A/B Testing Mastery: From Beginner to Pro in a Blog Post

By Alex Birkett · cxl.com · 22 min

View Original

A/B testing—for all the content out there about it, people still mess it up. From testing the wrong things to running A/B tests incorrectly, there are lots of ways to get it wrong.

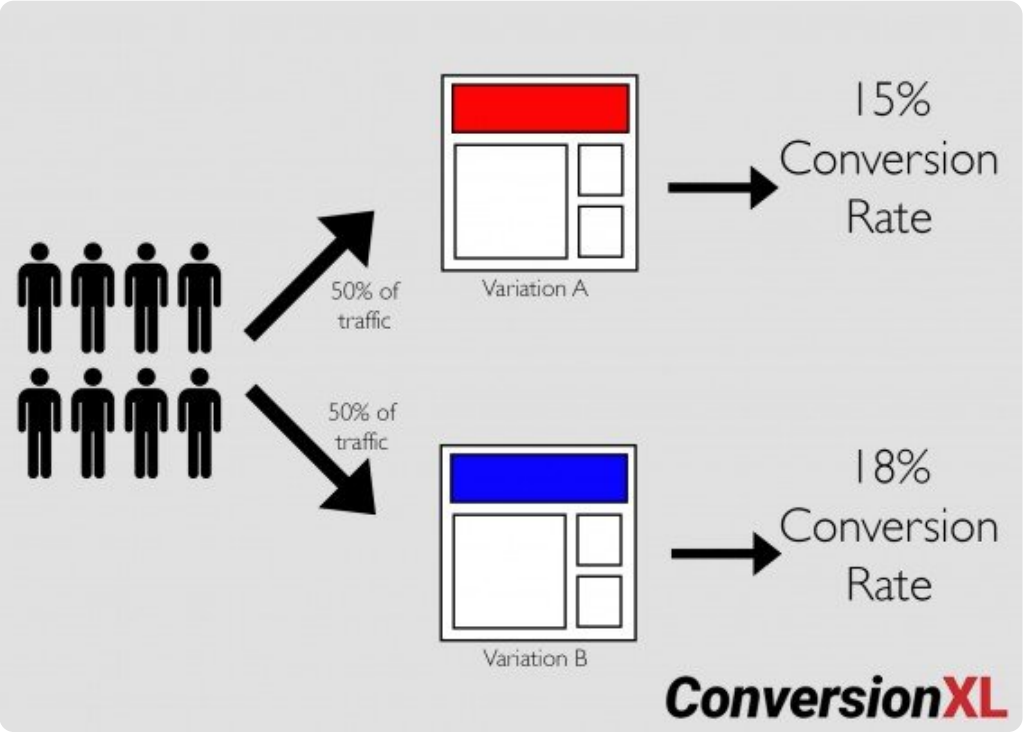
Here's what we'll cover in this tutorial:

What is A/B testing?

A/B testing splits traffic 50/50 between a control and a variation. A/B split testing is a new term for an old technique—controlled experimentation.

test," complete with a hypothesis, a control, a variation, and a statistically calculated result.

That's it. For example, if you ran a simple A/B test, it would be a 50/50 traffic split between the original page and a variation:



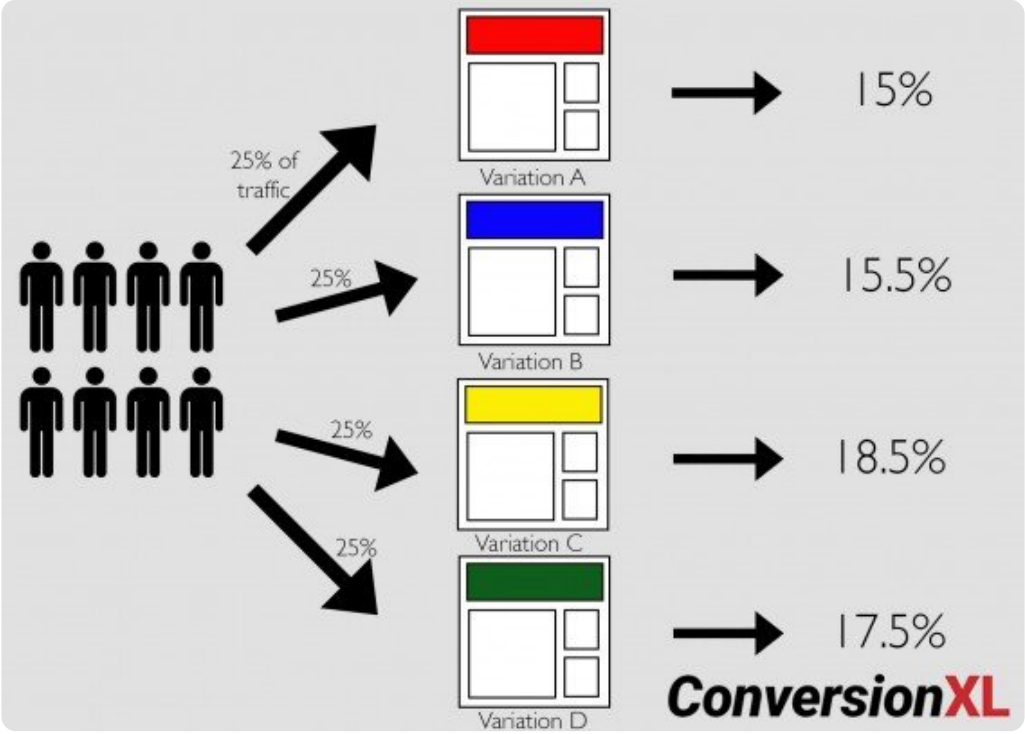
A/B testing splits traffic 50/50 between a control and a variation.

