DYNAMIC YIELD

# The Power of Dynamic Experiments

## Taking Your Testing to the Next Level

# Table of Contents

# Intro

A/B testing is one of the most widely used techniques for conversion optimization. It allows marketers to make smarter and more data-driven decisions about their creative ideas. Proposed changes are not implemented based purely on intuition (or by the person with the strongest persuasive skills), but rather based on specific desired goals – whether immediate or longer-term. At the same time, testing protects against making poor business decisions that may dramatically harm the user experience and bottom-line revenue. However, there are many pitfalls involved when conducting controlled experiments. If not done correctly, tests can fail to produce meaningful, valuable results.

In this eBook, we delve deeper into the mechanics of A/B testing, understanding the actual meaning of statistical significance and discovering some of the major pitfalls threatening the validity of test results. We'll review some of the alternative optimization methods to classic A/B testing and finish with presenting the ten golden rules for running successful and meaningful tests.

Armed with the right set of tools and testing models, marketers can harness the power of data to leverage historical and behavioral data per user, ultimately leading to more successful optimization initiatives.

# Taking a Leap of Faith

A classic A/B testing procedure is simple: First, we decide what we would like to test and what our goal is. Then, we create one or more variations of our original web element (a.k.a. the baseline). Next, we split the visitors' traffic randomly between two variations (i.e. we randomly allocate visitors according to some probability) and finally, we collect data regarding our web page performance (metrics). After a period of time, we look at the data, pick the variation that performed best and cancel the one that performed poorly. Sounds intuitive and straightforward? Not at all. We must keep in mind that the moment we pick a variation, we are generalizing the measures we collected up to that point to the entire population of potential visitors. This is a significant leap of faith, and it must be done in a valid way. Otherwise, we are eventually bound to make a bad decision that will harm the web page in the long run. The process of gaining validity is called hypothesis testing, and the validity we seek is called statistical significance.

In hypothesis testing, we begin by phrasing a claim called "the null hypothesis," which states the status quo, such as "the original page (baseline) has the same CTR as our newly designed page." A procedure is then defined to see if we can reject that claim as being highly improbable.

**So how do we do that?**

First, we have to understand where we could miss the mark and make a mistake. This could occur in two ways: First, we could wrongly reject the null hypothesis. After a brief look at the data, we might believe that there is a difference in performance between the new and the old variation of our web page, while actually no such real difference exists and what we observed is due to pure chance. This type of mistake is called "type I error" or "false positive". The second possible pitfall is that, after a brief look at the performance of each variation up to that point, we won't see a major difference and thus wrongly conclude that no difference exists. This error is called "type II error" or "false negative."

## How do we avoid these mistakes?
## The short answer is:  Define a proper sample size.

In order to determine the proper sample size, we have to predefine several parameters for our test: To avoid "false positive" errors, we need to define the confidence level, also known as "statistical significance." This number should be a small positive number, often set to <0.05, which means that given a valid model, there is only a 5% chance, or less, of making a type I mistake. In plain words, there is a 5% chance of detecting a difference in performance between the two variations where actually no such difference exists (a 5% chance of mistake, or less). This common constant is commonly referred to as having ">95% Confidence".

Many people conducting tests would stop with that first parameter, but to avoid the "false negative"-type II mistakes - we actually need to define two more parameters: One is the minimal difference in performance we wish to detect (if indeed one exists), and the other is the probability of detecting that difference, if such exists. This last quantity is called "statistical power," and it is often set to a default of 80%. The required sample size is then calculated using these three quantities (an online calculator can be found here). Although this may seem exhausting and the resulting sample size often seems way too high, the standard approach to testing requires following this procedure carefully, otherwise we are bound to fail.  In fact, even if we do follow the above to the letter we might still observe an incorrect outcome. Let's understand why.

