepage\ loads} 1}$$

There was a statistically significant difference in the number of home page loads, with the treatment having 7.8% fewer home page loads then the control. Clearly, the set of page loads over which the metric was computed were different between the treatment and the control. In a situation like this the metric value cannot be trusted.

What happened is that in the control after users clicked a link on the homepage and then wanted to come back, they used the browser back button causing a homepage reload. In the treatment, there was no back button option after opening a link in the new tab, but the homepage remained open in the old tab so users could come back to it without a reload. The back button page reloads in the control were generally faster than the first page load in a session due to browser caching. With the faster back button page loads omitted in the treatment, treatment's average PLT was substantially worse.

We call an effect like this the metric sample ratio mismatch. It is similar to the Sample Ratio Mismatch (SRM) problem mentioned in Section 4, and similarly to how an SRM invalidates the results of the whole experiment, a metric sample ratio mismatch usually invalidates the metric: the treatment-control delta may change in an arbitrary direction, and the statement "the new feature X caused Y amount of metric change" is no longer valid.

There is a variety of causes for metric sample ratio mismatch. Sometimes, like in the example above, it is directly caused by a change in user behavior. Other times it could be due to indirect reasons such as different loss rates of telemetry between control and treatment, incorrect instrumentation of new features, etc. Some of the pitfalls described later in the paper may cause metric sample ratio mismatches.

Not being aware that a metric is affected by a sample ratio mismatch could lead to a wrong ship decision, send the experimenter down a wrong investigation path, or result in a wrong estimate of the experiment impact. For this reason, it is critical that the experimentation system automatically detects sample ratio mismatches and warns the user about them.

What to do when a metric sample ratio mismatch occurs? The general strategy is to decompose the metric with the goal of (1) understanding what parts of the metric differ, and (2) isolate the parts that are not affected by the mismatch and can still be trusted. The specifics of the decomposition vary case by case. A good start is to look separately at the numerator and the denominator of the metric. For the PLT example above, the breakdown could proceed as follows.

Homepage PLT →

1. Average homepage PLT per user
    a. Average homepage PLT per user with 1 homepage visit
    b. Average homepage PLT per user with 2+ homepage visits
2. Number of homepage loads per user
    a. Number of back-button loads
    b. Number of non-back-button loads
        i. PLT for non-back-button loads

First, two separate metrics at user level could be created for numerator and denominator: 1 and 2 (as discussed in section 5.2, it is a good idea to always have such a decomposition on the scorecard). Decomposing 1 into 1.a and 1.b would show that only 1.b part has a statistically significant difference between the treatment and the control. This may give a hint to decompose 2 into 2.a and 2.b. Seeing that only 2.a part changes, one can then compute 2.b.i – a metric that does not suffer from a sample ratio mismatch and gives useful information about whether there is any difference in PLT on the subset of page loads that could be compared fairly.

## 5.2 Misinterpretation of Ratio Metrics

Click Through Rate (CTR) of an online ad or content site is usually considered a good indicator that users find the ad or content relevant and helpful [23]. This metric is widely used in online advertising [24] and often used in content websites like MSN.com as well.

The main page of MSN.com consists of a series of modules, each of which contains links to pages under the same topic area such as entertainment, sports, etc. The positions of such modules have been frequently tested to optimize users' experience. In one experiment a module located close to the bottom of the main page was moved to a higher position which requires users to scroll less to reach it. The result showed a 40% decrease in CTR of the module!

The metric definition is given below, computing an average over users of the CTR of each user.

$$Avg\ CTR/User = \frac{\sum_{users}\left(\frac{\#\ clicks\ on\ the\ module}{\#\ impressions\ of\ the\ module}\right)}{\sum_{users} 1}$$

In our example, there was no metric sample ratio mismatch as the ratio of users in treatment and control was balanced. The movement in the average CTR/user was valid and caused by the treatment. The question was whether 40% drop in CTR/user of a module meant that the treatment was bad?

When the numerator, the number of clicks on the module, and denominator, the number of times that the module was displayed to the users, of the metric were checked separately, it turned out that both were improved significantly but the denominator moved relatively more than the numerator; 200% and 74%, respectively. CTR is usually interpreted as measure of quality of content or recommendation. This interpretation does not hold when there is a change in the number of times the module was displayed to the users or when there is a change in the population of users who saw the module. The whole page CTR remained flat, which is known to be a typical result of an experiment [8], because both total numbers of clicks and impressions for the whole page were not changed significantly. At the same time, revenue increased due to the promoted module being more monetizable than other modules that lost clicks. Thus, the result of the experiment turned out to be good.

