Seven (Not So) Simple Steps in AB Test Design

Why your AB Tests aren't Working (Part II)

Authors: Donny Lofland and Christie Marsh, Ph.D

Introduction

In Part I of this series, *Why your AB Tests aren't Working*, we flesh out the core challenges with AB testing at a high-level (*and they are probably aren't what you think!*). Specifically, we explored issues around:

- "What can AB Tests *actually* tell us?",
- Challenges with changing or unknown users,
- Complexities with tests,
- and Incorrect or expiring conclusions.

In Part II, we will take a deep dive into the issues that you may face at each step of the AB test process. As with Part I, our goal is to share what we've learned from years of combined experience running AB tests in industry settings to help you identify the myriad challenges inherent to AB testing you may face that prevent meaningful and actionable results.

At a high level, Experimental Design is often broken down into the conceptual steps:

1. Hypothesis
2. Experiment

3. Data Analysis
4. Conclusion

While this is a good starting point, we really need to break these into more concrete actionable steps that help us improve our AB Test process. To help frame this discussion, let's start by breaking these down into a series of distinct practical steps more relevant to AB testing as seen in industry.

Key Terminology defined below article.

Seven (Not So) Simple Steps in AB Test Design

1. Clearly articulate your question
2. Do you have any existing data that informs or clarifies your question?
3. Identify your target audience
4. Define what metrics you will collect and how you will collect them
5. Define how long to run your test and how many users will be needed
6. Define success criteria and how this will be measured or determined
7. Communicate results

1. Articulate your Question

Start by clearly articulating the question you hope to answer. Often we see folks throwing ideas at the wall without any clear thought or theoretical support for why there should be differences. For example, rather than asking "I wonder if 10 seconds or 10 minutes is a better timeout for refreshing banner ads?", instead, start by thinking through the user experience and coming up with a hypothesis, which you can then test. We might say, "Users need time to notice a new banner ad and time to read and react." This now might lead to an AB test to determine if there is some minimum threshold at which banner ads start seeing better click-through rates (CTR). We could pose a separate thought, "Users might stop paying attention to Banner Ads if they don't refresh frequently enough". This then leads to a separate possible AB test to see if there is some upper bound on display times at which CTR drops. We have now framed two clear hypotheses that we can test and we can come up with some reasonable values to test for each.

When articulating your question, we want to identify AB tests (or experiments) that are likely to result in meaningful and measurable changes. This is a classic signal-to-noise ratio problem - if you start trying to test for subtle changes, you probably won't see those over the inherent variability seen in most AB tests. If you are trying to optimize the banner ad display time, don't test 30 seconds, 32 sec, 34 sec, etc … instead, go big and test 30 seconds, 60 sec, 5 minutes, and 10 min. We often think of AB tests as *one-and-done's*, but how are we to know what options might meaningfully change the experience without either drawing from existing research or taking big swings first? With a larger spread, you are more likely to see something change. If nothing changes, then it’s time to move on to a different question.

2. Existing Data or Exploratory Analysis

In our experience, not every question requires running an AB Test. From a business perspective, testing takes time, resources, labor costs, and infrastructure to set up, run, maintain, and analyze ... all affecting your bottom line. You also run the risk of hurting your bottom line in the short-term with all underperforming variants. The last thing a business should do is spend large sums on tests that are ROI negative. Assuming the question needs answering, organizations should determine if there is any historic data that might help answer the question or, failing that, help ensure your test design is efficient and more likely to result in detectable differences. It’s always best to gamble with loaded dice and here is your chance to improve your odds of finding a winner.

> *“It’s always best to gamble with loaded dice and here is your chance to improve your odds of finding a winner.”*

Data teams are uniquely positioned to help content creators find opportunities for improving goods and services. Sometimes starting with exploratory analysis can suggest areas of opportunity or areas where there isn’t much opportunity. Exploratory analysis also allows you to test assumptions about user behavior before making feature changes to act on hypotheses that may not even be true. Perhaps you believe that users aren’t clicking on your shop button because it is too small. Do you have other buttons in your UI that vary in size? Does their CTR vary as a function of button size? If not, then your button size hypothesis may not be true. Can your exploratory analysis identify what button features **are** correlating with button CTR?