# Reaching and Protecting Statistical Significance

While hypothesis testing looks promising, in reality it is often far from bulletproof, because it relies on certain hidden assumptions that are often not satisfied in real-life scenarios. The first assumption is usually pretty solid: We assume that the "samples" (namely the visitors we expose to the variations) are independent of each other, and their behavior is not inter-dependent. This assumption is usually valid, unless we expose the same visitor repeatedly and count these occurrences as different exposures.

The second assumption is that the samples are identically distributed. Simply stated, this means that the probability of converting is the same for all visitors. This, of course, is not the case. The probability of converting may depend on time, location, user preferences, referrer, etc.

For example, if during the experiment some marketing campaign is running, it may cause a surge of traffic from Facebook, for example. This may cause a drastic and sudden change in CTRs (click through rates), which is based on the fact that people coming from that particular campaign have different characteristics than the general visitor population. In fact, some of the more advanced optimization techniques we provide with Dynamic Yield depend on these differences.

The third and last assumption is that the measures that we sample, e.g. the CTR or conversion rate, are distributed normally. It might sound like some obscure mathematical term to some, but the "magic" confidence level formulas depend on this assumption, which is much shakier than the first two and often does not hold true. In general, the bigger the sample size and the higher the number of conversions we have, the stronger this assumption holds true – thanks to a mathematical theorem called the central-limit theorem.

## OK, I get that the math is not 100% bulletproof, but what pitfalls should I really watch out for?

Two main pitfalls exist. First, A/B testing platforms often offer real-time display of the test results collected up to that point. On one hand, this gives the test operator transparency and a feeling of control. However, this may cause us to react to results that are not yet 'done'. Repeatedly looking at the intermediate results and stopping a test before the pre-defined sample size and required significance level are reached is a sure recipe for making a mistake. This not only creates a statistical bias toward detecting a difference that is not there, but also, that bias is not something we can quantify in advance and correct (for example by demanding a higher significance level).

The second pitfall is to have too-high expectations with regard to the performance of the winning variation well after the test is over. Due to a statistical effect called "regression toward the mean," the performance of the winning variation over time will not be as good as it was during the test. Simply put, the winning variation may actually have won not just because it is really better, but also because it was to some extent "lucky." That luck oftentimes ends up being averaged out over time, and as a result, performance seems reduced.

# The Bandit Approach

The term “multi-armed bandits” suggests a problem to which several solutions may be applied. Multi-armed bandits allow to dynamically allocate traffic to variations that appear to be doing well, while allocating less and less traffic to underperforming variations.

**Multi-armed bandits are known to produce faster results, since there’s no need to wait for a single winning variation.**

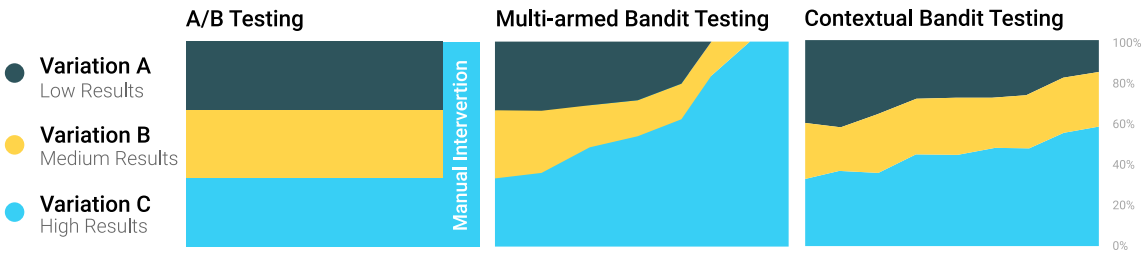
Bandit algorithms go beyond classic A/B/n testing, conveying a large number of algorithms to tackle different problems, all for the sake of achieving the best possible results. With the help of a relevant user data stream, multi-armed bandits become contextual. Contextual bandits for website optimization rely on an incoming stream of user context data, either historical or fresh, which can be used to make better algorithmic decisions, and of course, all this happens in real time.

## Contextual bandits are very powerful when trying to optimize different dimensions simultaneously, especially over short periods of time.

