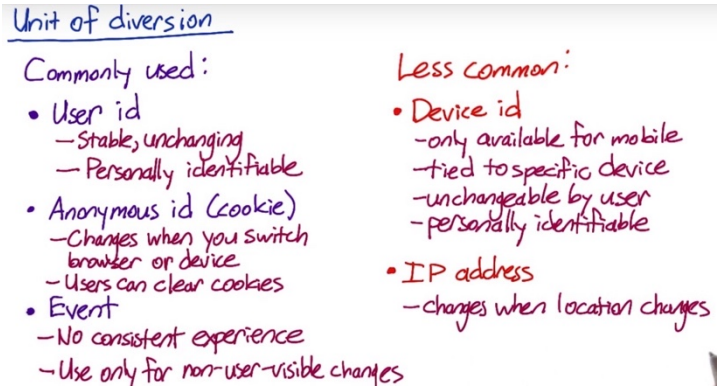
Design an Experiment

Now it's time to really apply what you've learned in working through the decisions you need to make in actually designing your experiment.

- First, we'll need to decide how you define what you use as a subject in your experiment and in your control. In other words, what are the units in the population that you're going to be running the test on and comparing? We call this the unit of diversion.
- Next, we'll need to choose the population. You'll need to decide which subjects are eligible. Everyone? Only subjects in the U.S.? When you're testing how to change and computing the evaluation metrics, you need to ensure that you're doing the test and computing the metric on equivalent populations.
- Then we'll use those decisions and what we learned in lesson three to properly size your experiment, before concluding with a few other decisions you need to finalize your experiment design, such as the duration of the experiment.

Unit of diversion: Typically, what you want to do for a user visible change is that you want to basically assign events people to either the control or the experiment. To do this, you're going to be using some imperfect proxy, like a cookie based, or a user ID for your people-based diversion. These are all what we call our unit of diversion.



User id and Anonymous id are different approximations to actual user or person, and event is just the singe event.

1. User id:

This would be something like the login that user created, such as user name or email. For example, your email address if you log into Facebook or Amazon, or your username, if create a username instead.
All the events correspond to the same user id are either in the control group or the experiment group, but they are not mixed between the two groups. Whether the user is using an app on their phone, visiting the website on their phone, or visiting the website on their desktop computer, it's a consistent experience.
User id is considered personally identifiable as it is usually associated with other personal information for an account, the user's email address or phone number, to help with account recovery.

2. Anonymous id:

An anonymous id is usually something like a cookie. On most websites, whenever a user visits the website, it will write a cookie, which is usually an anonymous random identifier to a file on that device.
The cookie is specific to a browser and a device though. If the user switches from Chrome to Firefox, or if they switch from their laptop to their phone, they'll get a different cookie.
Users can also choose to clear their cookies. In other words, it's much easier for a person to change their cookie.

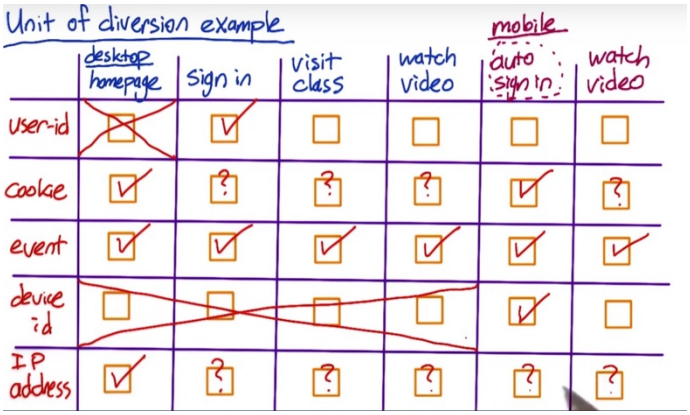
3. Event-based diversion:

Event-based diversion means that on every single event, you redecide whether that event is in the experiment or in the control. This means that a user may not get a consistent experience at all, so this is only appropriate in situations where the changes are not user visible. For example, if you have a ranked list, changes to the order of the list would fall in this category. Most users can't tell or won't notice.

There are also a couple of other less commonly used options for unit of diversion.

4. Device id: on mobile devices only, there's an option called a device id. It's also considered identifiable because it's immutable. But it doesn't have the cross device or cross platform consistency.
5. IP address: if the user changes location, then they often get a new IP address.

Now suppose you're running an experiment that would affect each of these different pages. For example, maybe you changed something about the navigation bar and it shows up on every page. For each of the different units of diversion we've talked about, user-id, cookie, event, device id and IP address, when would the user be assigned to the same group as before and when could they potentially be switched to the other group? For each case, check the box at the point or points, where the user could be switched from the experiment to control or vice-versa, including the first time that they are assigned to a group.



If you're doing diversion based on the user ID, the user would be assigned to a group when they first logged in.
If you did cookie-based diversion, you'd make a decision when the user first visits the home page, and again when they start the mobile app. But the users could clear their cookies at any point, meaning that they could be reassigned at any other point.
If you did event-based diversion, then on every single event, you'd re-decide whether that event was in the experiment group or the control group.