3. Identify your Target Audience

User Sources

Each user in your experiment should be independent of all other users. If your users are associated, linked, making decisions as a group, or influencing each other, then AB testing becomes difficult if not impossible to conduct. This problem isn’t always obvious. For example, we have no control over multiple people from a single household sharing the same device or login information. We might have a group of users connected via social apps or forums who are communicating outside the AB test. Furthermore, if we don’t aggregate once over the entire duration of the test and instead make multiple comparisons across, for example, sessions or days, those observations are necessarily dependent.

If you are running multiple AB tests in parallel and allow users into multiple tests at once, you can get interactions between the tests. Hypothetically, if the tests are influencing and measuring completely different aspects of the user experience, you might be ok. However, in practice, this can become hard to monitor. Ultimately, interactions have to be accounted for and interactions often increase variability or spread of measured KPIs.

If you are planning to leverage marketing or user acquisition to bring in users for your test, be aware that how you acquire users could skew your results. What is best for this group, might not be best for your usual sources of users.

Random Sampling

If you already run AB tests, then you probably have the infrastructure in place to randomly assign your users to variants. If you are new to AB testing, then the concept of *random sampling* is critical for running tests. Each user needs to be randomly and independently assigned to one of your groups. For each user, you roll a dice and use that to assign users to variants. Random sampling helps ensure factors outside your experimental variable(s) are evenly distributed across groups so there is less chance that the differences you see are coming from external factors.

In lab settings, we often repeat experiments to verify the reliability of the results, but in industry, we don’t want to unnecessarily spend time on repeating experiments. Instead, we often run AAB tests where we assign 25% to Control 1, 25% to Control 2, and 50% to the Variant. The dual controls should have the same setup and help us to see how well random sampling worked and how much variability is coming from the random assignment process itself. If Control 1 and Control 2 are very close, then we can trust that random assignment successfully mitigated differences between groups due to assignment to groups alone and we can have more confidence that differences between Control and our Variant are real. If Control 1 and Control 2 are different, then either there is a problem with the random sampling algorithm, your population variability on the KPIs of interest is very large, or you might have unusual data points randomly clustering in one group or the other. If Control 1 and Control 2 are quite a bit different, then analysis and picking the best variant may be difficult. Note that after confirming Control 1 and Control 2 are similar, we often combine both Controls back into a single Control group for analysis.

If your groups are not of similar sizes (e.g., 90% Control and 10% Variant), then you may have class imbalance problems that you will need to correct for during analysis. The more different your group sizes, the greater the difference in variance between groups. Outliers can also have an increased impact on your smaller group. Ideally, you should strive to have groups of similar sizes. Practically speaking, we often want to test some change on only a small portion of our users, say 10%, leaving 90% of our users in the *Control* group. This is a very common setup with the intent to mitigate risk, but unfortunately, it can lead to incorrect results if not addressed during the analysis step.

While random sampling helps mitigate external factors, it doesn't necessarily remove them all or their influences and you can still see large sampling variation that leads to incorrect conclusions about which group is best. Consider an example where most of your user base spends very little money, but a few users spend a huge amount. In mobile games, it's common to see the majority of players spending $1 or less, but we also see a few players spending $1000's. If during random sampling assignment, the few high spenders managed to all get assigned to *Control*, then *Control* might look like a clear winner over *Variant*. However, this was caused by these outliers and how they managed to get distributed. During analysis, you may need to compare results with and without outliers, or you might need to use more advanced Bootstrapping techniques or resampling techniques to help determine which group might have been best if the outliers were more evenly distributed.