Here is a visualization showing how the classic A/B/n testing approach would split the traffic between three different variations, vs. the multi-armed bandit and contextual bandit testing approaches. This visualization shows how bandit tests can yield better results over time. While the classic approach requires manual intervention once a statistically significant winner is found, bandit algorithms learn as you test, ensuring dynamic allocation of traffic for each variation.

With contextual bandits, the algorithms change the population exposed to each variation to maximize variation performance, so there isn't a single winning variation. Similarly, there isn't even a single losing variation. At the core of it all is the notion that each person may react differently to different content. For example, a classic A/B test for the promotional content of a fashion retail site that consists of 80% female customers would result in the false conclusion that the best performing content would be promotions targeted for females, although 20% of the customers would expect to have promotions targeted to males. A contextual-bandit approach allows us to push beyond the limitations of traditional A/B testing, to broaden the scope of optimization possibilities.

Variation Allocation in Different Test Methodologies Over Time



# Choosing the Right Objective

A conversion optimization process can fail for a variety of reasons. One of the biggest considerations when running experiments revolves around choosing the right optimization objective. Generally speaking, conversions are measured when a visitor takes actions that are defined as valuable to your business. A conversion can be measured as a click on an element, a goal completion or an online action which has a direct impact on revenue. Whether you're trying to improve display ad clicks using CTR-based optimization, generate more leads using goal-based optimization or increase sales using revenue-based optimization, choosing the right objective for your optimization initiatives can make a huge difference between failure and success.

Having a specific, measurable objective in mind, however, is not enough, and that's the first thing you need to realize before planning a test. Many other variables are involved, such as the type of experiment you're running, the audience sample size, experiment length, conversion attribution, the number of content variations that are being tested, and how prominent the changes being tested are. It's not just about measuring the results - it's about optimizing the data as effectively as possible using the right optimization method.

Here is a table that illustrates this point. Use it to get a better idea of what the optimization objective of your experiments should be, based on sample size and estimated length of the experiment. If you don't know what the sample size is, use a sample size calculator like the one Evan Miller offers. Of course, there is no silver bullet, and this table can only serve to guide you towards making the right decision.

## How to Choose the Right Optimization Objective:

| | Low Sample Size | Medium Sample Size | High Sample Size |
|---|---|---|---|
| **Short Experiment Length** | No Optimization | Goal Completion Optimization | Goal Completion Optimization |
| **Medium Experiment Length** | CTR Based Optimization | Goal Completion Optimization | Revenue-Based Optimization |
| **Long Experiment Length** | CTR Based Optimization | Revenue-Based Optimization | Revenue-Based Optimization |

Time is of great concern, and it's one of the main factors influencing the decision of choosing the right objective for your experiments. If you're running an eCommerce site and intending to get fast results for an experiment you're just about to launch, remember this: Revenue-based optimization requires greater purchase cycles than CTR or goal-based optimization.

With revenue-based optimization, it's far more complicated to attribute the actual conversion to a single experiment variation. It takes more time for the random visitor who got exposed to an experiment to complete a purchase than it takes to complete a click on an element. Some customers compare prices, some are just slow buyers. In fact, many things can happen between the first exposure to a live experiment and the final destination of completing a purchase. So as a general rule of thumb, if you're looking for fast results, forget about optimizing for revenue and stick with shorter and simpler conversion cycles, such as click-based optimization or goal completion optimization.

Drawing conclusions too quickly is also a major issue. It can be difficult to keep your cool and ignore an (allegedly) obvious winner a few days into a test. While results vary across industries and sample sizes, waiting for the optimization process to accumulate enough data to achieve statistical significance is absolutely crucial. Even if your testing tool forecasts incredible uplift, keep the test running for at least a few weeks or a few purchase cycles before starting to analyze results. While some experiments reveal major uplifts early in the test process, results can vary dramatically over time, causing many marketers to draw the wrong conclusions and even lose money over time if they implement changes based on the initial results.