If you are doing device id-based diversion, then you'd assign the group at the start of the mobile experience. Since you don't typically have device id's for non-mobile devices, you wouldn't be able to run the experiment on the events before the user switched to their mobile device.

Considerations for choosing user diversion.

A. Consistency

1. User consistency:

- If you're using user id, then user gets consistent experience as they change devices as long as they stay signed in. And so for a certain set of changes, the user will get a consistent experience across devices.
- Now, on the other hand, if you're testing a change that crosses the sign in, sign out border, then a user ID doesn't work as well. So for example, if you're changing the layout of the page or the location of the sign in bar In that case, you may want to use a cookie instead, so you get consistency across the sign in and sign out border but not across devices.

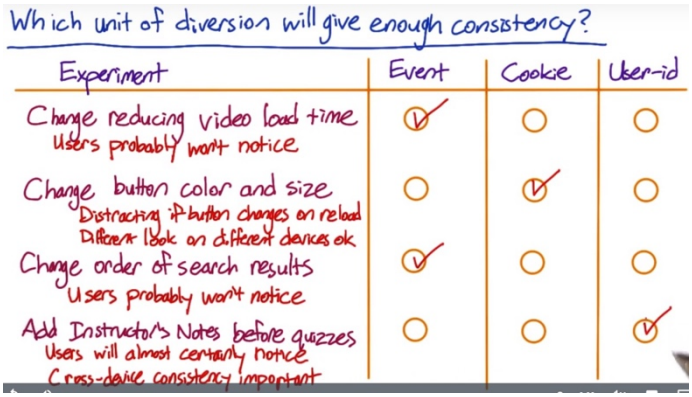
2. User Visibility:

- For user visible changes, you would definitely use a cookie or a user ID.
- There's probably a whole host of changes that are not visible to users. This can range from latency changes to backend infrastructure changes or honestly, ranking changes. For those changes, you can consider other user diversion.

3. What you want to measure:

- e.g., if you want to measure a learning effect, whether or not users adapt to change. In those cases, you also need a stateful unit of diversion like a cookie or user ID.
- if you're making a latency where, that you're making the site slower and you're trying to see whether or not the user uses the site less. In those cases, you need to use a cookie or a user ID to see what happens across time.
  So even when the user doesn't notice the change, depending on what you want to measure, you may also choose a user ID or cookie.

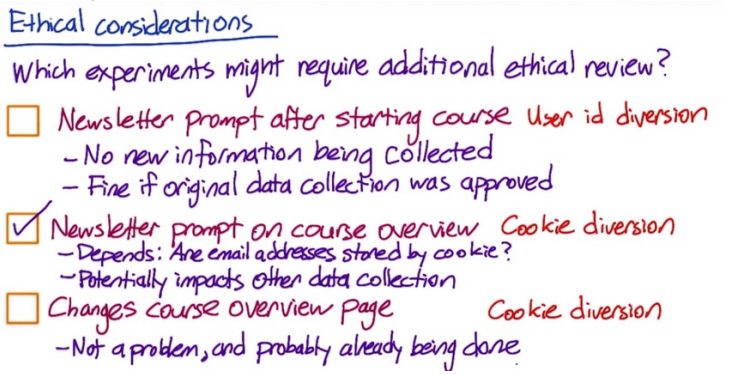
IP-based diversion
IP base diversion is not very useful generally speaking. You don't get the consistency because user's IP address could randomly change depending on the provider nor do you get the clean randomization that you get from event-based diversion.
There's a whole host of changes where IP based diversion may be your only choice. For example, your testing out an infrastructure change when you're testing out one hosting provider versus a different hosting provider to understand the impact of latency. In that situation, IP based diversion may really be your only choice.
What happens with IP based diversion is that you may not get a clean comparison between your experiment and your control. One example of this is modem dialups. For some providers, they all aggregate all of those modem dialup users into a single IP address. And so, then the question is how do I find that comparable population of users in my control? And so when you do IP based diversion is doing a lot of post analysis to try and find those good comparisons between your experiment and control.



For the 1st and 3rd example, there could be learner effect. You can start with event-based diversion, and switch to cookie-based in the future if necessary.
The fourth case is something that users will almost certainly notice. Cross-device consistency will also be important here, if you want to be able to determine whether the change impacts the pass rate of the quiz. If a student watches the video on their phone, then completes the quiz on their computer, for example, you'll need them to be in the same group both times. Because of this, you'll need to use user-ID based diversion here.

B. Ethical Considerations

If you use user id, then it is person identifiable, and there will be security and confidentiality concerns to address, and might need to get user consent.



Case #1. The newsletter signups were already stored by user ids.

Case #2. Audacity wants to test moving the newsletter prompt to as soon as a user views details about any particular course, and users who do not have an account yet are asked for their email address.

