Get to statistical significance faster with Stats Accelerator

Skip Ahead

Weighted improvement
How Stats Accelerator affects the Results page
Modify an experiment when Stats Accelerator is enabled
Technical FAQ
Additional resources

THIS ARTICLE WILL HELP YOU:

• Understand Optimizely's Stats Accelerator and how it affects your results
• Determine whether to use Stats Accelerator for your experiments
• Enable Stats Accelerator for your account

If you run a lot of experiments, you face two challenges. First, data collection is costly, and time spent experimenting means you have less time to exploit the value of the eventual winner. Second, creating more than one or two variations can delay statistical significance longer than you might like.

Stats Accelerator helps you algorithmically capture more value from your experiments by reducing the time to statistical significance, so you spend less time waiting for results. It does this by monitoring ongoing experiments and using machine learning to adjust traffic distribution among variations—in other words, it shows more visitors the variations that have a better chance of reaching statistical significance. In the process, it attempts to discover as many significant variations as possible.

If you're running a multivariate test, you can only use Stats Accelerator in partial factorial mode. Once Stats Accelerator is enabled, you can't switch directly from partial factorial to full factorial mode. If you want to use full factorial mode, you'll have to set your distribution mode to Manual.

Note also that Stats Accelerator cannot be used in Feature Management.
Weighted improvement

Stats Accelerator relies on dynamic traffic allocation to achieve its results. Anytime you allocate traffic dynamically over time, you run the risk of introducing bias into your results. Left uncorrected, this bias can have a significant impact on your reported results.

Stats Accelerator neutralizes this bias through a technique called **weighted improvement**.

Weighted improvement is designed to estimate the true lift as accurately as possible by breaking down the duration of an experiment into much shorter segments called **epochs**. These epochs cover periods of **constant allocation**: in other words, traffic allocation between variations does not change for the duration of each epoch.

Results are calculated for each epoch, which has the effect of minimizing the bias in each individual epoch. At the end of the experiment, these results are all used to calculate the estimated true lift, filtering out the bias that would have otherwise been present.

To illustrate this, let’s look at the charts below. The first chart shows conversion rates for two variations when traffic allocation is kept static. In this example, conversions for both variations begin to decline after each has been seen by 5,000 visitors. And while we see plenty of fluctuation in conversion rates, the gap between the winning and losing variations never strays far from the true lift.

The steady decline in the **observed** conversion rates shown above is caused by the sudden, one-time shift in the **true** conversion rates at the time when the experiment has 10,000 visitors.

In the next chart, we see what happens when traffic is dynamically allocated instead, with 90 percent of all traffic directed to the winning variation after each variation has been seen by 5,000 visitors. Here, the winning variation shows the same decline in conversion rates as it did in the previous example. However, because the losing variation has been seen by far fewer visitors, its conversion rates are slower to change.
This gives the impression that the difference between the two variations is much less than it actually is.

This situation is known as Simpson's Paradox, and it's especially dangerous when the true lift is relatively small. In those cases, it can even cause the sign on your results to flip, essentially reporting winning variations as losers and vice versa:

---

### How Stats Accelerator affects the Results page

When Stats Accelerator is enabled, the experiment's results will differ from other experiments in four visible ways:

- Stats Accelerator **adjusts the percentage of visitors** who see each variation. This means visitor counts will reflect the distribution decisions of the Stats Accelerator.

- Stats Accelerator experiments **use a different calculation** to measure the difference in conversion rates between variations: weighted improvement. Weighted improvement represents an estimate in the true difference conversion
rates that is derived from inspecting the individual time intervals between adjustments. See the last question in the Technical FAQ for details ("How does Stats Accelerator handle conversion rates that change over time and Simpson's Paradox?").

- Stats Accelerator experiments and campaigns use absolute improvement instead of relative improvement in results to avoid statistical bias and to reduce time to significance.