If you can't send enough traffic and generate conversions to all variations, your efforts will go unrewarded. Without sufficient data (e.g. traffic or conversions), the sample size will be too low, your experiments will never reach the statistical significance minimum of 95%, and the system will struggle to optimize and gain real uplifts. Thus, it will end up taking a lot of time and the test is bound to fail, so there's really no point in running short-term optimization initiatives for low-traffic sites.

Every test is fallible, and every test requires considerable resources (budget, time, and people), so stick to the elements you know make a difference, namely headlines, hero shot, and call-to-action. The fact is, the larger the sample size, the more data becomes available for the system to optimize. But make sure you test noticeable changes that are big enough to have a real impact on the bottom-line business. Start testing minor features and you'll find yourself lost in data you can't use.

As with any tool or methodology, a single experiment will not revolutionize your business, so don't expect the test to produce a "Silver Bullet". However, integrating testing over time and rolling out insights into your website will introduce incremental improvements that can (and do) reach a tipping point. Take a granular approach, choose the right objective and run your optimization initiatives intelligently and methodically. Plan for small wins, and they'll add up to a big difference in the long run.

# Revenue-based Optimization

**Running experiments with a revenue-based goal in mind is a pretty tricky business. There are several major challenges involved:**

- Revenue-based tests tend to have significantly smaller amounts of conversions compared to CTR (Click-through Rate) tests ("clicks are cheap", as they say…).
- Clicks tend to occur very shortly after impressions, as opposed to purchases, which typically take longer. As a result, revenue-based goals mean significantly more time elapses after users enter the test and receive a variation. This creates a delayed reward problem, leaving many "pending" users in limbo until they buy.
- As their name suggests, revenue-based experiments track the variance of revenue events. This can actually become very complex to implement when compared to the CTR case. Instead of just tracking the total impression and click count for CTR-based confidence calculation, all data-points (user-to-revenue pairs) need to be loaded to calculate the variance. High variance (i.e. purchases having a wide range of dollar values) also makes it difficult to produce conclusive results within a reasonable time frame.

By their very nature, revenue-based experiments inevitably mix between users who haven't converted for quite a while, and others who recently joined the test (and therefore are more likely to convert over time). This is simply the nature of the revenue-based beast, and that's OK – as long as all variations share the same mix.

However, once your shiny "Multi-arm Bandit" test tools periodically re-allocate each variation's traffic share based on performance, a fourth challenge surfaces: You have implicitly introduced bias among the user groups. Users have a better chance of "surviving" re-allocation if they're assigned to variations with a larger traffic share, and therefore have more time to convert.

As a result, these variations appear to show improved performance that could actually be misleading. Bias is hard to anticipate, and might ultimately lead to the wrong conclusions.