For conversion optimization, the main difference is the variability of Internet traffic. In a lab, it's easier to control for external variables. Online, you can mitigate them, but it's difficult to create a purely controlled test.

In addition, testing new drugs requires an almost certain degree of accuracy. Lives are on the line. In technical terms, your period of "exploration" can be much longer, as you want to be damn sure that you don't commit a Type I error (false positive).

Online, the process for A/B split-testing considers business goals. It weighs risk vs. reward, exploration vs. exploitation, science vs. business. Therefore, we view results through a different lens and make decisions differently than those running tests in a lab.

traffic, you can test as many variations as you like. Here's an example of an A/B/C/D test, and how much traffic each variation is allocated:



An A/B/n test splits traffic equally among a control and multiple page variations.

A/B/n tests are great for implementing more variations of the same hypothesis, but they require more traffic because they split it among more pages.

A/B tests, while the most popular, are just one type of online experiment. You can also run multivariate and bandit tests.

A/B Testing, multivariate testing, and bandit algorithms: What's the Difference?

A/B/n tests are controlled experiments that run one or more variations against the original page. Results compare conversion rates among the variations based on a single change.

Multivariate tests test multiple versions of a page to isolate which attributes cause the largest impact. In other words, **multivariate**

elements. For example:

Each element has a specific impact and use case to help you get the most out of your site. Here's how:

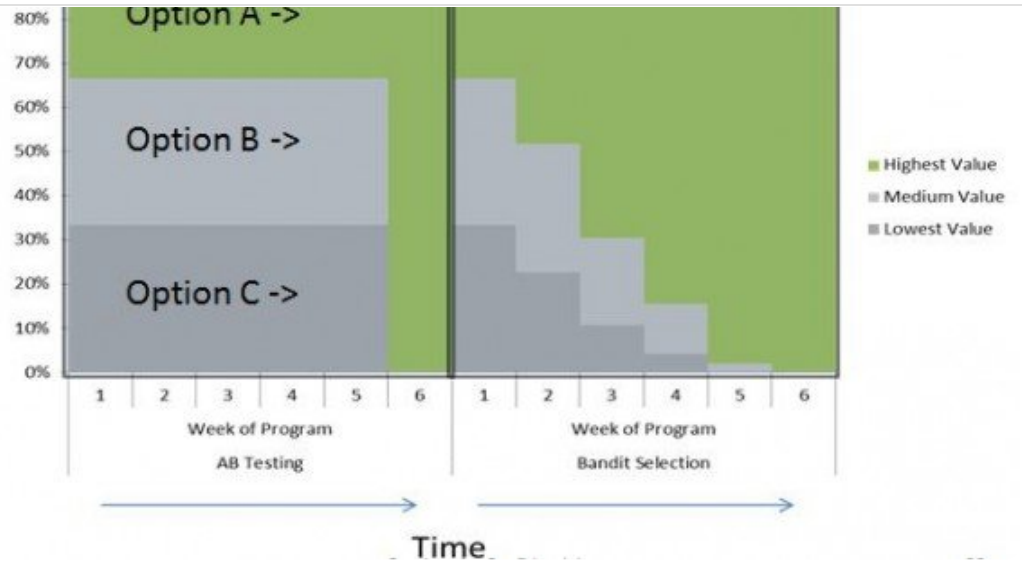
- Use A/B testing to determine the best layouts.
- Use multivariate tests to polish layouts and ensure all elements interact well together.