Ratio metrics are very common in experiment analysis, both due to intuitive interpretation and due to increase in sensitivity they usually provide compared to count metrics, because of their bounded variance. For example, the Avg CTR / User metric is usually more sensitive than Click Count / User, because the value of the CTR metric is between 0 and 1, while the value of Click Count / User is unbounded. In fact, for many count metrics the confidence interval does not shrink as the experiment runs longer [22]. However, as shown in the example above, ratio metrics can be misleading or potentially invalid if the denominator of the metric changed.

There are two ways to compute ratio metrics: (A) the average of ratios, and (B) the ratio of averages. For instance, the CTR metric can be computed as described above (method A) or as the ratio of the total number of clicks to that of page views from all users (method B). We find that method A has several practical advantages over method B. It tends to have higher sensitivity. Since it equally weighs each user regardless of one's activity level, it is more resilient to outliers. Since the denominator of method A is the number of users which is one of the controlled quantities in an experiment, it is less likely to suffer from having a metric-level SRM (Section 5.1). In addition, method A allows us to compute the variance of a metric in a simpler way compared to method B, because method A computes an average over users, which is typically the unit of randomization in an experiment. More detailed discussion on this can be found in [1].

To detect denominator mismatch in ratio metrics, we recommend to always define count metrics for the numerator and the denominator, and provide those in the result alongside the ratio metric.

## 5.3 Telemetry Loss Bias

Skype recently ran an experiment that evaluated changing to a different protocol for delivering push notifications to Skype's iPhone app, with the goal of increasing the reliability of notification delivery. While the experimenters could foresee some impact on message-related metrics, they did not expect it to impact call-related metrics in any way. Surprisingly, the results showed strong statistically significant changes in some call-related metrics, such as the fraction of attempted calls that were successfully connected. Even more puzzling was the fact that the pattern of movement did not follow a typical "improvement" or "degradation" pattern. Some metrics moved strongly while other, related metrics stayed flat. What went wrong?

First some background. An unavoidable aspect of analyzing mobile app experiments is that a substantial fraction of telemetry coming from mobile clients gets lost. To optimize bandwidth usage, most telemetry events are buffered on the client and then sent in batches only when the client is on wifi. The buffer is fixed size, and if the client is not on wifi for a long time, the buffer fills up and old events are dropped.

In this experiment, when the app was woken up by a push notification via the new protocol used by the treatment, it stayed up a few seconds longer, allowing more time for checking whether it is on wifi and, if so, preparing the telemetry batch and sending it over. Treatment showed reductions in loss rate of all the client events that were used to compute the metrics. For some metrics, this caused a metric SRM (see Section 5.1). For others, the impact was subtler.

Different rates of telemetry loss between the variants is a common way to bias experiment results. It is not limited to mobile experiments. In web site experiments, the tradeoff between the site performance and the reliability of click tracking [25] leads to lossy click tracking mechanisms being used, that could be affected by experiments. For example, opening links in a new tab, as in the MSN.com experiment described in Section 5.1, improves click tracking reliability due to the original page staying open and having more time to execute the javascript code that sends click tracking beacon [22].

Since the new events that make it (or events that are lost) are typically a highly biased set, telemetry loss bias practically invalidates any metrics that are based on the affected events. In the MSN.com example, this means that all click-based metrics were invalid. In the Skype example, all client event based metrics were invalid. Metrics based on the server-side events, however, remained valid. The puzzling pattern of movement in call-related metrics was the result of there being a mix of server and client event based metrics.

The results of overlooking telemetry bias could be severe, potentially leading to an incorrect ship decision. Therefore, for every client event used in metrics, a Data Quality metric measuring the rate of event loss should be created. The two common ways to measure event loss are

1. Compare to a corresponding server event. For example, in Skype there are both client and server versions of the "call" event (each containing different fields). The server event has almost no loss, so the number of client events received could be compared to that received from the server to compute the client event loss rate.
2. Use event sequence numbers. The client can assign a sequence number to every event. Gaps in the sequence can be used to compute a loss rate estimate.

Approach 1 is superior and should be used wherever possible, while approach 2 can be used where there is no server-side baseline event to compare to. Event loss rate should be tracked and a focused effort needs to be made to improve it via, e.g., reducing the size of the event and optimizing the timing and the priority of sending a different event. As much as possible metrics should be based on more reliable server-side events.

Telemetry loss bias is one of many ways how issues with instrumentation, data collection, and data processing – the steps leading up to the metric computation – can bias metric values. Having standard instrumentation guidelines, documenting the data flow for each metric, and implementing a rich set of Data Quality metrics are all helpful for preventing and detecting such issues.