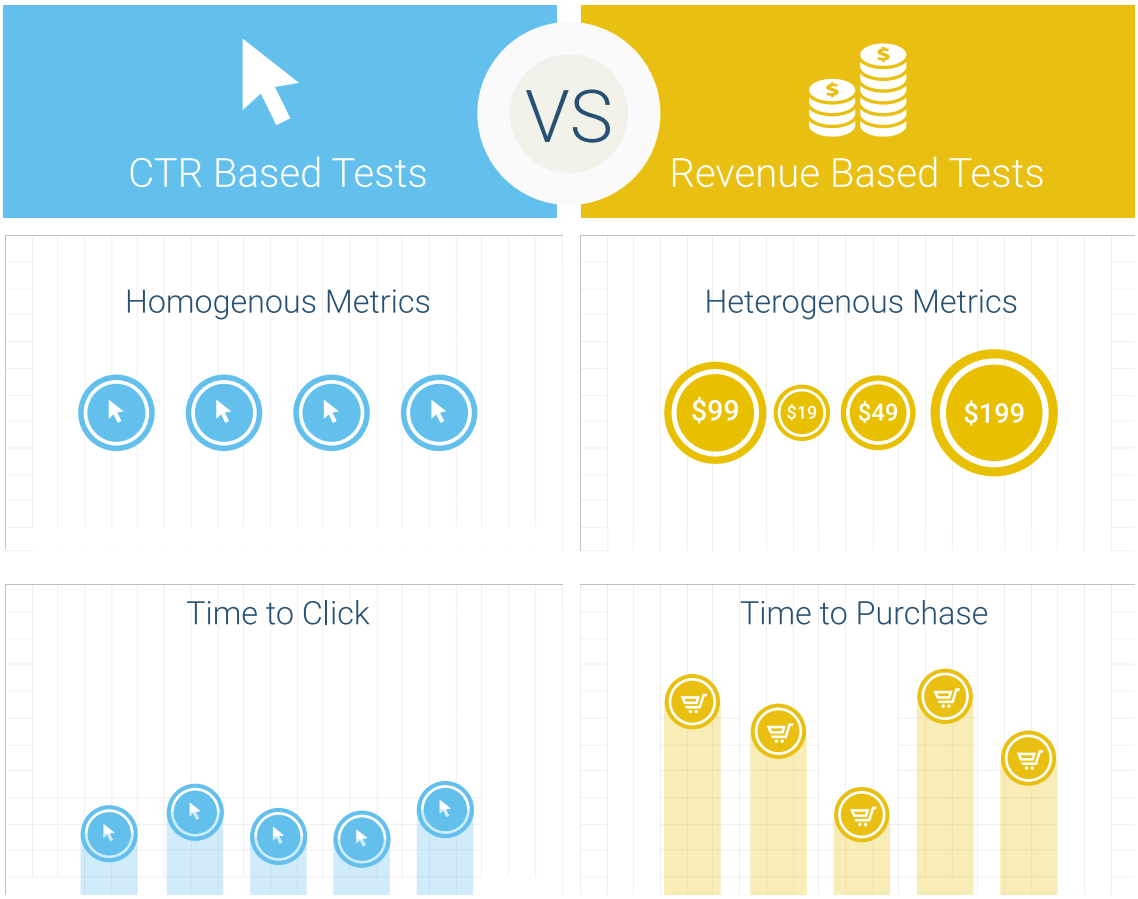
Of course, underlying variance and delayed rewards are both inherent properties of such tests. They're not going to magically disappear. The best solution is to reject the misleading revenue-per-user metric, and replace it with the more relevant revenue-per-visit unit of measurement. A visit is a widely-used but potentially vague term. In our context, a visit is considered to match a single unit of "user experience" in which the user is exposed to a particular variation under the assumption that the outcome represents either a conclusive conversion, or a decision not to convert.

As an example, suppose you only have enough screen real-estate to show one ad banner. To test which of four ads would yield the highest revenue, you assign each ad-click a different value (e.g. cost per click value, or the value of a lead successfully entering your funnel). In this case, the delay from impression to conversion is relatively short, so it makes sense to expose the user to a different variation upon each subsequent impression. The visit is therefore defined as the short time interval following a single impression until goal completion. Every impression marks a new unit of experience.

In contrast, assume your experiment begins on the homepage, but tests the more ambitious goal of an actual product purchase. That coveted purchase is typically generated at the end of the session. For our purposes, a session is defined as a single browser session. In this case, you would calculate the cumulative revenue generated for the entire duration of the browser session before recording the data point. In other words, each data point is assigned either a zero revenue value or the exact value of the purchase.

A visit is therefore a relatively flexible term designed to closely measure user behavior. However, to remain consistent, avoid relying on arbitrary logic such as "user-revenue per round-hour" (which might be tempting as it's easier to understand and perhaps implement). Such a definition would cut the revenue stream along arbitrary points in time uniformly (e.g. at the end of the hour), ignoring the complex variance of user behavior.

Once you settle on the above definition of a visit, you can measure each variation's performance by revenue-per-visit instead of revenue-per-user. During the experiment, the value of each visit is accumulated across multiple data points, and a given visit's outcome is not added as a new data point until that visit is considered concluded. With each new visit, a random variation is assigned to the user once – serving up the same variation for the duration of that visit.