It depends on exactly how the email addresses are being stored. The most natural implementation would be to store email address them by cookie, since the diversion is by cookie. That would make those cookies non-anonymous, which potentially impacts any other data that is being collected. For example, if another experiment is storing data by cookie that you wouldn't want re-identified, then that data has now become linked to an email address.

Case #3. Audacity changes some of the information on a course overview page, and measures the click through probability on the enroll button.

It doesn't require an additional review. Storing clicks by cookies is not a problem and is probably already being done elsewhere on the site.

<u>In general, you to watch out for what whether you are accidentally identifying data that would otherwise have been anonymous.</u>

C. Variability Considerations

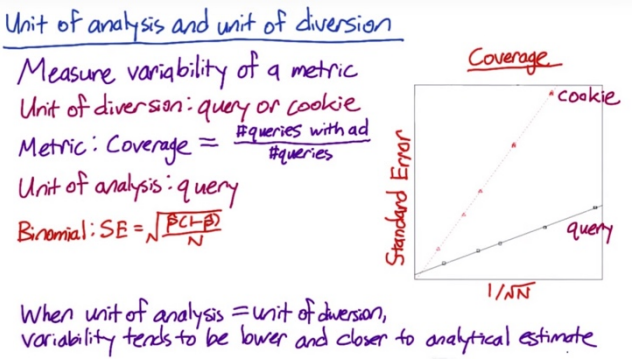
Unit of analysis is whatever the denominator of your metric is. And when your unit of diversion is the same as your unit of analysis, the analytically computed variability is likely to be very close to the empirically computed variability.

e.g. click-through rate = clicks / page views – page view is the unit of analysis. In the case of event-based diversion, page view is also the unit of diversion. Then, the analytical variability will be very close to the empirical variability.

However, if unit of diversion is cookie or user id, the actual variability might be a lot higher than what was calculated analytically. Sometimes by a factor of four, five, maybe even more. In those cases you really want to move to an empirically computed variability given your unit of diversion. This is because when calculating the analytical variability, you are assuming:

- The distribution of the underlying data
- What's going to be considered as independent

If you use event-based diversion, you assume each single event is independent. But if you use user id or cookie-based diversion, the independence assumption is no longer valid, as you are diverting groups of events and they are actually correlated.

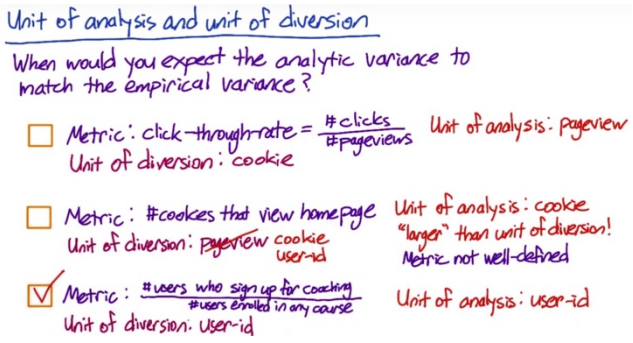


An experiment done by Google:

Diverting by query is a type of event-based diversion, since for a search engine, a query really is an event.

The metric they measured was called coverage, which is defined as the percentage of queries for which an ad is shown.

Notice that when the unit of diversion is a cookie, which is not the same as the unit of analysis, the variability is much higher. The variability might be higher by as much as four times, depending on the sample size



Case #2: There isn't really a denominator to this metric, but since the number of cookies is what's being computed, the cookie is the unit of analysis. The unit of analysis is larger than the unit of diversion in the sense that one cookie could generate multiple pageviews. This is a problem, given that the unit of diversion is a pageview, because it means the same cookie could have events in both the experiment group, and the control group. That means this metric is actually not well defined for this experiment design. In general, you need your unit of diversion to be at least as big as your unit of analysis.

In this case, cookie would work as the unit of diversion, and user-id would work, since one user-id can correspond to multiple cookies.

Choose a population: Inter- and intra- user experience

Questions to keep in mind when choosing a population:

1. You want to think about the fact that, in anything but event diversion, if you do cookie diversion, if you do device diversion, you're really looking at proxies for users. And that means you're going to have one group of users on the A side of your experiment and one group on the B side. Now if you do event-based diversion, you can end up with the mix of the same people on both sides.
   So, you have to be pretty careful in this case to make sure you haven't inadvertently mismatched your users.
2. There are some options:

Intra-user experiment: you expose the same user to this feature being on and off over time, and you actually analyze how they behave in different time windows.

This has some pitfalls, for example:

- You have to be really careful that you choose a comparable time window. You don't want to do this in the two weeks before Christmas and then have them behave very differently in the second part.
- With a lot of features, you might have a frustration or a learning problem, where people learn to use the particular feature in the first two weeks and then when you turn it off, they're like, why did my website change?

Interleaved experiment: where you actually expose the same user to the A and the B side at the same time for certain other types of applications like search ranking, preferences or, other things where you actually have a ranked order list.

