Overview of A/B Testing

Customer funnel:

Homepage visits → Exploring the site → Create account → Reach some sort of completion

Audacity example: change the "Start Now" button from orange to pink

Metric:

- Total number of courses completed? No. Using this metric would simply take too much time to be practical
- Number of clicks? No. What happens if more total users view the page in one version of the experiment?
- Click-through-rate: number of clicks / numbers of page views. No. It is used when you want to measure the usability of the site. It tells how often the users find that button.
- Click-through-probability: unique visitors who click / unique visitors to page. It is used when you want to measure the impact of the site. It tells how often the users went to the second level page on the site.

Hypothesis: changing the button will increase the click-through-probability of the button.

Click-through-probability： binomial distribution

- 2 types of outcomes (success, failure)
- Independent events
- Identical distribution (P same for all)

If we know the click-through-probability should follow binomial distribution, we can use the formula we have for sample standard error for the binomial to estimate how variable we expect our overall probability of a click to be. What that means is that for a 95% confidence interval, if we theoretically repeated the experiment over and over again, we would expect the interval we construct around our sample mean covers the true value in the population of 95% of the time.

The center of the confidence interval:

Estimated success probability: P(hat)= X/N.

A good rule of thumb to tell whether a binomial distribution is normal if the sample is large enough:

1. N*P(hat) > 5 (this's the more stringent condition for small probability problems); 2. N* (1-P(hat)) > 5

The width of the confidence interval if we can use the normal approximation:

The margin of error: m = z score of the confidence level * standard error

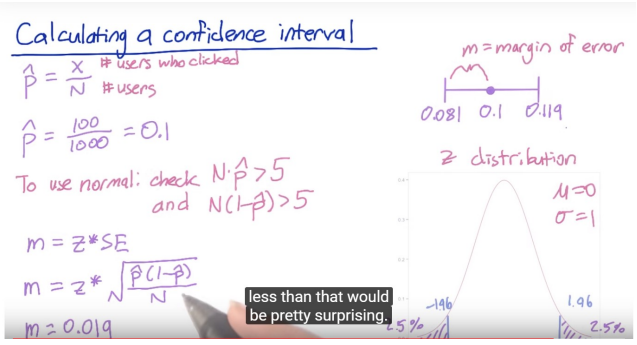
$= z * \sqrt{\frac{Phat(1-Phat)}{N}}$ (* standard error for binomial distribution)

Marginal error, which is the amount of random variation we expect in our sample and the width of the confidence interval, is a function of both the proportion of successes and the sample size. This means we need to consider the proportion of success when deciding how many samples to collect.

When the success probability p is farther from 0.5, SE and CI will be smaller, the distribution is tighter.

Similarly, if the number of samples is larger, the SE and CI will also be smaller.

For a normal distribution with a mean of 0 and a standard deviation of 1, with 95% confidence, the true value would be with 1.96 and -1.96 of the estimates we observed. Since we're doing a 2-tailed test, each tail will contain 2.5% of distribution. So, 1.96 is the z-score for 97.25%.



Hypothesis testing/ Statistical inference: a quantitative way to establish how likely it is that the results occurred by chance. We'll need to compare the proportion of clicks estimated on the control side and the experiment side and measure whether the difference we observed could have occurred by chance, or if it would be extremely unlikely to have occurred if the two sides were actually the same, which means that the difference is statistically significant.

In order to calculate the probability that your results are due to chance, you need to have a hypothesis about what the results would be if your experiment had no effect.