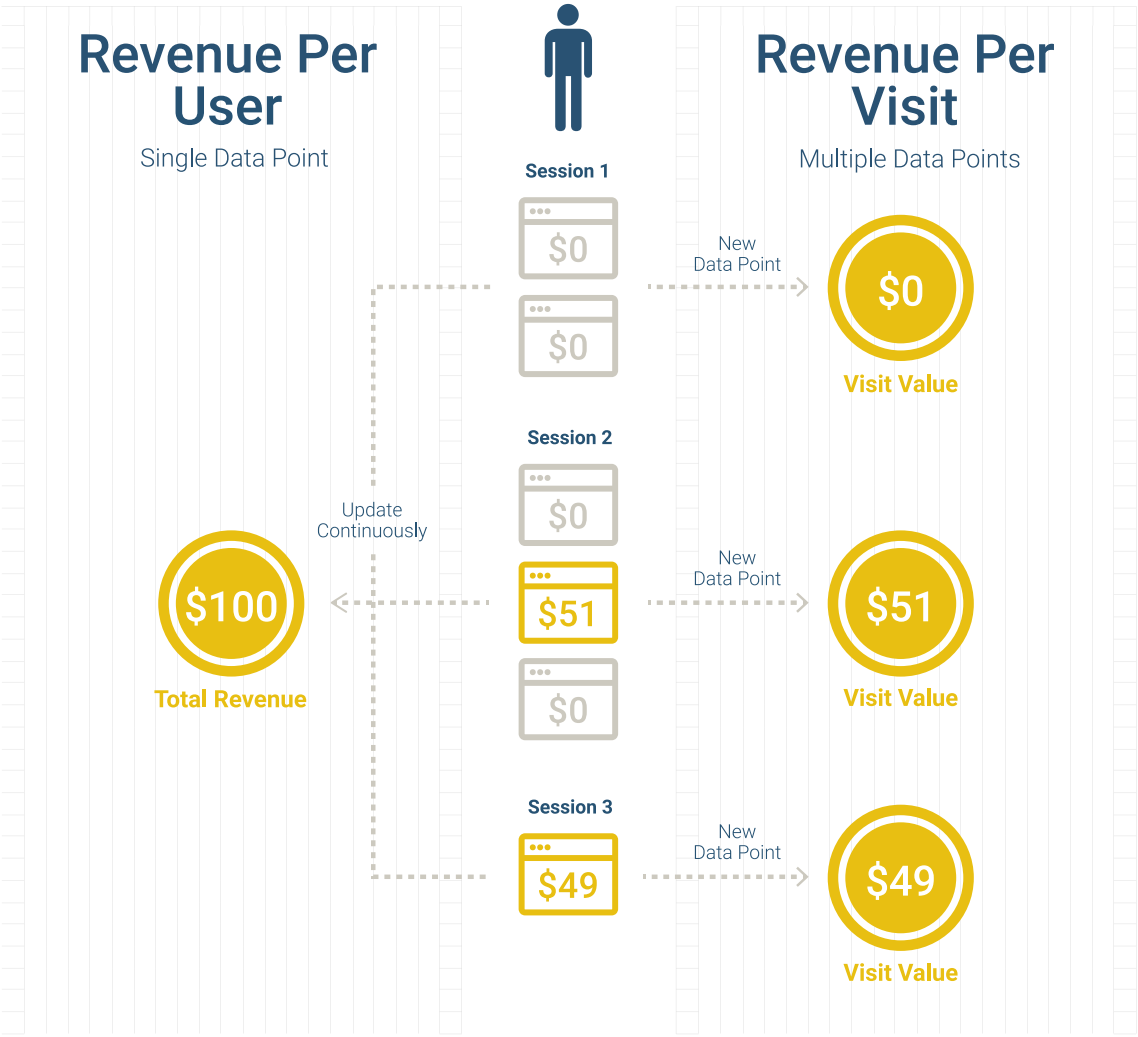


## So, how does this definition of visits mitigate our discussed challenges?

- In the case of very few conversions, it actually doesn't. As mentioned in Part I, when dealing with small amounts of data there is no alternative to running the experiment for a longer duration and letting sufficient data accumulate.
- The second challenge of delayed rewards is alleviated by uniformly adding data points after a visit is concluded. In this way, we avoid giving our variations a statistical down-bias from incomplete visits. Counting visits replaces the misleading user count with a more meaningful unit of measurement (even though visits still vary considerably in length).
- As for the third challenge of variance, the problem is now simpler. Once each visit is known to be concluded, you have a conclusive and consistent value (in contrast to per-user revenue, which might change at any time). On high-volume tests, the variance can be periodically updated, so that all data points up to that time can be discarded. Data points are time-capped, while users are not. This way, variance is better controlled, though of course not eliminated altogether (unless every user is limited to a single conversion of identical value). Note to advanced marketers: Replacing users with visits becomes even more powerful when you can dynamically re-allocate traffic. In this case, you eliminate user bias completely as your discrete measurement unit of work is visits.

Measuring revenue-per-user means updating a single data point that represents the cumulative revenue generated by a single user. In contrast, the proposed unit of revenue-per-visit creates a significantly richer and more uniform data set. The difference is especially dramatic when running dynamic experiments with revenue-based goals and allocating traffic in real time.

# Revenue-based Testing Methodology




Of course, if your definition of visit conflicts with how users behave, you can still end up with misleading test results. Knowing how to properly define a visit within the context of your use-case is a skill that comes with experience. However, as a general rule, the deeper your goal is buried in the funnel, the longer a "visit" duration should be. If a typical purchase on your website takes on average of three browser sessions to materialize, you should, in principle, define your visit to be "three browser sessions" (all happening within a reasonable time interval, after which the visit is ended).

# The Road to Success

*"I didn't fail the test, I just found 100 ways to do it wrong."*

Benjamin Franklin