You need to *a ton* of traffic to the page you're testing before even considering multivariate testing. But if you have enough traffic, you should use both types of tests in your optimization program.

Most agencies prioritize A/B testing because you're usually testing more significant changes (with bigger potential impacts), and because they're simpler to run. As Peep once said, "Most top agencies that I've talked to about this run ~10 A/B tests for every 1 MVT."

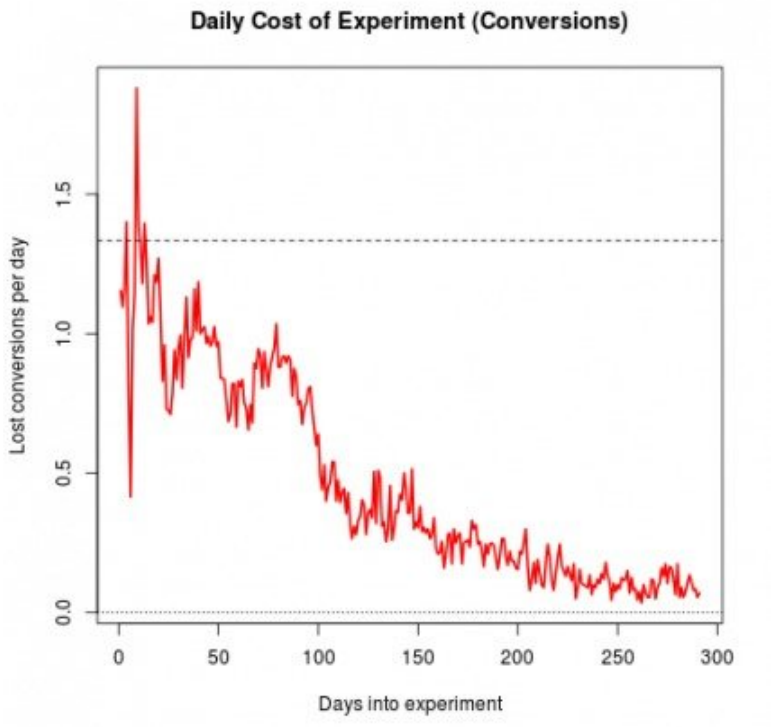
Bandit algorithms are A/B/n tests that update in real time based on the performance of each variation.

In essence, a bandit algorithm starts by sending traffic to two (or more) pages: the original and the variation(s). Then, to "pull the winning slot machine arm more often," the algorithm updates based on which variation is "winning." Eventually, the algorithm fully exploits the best option:



(Image Source)

One benefit of bandit testing is that bandits mitigate "regret," which is the lost conversion opportunity you experience while testing a potentially worse variation. This chart from Google explains that very well:



(Image Source)

- Headlines and short-term campaigns;
- Automation for scale;
- Targeting;
- Blending optimization with attribution.

No matter what type of test you run, it's important to have a process that improves your chances of success. This means running more tests, winning more tests, and making bigger lifts.

A/B Test Planning: How to Build a Process that Works

By Jaan Matti-Saul

A strong A/B testing plan will allow you to increase your revenue and learn valuable insights about your customers.

How to improve A/B test results

Ignore blog posts that tell you "99 Things You Can A/B Test Right Now." They're a waste of time and traffic. A *process* will make you more money.

in what Craig Sullivan calls the "Trough of Disillusionment." (Unless their results are littered with false positives, which we'll get into later.)

To simplify a winning process, the structure goes something like this:

1. Research;
2. Prioritization;
3. Experimentation;
4. Analyze, learn, repeat.

Research: Getting data-driven insights

To begin optimization, you need to know what your users are doing and why.

Before you think about optimization and testing, however, solidify your high-level strategy and move down from there. So, think in this order:

1. Define your business objectives.
2. Define your website goals.
3. Define your Key Performance Indicators.
4. Define your target metrics.

Once you know where you want to go, you can collect the data necessary to get there. To do this, we recommend the ResearchXL Framework.

Here's the executive summary of the process we use at CXL:

Heuristic analysis is about as close as we get to "best practices." Even after years of experience, you still can't tell exactly what will

Humility is crucial. It also helps to have a framework. When doing heuristic analysis, we assess each page based on the following:

- Relevancy;
- Clarity;
- Value;
- Friction;
- Distraction.