Null hypothesis (baseline):

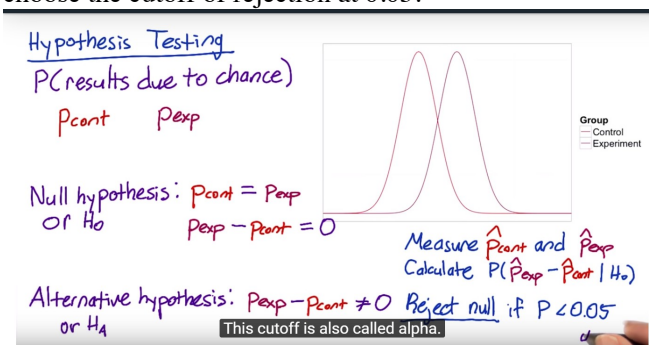
There's no difference in CTP between our control and our experiment, which means we would expect the 2 groups to have equivalent distributions, so they would be right on top of each other. Pcont – Pexp = 0

Alternative hypothesis:

The experiment does have an effect. We expect Pcont – Pexp != 0

We can estimate Pcont and Pexp from the data we collected and then calculate the difference between these (Pcont - Pexp) and compute the probability that this difference would have arisen by chance if the null were true (P(Pcont – Pexp | H0)).

Then we want to reject the null, and conclude that our experiment has an effect if this probability is small enough. We can choose the cutoff of rejection at 0.05.

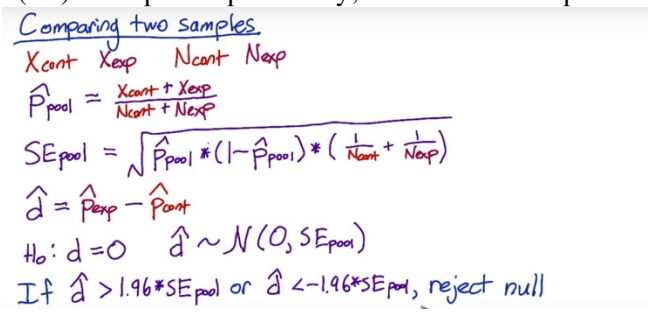


Because we have 2 samples, we'll need to choose a standard error that gives us a good comparison of both. The simplest thing we can do is calculate a pooled standard error.

X: the number of users who click in each group

N: total number of users in each group

P(hat) is the pooled probability, which is the total probability of a click across groups



We would expect our estimation of difference between groups to be distributed normally, with a mean of 0 and a standard deviation of the pooled standard error.

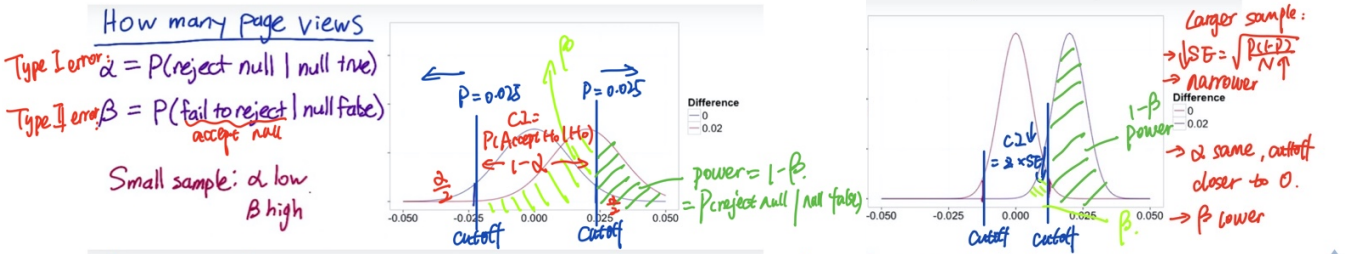
Then, we should decide what change in the probability is practically significant from a business perspective. What size change matters to us?

What you really want to observe is repeatability. And you want to make sure when you set up your experiment that you get that guarantee that these results are repeatable, so it's statistically significant. But you also want to make sure that if you can see change in your experiment that you're interested in from a business standpoint, so it's practically significant and also statistically significant. You need to size your experiment appropriately, such that the statistical significance bar is lower than the practical significance bar.

Design the experiment:

Statistical power is the main question we have to decide: given that we have control over how many page views go into our control and experiment, we have to decide how many page views we need in order to get a statistically significant result. If we see something interesting, we want to make sure that we have enough power to conclude with probability that the interesting result is statistically significant.