## 5.4 Assuming Underpowered Metrics had no Change

The total number of page views per user is an important metric in experiments on MSN.com. In one MSN.com experiment, this metric had a 0.5% increase between the treatment and control, but the associated p-value was not statistically significant. For a mature online business like MSN.com, 0.5% change of the total page views is often interpreted as a meaningful impact on the business. But since it was not statically significant can we assume that there was no impact on page views?

More careful look at the result revealed that the confidence interval of the metric lied over about ±5% and the experiment was not configured to have an enough power for the metric; it turned out that only 7.8% or larger change could be detected with 80% power, given the configuration of the experiment. Therefore, we cannot assume that we did not impact the underpowered metric.

Power is the probability of rejecting the null hypothesis given that the alternative hypothesis is true. Having a higher power implies that an experiment is more capable of detecting a statistically significant change when there is indeed a change. If an underpowered metric turns out to be statistically significant, then it is more likely that the observed change of the metric will be exaggerated compared to the true effect size [26], which is often referred to as an example of Winner's curse [27].

To avoid the aforementioned issues, a priori power analysis should be conducted to estimate sufficient sample sizes, at least for the OEC and the Guardrail metrics, and to allocate at least that number of samples to the experiment, so that changes which are small but meaningful to business can be detected as being statistically significant. The recommendation from such a priori power analysis can be simplified for typical scenarios of experiments for a given product. For example, for Bing experiments run in the US, it is recommended to run with at least 10% of the users for one week, if the feature being tested impacts most users. Sometimes, the recommended number of samples given an effect size to detect can be larger than the traffic availability of a product. In such a case, it is important to know the least effect sizes of the metrics of interest given the full usage of the traffic so that experimenters are aware of the limits in the detectability of changes imposed by configurations of experiments.

We recommend to have at least 80% power to detect small enough changes in success and guardrail metrics to properly evaluate the outcome of an experiment.

## 5.5 Claiming Success with a Borderline P-value

We ran an experiment in Bing.com where we observed a statistically significant positive increase for one of the key Bing.com OEC metrics with a p-value of 0.029. The metric tries to capture user satisfaction and is a leading indicator of user retention. Very few experiments succeed in improving this metric. Given that we have verified the trustworthiness of the experiment is there anything left but to celebrate such a good result?

In Bing.com whenever key metrics move in a positive direction we always run a certification flight which tries to replicate the results of the experiment by performing an independent run of the same experiment. In the above case, we reran the experiment with double the amount of traffic and observed that there were no statistically significant changes for the same metric.

Since the formal introduction of a p-value by Pearson [28] and its statistical significance boundary of 5% by Fisher [29], p-values have been very widely used as an indicator of the strength of evidences of scientific findings. However, as pointed out in [4] [30], p-values have been "commonly misused and misinterpreted". One of the notable cases is simply rejecting a null hypothesis when a p-value is close to the decision boundary without considering other supporting evidence.

The behavior of metrics changing from not being statistically significant to statistically significant or vice versa is often observed in repeated A/A experiments which are configured to have no differences between the treatment and control groups. By the definition of p-value, there is a chance, typically 5%, that a metric comes with a p-value being less than 5% in an A/A experiment. Thus, when one tracks many metrics simultaneously, some of those metrics will have p-values less than the decision boundary. However, again by the definition of p-value, such statistical significant p-values in A/A experiments are typically close to the boundary, e.g. between 1% to 5% [31].

As shown in the example above, when a metric comes with a borderline p-value, it can be a sign of a false positive, and there are many such cases in the world of online A/B testing due to a large number of metrics computed for an experiment. We recommended for experimenters to evaluate experiment results by placing emphasis on strongly statistically significant metrics and rerunning with larger traffic when metrics, in particular the OEC metrics, have borderline p-values. In cases where repeated reruns provide borderline p-values either due to small treatment effects or when we cannot increase the traffic we can use Fishers Method to obtain more reliable conclusion.

## 5.6 Continuous Monitoring and Early Stopping

Example 1: A two-week long experiment was run in Bing, evaluating a new ranking algorithm. After the first week, the key success metric showed statistically significant improvement. Can the experiment owner stop the experiment after the first week, since the success criteria were already met?

Example 2: A two-week long experiment was run at Xbox to evaluate teaching tips for users who get suspended from playing multiplayer games due to bad behavior. The goal of the experiment was to decrease customer support service calls about the suspensions. At the end of two weeks, we observed no such change in the number of calls made between the treatment and the control groups. Can the experiment owner keep running the experiment beyond the two weeks, to see if with increased power key metrics improve in a statistically significant way?