Technical analysis is an often-overlooked area. Bugs—if they're around—are a conversion killer. You may think your site works perfectly in terms of user experience and functionality. But does it work equally well with every browser and device? Probably not.

This is a low-hanging—and highly profitable—fruit. So, start by:

- Conducting cross-browser and cross-device testing.
- Doing a speed analysis.

Web analytics analysis is next. First thing's first: Make sure everything is working. (You'd be surprised by how many analytics setups are broken.)

Google Analytics (and other analytics setups) are a course in themselves, so I'll leave you with some helpful links:

Next is **mouse-tracking analysis**, which includes heat maps, scroll maps, click maps, form analytics, and user session replays. Don't get carried away with pretty visualizations of click maps. Make sure you're informing your larger goals with this step.

Qualitative research tells you the *why* that quantitative analysis misses. Many people think that qualitative analysis is "softer" or easier than quantitative, but it should be just as rigorous and can provide insights as important as those from analytics.

Finally there's **user testing**. The premise is simple: Observe how actual people use and interact with your website while they narrate their thought process aloud. Pay attention to what they say *and* what they experience.

After thorough conversion research, you'll have lots of data. The next step is to prioritize that data for testing.

How to prioritize A/B test hypotheses

There are many frameworks to prioritize your A/B tests, and you could even innovate with your own formula. Here's a way to prioritize work shared by Craig Sullivan.

Once you go through all six steps, you *will* find issues—some severe, some minor. Allocate every finding into one of five buckets:

1. **Test.** This bucket is where you place stuff for testing.
2. **Instrument.** This can involve fixing, adding, or improving tag/event handling in analytics.
3. **Hypothesize.** This is where you've found a page, widget, or process that's not working well but doesn't reveal a clear solution.
4. **Just Do It.** Here's the bucket for no-brainers. Just do it.
5. **Investigate.** If an item is in this bucket, you need to ask questions or dig deeper.

Rank each issue from 1 to 5 stars (1 = minor, 5 = critical). There are two criteria that are more important than others when giving a score:

1. **Ease of implementation** (time/complexity/risk). Sometimes, data tells you to build a feature that will take months to develop. Don't start there.

Create a spreadsheet with all of your data. You'll have a prioritized testing roadmap.

We created our own prioritization model to weed out subjectivity (as possible). It's predicated on the need to bring *data* to the table. It's called PXL and looks like this:

Grab your own copy of this spreadsheet template here. Just click File > Make a Copy to make it your own.

Instead of guessing what the impact might be, this framework asks you a set of questions about it:

- **Is the change above the fold**? More people notice above-the-fold changes. Thus, those changes are more likely to have an impact.
- **Is the change noticeable in under 5 seconds?** Show a group of people the control and then the variation(s). Can they tell a difference after 5 seconds? If not, it's likely to have less of an impact.
- **Does it add or remove anything?** Bigger changes like removing distractions or adding key information tend to have more of an impact.
- **Does the test run on high-traffic pages?** An improvement to a high-traffic page generates bigger returns.

Many potential test variables require data to prioritize your hypotheses. Weekly discussions that ask these four questions will help you prioritize testing based on data, not opinions:

1. Is it addressing an issue discovered via user testing?
2. Is it addressing an issue discovered via qualitative feedback (surveys, polls, interviews)?
3. Is the hypothesis supported by mouse tracking, heat maps, or eye tracking?

We also put bounds on Ease of implementation by bracketing answers according to the estimated time. Ideally, a test developer is part of prioritization discussions.

Grading PXL

We assume a binary scale: You have to choose one or the other. So, for most variables (unless otherwise noted), you choose either a 0 or a 1.

But we also want to weight variables based on importance—how noticeable the change is, if something is added/removed, ease of implementation. For these variables, we specifically say how things change. For instance, on the Noticeability of the Change variable, you either mark it a 2 or a 0.

Customizability

We built this model with the belief that you can and should customize variables based on what matters to your business.

For example, maybe you're working with a branding or user experience team, and hypotheses must conform to brand guidelines. Add it as a variable.

Maybe you're at a startup whose acquisition engine is fueled by SEO. Maybe your funding depends on that stream of customers. Add a category like, "doesn't interfere with SEO," which might alter some headline or copy tests.

All organizations operate under different assumptions. Customizing the template can account for them and optimize your optimization program.

Whichever framework you use, make it systematic and understandable to anyone on the team, as well as stakeholders.

First rule: **Don't stop a test just because it reaches statistical significance.** This is probably the most common error committed by beginner optimizers with good intentions.