New Users vs. Existing Users

When setting up an AB Test, you will often choose to target only *new users* to avoid changing the experience out from under your *existing users*. This strategy helps mitigate risk by not negatively impacting your current users.

To collect enough data for analysis, you will often run tests for 1–2 weeks, though longer is certainly possible depending on your conditions. However, the longer you run a test, the greater the chance market fluctuations, seasonal variability, trends, or random fluctuations might start skewing results. We want tests to run as short a possible while collecting enough data to be relevant.

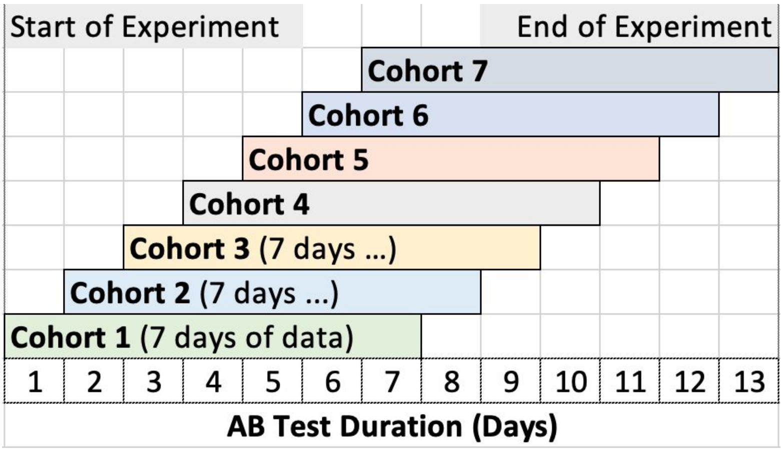
A challenge is that an AB Test may impact users differently over time. For example, your *Control* might be the winner for the first few days, but the *Variant* might then provide a more optimal user experience and take the lead as your winner. Extending this further, your *Control* might again perform better at a later point. The current winning variant will entirely depend on the levers you are pulling in the test, whether those levers can have different impacts at different points of a users' lifecycle, and the timing of when you call the winner and close your test.

This is especially a problem if your user activity is temporally biased, that is, most of your data points are concentrated at a certain point in the experience because that's when most of your users are active. If activity is front-loaded, you end up optimizing the early experience at the expense of the late experience. This begs the question of whether you'd retain more users if the experience were temporally optimized.

Another subtle complexity relates to the portions of *new* vs. *existing* users in your test. We often run a test for a fixed number of days and evaluate KPIs daily. If you use this approach, your measured KPIs may be dominated by *new* users during the first few days. As your test progresses, the *existing* retained users will add up and begin dominating measured KPIs. This imbalance in users may skew your results and interpretation. Here is an example of what your user base might look like over seven days. Notice how the portion of new users decreases over time.

Day	New Users	Existing Users	Total Users	% New
1	1000	0	1000	100%
2	1000	500	1500	66.7%
3	1000	800	1800	55.6%
4	1000	1010	2010	49.8%
5	1000	1178	2178	45.9%
6	1000	1329	2329	42.9%
7	1000	1465	2465	40.6%

Ideally, you should measure KPIs by the *user*, not *days*. You should also ensure that all users in your experiment have the same number of days of data. If you want to run an experiment for seven days, you would stop adding new users on day eight but continue to collect data until you have seven complete days of data for the last cohort of users that joined on day seven.