The answer to both questions, in the standard scenario where null hypothesis testing (NHST) is used, is "no". Continuously checking the results and stopping as soon as statistical significance is achieved leads to an increase in Type-I error, or the probability of a false positive. The reason for this is that allowing extra opportunities for evaluation, in addition to the evaluation at the end of the pre-defined experiment period, can only increase the chance of rejecting the null hypothesis. This may lead to shipping a feature that did not improve the metrics the experiment owner thought it improved, or assuming the feature had negative impact when it actually did not.

In our experience, stopping early or extending the experiment are both very common mistakes experiment owners make. These mistakes are subtle enough to even make it into recommended practices in some A/B testing books [32]. The issue is exacerbated by the fact that in practice continuous experiment monitoring is essentially a requirement to be able to shut down the experiment quickly in case a strong degradation in user experience is observed.

Making experiment owners aware of the pitfall, and establishing the guidelines that require making ship decision only at the pre-defined time point, is one approach to avoid this pitfall. Another approach is to adjust the p-values, to account for extra checking [33]. Finally, in [34] a Bayesian framework is proposed that, unlike NHST, naturally allows for continuous monitoring and early stopping.

## 5.7 Assuming the Metric Movement is Homogeneous

Multiple prior works [11] [12] discussed the revenue-relevance tradeoff in web search. More and/or lower quality ads shown to users typically lead to a degradation in user experience, and vice versa. We ran an experiment in Bing that evaluated a new ad auction and placement algorithm, attempting to increase revenue by improving ad quality and keeping the number of ads shown roughly the same. The experiment seemed to be very successful, increasing revenue by 2.3%, while at the same time decreasing the number of ads shown per page by 0.6%.

However, an interesting insight was obtained by segmenting the pages by whether it was the first page returned after the user typed the query (original), or one of the subsequent pages returned after the user clicked on a result and pressed the back button (dup). Dup pages are reloaded and are generally different from the original, potentially showing a different number of ads and web results. We found that while on dup pages the number of ads per page decreased dramatically, by 2.3%, the number of ads per page on the original pages actually increased by 0.3%. While the number of original pages is higher, the large delta on dup pages led to the overall decrease. Given the importance of original pages, it was determined that the experiment actually did not meet its goal of keeping the ad load the same.

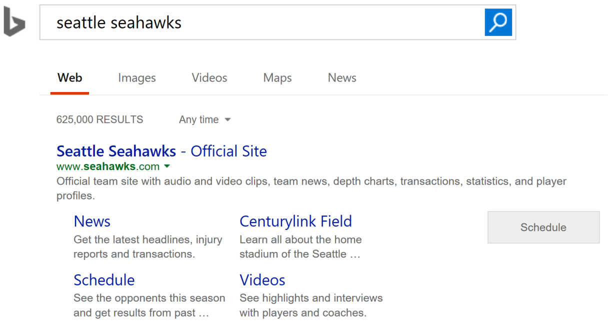
The lesson here is to avoid the pitfall of assuming that treatment effect is homogeneous across all users and queries. We observe heterogeneous treatment effects quite often in our experiments, such as users in different countries reacting to the change differently, or the feature being tested not working correctly only in a certain version of a certain browser. Not being aware of such effects may result in shipping a feature that, while improving user experience overall, substantially hurts experience for a group of users. Conversely, a feature that is negative overall may actually be positive for a certain group.

Due to a large number of metrics and segments that need to be examined in order to check whether there is a heterogeneous treatment effect, it's pretty much impossible to reliably detect such effects via a manual analysis. Developing automated tools for detecting heterogeneous treatment effects is an active area of research [35]. Our system automatically analyses every experiment scorecard and warns the user if heterogeneous treatment effects are found. While there is a lot of value of monitoring running experiments, the experimenters should understand the caveats around continuous monitoring and build robust systems to help make these decisions easier.

## 5.8 Segment Interpretation

It is a common practice to segment the users in the experiment by various criteria. For example, in mobile app experiments commonly used segments are the user's country, device type, operating system, app version, date the user joined the experiment, etc. Segmentation is useful for debugging, for example when trying to understand if a data quality issue applies to all users or is limited to a specific user segment, for understanding if some segments show stronger metric movement than others, and for detecting other types of heterogeneous treatment effects (see Section 5.7). However, one needs to interpret metric movements on segments with care.

In a Bing experiment testing a new ranking algorithm, one of the segments used was whether the user saw a "deeplink" – one of the extra links that show up for some navigational queries. Figure 1 shows an example. Both groups of users, those who saw a deeplink ($U_1$) and those who did not ($U_2$), showed a statistically significant increase in Sessions per User, the key Bing metric. However, the combination ($U_1$ + $U_2$) did not show a statistically significant change in the metric. How can this be possible?