If you call tests when you hit significance, you'll find that most lifts don't translate to increased revenue (that's the goal, after all). The "lifts" were, in fact, imaginary.

Consider this: When 1,000 A/A tests (two identical pages) were run:

- 771 experiments out of 1,000 reached 90% significance at some point.
- 531 experiments out of 1,000 reached 95% significance at some point.

Stopping tests at significance risks false positives and excludes external validity threats, like seasonality.

Predetermine a sample size and run the test for full weeks, usually at least two business cycles.

How do you predetermine sample size? There are lots of great tools. Here's how you'd calculate your sample size with Evan Miller's tool:

In this example, we told the tool that we have a 3% conversion rate and want to detect at least 10% uplift. The tool tells us that we need 51,486 visitors *per variation* before we can look at statistical significance levels.

In addition to significance level, there's something called statistical power. Statistical power attempts to avoid Type II errors (false negatives). In other words, it makes it more likely that you'll detect an effect *if there actually was one.*

For practical purposes, know that 80% power is the standard for A/B testing tools. To reach such a level, you need either a large sample size, a large effect size, or a longer duration test.

A lot of blog posts tout magic numbers like “100 conversions” or “1,000 visitors” as stopping points. Math is not magic. Math is math, and what we’re dealing with is slightly more complex than simplistic heuristics like those figures. Andrew Anderson from Malwarebytes put it well:

Andrew Anderson:

“It is never about how many conversions. It is about having enough data to validate based on representative samples and representative behavior.

One hundred conversions is possible in only the most remote cases and with an incredibly high delta in behavior, but only if other requirements like behavior over time, consistency, and normal distribution take place. Even then, it is has a really high chance of a Type I error, false positive.”

We want a representative sample. How can we get that? Test for two business cycles to mitigate external factors:

- **Day of the week.** Your daily traffic can vary a lot.
- **Traffic sources.** Unless you want to personalize the experience for a dedicated source.
- **Blog post and newsletter publishing schedule.**
- **Return visitors.** People may visit your site, think about a purchase, then come back 10 days later to buy it.
- **External events.** A mid-month payday may affect purchasing, for example.

Be careful with small sample sizes. The Internet is full of case studies steeped in shitty math. Most studies (if they ever released full numbers) would reveal that publishers judged test variations on 100 visitors or a lift from 12 to 22 conversions.

result in calling a result early due to "spotting a trend" (impossible). What you'll find is that many test results regress to the mean.

Regression to the mean

Often, you'll see results vary wildly in the first few days of the test. Sure enough, they tend to converge as the test continues for the next few weeks. Here's an example from an ecommerce site:

- First couple of days: Blue (variation #3) is winning big—like $16 per visitor vs. $12.50 for Control. Lots of people would (mistakenly) end the test here.
- After 7 days: Blue still winning, and the relative difference is big.
- After 14 days: Orange (#4) is winning!
- After 21 days: Orange still winning!
- End: No difference.

If you'd called the test at less than four weeks, you would have made an erroneous conclusion.

There's a related issue: the novelty effect. The novelty of your changes (e.g., bigger blue button) brings more attention to the variation. With time, the lift disappears because the change is no longer novel.

It's one of many complexities related to A/B testing. We have a bunch of blog posts devoted to such topics:

Can you run multiple A/B tests simultaneously?

You want to speed up your testing program and run more tests—high-tempo testing. But can you run more than one A/B test at the same time?Will it increase your growth potential or pollute your data?

multiple simultaneous tests; extreme interactions are unlikely.

Unless you're testing really important stuff (e.g., something that impacts your business model, future of the company), the benefits of testing *volume* will likely outweigh the noise in your data and occasional false positives.

If there is a high risk of interaction between multiple tests, reduce the number of simultaneous tests and/or let the tests run longer for improved accuracy.

If you want to learn more, read these posts:

How to set up A/B tests

Once you've got a prioritized list of test ideas, it's time to form a hypothesis and run an experiment. **A hypothesis defines *why* you believe a problem occurs.** Furthermore, a good hypothesis:

- **Is testable.** It is measurable, so it can be tested.
- **Solves a conversion problem.** Split-testing solves conversion problems.
- **Provides market insights.** With a well-articulated hypothesis, your split-testing results give you information about your customers, whether the test "wins" *or* "loses."

Craig Sullivan has a hypothesis kit to simplify the process:

1. Because we saw (data/feedback),
2. We expect that (change) will cause (impact).
3. We'll measure this using (data metric).