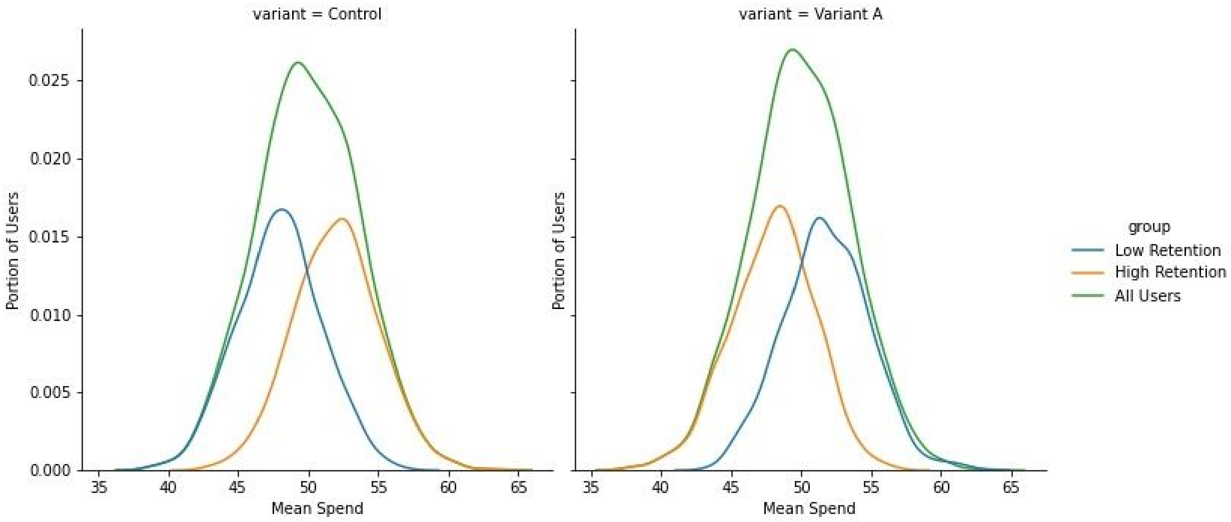
Equivalent Cohorts and Weekly Cycles

For most apps, websites, and even retail in general, there are clear weekly seasonal trends. When running AB Tests, you need to ensure that you collect cohorts for every day of the week and collect an equivalent number of days data for each cohort. Also, make sure to understand any longer seasonal, trend, or cyclic patterns in the KPIs you are using to make decisions. Ideally, you want a KPI that is relatively stable so you can more easily measure differences between your variants and attribute changes to the test itself. If your KPI of interest was already trending up or down and you then run an AB Test, this can bias interpretation. Yes, in theory, if each group is a random sample, then overall trends in a KPI should be equally distributed between variants. However, in practice, non-stable KPIs can increase variability, and observed sampling error could be more pronounced. Any interactions between what is causing the trend in the first place and the KPIs you measure in your AB test can also result in variants appearing better or worse than they are. If you can simplify your upfront design through careful planning, you'll have fewer hoops to jump through during the analysis phase.

Result May Vary Across Segments or Subgroups

No user base is homogenous. Your user base is probably a mixture of many distinct groups, each containing users who behave in similar ways or have similar characteristics. We usually define *dimensions,* or specific user characteristics, (e.g., spending habits, time-in-app, retention, willingness to engage with specific app features, etc.) to help categorize users based on behaviors. For example, subgroups might include *high spenders who churn quickly*, *low spenders who stick around for a while*, *users willing to engage with ads*, *users who link a social profile*, etc. When we run an AB Test, *Control* might be a clear winner with some subgroups, whereas your *Variant* might win with others. If looking at the aggregate results for each variant, we might not see a winner, but there could be clear winners when we dig into subgroups. This can complicate the interpretation of "winner."

