MODE

WHITEPAPER

How to Make Sure A/B Tests Aren't Leading You Astray

for trustworthy business decisions

Julia Glick

Table of Contents

Introduction

We use A/B tests to help us make decisions that measurably improve our product and create a better experience for the majority of our users.

When we run an A/B test, we choose a random subset of users to receive an intervention, such as a new feature or a modified user interface, and look to see whether it improves a metric such as user engagement. We use that test result to figure out what will happen if we take one course of action or another, such as rolling out the new feature globally versus cancelling the launch of that feature. Then we make a business decision, and then act.

In formal terms, A/B tests help us learn about the *treatment effect*—the impact on our key metrics from the treatment that we gave to the test group but not the control group. We use *estimators* to estimate the treatment effect. And many traditional estimators of treatment effects are unbiased, including most of the ones we learned about in statistics classes and that most businesses use day to day.

An *unbiased estimator* is one where the expected value of the estimator is equal to the true value that it's trying to estimate. This is usually considered a good and very important property. Any single estimate will be above or below the true value, but an unbiased estimate is correct on average. While unbiasedness is a great property to have, it matters for some kinds of tasks a lot more than for others.

In contrast to unbiased estimators, *regularized estimators* add information to the model beyond what is in the observed data. We usually use regularization to make our estimates smaller, meaning closer to 0 (whether positive or negative).

Regularized estimates are therefore biased, but also have less uncertainty because we're including extra information; we're making a *bias-variance tradeoff*. Regularization is used extensively in machine learning contexts to improve the generalizability of our models.

Regularization can be used for pretty much the same purpose in A/B testing. **To drive more accurate business decisions, we should give up pure unbiasedness and start using some form of regularization in our A/B testing as a matter of course.**

This is because we're not always learning the right things from these tests if we continue to use unbiased estimators. For any unbiased estimator, the mathematical proof that it is unbiased relies on certain assumptions about how we *use* the estimators. And those assumptions often just don't hold in a business context. Holding too hard onto supposed "unbiasedness" can lead us to make bad decisions that hurt our customers' experience and our business outcomes.

In this paper, we'll explore why accepting biased estimates will help us make better business decisions, and suggest some methods you can use to accomplish that.

The purpose of A/B testing

Before we dive in further, let's review the purpose of A/B testing. The goal of running an A/B test is to learn about the relative impact of one course of action, the *treatment*, compared to another course of action, the control or *baseline*. We randomize users (usually) into treatment or control conditions and measure one or more outcomes of interest. *The treatment effect* is the difference between what actually happened to the treatment group and what would have happened if we'd given those users the baseline experience instead.

The test analysis we run tells us about what *happened*, in that specific test, to the specific users who were randomized, at that specific time. But we want to generalize the results beyond those users. We want to know about what *will happen* in the future based on business decisions that we have not yet made and actions that we have not yet taken.

The primary job of A/B testing in a business context is almost always to help us make *predictions*. Predictions, specifically, about possible futures, where we as a business choose to do one thing or another thing. (That includes choosing which A/B tests to run next.)

If we run A/B tests but don't use them to make predictions, then we're stuck in the past, and we don't have a way of making better decisions. And then we would be missing out. We would be spending resources on A/B testing, but not gaining all of the benefits that we could if we used those tests to make predictions about how to act, which is to say, to *learn*.

The problem with typical A/B tests

When we run a traditional linear regression analysis or a t-test, as we do for most A/B tests, we assume that the estimate of the difference between the treatment and control groups will be equal to the true difference in *expectation*.

Of course, when we actually run and analyze the test, the estimate will be off by some amount (on average, by one standard error); half the time it will be too low, and half the time it will be too high. But this all averages out in the long term, right?

Well, not always.

For most companies, the process for analyzing A/B tests is based on "null hypothesis significance testing" (NHST), but this process often comes up short.

We're going to have to get precise with our language in order to find the problems. So there's some statistical jargon here, including formal definitions of things many of us are used to thinking about informally. If you have experience with A/B tests in practice, then you probably understand the concepts just fine, whether you're fluent with the technical terms.