And the advanced one:

1. Because we saw (qualitative and quantitative data),

3. We expect to see (data metric[s] change) over a period of (X business cycles).

Technical stuff

Here's the fun part: You can finally think about picking a tool.

While this is the first thing many people think about, it's not the most important. Strategy and statistical knowledge come first.

That said, there are a few differences to bear in mind. One major categorization in tools is whether they are server-side or client-side testing tools.

Server-side tools render code on the server level. They send a randomized version of the page to the viewer with no modification on the visitor's browser. Client-side tools send the same page, but JavaScript on the client's browser manipulates the appearance on the original and the variation.

Client-side testing tools include Optimizely, VWO, and Adobe Target. Conductrics has capabilities for both, and SiteSpect does a proxy server-side method.

What does all this mean for you? If you'd like to save time up front, or if your team is small or lacks development resources, client-side tools can get you up and running faster. Server-side requires development resources but can often be more robust.

While setting up tests is slightly different depending on which tool you use, it's often as simple as signing up for your favorite tool and following their instructions, like putting a JavaScript snippet on your website.

Beyond that, you need to set up Goals (to know when a conversion has been made). Your testing tool will track when each variation converts visitors into customers.

craft variations. Some tools allow use of a visual editor, but that limits your flexibility and control.

How to analyze A/B test results

Alright. You've done your research, set up your test correctly, and the test is finally cooked. Now, on to analysis. It's not as simple as a glimpse at the graph from your testing tool.

One thing you should always do: Analyze your test results in Google Analytics. It doesn't just enhance your analysis capabilities; it also allows you to be more confident in your data and decision making.

Your testing tool could be recording data incorrectly. **If you have no other source for your test data, you can never be sure whether to trust it. Create multiple sources of data.**

What happens if there's no difference between variations? Don't move on too quickly. First, realize two things:

1. Your hypothesis might have been right, but implementation was wrong.

Let's say your qualitative research says that concern about security is an issue. How many ways can you beef up the perception of security? Unlimited.

The name of the game is iterative testing, so if you were on to something, try a few iterations.

2. Even if there was no difference *overall,* the variation might beat the control in a segment or two.

If you got a lift for returning visitors and mobile visitors—but a drop for new visitors and desktop users—those segments might cancel each other out, making it seem like there's "no difference." Analyze your test across key segments to investigate that possibility.

The key to learning in A/B testing is segmenting. Even though B might lose to A in the overall results, B might beat A in certain segments (organic, Facebook, mobile, etc).

There are a ton of segments you can analyze. Optimizely lists the following possibilities:

- Browser type;
- Source type;
- Mobile vs. desktop, or by device;
- Logged-in vs. logged-out visitors;
- PPC/SEM campaign;
- Geographical regions (city, state/province, country);
- New vs. returning visitors;
- New vs. repeat purchasers;
- Power users vs. casual visitors;
- Men vs. women;
- Age range;
- New vs. already-submitted leads;
- Plan types or loyalty program levels;
- Current, prospective, and former subscribers;
- Roles (if your site has, for instance, both a buyer and seller role).

At the very least—assuming you have an adequate sample size—look at these segments:

- Desktop vs. tablet/mobile;
- New vs. returning;
- Traffic that lands on the page vs. traffic from internal links.

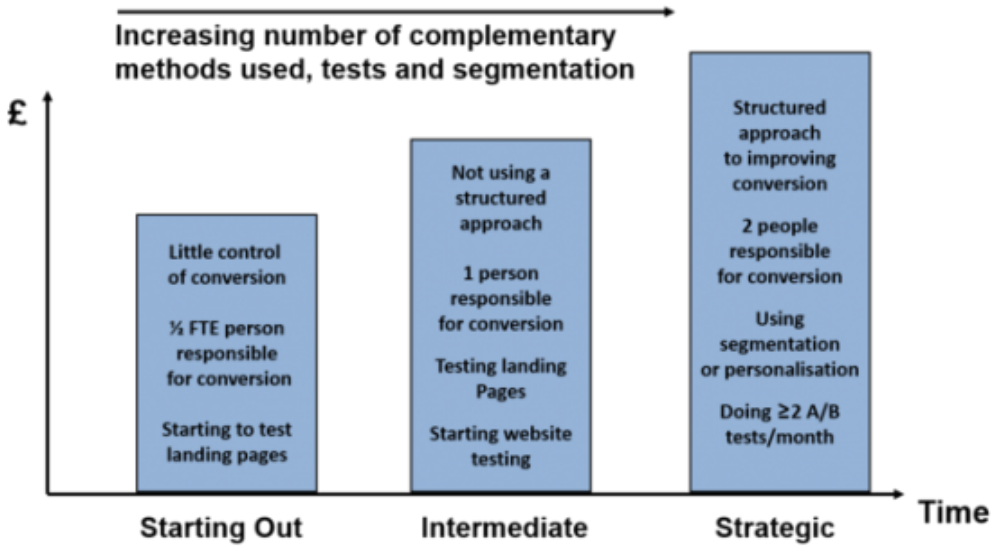
conversions *per variation* within in a given segment.