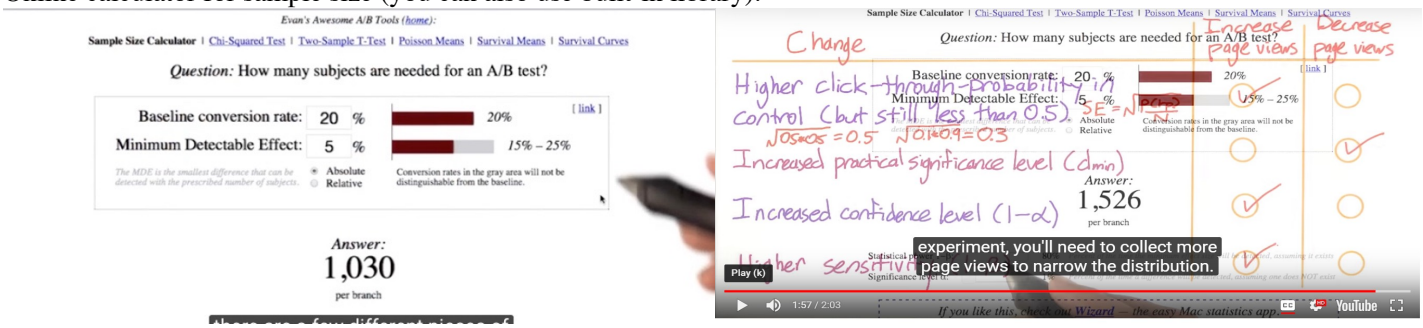
Power has an inverse trade-off with size. The smaller the change that you want to detect, or the increased confidence interval that you want to have in the result, means that you have to run a larger experiment, so more page views.



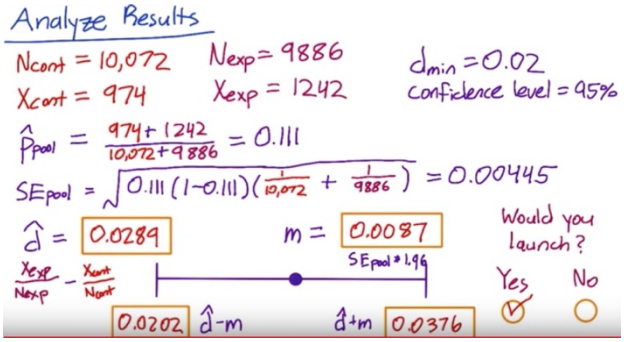
Blue curve is the distribution of results would look like if you collected 1000 samples and there was no true difference between the groups because of a 0 mean. Your probability of falsely concluding there was a difference is alpha: 0.05 (Type 1 error, false positive).

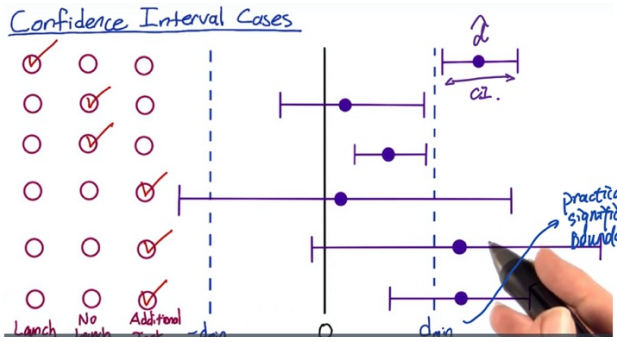
If you increased your sample size, the SE would decrease, so the distribution will look narrower. To keep alpha the same, the cutoffs for rejecting will be closer to 0.
Red curve refers to the case where there was difference. The probability of failing to reject the null when the null is false is beta (Type 2 error, false negative), which is pretty high with small sample size.
By collecting a small sample, alpha is low, that is, you're unlikely to launch a bad experiment. But beta is high, that is, you're likely to fail to launch an experiment that actually did have a difference you care about. For normal distribution, as your true change gets larger, beta will go down, that is, a lower chance of error.
1 – beta = = P (reject H0 | Ha) = sensitivity/power. In general, you want your experiment to have a higher level of sensitivity at the practical significance boundary, which is always set as 80%
For larger samples, alpha doesn't change. In the case where there's a true difference, you're much less likely to fall within the range of failing to reject the null, that is, you're more likely to reject the null and conclude there was a difference. Beta has gone down and power increased.