To analyze a test, we construct an estimator using the observed data. Typically we'll use a t-test or linear regression, which are both versions of "ordinary least squares" (OLS). We also estimate the uncertainty around that estimator. Together, these give us a test statistic, such as a t or F statistic, which has a known distribution *if the test effect was 0*, that is, under the "null hypothesis" of no difference.

If the observed test statistic falls into an extreme, low-probability region under the null distribution, then we reject the null hypothesis and declare that there was a difference between the test and control groups. If the statistic is in a high-probability region of the null distribution, then we fail to reject the null hypothesis.

The *power* of a test to reject the null hypothesis is the probability (in advance of running the test) that the test statistic will reject, given the true difference between the groups. For a test with no difference between the groups, that probability will be equal to the probability of a "false positive" result, called α.

Test power depends on both the true and unknown difference between groups and on the uncertainty around the estimator. Since uncertainty decreases as the number of users in the test increases, we need to be sure to run tests that are large enough to detect results of business interest. But running excessively large tests incurs business costs, including opportunity costs (failing to run other tests that we could be running alongside this one) and risks (assigning too many users to a test condition that turns out to be harmful). Test power is very important, and we're going to come back to it several times in this paper; unfortunately, it's also a tricky and challenging topic in practice.

No matter what, the estimate of the effect size that we get from OLS is the Best Linear Unbiased Estimator for the difference between the test and control groups. That's true whether or not the test statistic rejects the null hypothesis.

In most business contexts, the process of analyzing a test is not used exactly as the assumptions of NHST dictate because we have many business questions that need answering. Two of those questions are usually, "Is this effect 'real'?" and "If it's real, how big is this effect?"

Unfortunately, if we answer those questions in a naive way, we end up with what Andrew Gelman calls the "statistical significance filter." It goes like this:

First, we ask the "is it real?" question, and we use NHST to answer the question: the effect is "real" if $p < .05$ (or $p < \alpha$ for whatever our α is). If the effect is not "real," then we decide it's not worth bothering about; the effect might as well be 0 for our decisions.

If the effect is "real," i.e., statistically significant, then we use the estimate of the effect size from the test to guide our decisions about whether to go forward with any new change, such as rolling out the test condition to all users.

The problem is that now we're not looking at an unbiased estimator anymore. We're not using the effect size estimate itself, we're using the effect size estimate *conditional on the test being statistically significant*, and that is a totally different situation.

What happens as a result? Our effect size estimates become inflated away from 0 and we get results that are, on average, too extreme. And tests that are less powerful, but that come out significant anyway, give even more inflated results. **This can make us overly optimistic about the future impact of a change, or, worse, keep us chasing product changes that do almost nothing at all instead of switching to more fruitful avenues of exploration**.

If we understand and account for the statistical significance filter, however, we can avoid being led astray. Let's walk through the problem in detail.

How statistical significance can inflate A/B test results

Testing the null hypothesis of t-tests

Let's think about a t-test where we're testing the null hypothesis of $\theta = 0$ as usual. (Once again, this is the stats textbook material behind what we all do regularly. Don't panic; you know this.)

The t-test gives us an estimate $\hat{\theta}$ of θ, and a good estimate of the standard error of θ, se; the ratio $t = \frac{\theta}{se}$ is our test statistic. You can assume we have high degrees of freedom, so t is approximately normal; if $t > 2$ (approximately) then we'll reject the null hypothesis with $\alpha = .05$.

Our estimate $\hat{\theta}$ is a noisy observation, and it comes from a distribution that is close to $N(\theta, se^2)$, a Gaussian centered at the true value of the parameter θ. The power of the test is the probability that the t-test will reject the null hypothesis, which depends on both θ and se. For a given se, power depends on θ, but of course we don't know θ; that's why we're running the test.

Think of a test with true power = 0.5, that is, one that's a coin flip whether our result will be statistically significant or not; θ is just about equal to $2 \times se$. (Most people would consider this test to be underpowered. A more usual goal would be 80% power or higher.)

This test is illustrated in the following figure: the green distribution is the true distribution of θ, and the blue distribution is our assumed distribution of $\hat{\theta}$ under the null hypothesis of no difference. The vertical lines at $\pm r$ (in this example, $r = 1.96$) show the rejection regions; values of $\hat{\theta}$ more extreme than those lines will lead us to reject the null.