625,000 RESULTS Any time

Seattle Seahawks - Official Site
www.seahawks.com
Official team site with audio and video clips, team news, depth charts, transactions, statistics, and player profiles.

News
Get the latest headlines, injury reports and transactions.

Centurylink Field
Learn all about the home stadium of the Seattle …

Schedule
See the opponents this season and get results from past …

Videos
See highlights and interviews with players and coaches.

**Figure 1: "News", "Schedule", etc. are deeplinks for the "seattle seahawks" official site search result**

What happened here is that the experiment did not impact the Sessions per User metric. However, the fraction of users in U1 (those who saw a deeplink) decreased in the treatment. The users who "dropped out" from U1 were less active than average in that group, resulting in higher treatment Sessions per User for U1. These users joined the U2 group (those who did not see a deeplink), where they were more active than average, resulting in higher treatment Sessions per User for U2 as well. This situation is known as Simpson's paradox [36].

The key lesson from this example is to ensure that the condition used for defining the segment is not impacted by the treatment. This can be tested by conducting an SRM test (see Section 4) for each segment group. Indeed, for the experiment described above, the sample ratio of users in both deeplink and no-deeplink segments was statistically significantly different. If statistically significant difference is observed, the results for that segment group, and often for all other groups in the segment, are invalid and should be ignored.

A common way to run into a Simpson's paradox is to keep recursively segmenting the users until a statistically significant difference is found. Experiment owners are particularly prone to this pitfall when the metric they were hoping to improve in the experiment did not show a statistically significant improvement. Trying to see if the metric improved at least for some subgroup of users, they keep recursively segmenting the user population until they see the desired effect.

Aside from increasing the possibility of encountering a Simpson's paradox, this is also an instance of multiple testing problem [37] and will result in an increased Type-I error. For example, for a commonly used Type-I error threshold of 5%, assuming segment groups are independent, 1 out of 20 segment groups would show a statistically significant change in the metric simply by chance. To control the Type-I error, correction procedures such as Bonferroni correction [38] should be used when analyzing segments.

## 5.9 Impact of Outliers

Infopane is the module showing a slide show of large images at the top of MSN.com homepage. There was an experiment to increase the number of slides in the infopane from 12 to 16. Contrary to the expectation that increasing the number of slides would improve user engagement, the experiment showed significant regression in engagement. The experiment also had an SRM, with fewer users than expected in the treatment.

Investigation showed that the cause of the SRM and the puzzling drop in engagement was bot filtering logic. Number of distinct actions the user takes was one of the features used by the bot filtering algorithm, resulting in users with outlier values labeled as bots and excluded from experiment analysis. Increasing the number of slides improved engagement for some users so much that these real users started getting labeled as bots by the algorithm. After tuning the bot filtering algorithm, SRM disappeared and experiment results showed a big increase in user engagement.

Outliers can skew metric values and increase the variance of the metric making it more difficult to obtain statistically significant results. Because of this, it is common to apply some kind of filtering logic to reduce the outlier impact. As the example above shows, however, one needs to ensure that all variants in the experiment are impacted by the outlier filtering logic in the same way.

Common outlier handling techniques include trimming (excluding the outlier values), capping (replacing the outlier values with a fixed threshold value), and windsorizing (replacing the outlier values with the value at specified percentile). For all these approaches, we always recommend having a metric that counts the number of values affected by the outlier handling logic in the Data Quality section, to be able to detect situations like the one in our example. Changing the metric to compute a percentile (25th, median, 75th, 95th are common) instead of the average is another way to handle outliers, though percentile metrics are often less sensitive than averages, and are more expensive to compute. Applying a transformation, e.g. taking a log, also helps to reduce outlier impact, but makes the metric values difficult to interpret.

## 5.10 Novelty and Primacy Effects

The edge browser new tab page has a section at the top that shows frequently visited sites for a user (referred to as top sites) to allow the user to quickly navigate to that site. A user can also add a site to the top sites list. While top sites feature was used by many users, many did not have a full list of top sites, and never added a top site. An experiment was run for four weeks to test the idea of showing a coach mark [39] to such users asking them to add a top site. This coach mark was shown just once to only those users who had an empty top site and had never added a top site before.

The experiment result showed 0.96% increase in whole page clicks and 2.07% increase in clicks on the top sites. This result seemed positive. The question was how durable is the effect? Will it lead to long term increase in user engagement?