Inter-user experiments: which is used most A/B testing. That means you've got different people on the A side and on the B side. There is a refinement of that called a cohort. In a cohort, you try to match up your entering class so at least you have roughly the same parameters in your two user groups.

Target Population

Assuming that we're doing an inter-user experiment. That is, there are different users in the different groups. And then we need to decide our target population.

There are some easy divisions of your user space, such as what browser they're on, what geo location they come from, what country, what language they're using, how long they've been using your websites, etc. You may even have, depending on what you're doing, demographic information, such as their age, that you could use to target a very specific population of, of your user space.

Why decide targeting population in advance:

- If you're running a feature and you're not sure if you're gonna release it and it's a pretty high-profile launch, you might want to restrict how many of your users have actually seen it. So, you don't get any press coverage or blog coverage.
- If you want to release it internationally, you need to check is this language right.
- If you are not sure that your feature works on old browsers, and you might want to just restrict it to say modern browsers.
- If you're running a couple of different experiments at your company at the same time, you might not want to overlap. You might want to have, you know, oh, I'm just going to take this section of traffic, and you guys can run that other experiment in Korean, and it'll be fine.
- You may not want to dilute the effect of your experiment across a global population. So you may only run your experiment on the affected traffic.

Cases in which don't choose particular traffic:

- You cannot ID who a particular feature is going to affect.
- You may want to test the effect across your global population because you not sure if your targeting is exact, the way you want.
- You may just not care that much because it could be a feature that effects 90% of your traffic.

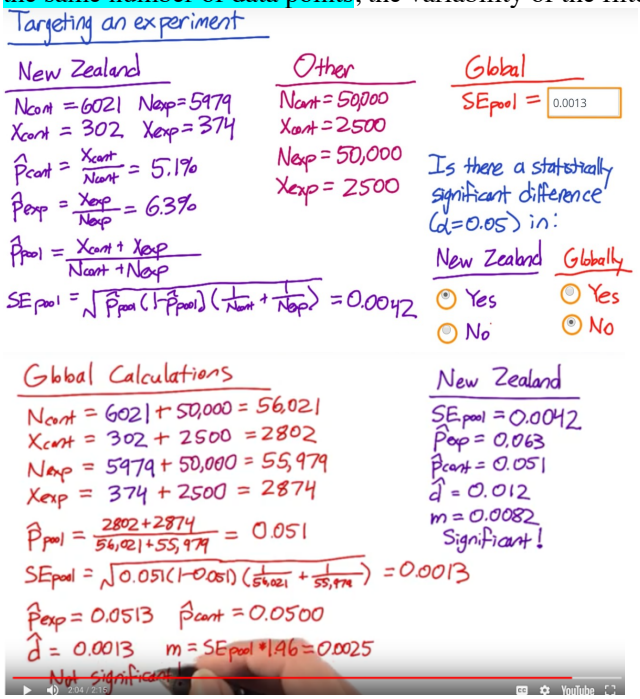
What you need to do to decide your targeting population:

- You need to talk to your engineering team first, or whoever implemented the feature, to better understand the features. Like are we sure that this is not going to trigger for this particular browser? Is

our targeting exactly right? Are we actually concerned about potential interactions so we might want to run a global experiment.

- You always want to make sure that you have the same filters on the targeted and untargeted parts of your experiment. So you don't want to do accidentally include only logged-in users on the targeted bit. And then when you go to compare it to your global population you realize that there's something completely wrong. So you want to make sure that everything's lined up.
- Before you launch a big change, you may actually want to go back and run a global experiment and make sure that you don't have any unintentional effects on the traffic you weren't targeting because that can be a real issue.

Example for diluting the results:
In this case the variability of the global data as measured by the pooled standard error is lower than the filtered data. Mostly because there is so much more data globally. This will often be the case in practice. But it's also good to keep in mind that, in practice, your data will actually be a mix of different populations almost every time. When you filter, you're going to get a smaller but also more uniform population. Which means that for the same number of data points, the variability of the filtered data is likely to be lower.