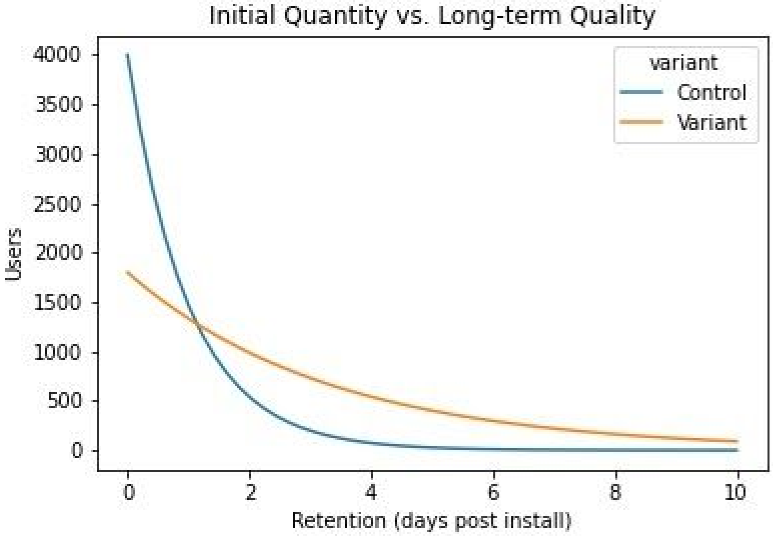
Say you choose *Control* as the winning variant, that decision might improve half your users' experience but hurt it for the other half. This is especially problematic should your user base start shifting to be more like the half that was hurt. In that case, your AB test ultimately ended up making your app generally worse overall instead of better. Most vendors offering off-the-shelf AB test solutions don’t have the functionality to dig into your test results to understand this nuance or to close AB tests to different variants for different subsets.



4. Picking your Metrics

We ideally want simple qualitative answers like “Control Wins” or “Variant B Wins.” However, there is a complex interaction between the users and your content. Often pulling a simple lever within an experiment can cause different KPIs to change in opposite directions. Choosing the “winner” can require judgment calls on which KPIs are most important given the test’s hypotheses and performance goals. This is why it is important to measure more than just one KPI as part of your AB test. If your test is aimed at maximizing click-thru-rate (CTR), but the users who click through are 75% more likely to quickly churn, then you’ll never know your variant that appears better on the surface might be worse. Measuring multiple metrics beyond the simple KPI of interest can help ensure you aren’t missing parts of the picture.

It’s also vital to ensure a mixture of “winner because of quantity” and “winner because of quality.” Suppose you are testing different creatives on your landing page, and you select the winning variant that maximizes the number of *installs*. In that case, you may end up ultimately hurting your app in the end if this variant also reduces the number of users retaining for more than a day by 300%.



We often want to pick easy to understand and interpret metrics, and conventional AB Testing often misses out by choosing these single point aggregate measures (e.g., mean or median), which mask underlying patterns. When designing an AB test, sometimes the metric of interest should be the shape of the distribution. We can then ask how the distribution changes between variants and are there any actionable information or insights.

5. Define Test Length and Number of Users

If you don't see a clear winner in your AB Test, you may be tempted to just run the test longer and collect more data until you do find a winner. Scientists understand that given a large enough dataset, "statistically significant" differences are virtually guaranteed; however, those differences are suspect and more likely to be meaningless or so small as to not effectively matter.   There is a term for this and it's called "p-value fishing". When we run AB tests, we want to find **meaningful** results or it's not worth our trouble.

When designing tests, you need to determine upfront whether you will be able to actually detect differences and if not, rethink your test. You should use a sample size calculation to estimate how many users you will need in each group to detect some threshold difference with some confidence level. Then, run your test long enough to gather those users and stop. If you don't find a statistical winner, then call it a tie and move on to another question.

6. Define Analysis and Success Criteria

Assumptions

Many standard statistical techniques used to analyze AB Tests and pick "winners" come with assumptions — common ones include:

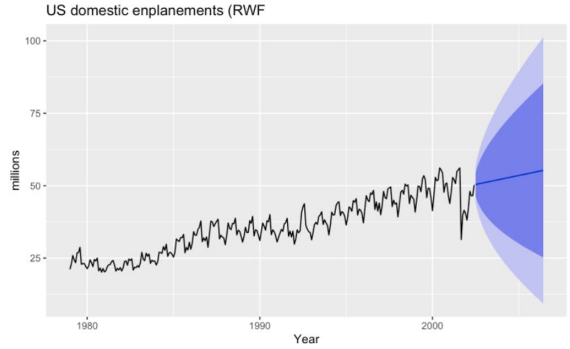
- **Normally distributed data** - ”Normal” data looks like a classic bell curve. If your data is skewed (not symmetrical around the mean, such as a distribution with a “long-tail”) or has any other shape that is non-normal, your data must be transformed to make it normal. That, or you have to use different summary metrics and statistical techniques to understand what is going on.
- **Independence of observations -** your data points within variants and between variants must be independent. What does this mean? The latter is straightforward: users in an experiment must be assigned to only one variant. The former is more complicated. Each measurement within the variant must be independent, i.e., they must not correlate with each other. If we run many AB tests at once, then it is virtually impossible to guarantee that users are entirely independent of each other, not to mention we have no control over multiple people from a single household using the same device or sharing login information. Furthermore, if we don’t aggregate once over the entire duration of the test and instead make multiple comparisons across, for example, sessions, those observations are necessarily dependent
- **Linearity** - Standard tests assume that the relationship between the features that have changed and the KPIs that are being measured is *linear.* This means we expect to see a simple relationship - as a feature increase by *x*, the KPI increases by *y*. Everyone learned the equation for a line, *y* = *m* * *x* + *b* in Algebra; however, sometimes features and KPI’s have more complex relationships. Common non-linear relationships include:
    - logarithmic and power curves,
    - quadratic and other polynomials,
    - cyclic relationships based on trigonometric functions (e.g., sin or cos).
- **Equality of variance in standard deviation** - the extent of the variation is the same across all variants, but with the vast diversity that exists in real-world populations as opposed to hand-selected experiment participants, this is impossible to guarantee.
- **P-values -** Statistics is all about probability. In experiments, we specifically want to know how probable our conclusions are correct and not from pure chance. Historically, *p-values* were used to help score how confident we are in our findings. In simple terms, the lower the p-value, the less chance we are wrong, and the higher the p-value, the greater the change we are wrong. Academic research is generally moving away from the use of p-value as p-values come with a host of interpretation challenges and can be very misleading.

Very often, we see analyses where the data do not follow these assumptions, and incorrect approaches are used to make decisions.

Extrapolation Problems

We cannot merely extrapolate or project results from an AB test now on future versions of your app or future populations of users. When you try to project trend lines into the future, there are unknowns and uncertainty. The best example is to visualize the projected path for a hurricane. While we have perfect knowledge of where it has been, the moment we try to forecast future locations, we are faced with wide confidence intervals of where it might go. The same is true of any measured KPI. Each separate hurricane we see is a random sample of all possible hurricanes. Even knowing the last hurricane's perfect track, while it might generally inform a likely trajectory for the next hurricane, there will be variation.

Often, business analysts will take 14 days of test data and attempt to forecast 60, 90, or even 365 days into the future and declare how profitable choosing a variant will be. Realistically you need a minimum of twice as much data as you wish to forecast (Saffo, 2007), so a forecast of 60 days requires a minimum of 120 days of historical data. This assumes nothing at all changes and the makeup of your user base stays the same.



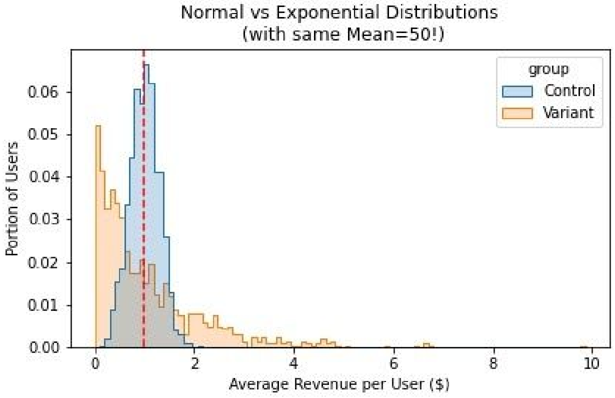
Aggregate KPIs Mislead (though some can be useful)