When running an A/B test, using a valid methodology is crucial for our ability to rely on the test results to produce better performance long after the test is over. In other words, we try to understand if tested changes directly affect visitors' behavior or occur due to random chance. A/B testing provides a framework that allows us to measure the difference in visitor response between variations and, if detected, establishes statistical significance, and to some extent causation.

One of the fundamental goals of statistical inference is to be able to make general conclusions based on limited data. When performing A/B tests, the scientific phrase "statistically significant" sounds so definitive that many marketers and users of A/B testing solutions rely on it to conclude the observed results of the tests. Sometimes, even a tiny effect can make a huge difference, eventually altering the "significance" of the conclusions. Sticking to a strict set of guidelines will deliver more reliable results, and thus more solid conclusions:

1

## High confidence level

Try to get as close to a 99% confidence level as possible in order to minimize the probability of reaching the wrong conclusions.

2

## Be patient

Don't jump to conclusions too soon, or you'll end up with premature results that can backfire. You know what? Stop peeking at the data as well! Wait until the predefined sample size is reached. Never rush your conversion rate optimization, even if your boss pushes you to get results too fast. If you can't wait and need potentially faster results, choose tools that can actually achieve reliable results faster, as a result of a mathematical prediction engine, or a multi-armed bandit approach. That being said, there's no real magic. Be patient.

3

## Run continuous or prolonged tests for additional validations

If you don't trust the results and want to rule out any potential errors to the test validity, try running the experiment for a longer period of time. You'll get a larger sample size and that will boost your statistical power.

4

## Run an A/A test

Run a test with two identically segmented groups exposed to the same variation. In almost all cases, if one of the variations wins with high statistical confidence, it hints that something may be technically wrong with the test. Most A/B testing platforms use a standard p-value to report statistical confidence with a threshold of 0.05. This threshold is problematic, because when not enough data is collected, these tools may reach statistical significance purely by chance - and too soon (this is often due to the fact that not all assumptions of the model are valid).

5

## Get a larger sample size or fewer variations

If you're able to run the test on a larger sample size, you will get higher statistical power, which leads to more accurate and more reliable results. On the other hand, if you're using more than two variations and don't have enough traffic volume for proper, valid results, try reducing the number of content variations.