When we looked at the experiment results for each day segment, we found that the percent delta in clicks on top sites between treatment and control declined quickly, suggesting a novelty effect. To test the hypothesis, we looked at the effect during the visit the user was shown the coach mark, and the effect during the subsequent visits, separately. While the former segment showed a large increase in user engagement and clicks on the top sites, the latter segment did not show any statistically significant movement in metrics. This indicated that the treatment effect did not last beyond the first visit. This result saved the feature team from spending too many resources on creating more coach marks and also saved the users from getting multiple coach marks.

While experiments are typically run for a short duration, we are trying to estimate the long-term impact of the treatment on the business and users. To properly assess the long term impact, we need to consider if the treatment effect would increase or decrease over time. In the example above, the change appeared positive in the short period but it was flat in the long term. This is a novelty effect. Ignoring this effect could lead the product team to invest in wrong areas which do not produce durable positive treatment effects. It may also lead to an increase in features that capture user attention and engage with the product while distracting the users from the actual task they want to finish, increasing user unhappiness in the long term. Pop-up ads and rapidly changing UI elements on the page fall into this category.

Conversely, some treatment effects might appear small in the beginning but they increase over time as user learning occurs over time or a machine learning system better adapts to the treatment. This is called a primacy effect. Typical examples here include decrease in users ignoring ads (ads blindness) when fewer and more relevant ads are shown to users [12], or increase in user engagement when a new content recommendation system gathers more information from users over time and adapts.

It is not always easy to detect novelty and primacy effects in all cases. When it is known that the feature has primacy effect, for example when testing a new personalization algorithm that needs to gather some data about the user, we can mitigate these effects by doing a warm start so the marginal improvement in performance over time is smaller. We also recommend segmenting treatment effect by different days of the experiment, or different user visits, to see if the treatment effect changes over time. One can also run long-term experiments to avoid the impact of novelty and primacy effects, though there are many pitfalls with analyzing long running experiments correctly [12] [11].

## 5.11 Incomplete Funnel Metrics

We ran an experiment in Xbox to test various promotion strategies for products that were on sale. Several tests were conducted wherein the images and messaging for the sale were changed to increase revenue. The results showed that such small changes have large positive impacts on the number of users who click through to the sale. What should we do next?

While it is easy to increase the number of clicks we obtain on specific locations it does not directly relate to our aim of increasing revenue. In some of these experiments we did not see any corresponding revenue increase even when it had sufficient power. This was the primary goal of this experiment. On the other hand, it cost our users wasted time and effort.

Several online user experiences can be modelled as a funnel process. The two main experiences for these include sign-ups to online services and online shopping on e-commerce sites. In these processes, users need to perform several actions in a row until an eventual success goal is met. In real scenarios, the final success rates of less than 1% are not uncommon making it crucial to understand and optimize these processes.

For funnel based scenarios it is crucial to measure all parts of the process. At every step of the funnel we need to ensure that the success rates are compared and not just the raw clicks or user counts. In addition, we should measure both conditional and unconditional success rate metrics. Conditional success rates are defined as proportion of users that complete the given step among users that attempted to start the step. Unconditional success rates compute the success rate taking into consideration all users who started at the top of the funnel. The combination of these two types of metrics along with detailed breakdowns at each step is the best guiding light towards improving our main goal while maintaining at least the current success rate standards.

While large ecommerce or social websites have millions of users attempting to perform these actions most smaller sites do not have such a luxury. Even with thousands of users starting the process we could lead to only a few hundred users successfully completing the final step. Hence it becomes crucial to ensure that our metrics at every step are sufficiently powered to detect any changes we deem significant to our business.

## 5.12 Failure to Apply Twyman's Law

We recently ran an experiment on the MSN.com homepage where the Outlook.com button on the top of the page was replaced with a mail app button which, when clicked, would open the desktop mail app in treatment instead of navigating the user to outlook.com as it did in control. For this experiment we saw a 4.7% increase in overall navigation clicks on the page, and a 28% increase in the number of clicks on the button. There was also a 27% increase in the number of clicks on the button adjacent to the mail button. It seemed like we had hit a jackpot. The treatment was doing much better than we expected. What else was left than to ship this treatment to all users?

Such a metric movement was too good to be true. While there was a massive increase in number of clicks, we did not see any stat-sig change in metrics related to user retention and satisfaction. Also looking at each day segment we found that the number of clicks on the mail app button were decreasing rapidly day over day (Figure 2). We believe that the treatment caused a lot of confusion to the users who were used to navigating to outlook.com from the msn homepage. When the button instead started opening the mail app, some users continued to click on the button expecting it to work like it used to. They may have also clicked on the button adjacent to mail button to check if other buttons are working. This is a likely reason to see such an outcome. Had this treatment been shipped to all users, it would have caused a lot of user dissatisfaction. In fact we shut down the experiment mid-way to avoid user dissatisfaction.