Online calculator for sample size (you can also use built-in library):



Baseline: the estimated CTP before making the change
Minimum dateable effect: practical significance level
#1. Probability level gets closer to 0.5 → sqrt(p*(1-p)) increases → standard error increases → means that I'd also need to increase the number of page views(N) in order to reduce the SE back to its original level.
#2. Larger changes are easier to detect than smaller changes, so do not need that many samples.
#3. Increase confidence interval – you want to be more certain that a change has occurred before you reject the null. More conservative – more samples needed (or you can reject the null less often, but power will decrease)
#4. Increase the power – need to collect more samples to narrow the distribution.





Case #2: Neutral case: CI includes 0. There's not a practically significant change.
Case #3: Upper bound of CI is less than practical significance, and CI does not include 0. The change is statistically significant, but the magnitude is not large enough for you to care about.
Case #4 - #6: We need to run additional tests with greater power to draw conclusions if time allows. If don't have time, we should communicate to the decision-makers when they're going to have to make a judgement, and take a risk, because the data is uncertain. They're going to have to use other factors, like strategic business issues, or other factors besides the data.

Metrics for experiments

Define the metric

Before you decide the metric, you should think about how you're going to use the metric. There're really 2 main use cases：

- Invariant checking metrics: these are the metrics that shouldn't change across your experiment and your control. For example, if you're running an experiment and a control, one major term of comparison is, are the populations the same? You're going to check number and distribution of users. All of these things are sanity check to make sure that your experiment is actually run properly.
- Evaluation metrics:
  - High level business metrics: how much revenue you make, what your market share is, how many users you have
  - Detailed metrics: the user experience with actually using your product, how long they stay on the page, finish a class, etc.

Steps for making a definition:

Step 1: high-level concepts

1. Come up with a high-level concept of that metric, like a one sentence summary, "active users", "click-through-probability", etc.

Step 2: Define the details

2. Figure out all the nitty gritty details. How do you define what active is? Is it a 7-day active or 28-day active? Which events count towards activity.

Step 3: Summarize all the individual data measurements

3. Summarize all the individual data measurements into a single metric, like a sum, count, or average, median.
   If you have multiple metrics, you can create a composite/combine metric, which stands for an overall evaluation criterion. This term is used by Microsoft for when they come up with a weighted function that combines all the different metrics.
4. Think about how generally applicable the metric is. Better to design a less optimal metric applicable to the whole suite of AB tests, than a perfect metric. You will introduce more risk when doing something custom.

Evaluation – single or multiple metrics??

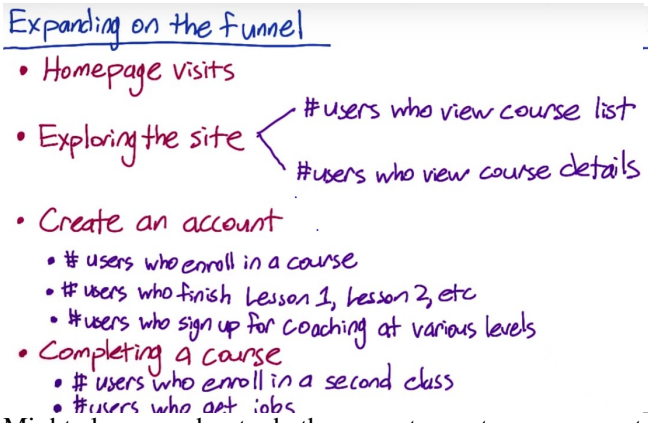
Depend on company culture and how comfortable people are with the data. The leader may be more comfortable with a whole suite of metrics where they can see things move.