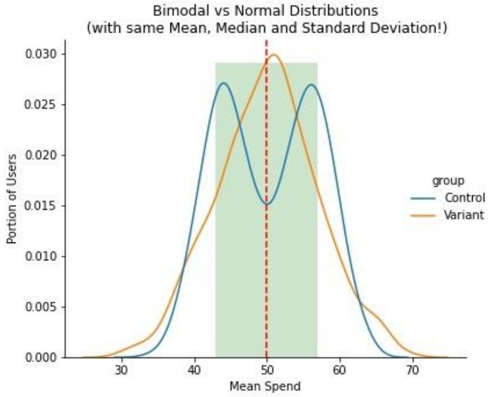
Aggregate or summary metrics, e.g., average revenue per daily active user (ARPDAU), come with assumptions on the data's shape. Using additive mean (or "average") with skewed data is NOT usually appropriate. Aggregate KPIs also mask essential information and bias towards the greatest common denominator when that may not be who or what you want to target. Before choosing summary metrics, you have to explore the data to ensure that summary metrics capture and communicate what is going on.

In the first chart, we illustrate a *Control* and *Variant* that have the same "average" (additive mean) labeled as the red line. If you were to only look at this simple summary metric and nothing else, you would miss the fact that *Control* and *Variant* actually look very different. *Control* has far more low-value users, but a few high-value users as well (notice the long tail to the right). Our *Variant*, with the same "mean," has moved more users away from the low value, but also reduced the number of high-value users. Since the *Variant* is normally distributed, using mean is appropriate. Our *Control,* however, is an exponential curve, where "mean" is misleading and doesn't convey that the bulk of our users are in the low-value category. A *median* or *geometric mean* might give us a more representative summary statistic when looking at skewed data.

Notice also that we can get the same mean with right- or left-skewed data and even bimodal data. To really understand what is going on, we need to plot our data and then decide how best to summarize it. Median might help, but figure two illustrates a normal curve and bimodal curve with the same median, mean and standard deviation! No one single metric is sufficient … we need to do some exploratory data analysis to see what is really going on.

Lastly, outliers can artificially raise or lower curves. The challenge with outliers is that we cannot rely on them to always show up so we need to be careful how we include them when summarizing data. In fact, sometimes outliers are the more interesting group and may show us areas of opportunity.





More is Sometimes Less

It’s a common misconception that if you didn’t find significance during the test period, you can collect more data until you see a *significant difference*. Yes, if you collect enough data, you will almost always find a point where one group becomes “significantly different.” However, the actual difference between the groups, or *effect size*, becomes smaller so that the *significant difference* holds little meaning. Keep in mind there is a cost to collecting more data — data costs, infrastructure costs, labor costs, and opportunity costs (not being able to do other work). With testing, we are hoping to find a *significant difference* AND large *effect size*. If both conditions aren’t met, we cannot reliably determine a “winner”.

If a winner cannot be reasonably found, have your domain experts pick a winner using their intuition or just close to the Control group and move on to your next question.

7. Communicate Results

A final challenge is that all the best statistics and graphs in the world may not be sufficient to communicate subtleties. Your audience will bring their background, assumptions, and biases into the picture. The most common misconception is that *AB Testing should be easy and there must be a winner*. Remind your audience that the harder you have to look for differences, the less valuable any conclusions are likely to be.

Conclusion

As we've seen, there are many subtle challenges to designing, running, and analyzing AB Tests. This article isn't to say AB Testing cannot or should not be performed, but rather, you should be aware of challenges when designing an AB Testing Framework. If you are embarking on AB testing or have an existing AB test program, here are some tips that might help you find more success.