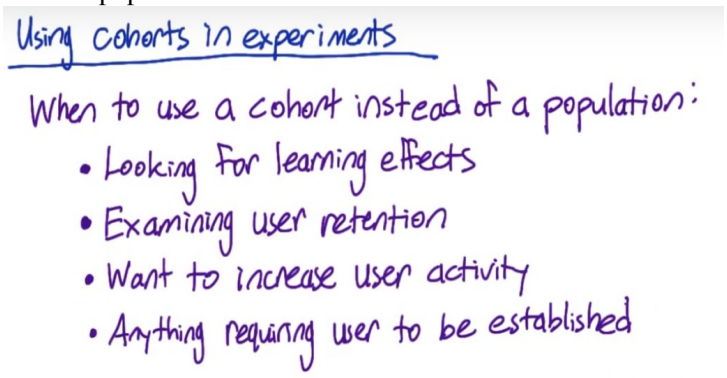


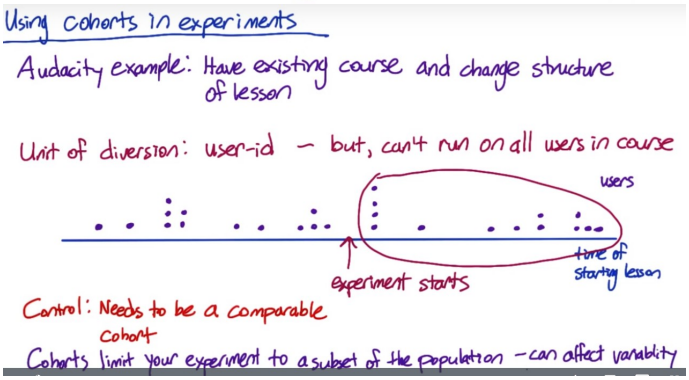
Note that even though the New Zealand only data had a higher variability, and thus a wider margin of error, or a wider confidence interval, the New Zealand results were significant whereas the global results were not, because the observed difference was so much higher in New Zealand, 0.012 versus 0.0013. Adding all of the unaffected traffic that was outside of New Zealand diluted the difference in the global data, causing the result not to be significant.

Populations vs. Cohort

- Typically, cohort is a subset of populations for users entering the experiment at the same time. It usually means that you define an entering class and only look at users who entered your experiment on both sides around the same time, and you go forward from there.
- You may have all kinds of problems during the span of your experiment. So you can lose users, and gain users, and have users who've been exposed to the experiment for different period of time.
- You can also use other information to define cohort – e.g. users have been using your site consistently for 2 months, users with both laptop and mobile associated with their user ID, etc.
- Cohorts are harder to define and require more data as we are losing some users.

- Typically, cohorts are used when looking for user stability (e.g. measure learning effects, increased usage of your site/device, etc.), when you want to observe how your change affects users' behaviors relative to their history. If you don't need those types of metrics, then you can probably stick with the population.





Suppose Audacity have an existing course that's already up and running. Some students have completed the course, other students are midway through, and there are students who have not yet started. They want to try changing the structure of one of the lessons to see if it improves the completion rate of the entire course.
Now, because they want to see, what happens throughout the course where students can pause or unpause the lessons, switch devices, etc., the unit of diversion will need to be a user-id. That said, it doesn't make sense to just run the experiment on all the users in the course. To see that, suppose that this blue line shows the time that students start the lessons that Audacity is changing with later times to the right. Each purple dot represents a user or student.
Now, suppose that Audacity starts running the experiment at this time, for students who started the lesson a while ago, they may actual have finished the lesson already. So, they're already past that lesson, and they're not even going to see the change. Instead, it would make sense to use a cohort, and only include users who started the lesson after the experiment was started in the experiment. That is, it's a subset of the population, who have the shared experience of receiving the new lesson, and not seeing the old lesson.
Now, for the control, Audacity needs to create a comparable cohort. They cannot just use these users, who are not included in the experiment as the control because there may have been other system changes in that time that affected the new users.
So, instead, Audacity will need to split this cohort into an experiment cohort and a control cohort, so that they all have the same timing of when they started the lesson.

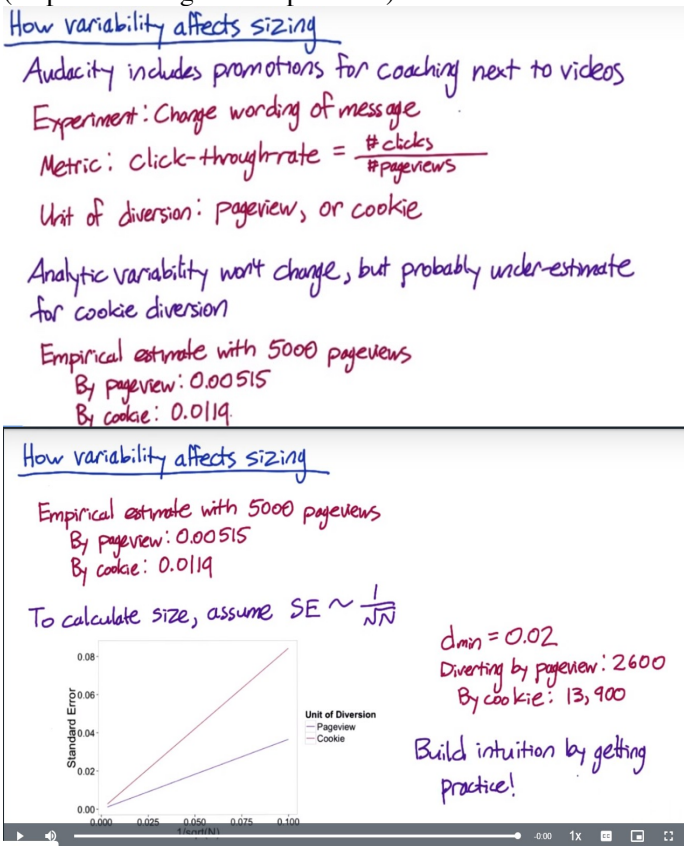
Sizing:

Sizing our experiment and control is an iterative process. What we're going to do is we are going to try out some decisions for our unit of diversion and our population, see what the implication is on both the size as well as the duration of our experiment.
And then if we don't really like those results, we'll need to revisit our decisions and iterate.