In another experiment conducted in Windows Store where a new configuration of top-selling apps page was tried, an unexpected 10% increase in the number of views of one type of page was observed. The investigation revealed that the affected type of pages in the treatment was not properly configured and no content was displayed to the users, which led users to try opening those pages multiple times and resulted in 10% increase.

Twyman's law says that any figure that looks interesting or different is usually wrong. Applying it to analyzing the results of online experiments, we can rephrase it as follows: any unexpected metric movement, positive or negative, usually means there is an issue. It is a common bias in all of us to view surprising negative results with a lot of skepticism as compared to surprising results that appear positive. Overlooking such issues, however, can lead us to ship a harmful feature or bug to our customers.

Unexpected metric movements (positive or negative) should be investigated to prevent harm to customers and the business. Having a comprehensive set of metrics, segments and OEC helps in investigating such issues faster. At Microsoft we have configured automated alerts and auto-shutdown of the experiments if we detect unexpected large metrics movements in both the positive and the negative directions.

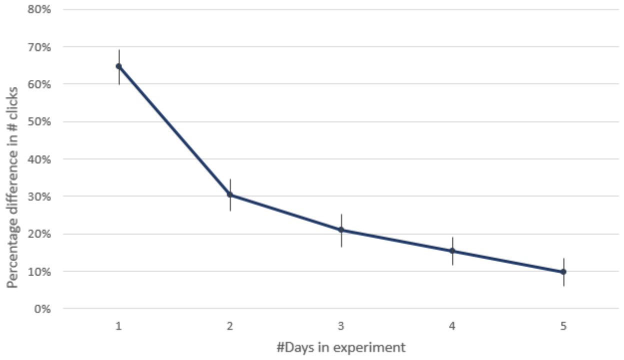


**Figure 2: The percentage difference in number of clicks, each day, between treatment and control on the mail/outlook button in the MSN.com experiment.**

## 5 CONCLUSION

In this paper we shared twelve common metric interpretation pitfalls we observed while running online controlled experiments across different teams at Microsoft. We hope the knowledge of these pitfalls and the mitigation approaches we shared will help others engaged in running controlled experiments avoid these pitfalls and increase trustworthiness of their decisions.

The twelve pitfalls discussed in this paper are by no means the complete list. The mitigation techniques we discussed, however, such as having a comprehensive set of Data Quality metrics, as well as alerts and power calculation tools, apply to a broader range of issues and have proved to be helpful in detecting other, less common pitfalls.

We also hope that this paper will stimulate more research and sharing of best practices among the academia and industry in the rapidly evolving area of online experimentation.

## ACKNOWLEDGMENTS

We wish to thank Ronny Kohavi, Greg Linden, and Widad Machmouchi for great feedback on the paper. Multiple co-workers on the Analysis and Experimentation team at Microsoft helped crystalize these ideas.

## REFERENCES

[1] A. Deng and S. Xiaolin, "Data-driven metric development for online controlled experiments: Seven lessons learned," in *KDD*, 2016.

[2] W. Machmouchi and G. Buscher, "Principles for the Design of Online A/B Metrics," in *Proceedings of the 39th International ACM SIGIR*, 2016.

[3] P. Dmitriev and W. Xian, "Measuring Metrics," 2016, Proceedings of the 25th ACM International on Conference on Information and Knowledge Management.

[4] S. Goodman, "A Dirty Dozen: Twelve P-Value Misconceptions," in *Seminars in Hematology*, 2008.

[5] R. Kohavi and R. Longbotham, "Online Controlled Experiments and A/B Tests.," in *Encyclopedia of Machine Learning and Data Mining*, 2017.

[6] "Microsoft Experimentation Platform," [Online]. Available: http://www.exp-platform.com.

[7] R. Kohavi, A. Deng, B. Frasca, T. Walker, Y. Xu and N. Pohlmann, "Online Controlled Experiments at Large Scale," in *KDD*, 2013.

[8] R. Kohavi, A. Deng, R. Longbotham and Y. Xu, "Seven Rules of Thumb for Web Site Experimenters," in *KDD*, 2014.

[9] R. Kohavi, R. Longbotham, D. Sommerfield and R. M. Henne, "Controlled experiments on the web: survey and practical guide," *Data Mining and Knowledge Discovery*, vol. 18, no. 1, pp. 140-181, February 2009.

[10] A. Deng, Y. Xu, R. Kohavi and T. Walker, "Improving the Sensitivity of Online Controlled Experiments by Utilizing Pre-Experiment Data," in *Sixth ACM WSDM*, Rome, Italy, 2013.