Tips for Success

1. Plan, plan, plan - the more thought you put into a test before you run it, the less work you'll face during analysis, and the better the chance you'll find meaningful results.
2. Treat AB testing as a short-term optimization problem - if a question is critical or shows meaningful results, plan to circle back later and see if it's still a winner.
3. Simple dashboards and simple metrics in most cases will not tease out meaningful results. AB test analysis requires more sophisticated statistical techniques.
4. Be willing to accept that sometimes there is NO winner. In those cases, use your instinct to choose what you feel makes the most sense in the context of other changes you might want to make or other tests you might want to run. You are the domain expert and your intuition probably performs better than chance. If you are not seeing "winners", then ask whether you should be AB testing that aspect of your product or whether you should focus attention on other areas where you might have a better chance to move the needle.
5. If you aren't doing segment analysis yet, consider exploring that direction. Find the subgroups that are most valuable or that have the most potential for improvement. For these groups, think about AB tests that might improve their experience and pay less attention to parts of your user base where you don't see value or potential.

Looking to the Future

AB testing is not the only way to improve your products or services. Outside of simple questions like "Did the feature break something? Does the feature work?", AB testing is a costly, time-consuming endeavor and just not the right tool for tuning, optimizing, or finding differences, especially nuanced or subtle differences. That said, it's the tool most organizations have now and in some cases, it's better than nothing.

Looking ahead, we suspect more platforms and analytics providers will move towards individual and group-based optimization. Running AB tests with an assumption that all users will benefit from a single winning configuration will eventually be replaced with the more sophisticated techniques that allow you to target an optimal experience per user. Multi-arm bandit and real-time machine learning offer the benefits of on-going self-optimization so you do not have to "rerun" tests in the future. You will put in hooks in your product and let the system self optimize.

As you make changes or add features, those hooks will adjust so that users continue to get an optimal experience no matter what changes occur in your product or user base.

Terminology

Here is a list of terms we'll use during our discussion. These are commonly seen in marketing, especially when talking about customers or users.

- **App** - "Apps," short for applications, but could be websites, web applications, mobile apps, desktop apps, video games, or any interactive interface where users interact and make choices. When we use the term app, it could refer to any product or service offer.
- **Churn** - When a user leaves your app, never to return. Churn could also mean a customer that won't return to a retail establishment.
- **Cohort** - Typically, a group of users that all started on the same calendar day, for example, installing a mobile application or signing up for a service.
- **Engagement** - Any measure of how much someone is interacting with your service, app, or content. For example, the *number of views*, click-thru-rate (CTR), *time spent reading*, *time in-app*, or *progress through fixed user experiences*.
- **KPI** - This stands for "*Key Performance Indicator*" and is something we measure and use to quantify users or groups in some way. Common KPIs can be related to the user **value** (e.g., *lifetime value (LTV)*, *average revenue per user*), **engagement** (e.g., *in-app time*, *sessions per day*), **retention** (e.g., *days since install*), or any **app-specific measures** (e.g., *the level reached*, *screens visited*, *links clicked*).
- **Retention** - How long a user remains a potential customer or sticks around in your application or service.
- **Variability -** This is a measure of how much random noise is in your data. With any experiment, we expect there to be random factors outside our control. If everyone behaved in the exact same way, we'd expect little to no variability. However when running tests with real people, you will see a mix of player types making different choices.

- **Control** - The specific variant group that has seen no changes. This group serves as a baseline for comparing other groups. If you are doing something for the first time and don't have a baseline or control group, you can arbitrarily assign one group to be your control.
- **Variant -** Each distinct group of users in an experiment. If an AB Test consists of 4 groups, each group is called a *variant*. Typically, each group has a slightly different experience and we are trying to measure outcomes between groups.

References

Kenny, D. A. (1979). *Correlation and causality*. Wiley-Interscience, New York.

Saffron, Paul. 2007. “Six Rules for Effective Forecasting” Harvard Business Review, July–August 2007 Issue.

Thaler, Richard H. 2016. "Behavioral Economics: Past, Present, and Future." American Economic Review, 106 (7): 1577-1600.