Relative improvement is computed as:

\[
\frac{(\text{Variation conversion rate} - \text{Baseline conversion rate})}{\text{Baseline conversion rate}}
\]

Absolute improvement is computed as:

\[
\text{Variation conversion rate} - \text{Baseline conversion rate}
\]

Stats Accelerator reports absolute improvements in percentage points, denoted by the "pp" unit:

Additionally, the winning variation displays its results in terms of approximate relative improvement as well. This can be found just below the absolute improvement (in this example, the relative improvement is -12.15%), and is provided for continuity purposes, so that customers who are accustomed to using relative improvement can develop a sense of how absolute improvement and relative improvement compare to each other.

Because traffic distribution will be updated frequently, Full Stack customers should implement sticky bucketing to avoid exposing the same visitor to multiple variations. To do this, implement the user profile service. See our developer documentation for more detail.

Modify an experiment when Stats Accelerator is enabled

It is possible to modify an experiment if you have Stats Accelerator enabled. However, there are some limitations you should be aware of.

Prior to starting your experiment, you can add or delete variations for Web, Personalization and Full Stack experiments as long as you still have at least three variations.

You can also add or delete sections or section variations for multivariate tests, provided that you still have the minimum number of variations required by the algorithm you’re using.

Once you’ve started your experiment, you can add, stop, or pause variations in Web, Personalization, and Full Stack
experiments. However, for a multivariate test, you can only add or delete sections. You cannot add or delete section variations once the experiment has begun.

When Stats Accelerator is enabled for a test, it will periodically re-publish the Optimizely snippet, so the variation traffic distribution changes can go live. This is the same as a regular publish. When this happens, any pre-existing unpublished changes will be published as well.

Technical FAQ

How does Stats Accelerator work with Stats Engine?
Stats Engine will continue to decide when a variation has a statistically significant difference from the control, just as it always has. But because some differences are easier to spot than others, each variation will require a different amount of samples allocated to it in order to reach significance.

Stats Accelerator decides how many samples each variation should be allocated in real-time to get the same statistically significant results as standard A/B/n testing, but in less time. These algorithms are only compatible with always-valid p-values, such as those used in Stats Engine, that hold with all sample sizes and support continuous peeking/monitoring. This means that you may use the Results page for Stats Accelerator-enabled experiments just like any other experiment.

What algorithms or frameworks does Stats Accelerator support?
Optimizely draws from the research area of multi-armed bandits. Specifically, for pure-exploration tasks, such as discovering all variants that have statistically significant differences from the control, algorithms in use are based on the popular upper confidence bound heuristic known to be optimal for pure-exploration tasks (Jamieson, Malloy, Nowak, Bubeck 2014).

Can I use my own algorithm?
Using the REST API, you can programmatically adjust Traffic Allocation weights as needed. Optimizely’s out-of-the-box Stats Accelerator feature was finely tuned based on millions of historic data and state-of-the-art work in the field of bandits and adaptive sampling.

How much time will I save with Stats Accelerator?
Users typically achieve statistical significance two to three times faster than standard A/B/n testing when using Stats Accelerator. This means with the same amount of traffic, you can reach significance using two to three times as many variants at a time as was possible with standard A/B/n testing.

How often does Stats Accelerator make a decision?
The model that dictates Stats Accelerator is updated hourly. Even for Optimizely users with extremely high traffic, this is more than sufficient to get the maximum benefits of a dynamic, adaptive allocation. If you require a greater or lower frequency of model updates, please let us know.

What happens if I change the baseline on the Results page?
There is no adverse impact to selecting another baseline, but the numbers may be difficult to interpret. We suggest keeping the original baseline when you interpret Results data.

What happens if I change my primary metric?
The Stats Accelerator scheme reacts and adapts to the primary metric. If you change the primary metric mid-experiment, the Stats Accelerator scheme will change its policy to optimize that metric. For this reason, we suggest you do not change the primary metric once you begin the experiment or campaign.

What happens when I pause or stop a variation?
We recommend that you refrain from doing this. Though Stats Accelerator will ignore those variations’ results data when adjusting traffic distribution amongst the remaining live variations, the variation that’s been paused/stopped will exhibit conversion events due to delayed conversions.

Since Weighted Improvement is a weighted sum of the observed effect size within epochs, subsequent periods—after the variation’s been stopped—will yield a larger effect size due to delayed conversion events but no decisions, resulting in a
skewed Weighted Improvement that is misleading.