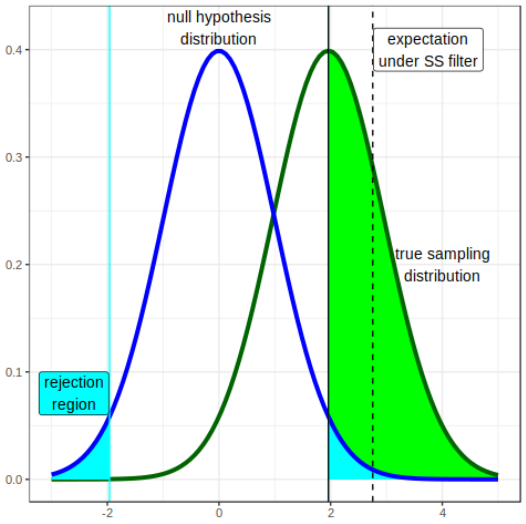


Figure 01: R code for this plot

So what happens? If $|\hat{\theta}|$ then we fail to reject the null; if $|\hat{\theta}| < r$ we reject. But because power is exactly equal to 0.5, therefore $r = \theta$, as shown in the diagram.

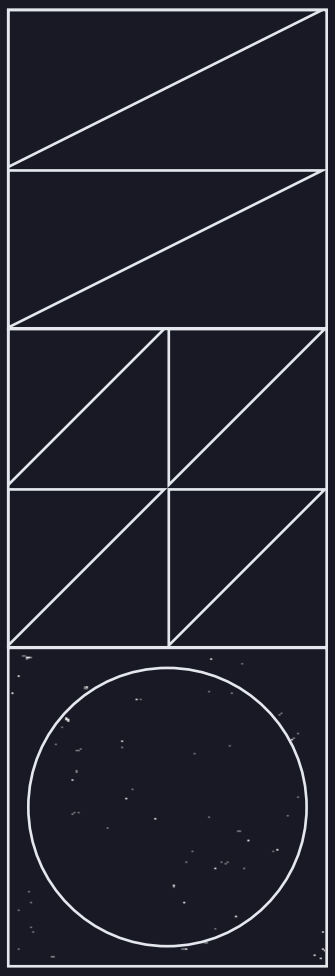
We reject exactly when our estimate is greater than the true value of θ; we fail to reject when the estimate is less than θ. **If we're applying the statistical significance filter, that means that we'll only ever pay attention to estimates that are *strictly too big*.**

When power ≤ 0.5, it's obvious that the situation is terrible; but even with power > 0.5, the same result still holds in expectation. By conditioning on statistically significant results, we drive our estimates away from zero.

When power > 0.5, we might randomly observe a result that's below rather than above the true value, but with power ≤ 0.5, we will always overstate the results. Businesses tend to run tests with fewer users than might be ideal, to get results more quickly. Remember that we don't know the true power of any single test, and many tests will have little or no true effect, so they are guaranteed to be underpowered.

Put another way, the statistical significance filter encourages us to pay attention to outliers—results that are too big just by chance.

The primary danger is not that we get *wrong* results, i.e., results that are in the wrong direction, unless power is very very low. Instead, the danger is that we get results that are in the correct direction but far too big. That can lead us to shift the direction of future testing, or even the direction of the business, trying to chase an illusory lift.



Example: Retention numbers with no regularization

To make things concrete, let's look at an example. This test is imagined and the data are fake, but the process and interpretation is inspired by real life experience.

Suppose we're testing a user-facing improvement to our website that we think will make our site a little easier and more pleasant to use. Let's imagine our business model is subscription-based and billed monthly. Our hypothesis is that this change will improve user engagement, which we measure in two ways: the ratio of days on which a user returns to the site divided by the total number of days we run the A/B test, and the average number of minutes of use per day that the user comes to the site at all.

We also expect this to impact user retention—the proportion of paid users who continue to pay next month. We run the test for at least a full month to ensure that all users have a billing period during the test. In total, we have three outcomes we want to include in the test.