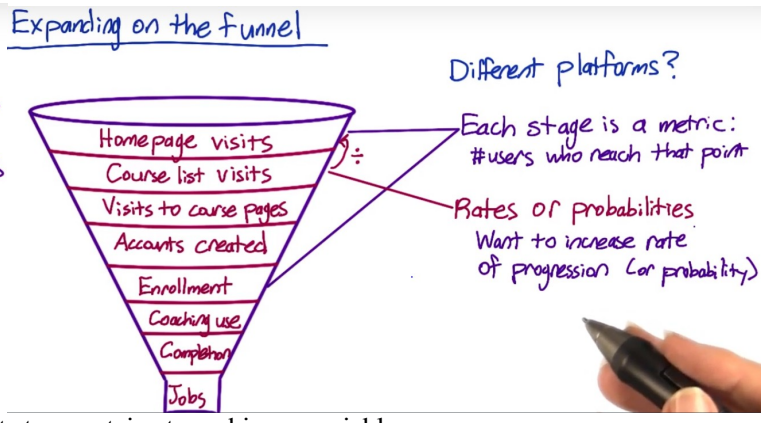
For PR purposes, external reporting, you may have to settle on a single overall objective. In large companies, people may want different teams to move towards the same goal, therefore in that case might want a single metric.

Do not suggest using composite metric because:

1. hard to define and agree on a definition
2. can run into problem if you over-optimize looking into one thing without looking at others
3. when the metric moves, people will come in and ask why it moves, and you have to go back and check individual metric anyway

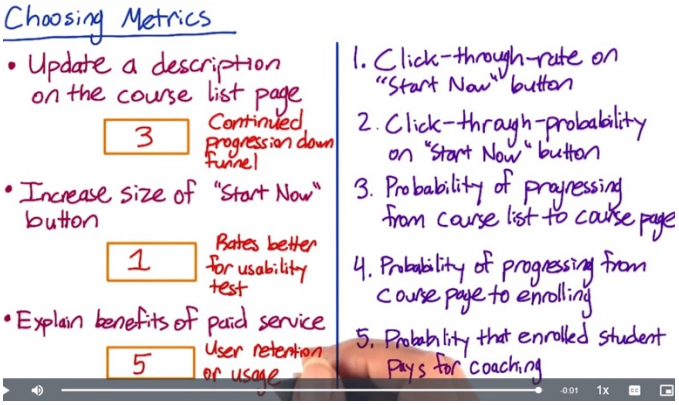
Refine the customer funnel



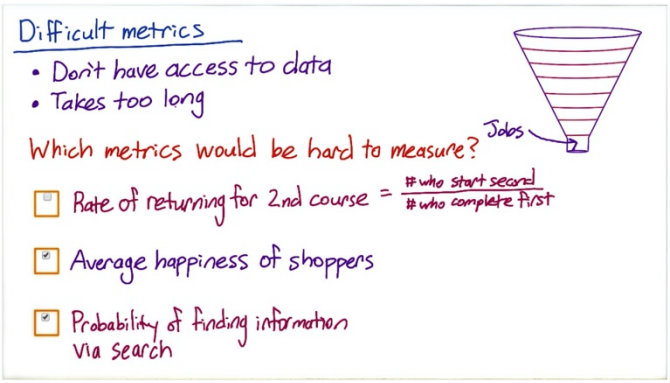


Might also care about whether or not a customer ever gets to a certain step – binary variable.

Probability – unique user that progress across the funnel.



1. The rate at which users progress from the top level to the second level. Rates are often better than probabilities for measuring the usability of a button, and increasing the size of the "Start Now" button is probably an attempt to increase the usability
2. The probability from the 1st step to the 2nd step
3. The probability from the 2nd to the 3rd, but should determine whether to measure a specific course page or any course pages.
4. Also should determine whether a specific course page or any course pages.