If you believe that a variation is underperforming, we recommend letting Stats Accelerator determine this, after which it will minimize traffic to this variation (because it's reached statistical significance) so it can funnel remaining traffic to the other variations. Otherwise, create a new A/B test with the variation removed.

How does Stats Accelerator handle revenue and numeric metrics?

For numeric metrics like revenue, the number of parameters to fully describe the distribution may be unbounded. In practice, Optimizely uses robust estimators for the first few moments (for example, the mean, variance, and skew) to construct confidence bounds that are used, just like those of binary metrics.

How does Stats Accelerator work with Personalization?

In Personalization, the Stats Accelerator option can be found in the settings for an individual experience. Stats Accelerator will automatically adjust traffic distribution between variations within campaign experiences. This will not affect the holdback. To maximize benefit, you should increase your holdback to a level that would normally represent uniform distribution. For example, if you have 3 variations and a holdback, consider a 25% holdback.

What is the mathematical difference between Stats Accelerator and Multi-Armed Bandit?

In simple terms, if your goal is to learn whether any variations are better or worse than the baseline and take actions that have longer-term impact to your business based on this information, use Stats Accelerator. If, on the other hand, you just want to maximize conversions among these variations, choose Multi-Armed Bandit.

In traditional A/B/n testing, a control schema is defined in contrast to a number of variants that are to be determined better or worse than the control. Typically, such an experiment is done on a fraction of web traffic to determine the potential benefit or detriment of using a particular variant instead of the control. If the absolute difference between a variant and control is large, only a small number of impressions of this variant are necessary to confidently declare the variant as different (and by how much). On the other hand, when the difference is small, more impressions of the variant are necessary to spot this small difference. The goal of Stats Accelerator is to spot the big differences quickly and divert more traffic to those variants that require more impressions to attain statistical significance. Although nothing can ever be said with 100% certainty in statistical testing, we guarantee that the false discovery rate (FDR) is controlled, which bounds the expected proportion of variants falsely claimed as having a statistically significant difference when there is no true difference (users commonly specify to control the FDR at 5%).

In a nutshell, use Stats Accelerator when you have a control or default and you’re investigating optional variants before committing to one and replacing the control. With Multi-Armed Bandit, the variants and control (if it exists) are on equal footing. Instead of trying to reach statistical significance on the hypotheses that each variant is either different or the same as the control, Multi-Armed Bandit attempts to adapt the allocation to the variant that has the best performance.

How does Stats Accelerator handle conversion rates that change over time and Simpson’s Paradox?

Time variation is defined as a dependence of the underlying distribution of the metric value on time. More simply, time variation occurs when a metric’s conversion rate changes over time. Stats Engine assumes this distribution is identically distributed.

Time variation is caused by a change in the underlying conditions that affect visitor behavior. Examples include more purchasing visitors on weekends; an aggressive new discount that yields more customer purchases; or a marketing campaign in a new market that brings in a large number of visitors with different interaction behavior than existing visitors.

Optimizely assumes identically distributed data because this assumption enables continuous monitoring and faster learning (see the Stats Engine article for details). However, Stats Engine has a built-in mechanism to detect violations of this assumption. When a violation is detected, Stats Engine updates the statistical significance calculations. This is called a “stats reset.”

Time variation affects experiments using Stats Accelerator because the algorithms adjust the percentage of traffic exposed to each variation during the experiment. This can introduce bias in the estimated improvement, known as Simpson's Paradox. The result is that stats resets may be much more likely to occur. (See the weighted improvement section above for more information.)
The solution is to change the way the improvement number is calculated. Specifically, Optimizely compares the conversion rates of the baseline and variation(s) within each interval between traffic allocation changes. Then, Optimizely computes statistics using weighted averages across these time intervals. For example, the difference of observed conversion rates is scaled by the number of visitors in each interval to generate an estimate of the true difference in conversion rates. This estimate is represented as weighted improvement.

**Additional resources**

- [FDR Control with Adaptive Sequential Experimental Design](#) is a technical white paper on the mathematical foundation of Stats Accelerator.