Your choice of metric, unit of diversion, choice of population – all these can affect the variability of the metric. So you want to take all the stuff into account and then start to determine and determine the size. You're going

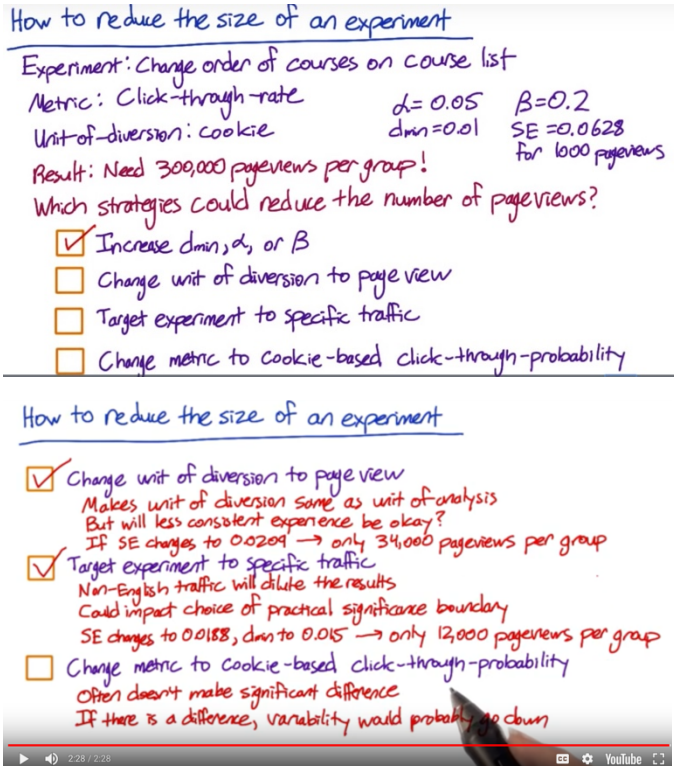
to have to figure out whether what you plan to do is realistic given how long it takes to run the experiment and the variability of the metric.
e.g. the page load time, the 90th percentile of latency. Originally you could measure that in an event-based diversion because you just measure each page load time.
If we want to measure if users increase the use of the site more based on the latency they experience, then we need to look at user id diversion. This will require a fair amount of user data. And if you're originally planning to run this globally, you may realize looking at the variance of your metrics, that that's just not really realistic. It's going to take a very long time to get a lot of data, it's a big investment.
You may think that I'm really affecting the 90th percentile here, that's what I'm targeting. So, let's look at people with slow connections. And then maybe, because I need to get enough data, I want to look at a cohort of users who've used my site fairly regularly over the past two months. And that way, I can get more data about them more quickly.
While this restriction may give you a smaller scope to your project, it can really give you a better sense of whether you're going to get a signal out of this experiment at all before you invest the time and the user time in actually running a larger experiment.

(empirical-sizing R code provided)



If you calculate the variability analytically, it won't change between the two units of diversion but for the cookie-based diversion, the analytic estimate is likely to be an under-estimate. So Audacity does an empirical estimate of the variability with 5,000 page views in each group.
In order to calculate the size, we can assume that the standard error for the experiment is proportional to one over the square root of the sample size.
In this case, let's say that the practical significance boundary, or d min is 0.02. Then if Audacity used pageview as the unit of diversion, they would need about 2,600 page views to get enough power. But if they diverted by cookie, they would need about 13,900 pageviews.

This formula calculates sample size needed for both control and experiment group, so it uses 2*. For each group, we don't need 2*.

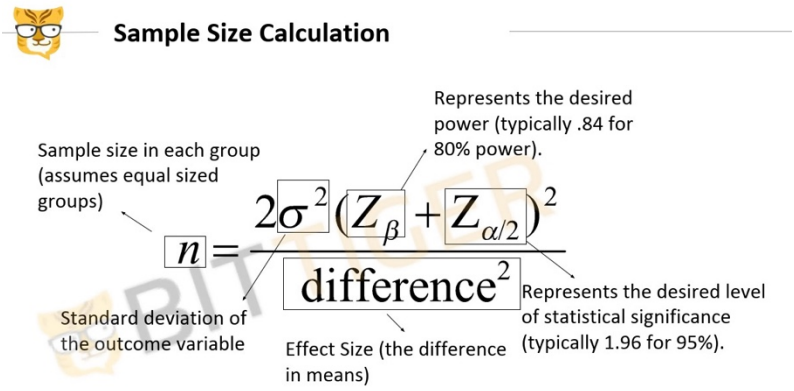


How to reduce the size of an experiment

Now let's say Audacity does another experiment, this time changing the order courses appear on their course list page. The metric they use is the overall click-through-rate to individual course pages.
That is, the total number of times the user clicks on any course, divided by the number of page views. To give a consistent user experience of the course list, while still including non-logged in traffic in the experiment, Audacity chooses cookie as the unit-of-diversion.
#1. They could increase the practical significance boundary (dmin) to not try to detect a smaller change. Or, alpha or beta, (decrease z-value of alpha and beta) that is, accept a higher probability of a false positive or a false negative.