If your treatment performed well for a specific segment, it's time to consider a personalized approach for those users.

How to archive past A/B tests

A/B testing isn't just about lifts, wins, losses, and testing random shit. As Matt Gershoff said, optimization is about "gathering information to inform decisions," and the learnings from statistically valid A/B tests contribute to the greater goals of growth and optimization.

Smart organizations archive their test results and plan their approach to testing systematically. A structured approach to optimization yields greater growth and is less-often limited by local maxima.



(Image Source)

So here's the tough part: There's no single best way to structure your knowledge management. Some companies use sophisticated, internally built tools; some use third-party tools; and some use Excel and Trello.

It's important to communicate across departments and to executives. Often, A/B test results aren't intuitive to a layperson. Visualization helps.

Annemarie Klaassen and Ton Wesseling wrote an awesome post on visualizing A/B test results. Here's what they came up with:

A/B testing statistics

Statistical knowledge is handy when analyzing A/B test results. We went over some of it in the section above, but there's more to cover.

Why do you need to know statistics? Matt Gershoff likes to quote his college math professor: "How can you make cheese if you don't know where milk comes from?!"

There are three terms you should know before we dive into the nitty gritty of A/B testing statistics:

1. **Mean.** We're not measuring *all* conversion rates, just a sample. The average is representative of the whole.
2. **Variance.** What is the natural variability of a population? That affects our results and how we use them.
3. **Sampling.** We can't measure the true conversion rate, so we select a sample that is (hopefully) representative.

What is a p-value?

Many use the term "statistical significance" inaccurately. Statistical significance by itself is not a stopping rule, so what is it and why is it important?

To start with, let's go over p-values, which are also very misunderstood. As FiveThirtyEight recently pointed out, even scientists can't easily explain p-values.

probability that B is better than A.

Similarly, it doesn't tell us the probability that we will make a mistake in selecting B over A. These are common misconceptions.

The p-value is the probability of seeing the current result or a more extreme one *given that the null hypothesis is true*. Or, "How surprising is this result?"

To sum it up, statistical significance (or a statistically significant result) is attained when a p-value is less than the significance level (which is usually set at 0.05).

Significance in regard to statistical hypothesis testing is also where the whole "one-tail vs. two-tail" issue comes up.

One-tail vs. two-tail A/B tests

One-tailed tests allow for an effect in one direction. Two-tailed tests look for an effect in two directions—positive or negative.

No need to get very worked up about this. Gershoff from Conductrics summed it up well:

Matt Gershoff:
"If your testing software only does one type or the other, don't sweat it. It is super simple to convert one type to the other (but you need to do this BEFORE you run the test) since all of the math is exactly the same in both tests. All that is different is the significance threshold level. If your software uses a one-tail test, just divide the p-value associated with the confidence level you are looking to run the test by two. So, if you want your two-tail test to be at the 95% confidence level, then you would actually input a confidence level of 97.5%, or if at a 99%, then you need to input 99.5%. You can then just read the test as if it was two-tailed."

Confidence intervals and margin of error

of utmost importance to understanding your test results.

VARIATION	VISITORS	CONVERSIONS	CONVERSION RATE	IMPROVEMENT
Original	7965	98	1.23% (±0.24)	---
Variation #1	8072	263	3.26% (±0.39)	+164.8%

An example of confidence intervals. (Image source)

In A/B testing, we use confidence intervals to mitigate the risk of sampling errors. In that sense, we're managing the risk associated with implementing a new variation.

So if your tool says something like, "We are 95% confident that the conversion rate is X% +/- Y%," then you need to account for the +/- Y% as the margin of error.