When the results come back, they look like this:

Measure	Control	Test Group	Lift	Std Err	t	p
Days active ratio	0.400	0.405	.01	.006	.0.83	.4
Minutes/visits	39	37	-2	8	-0.25	.8
Retention	0.750	0.765	.015	.005	3	3

We see a significant lift to retention. This is a huge win for the business, when it's compounded over many months of improved user retention.

Or at least, that's what we might initially conclude. But there are some problems with that.

One issue is that we're immediately drawn to look at and think about the retention lift, because it is significant, without thinking much about the two engagement metrics, which are non-significant and therefore "unimportant" or "boring." We'll talk more about how to take a broader view of the situation later on. But for right now, let's set that problem aside and think about just the retention measurement.

We're excited by this retention win because it's statistically significant, and also because it's much larger than any previous lifts we've seen so far. We've run a dozen tests, and some of them have indeed increased engagement, but any retention lifts before have been in the range of .002-.006, and none of them were statistically significant.

Are you seeing the problem yet? When we count the new test with the significant retention win, we've now run thirteen tests, each with a 5% chance of having a measured lift fully two standard errors away from the *true* lift.

On this particular occasion, we got "lucky". Although as experimenters we could never know for sure, let's suppose that the true lift was .005, right in line with our previous tests, but with a standard error of .005, we happened to draw a +2 s.d. measurement error and read a .015 lift instead. (In 13 tests, there is almost a 50/50 chance of this happening at least once—for each outcome metric.) And because it's the one that came out statistically significant, we pay attention to it and celebrate it. Only... we think the lift is *three times bigger* than it actually is (observed 0.015 vs true 0.005).

The primary danger is not that we get wrong results, i.e., results that are in the wrong direction, unless power is very very low. Instead, the danger is that we get results that are in the correct direction but far too big. That can lead us to shift the direction of future testing, or even the direction of the business, trying to chase an illusory lift.

How we can get less inflated results

So how can we address the statistical significance filter and make better decisions? One approach is to apply *regularization* to our estimates in a consistent and principled way. Regularization pulls our estimates toward 0 (or toward some other central point, but we'll use 0).

Regularization is widely used in machine learning because it's a way of addressing *overfitting* in predictive models: it's easy to build a model that does a very good job of fitting all the details of the training data, but then falls apart when it comes to making predictions on new data.

The statistical significance filter pushes us toward estimates that are inflated—which is a kind of overfitting—and regularization pulls us back down to a more reasonable estimate that is better for making predictions about what we'll do next time. Of course, that means we're getting *biased* estimates when it comes to figuring out what actually happened in that past test. But we don't need to care about that very much. A/B tests should be for learning and prediction; we're doing inferential statistics, not descriptive statistics.

If you've ever applied regularization in an ML context, you're probably already asking the next critical question: *How much* regularization should we be applying, and what kind? Figuring out how strongly to regularize is itself an additional parameter that needs to be estimated when building a model. It is definitely not an easy question to answer, especially not here in our A/B testing context.

In the face of these challenges, a Bayesian approach lets us chart a way forward. It's not the only way, but I've had success with it in the past.

A Bayesian solution

This model is motivated by the idea that our A/B testing and subsequent predictions take place in a specific context. Our tests aren't pure one-offs that have no relationship to any past or future tests (or, at least, they *shouldn't* be in a healthy testing regimen). Rather, we test different but related things.

Maybe for our product, we run a lot of tests that try to improve user engagement and retention; some of them hit various different parts of the experience, but they all have the same fundamental goal.

We should think of these tests as similar to each other in many relevant ways, far more similar than most of us usually expect.

Here's an imagined, but realistic, sequence of the dozen user retention tests we ran before the UI improvement described in our first example:

No two of these tests will have the same effect on retention, and many won't even affect retention through the same causal mechanisms. Tests 11 and 12, if they affect retention, should only really help new users; long-time users won't be seeing the tutorials. Tests 3 through 7 might affect heavier users more, because they spend more days on the site and are more likely to notice the upgrades, while test 8 is more likely to have an impact on occasional or near-lapsed users. Tests 1 and 2 won't affect users who stay logged in for the duration of the test.