#2. By changing the unit of diversion to be the same as the unit of analysis, the variability of the metric will probably decrease and be closer to the analytical estimate. By decreasing the variability of the metric, you decrease the number of pageviews you need to be confident in your results.
The main question here is whether the less consistent experience will be acceptable.
In this case, if audacity recalculated the empirical estimate of the standard error using the pageview as the unit of diversion, they might find that the new standard error was 0.0209 for the same sample size. 34,000 pageviews per group would be necessary.

#3. Targeting the experiment to English traffic will also reduce the total number of pageviews needed. Since the non-English traffic is not effected, including it will dilute the results of the experiment, which would increase the number of pageviews needed.
Of course, there are fewer non-English page views available than total page views. So this might not reduce the time frame of the experiment, but other experiments could be run on the non-English traffic in the meantime. So this could still be worth doing.
Filtering the traffic could also impact your choice of practical significance boundary. First, since you're only looking



at a subset of your traffic, you might need a bigger change before it matters to the business. Or since your variability is probably lower, you might want to take advantage of that and detect smaller changes rather than decreasing the size of the experiment.
Because the practical significance boundary could move in either direction, your size could really move in either direction.
But it's likely that the variance will go down and the practical significance boundary will increase, so it's likely that the size will be smaller.
In this case, suppose that audacity keeps pageviews as the unit of diversion and then targeting the experiment to English only traffic further reduces the standard error to 0.0188. And they also decide to increase their practical significance boundary to 0.015 for the English traffic only. At this point, they would only need 12,000 pageviews per group.

#4. It will often not make a significant difference to the variability, especially if you're using a short time window for the probability. If there is a difference, the variability will probably go down. Since the unit of analysis would be the same as the unit of diversion in this case. So this could reduce the number of pageviews needed, but it also might not help much.

Sizing trigger: there might be cases in which you don't know which fraction of the population is going to be affected by the changes of the feature. So you need be conservative about the time needed for the experiment. You can run a pilot experiment, or just observe the experiment for the first couple of weeks to check which fraction is affected.

Duration vs. Exposure

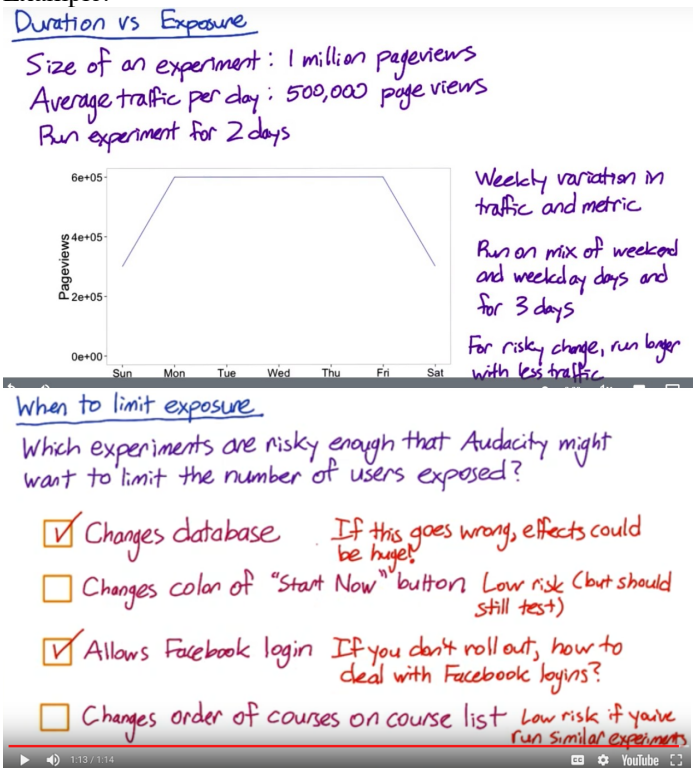
- First of all, what's the duration of the experiment that I want to run?
- Secondly, when do I want to run the experiment? Is back to school a good time to run it? What about holidays? Is it going to overlap something that's important?
- Third you have to think about what fraction of your traffic you're going to send through the experiment.

Those are all interrelated as they get you to the ideal size but you need to think about them a little bit separately.
For the 3rd question, what we're really asking is on any given day what proportion or what percentage of the cookies are you sending to your experiment and your control? Let's say we're and we need 1 million cookies in our experiment and our control combined. Now, if you only get a 100,000 cookies visiting your site on any given day. That means that if you want to run 50% of your traffic through the experiment and 50% through the control, you need to run your experiment control for ten days.