How confident you are in your results depends largely on how large the margin of error is. **If the two conversion ranges overlap, you need to keep testing to get a valid result.**

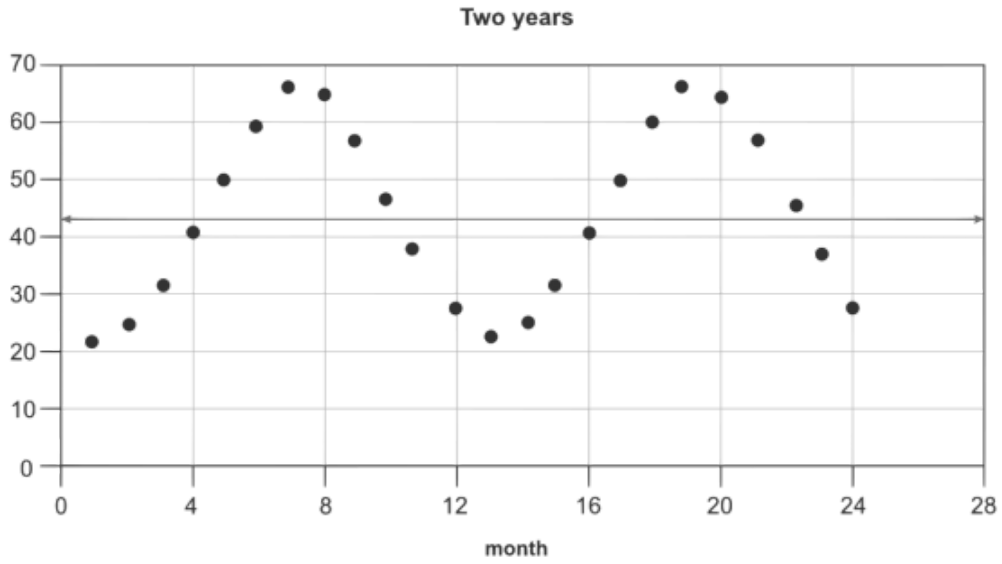
Matt Gershoff gave a great illustration of how margin of error works:

Matt Gershoff:

"Say your buddy is coming to visit you from Round Rock and is taking TX-1 at 5 p.m. She wants to know how long it should take her. You say I have a 95% confidence that it will take you about 60 minutes plus or minus 20 minutes. So your margin of error is 20 minutes, or 33%.

If she is coming at 11 a.m. you might say, "It will take you 40 min, plus or minus 10 min," so the margin of error is 10 minutes, or 25%. So while both are at the 95% confidence level, the margin of error is different."

There's a challenge with running A/B tests: Data isn't stationary.



Sinusoidal data

A stationary time series is one whose statistical properties (mean, variance, autocorrelation, etc.) are constant over time. For many reasons, website data is non-stationary, which means we can't make the same assumptions as with stationary data. Here are a few reasons that data might fluctuate:

- Season;
- Day of the week;
- Holidays;
- Positive or negative press mentions;
- Other marketing campaigns;
- PPC/SEM;
- SEO;
- Word-of-mouth.

Others include sample pollution, the flicker effect, revenue tracking errors, selection bias, and more. (Read here.) These are things to keep in mind when planning and analyzing your A/B tests.

Bayesian or Frequentist A/B testing is another hot topic. Many popular tools have rebuilt their stats engines to feature a Bayesian methodology.

Here's the difference (very much simplified): **In the Bayesian view, a probability is assigned to a hypothesis. In the Frequentist view, a hypothesis is tested without being assigned a probability.**

Rob Balon, who carries a PhD in statistics and market research, says the debate is mostly esoteric tail wagging from the ivory tower. "In truth," he says, "most analysts out of the ivory tower don't care that much, if at all, about Bayesian vs. Frequentist."

Don't get me wrong, there are practical business implications to each methodology. But if you're new to A/B testing, there are much more important things to worry about.

A/B testing tools and resources

Littered throughout this guide are tons of links to external resources: articles, tools, books, etc. To make it convenient for you, here are some of the best (divided by categories).

A/B testing tools

There are a lot of tools for online experimentation. Here's a list of 53 conversion optimization tools, all reviewed by experts. Some of the most popular A/B testing tools include:

A/B testing calculators

A/B testing statistics resources

A/B testing/CRO strategy resources

A/B testing is an invaluable resource to anyone making decisions in an online environment. With a little bit of knowledge and a lot of diligence, you can mitigate many of the risks that most beginning optimizers face.

If you really dig into the information here, you'll be ahead of 90% of people running tests. If you believe in the power of A/B testing for continued revenue growth, that's a fantastic place to be.

Knowledge is a limiting factor that only experience and iterative learning can transcend. So get testing!