Test #	What we tested
1	Make the login button more visible
2	Improved login flow
3	Improved site responsiveness #1
4	Improved site responsiveness #2
5	Improved site responsiveness #3
6	New feature #1, but adds UI complexity
7	New feature #2, but adds UI complexity
8	Reminder emails to re-engage users
9	Payment flow improvements
10	Recurring billing improvements
11	Tutorial for new users
12	Improved tutorial for new users
13	*UI change from the example*

But at a deeper level, these tests are united by our business model. We have a product. People like using it. People are willing to pay us for the opportunity to use it. If we can make the product better, easier, or more pleasant to use, or if we can remind people to use it to solve more of their problems, or if we can make it easier for them to pay us, then we can increase the rate at which people continue to pay us to use it. There aren't any retention tests that *don't* center around the fact that we have a subscription product and it's hard to imagine what such a test would look like. (The same general point holds, *mutatis mutandis*, for other business models.)

Most of all, these tests are similar in that, most likely, they all have similar plausible effect sizes, certainly in terms of order of magnitude. If we've changed a dozen very different parts of the user experience and have typically seen no more than a .06% lift to retention from any one change, then we shouldn't expect that the thirteenth change could lead to a .60% lift, 10x the others, even if it's not something we ever tested before. Of course, it's not a guarantee. We can be surprised! Sometimes small changes have a big impact. Or sometimes we know in advance that we'll get a big impact, like when we reduce the price of our subscription. But usually, we should trust our past tests to help inform us of the plausible kinds of lift we could get, from the kinds of interventions that we usually test.

That leads us to a *partial pooling* approach to estimating test effect sizes. We can use our past test results as *prior information* in a Bayesian model, giving us information about what kinds of effect sizes to expect from the next test. Yes, this requires us committing whole-heartedly to the line of argument above. And getting informed buy-in from stakeholders for such a change to our testing methods is not an easy task. But the benefits can be worth it.

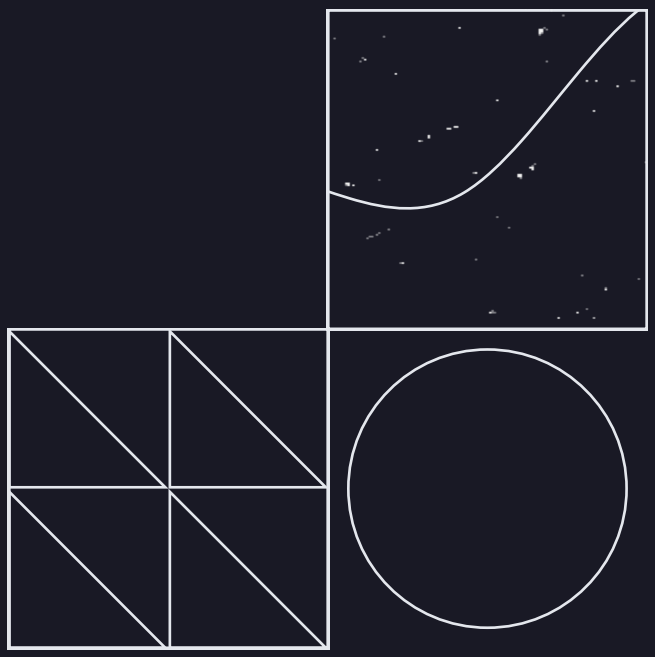
Specifically, I like to use a model that says that each new A/B test has a true effect size which is drawn from some common distribution, centered at 0. That common distribution might be Gaussian (leading to something like L2 or "ridge" regularization). Or maybe we want to leave more room for those surprisingly large effects I mentioned; we could pick a distribution with fatter tails, such as the t distribution with 4 degrees of freedom (see Bayesian Data Analysis p.437).

Then, a test observation is assumed to be the true (but unobservable) effect of the test, plus Gaussian error with standard deviation equal to the standard error of the estimate. (We could use t distributions but our tests all have high degrees of freedom so they're approximately Gaussian anyway.)

We can only use this model when we have some past tests to include as prior data. What happens is, the distribution of latent effect sizes is rather wide, but still has a mode at 0. Tests that are highly powered have low standard errors on the estimator, so the data overwhelms the prior, and the results are shrunk only a little bit towards 0.