Has data, but too long

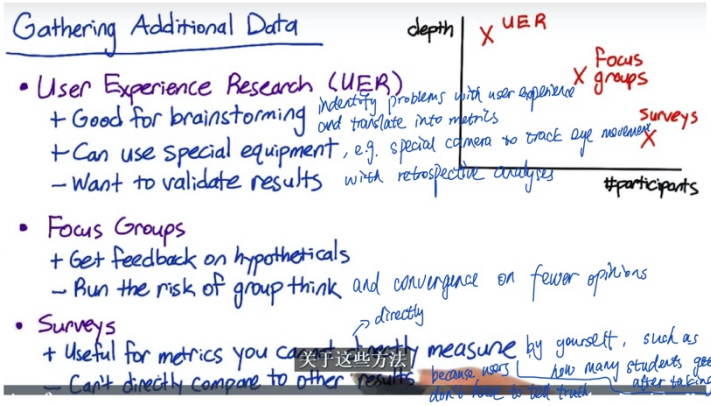
Doesn't have data

Doesn't have data

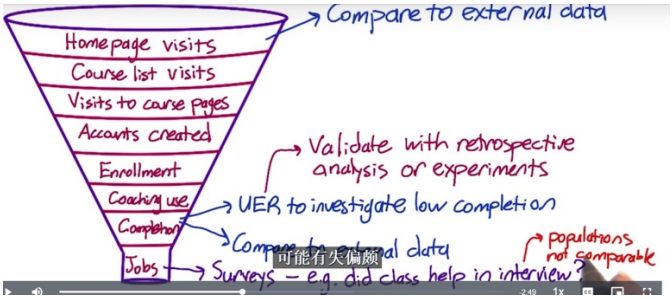
Techniques for generating new ideas of metrics and digging deeper into a user experience:
We can use different techniques, which ranges from surveys to retrospective analyses, to focus groups. These techniques can be used for both brainstorming new metrics, as well as validating possible metrics.

1. External data from:
   - Companies that collect very granular data like market shares, websites with visitor data, etc.
   - Companies run surveys of users about, like how many devices are you using? How much time are you spending on each of these devices?
   - Academic research to establish metrics
2. Use existing data including logging data, or data capture, from your site:
   - Retrospective or observational analysis with existing data to get a baseline and help you develop theories.
   - Use this data in conjunction with other methods, such as surveys or user experience research, to get correlation, not necessarily causation.
3. Gather new data to answer questions that you can't really answer from your existing data. Techniques include user experience research, surveys and focus groups
4. Talk to your colleagues about what ideas they think make sense for metrics. And think about company culture.

Gather additional data



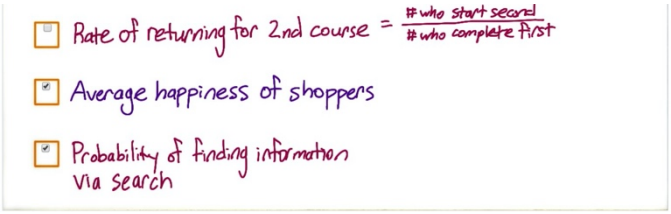
These additional data vary along two major axes:
Some give more in-depth customized data; some will be possible to run on a greater total number of participants
1. UER: you can go really deep with just a few users, by observing them doing tasks of interests, like taking a lesson of a course or asking them questions
2. Focus group: bring a bunch of users or potential users together for a group discussion. You can talk to more users than UER but can't go as deep with each person
3. Survey: you recruit a population and ask them a bunch of questions, either online, or in person, or via telephone. Surveys are pretty cheap to run on a whole bunch of users, and the data you get is much more quantitative, but it's not very deep or individually customized



UER for a particular course with low completion:
Watch the students complete the lesson – understand where to click? Can they find all the info on the screen? Do they follow the order? How are they interacting with the coach? Waiting for video to load – latency? Link for the additional materials used?
For potential metrics identified via UER – use retrospective analysis to examine how the metric varies over time, or run some new experiments to see how that metric vary as you make changes

Unmeasurable metric – Survey
e.g. send emails to ask whether interview questions were covered by the course before.
Surveys are very helpful for the metrics hard to measure, but cannot compare the numbers from the survey directly with numbers from the other measurement as the populations are not the same. Survey population might be biased compared to internal data.