6

## Test noticeable changes

Testing minor changes to elements on your site may lead you farther away from any statistically significant conclusions. Even if you're running a high-traffic site, test prominent changes.

7

## Don't jump into behavioral causation conclusions

As marketers, we often base decisions on our intuition regarding the psychology of the visitor. We believe we know the reason for the visitor's positive/negative reaction to variations. A/B testing comes in to help us rely a bit less on our instincts and a bit more on concrete evidence. Good marketing instincts are useful for creating testing ideas and content variations. The A/B test will take us the extra mile by enabling us to base our instincts on data.

8

## Don't believe everything you read

Although reading case studies and peer testing recommendations is great fun, find out what really works for you. Test for yourself. Remember that published statistics sometimes tend to be over optimistic and not representative.

9

## Keep your expectations real

More often than not, following the end of a successful A/B test, there's an observed reduction in the performance metrics of the winning variation. This phenomenon is called regression toward the mean, and it is not something that can be quantified and corrected in advance. So, to avoid making the wrong conclusions, lower your expectations once a test is over.

10

## Test continuously and never stop thinking and learning

Websites are dynamic, so your ideas and thoughts should be, too. Evolve and think forward. Remember that the downside of all traditional A/B testing tools is that, eventually, these tools direct you to make static changes to your site, which may or may not fit all of your potential users in the long run. In other words, you're not necessarily maximizing your conversion rates by serving only one winning variation to all of your visitor population or by conducting short-term tests. Some tools allow you to personalize the delivery of variations based on machine-learning predictions (that require a relatively large amount of data) instead of waiting for one variation to win globally. With these tools, instead of catering your website to the lowest common denominator, you can actually deliver a better user experience by dynamically choosing to display the right variation to the right users.

# The Future of A/B Testing (is Actually Here)

**The term "Dynamic Experiments" represents a new approach for achieving sustainable value more quickly and efficiently. With the limitations of traditional A/B/n testing in mind, it offers a new way to conduct sophisticated experiments. In particular, this approach allows marketers to:**

- Dynamically deliver the best performing variation at the user level, ensuring each customer sees the best performing personalized content variation, instead of a generic, one-size-fits-all (and therefore misleading) winner.
- Directly optimize experiments for maximum revenue-based performance by migrating from a static A/B/n approach to a flexible platform that can automatically fine-tune every element in order to maximize value.
- Gain full control over every piece of data, all in real time, allowing marketers to make better, more effective data-driven decisions.
- Run complicated optimization initiatives with minimum need for IT, even when setting up complex experiments in ever-changing dynamic environments.

As effective as these tests can be, they can also be misleading if not properly conducted. Reaching proper statistical significance is critical for reliable results. To get there, we need to determine the required parameters, and estimate and then stick to the required sample size. Sampling the results frequently ahead of reaching the sample size target is a classic recipe for losing validity.

If you are looking for shorter-term optimization, you should look into other optimization methods, such as multi-armed bandit or machine-learning-based personalization, both featured by Dynamic Yield. In fact, the approach we take is multi-layered: Tests start with automatic traffic tweaks for all visitors, then personalization kicks in when there's enough data. All the while, you can apply manual rules, as simple or as complex as you need, to get better control of who gets what and when.

**Contact Us**
to Take Your Testing
to the Next Level

# About Dynamic Yield

Dynamic Yield's unified customer engagement platform helps marketers increase revenue by automatically personalizing each customer interaction across the web, mobile web, mobile apps and email. The company's advanced customer segmentation engine uses machine learning to build actionable customer segments in real time, enabling marketers to take instant action via personalization, product/content recommendations, automatic optimization & real-time messaging.

Dynamic Yield personalizes the experiences of more than 500 million users globally and counts industry leaders like like Sephora, Urban Outfitters, Europe's fashion leader Lamoda, MediaDC, and Liverpool Football Club among its many customers. Based in New York, the company has more than 100 employees in eight offices worldwide.