Badly underpowered tests, on the other hand, have wide standard errors and a lot of probability mass far away from any plausible effect sizes, but also a lot of probability mass near 0; these get regularized heavily. Yes, it means that some "big wins" (that aren't real anyway) vanish into disappointingly small results. And that can be hard to swallow, but those "wins" were illusory, and they were only going to lead to poor decisions in an attempt to chase spurious effects. We're better off letting them shrink away toward 0.

I've implemented the model in a Mode Notebook using Stan. (I used R for the "glue code," but the Stan model will run just as well in a Python environment if that's what you prefer.) Please note that the prior distribution is constrained to be symmetric around 0, so we don't need to worry about which group is "test" and which is "control." If you replace the fake "past test" data with your own tests (after making sure they are all on the same scale), then you can use this model as-is, or adapt it to your own preferences.



Example: Retention numbers after applying a Bayesian model

Let's take a look at what happens to the retention lift in our example test if we apply this model. Of course, the results depend not only on this specific test but also on the other tests we include in the dataset; those other tests (also fake data) are in the CSV file included with the notebook. After running the model, we see the following results:

This test is strongly affected by the regularization because it was badly underpowered. The standard error of the estimator is large relative to the effect sizes we've seen in the past, and furthermore, the confidence interval does not exclude those typical effect sizes. So the model heavily regularizes the result; it's very likely, given the evidence we have, that the outsized retention lift we see comes from observation error.

	Unregularized	Regularized
Lift	.015	.0087
Standard error	.005	.005
t	3	1.7
p	.003	.09

If, on the other hand, we ran a test that had the same lift but a standard error ⅓ as large (say, because we ran a test with 10x as many people in our test and control groups), then the model would regularize that result much less, because we'd have overwhelming evidence that the lift is not just a fluke. (Try it for yourself and see!)

At a practical level, this changes the business decisions we would make as a result of this test. There's still evidence that this new UI is favorable for retention, and we should go ahead and roll it out. But the size of that lift is not very well determined by the test; it's more likely than not to be on the high end relative to our past tests, likely at or above the .006 range, but the test wasn't precise enough to let us say just how big.

UI improvements are something we should keep investigating, but not at the expense of other kinds of work.

Even if you're not about to roll out this model—and honestly, doing something like that on the strength of one white paper is probably a bit too impulsive—it's still worth playing with the Bayesian model to get a sense for what it does.

If you spend some time with it, you'll begin to get a sense of how underpowered tests are affected by the partial pooling. Well-powered tests, in contrast, shift only slightly toward 0. Using prior test results in a Bayesian model is a way of regularizing uncertain and underpowered test results in a formal and consistent way.

What to consider if your team isn't yet open to Bayesian models

Getting buy-in from all the relevant stakeholders to roll out a fully Bayesian A/B testing framework with pervasive regularization is a challenge. Getting stakeholders to buy into a system that:

- makes all your lifts look smaller
- forces you to run fewer tests just so that any of them can be well powered, instead of letting you run lots of (underpowered) tests and pick out several (illusory) wins each quarter
- ties together the fate of all the tests you run, instead of letting each succeed or fail on its own
- is substantially more complicated and opaque than simple t-tests

...is *extraordinarily difficult*.

But it's possible. I've seen it done and have helped roll out such a system myself. It was limited in scope, a specialized A/B testing domain that was already siloed off from our main product tests, but we did it. And, as should be obvious by now, I believe that it helped. It made a difference. It let us learn better, slower in the short term but more accurately, and thus faster, in the long term.

But not every team in every company will be open to this kind of solution. Nor is this kind of solution right for every team in every company. Different companies have different circumstances and it is your job to help your company make the best decisions for your company. (Be advised that non-Bayesian approaches to regularization are going to have a lot of the same problems.

Even if you're not explicitly doing partial pooling across tests, you're still going to need to pick a regularization parameter based in part on the scale of test effect sizes you've seen in the past, and the practical impact of the regularization itself is going to be basically the same as in this Bayesian model.)

So what else might you do?

01 - Avoid violating the assumptions of NHST and OLS

If we want our estimates to be unbiased while using NHST, we need to pay just as much attention to statistically non-significant results as to the statistically significant results.

But this is difficult in practice. We're interested in improving our products, and that means paying attention to signals either that there's an improvement to be made or that there is real harm being done.