[11] P. Dmitriev, B. Frasca, S. Gupta, R. Kohavi and G. Vaz, "Pitfalls of Long-Term Online Controlled Experiments," in *IEEE International Conference on Big Data* , 2016.

[12] H. Hohnhold, D. O'Brien and D. Tang, "Focusing on the Long-term: It's Good for Users and Business," in *KDD*, 2015.

[13] V. F. Ridgway, "Dysfunctional Consequences of Performance Measurements," *Administrative Science Quarterly,* 1956.

[14] R. W. Schmenner and T. E. Vollmann, "Performance Measures: Gaps, False Alarms and the "Usual Suspects"," *International Journal of Operations & Production Management,* 1994.

[15] R. S. Kaplan and D. Norton, "The Balanced Scorecard - Measures that Drive Performance," *Harvard Business Review,* 1992.

[16] J. R. Hauser and G. M. Katz, "Metrics: you are what you measure!," *European Management Journal,* 1998.

[17] A. Deng, J. Lu and S. Chen, "Continuous monitoring of A/B tests without pain: Optional stopping in Bayesian testing," in *DSAA*, 2016.

[18] A. Deng, "Objective Bayesian Two Sample Hypothesis Testing for Online Controlled Experiments," in *Proceedings of the 24th International Conference on World Wide Web (WWW '15 Companion)*, 2015.

[19] R. Johari, L. Pekelis and D. J. Walsh, "Always valid inference: Bringing sequential analysis to A/B testing.," *In submission. Preprint available at arxiv.org/pdf/1512.04922,* 2015.

[20] R. Kohavi, "Lessons from running thousands of A/B tests," 2014. [Online]. Available: http://bit.ly/expLesssonsCode.

[21] Z. Zhao, M. Chen, D. Matheson and M. Stone, "Online Experimentation Diagnosis and Troubleshooting Beyond AA Validation," in *DSAA, 2016.*

[22] R. Kohavi, A. Deng, B. Frasca, R. Longbotham, T. Walker and Y. Xu, "Trustworthy online controlled experiments: Five puzzling outcomes explained," *in KDD*, 2012.

[23] Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Click-through_rate.

[24] "Clickthrough rate (CTR): Definition," AdWords Google, 2016. [Online]. Available: https://support.google.com/adwords/answer/2615875.

[25] R. Kohavi, Messner, S. Eliot, J. L. Ferres, R. Henne, V. Kannappan and J. Wang, "Tracking Users' Clicks and Submits: Tradeoffs between User Experience and Data Loss," October 2010. [Online]. Available: http://bit.ly/expTrackingClicks.

[26] J. P. Ioannidis, "Why most discovered true associations are inflated," *Epidemiology,* vol. 19, no. 5, pp. 640-648, 2008.

[27] R. H. Thaler, "Anomalies: The winner's curse," *The Journal of Economic Perspectives,* vol. 2, no. 1, pp. 191-202, 1988.

[28] K. Peasron, "On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling," *Philosophical Magazine,* vol. 50, no. 5, p. 157–175, 1900.

[29] R. Fischer, Statistical Methods for Research Workers, Edinburgh: Oliver & Boyd, 1925.

[30] R. L. W. a. N. A. Lazar, "The ASA's Statement on p-Values: Context, Process, and Purpose," *The American Statistician,* vol. 70, no. 2, pp. 129-133, 2016.

[31] "Fisher's Method," [Online]. Available: https://en.wikipedia.org/wiki/Fisher%27s_method .

[32] R. Kohavi, "Online Controlled Experiments: Lessons from Running A/B/n

Tests for 12 years," 2015. [Online]. Available: http://bit.ly/KDD2015Kohavi.

[33] R. Johari, P. Leo and J. W. David, "Always valid inference: Bringing sequential analysis to A/B testing," 2015. [Online]. Available: https://arxiv.org/abs/1512.04922.

[34] A. Deng, J. Lu and S. Chen, "Continuous monitoring of A/B tests without pain: Optional stopping in Bayesian testing," in *DSAA, 2016.*

[35] A. Deng, P. Zhang, S. Chen, D. Kim and J. Lu, "Concise Summarization of Heterogeneous Treatment Effect Using Total Variation Regularized Regression," in *In submission.*

[36] "Simpson's paradox," [Online]. Available: https://en.wikipedia.org/wiki/Simpson%27s_paradox.

[37] "Multiple Comparisons problem," [Online]. Available: https://en.wikipedia.org/wiki/Multiple_comparisons_problem.

[38] "Bonferroni correction," [Online]. Available: https://en.wikipedia.org/wiki/Bonferroni_correction] .

[39] "Mobile Patterns," [Online]. Available: https://mobilepatterns.wikispaces.com/Coach+Marks.