Now, another choice is to run your experiment at 25% each, say, it's because you want to run another test, then you'd have to run your experiment for 20 days as opposed to 10. And that's how, the duration of your experiment, is related to the proportion of traffic that you're sending through your experiment.
Why wouldn't you always run on all of your traffic so you can get results quicker?

- Safety consideration: basically, you may have a new UI feature, and you're not sure either how well it functions in all browsers, or how your users are going to react. They might get frustrated with it. So you might want to actually keep the site mostly the same, and only expose a few people to it until you feel more comfortable with it.
- Press: you want to limit the coverage of new features if you're not sure it's even gonna be the way you go with the site.
- If you're running a 50-50 experiment, then you can gather all data on a single day, would you actually want to make a decision based on a single day if it was a holiday? Well, a more common scenario is that you have to have very different behavior on weekdays and weekends. And so you might actually prefer to run at a smaller percentage across multiple days to get a sense for how the differences are by week day and weekend, across holidays, by different times of day, all of those different types of things that you are actually accounting for those other sources of variability.

Example:



For which experiments do you think it would make sense to run the experiment for longer, but expose fewer total?

#1. In practice though, if this kind of change goes wrong, the effects could be huge. Your site might go down, or not work at all. Of course, you should always be testing changes like this, and all changes in the controlled development environment before exposing it to users. But sometimes new bugs appear when the change is exposed to real traffic. Because of this, it's often a good idea to roll out this kind of a change to a small percentage of users, make sure nothing goes wrong, before rolling it out to everyone.

#2. The second case is low risk. Changing the color button is innocent enough that even if all your users saw the change, it would probably be fine. But you still need to test the change before rolling it out to users.

#3. The third case is higher risk, particularly if you end up not rolling out the experiment. How are you going to deal with all these Facebook logins that you're not supporting? Keeping the affected users to a small number, so that you won't have very many of these to deal with, would be a good idea.

#4. Assuming that you've run similar experiments in the past, this last case is low risk also, since most users won't notice ranking changes. If this is the first time you've tested a ranking change, though, then this might be risky for the same reason as the database change. If there's a bug, the courses might not appear at all, for example.

Learning effects

Learning effects is basically when you want to measure user learning or whether a user is adapting to a change or not.

Two different types of learning effects:

1. Change aversion, where when users first see a change they're like, what is this? I don't like anything.
2. Novelty effect which is the exact opposite. Oh, this is a new thing.

In both situations, what happens is that when a user first sees a change, they're going to tend to react in one of these two ways. But over time they're going to probably plateau to a very different behavior. Now, the key issue with trying to measure a learning effect is time. It takes time for you just to actually adapt to a change and often times you don't have the luxury of taking that much time to make a decision.

Things to keep in mind if you want to measure user learning:

1. You need a stateful unit of diversion like a cookie or a user ID.
2. Because a lot of the learning is based on not just a slight time but how often they see the change so we call that a dosage. Then you probably want to be using a cohort as opposed to just a population. And so you would choose a cohort in both the experiment and the control based on either how long they've been exposed to the change or how many times they've seen it.
3. From a duration perspective, because you want to measure a learning effect, this is going to take some amount of time to basically see what's going to be happening. Now, the other thing though, is that it's going to take a long, a long period of time. You don't want to be putting a lot of your users through a change that you're testing over a long period of time because maybe you end up testing other changes. From a risk perspective, if you're actually wanting to measure a user learning effect, that means that you're probably a little uncertain about what the effect is going to be, which means that it's probably a higher risk change. Now, both of those mean that you're probably going to want to run it through a small proportion of your users for a longer period of time.
4. Pre-periods and post-periods, which are uniformity trials. There's A/A test that we discussed back in lesson three. But what we're doing is instead of using it across the entire system, we're using it in a way that's specific to your experiment and your control. And so, what happens is that before you run your A/B test on your experiment and control, and you have those populations, you're on a pre-period on the exact same populations but they're receiving the exact same treatment. It's an A/A test on the same set of users. And what happens in the pre-period is that if you measure any difference between your experiment and your control populations that difference is due to something else. Maybe system variability, user variability, things like that. Now a pre-period I would note, is useful not just for when you want to test user learning, but sort of across the board. So that you know that any difference that you measure in your experiment and control is due to the experiment, and not due to any preexisting and inherent differences in your population. Now, that's what a pre-period is, and that basically says, okay, I don't have any differences in my populations. A post-period is saying, after I run my experiment, my control, I'm going to run another A/A test. And then, what, what we can say is that if there are any differences in the experiment and the control populations after I've run my experiment, then I can attribute those differences to user learning that happened in the experiment period. And so, that's what we basically do. Now, the key thing that I sort of note is that these are pretty advanced techniques. If you're really trying to measure user learning, hopefully you've run tens, if not, hundreds of experiments already. If not, I'd probably stick to some of the simpler techniques.