It's painful and boring to think just as hard about every near-0 "failed" test as we do about every exciting large significant result. We naturally slip into paying attention just to the significant lifts and that's when the statistical significance filter creeps back in.

First, if we only look at effect size estimates (and confidence intervals or credible intervals) but don't do significance testing, then no filtering occurs. This approach certainly has its advocates, and I do believe it has a place.

But there are challenges. Looking to see whether the credible interval includes or excludes 0 is a backdoor into statistical significance testing, and then we might introduce the statistical significance filter while denying that we're doing it.

02 - Design and run experiments with high statistical power

We can also protect ourselves from the statistical significance filter implicitly, by carefully designing our tests up front. Even if we don't treat them as prior information in a Bayesian sense, those past tests still help inform us about the scale of effect sizes that are plausible. Therefore, we can use them during the design phase, rather than the analysis phase, to improve our power analysis.

The impact of the statistical significance filter is very large for underpowered tests, as we've seen. But it's not so bad for tests with high power. The figure below shows the situation for a test with power $= 0.9$, what most of us would consider a well-powered test. The area of the true distribution that's close to 0, where we'd fail to reject, is quite small.

Now, this still does lead us to inflated estimates, but not very inflated; the solid and dotted lines show the true and conditional expected values of $\hat{\theta}$ if we use the statistical significance filter, and they're quite close together. But running a test with even 90% power for a plausible effect size is shockingly difficult in many business contexts.

And of course, it's impossible to perfectly predict power anyway, because it depends on the unknown θ.

Compare this to the figure earlier, where power $= 0.5$ and the effect size is substantially inflated in expectation. As power gets even worse, approaching, the inflation can become arbitrarily large.

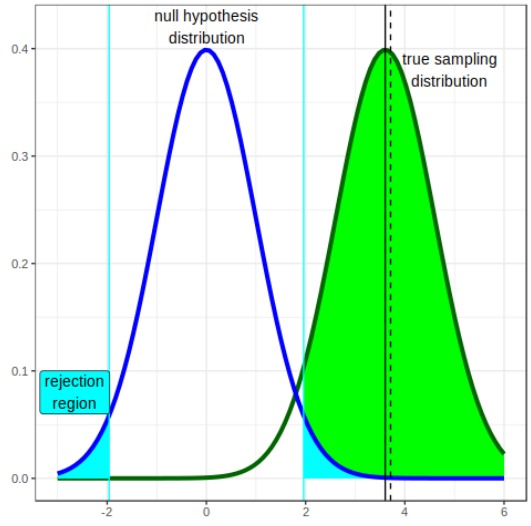


Figure 02: R code for this plot

It's often that we power our tests for the lifts we wish we could get rather than the lifts we'll probably see, and that's a recipe for being led astray by the statistical significance filter.

If all our tests are well-powered to detect the effect sizes we're likely to actually see, then we'll end up analyzing tests that would be minimally regularized under the Bayesian model anyway, and we can, in some sense, dispense with the formal regularization.

> **"It's often that we power our tests for the lifts we wish we could get rather than the lifts we'll probably see, and that's a recipe for being led astray.**

Improving our power analyses is one of the biggest things we can do to improve our A/B testing anyway; it just carries special importance if we're not regularizing during the analysis step. So even if you do use the Bayesian model or some variation on it, this is worth thinking about.

That said, it's just not always possible to run highly-powered tests. For a wide variety of business reasons—cost, duration, negatively affecting too many of our users, opportunity costs for other tests, taking advantage of an unplanned quasi-experiment or observational inference opportunity, a true effect size that is near zero, and many others—sometimes we just *need* to analyze a test that isn't at the level of power that we'd prefer. It happens. If we have a regularization procedure planned in advance it gives us another layer of protection against being misled by those tests. It's not the only way forward, but it helps.

(As a side note: You might have noticed that the example had three different outcome metrics and wondered whether applying a multiple comparisons correction to the p value would help avoid the statistical significance filter. But no! This actually makes the problem even worse, because it decreases test power.

We'll need to run a larger number of underpowered tests before one happens to come out significant, but when we do get "lucky," the effect size will be even *more* inflated. Multiple comparisons corrections have their place in null hypothesis significance testing, but they don't help here.)

03 - Do informal regularization after the mathematical analysis

Ensuring your tests are well powered is important, but it doesn't help with the analysis you're doing right now for the test you finished running last week.

We, as data scientists, and our stakeholders, as business decision-makers, are all making judgments all the time. We judge business context, and we judge our A/B test results in that context. We rarely follow our A/B tests "blindly." Sure, maybe there's a bandit algorithm in there somewhere that decides "by itself" which content to push forward on our homepage; but even when we roll something like that out, we're still making a judgment that this is an appropriate and safe domain in which to follow the test results without additional human judgment. (And we still build in manual overrides, for the times when the algorithm gets something wrong.)

And when an A/B test result is just not plausible, when it's unbelievable, we're especially careful about our decisions. We don't completely throw away the test result, because we might be wrong about what's going on: Maybe the new homepage design really does hurt retention rates, maybe we forgot to include the login button... oops. But we pay attention and look for explanations and, in general, use our best judgment about how to go forward.

This feels like a kind of Bayesian updating. We have lots of background (or "prior") information about our business that we can't encode into our A/B test analysis, so we read the analysis in light of that background context. When results are unreasonably large, or in an unexpected direction, or otherwise surprising, we don't—or shouldn't—necessarily just say, "Well, guess the world works completely differently to how we thought it did!" and entirely change the direction of the company. This is a kind of implicit, pragmatic *mental regularization*.

What are we regularizing to? I'd say, we mentally regularize to *our best understanding of the causal mechanisms that make our business function*. So having clear causal stories behind our tests, both in advance during the design phase and while seeking out new causal understanding when the results are surprising, is critical for us to do a good job with this. And paying attention to every test result is an important part of this process, so that we can have a good mental model of typical lifts and discover unexpected patterns in the results.

Example: Making good judgments about retention

Let's think back to our example test. We changed the UI and saw a significant retention lift, but with little corresponding impact on user engagement:

Measure	Control	Test Group	Lift	Std Err	t	p
Days active ratio	0.400	0.405	.005	.006	.0.83	.4
Minutes/visits	39	37	-2	8	-0.25	.8
Retention	0.750	0.765	.015	.005	3	.003

Does that seem plausible? Well, perhaps not so much. Our implicit causal story is that the UI should make the product more usable, which leads to improved retention. But if that's our story, then at least one of the two engagement measures is *probably* (not guaranteed, but probably!) going to reflect the improvement as well.

In this case, we see a non-significant lift to the proportion of days on which users are active, with a tiny and non-significant decrease in the minutes per visit. We might want to think that the retention lift is in line with the users who come back more frequently—and mentally shrink the effect size to something commensurate with that lift, despite it not being significant.

If, on the other hand, this hadn't been a UI test but instead a change to our payments flow, it might be entirely plausible that we'd see a retention lift with no corresponding engagement impact. And we'd want to take that into account in deciding how to act on the results.

A/B tests are the beginning, not the end of solving a problem

Done poorly, mental regularization means injecting human judgments into our A/B test analyses. And that can completely destroy the mathematical assumptions that traditional statistical significance testing is built on. This is practically a recipe for manufacturing violations of the assumptions for null hypothesis significance testing.

But this is only inherently wrong if NHST is the goal of our work, and it's not. Making good decisions is the goal. We *should* be using this kind of judgment, and working to get better at this kind of judgment —recognizing what we're trying to do and what kind of effects it can have. If we do some formal regularization, that can help us with this. We might even think of formal regularization as an application of informal regularization, rather than thinking of informal regularization as an approximation to formal regularization. The decision of which statistical methods to use does not itself have a statistical answer.

And so, whether we use unbiased OLS or do Bayesian partial pooling across tests or apply some other kind of formal regularization, reflecting on why we made past decisions is worth the effort. Statistics is the beginning, not the end, of an A/B test analysis; running an A/B test is the beginning, not the end, of solving a problem. If we're aware of what we're doing and why—and that includes when we're letting statistical significance guide our attention to some test results over others, but also when we're letting our background knowledge override what looks like an unambiguous test result—we can improve our business decision-making. We can do a better job of learning from our A/B tests.

And our business can flourish.