Additional techniques for difficult metrics:

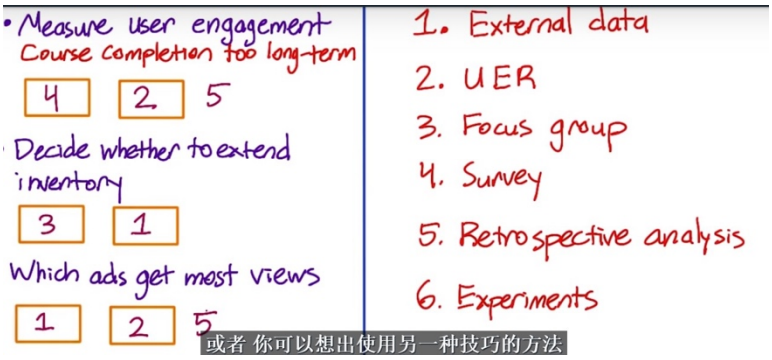


Rate of returning to 2nd course: Follow up with a survey to find out the reason to return to 2nd course. If something measurable that predicts the return, can use it as proxy.
Average happiness of shoppers: Find things correlated – survey at the end of purchase, or UER.

Searcher find information they were looking for?
Possible proxies:

- Length of time spent on search page
- Whether the client clicks on the results shown on the page
- Whether there are any follow-up queries to try to find information in different way

You can identify which of those proxies are more promising by looking at external data about information finding research or by running an UER study, or by human evaluation where pay human raters to evaluate your site



Student's engagement in class:
1. Survey: to ask students how engaged they are
2. UER: to observe how students interact with the course, and find whether the engagement correlates to something easier to measure, such as the time spent on a page, or clicking more links on extra materials
3. Retrospective analysis: to see whether user behaviors are correlated with student's engagement from history data
Ads:
1. Check external research and find proxy
2. UER: Use special camera to observe which ads they are paying attention to
3. Clicks relative to views: use clicks / or the lowest position that was ever clicked. And do retrospective analysis to find correlated factor.

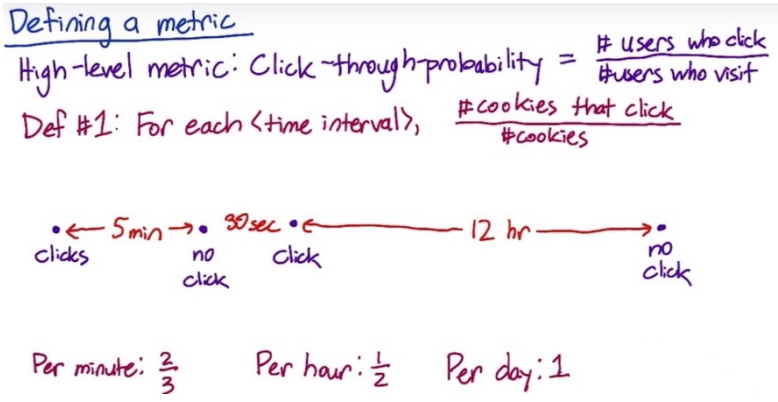
Build intuition about your metric, your data, and your system. A good data analyst should be able to understand what changes in your data and metric, your system can produce? You need to first decide, given the events that we observed, which are the ones that should count for those metrics and how to combine them (e.g. numerator, denominator)
For example: click-through-rate

- Total number of clicks/ total number of views
- A more nuanced version for probability: there's something called a cookie which is an anonymous identifier for a user. What we can do instead is say, did a cookie visit the site, and then, given that a cookie visited, did they click or not?

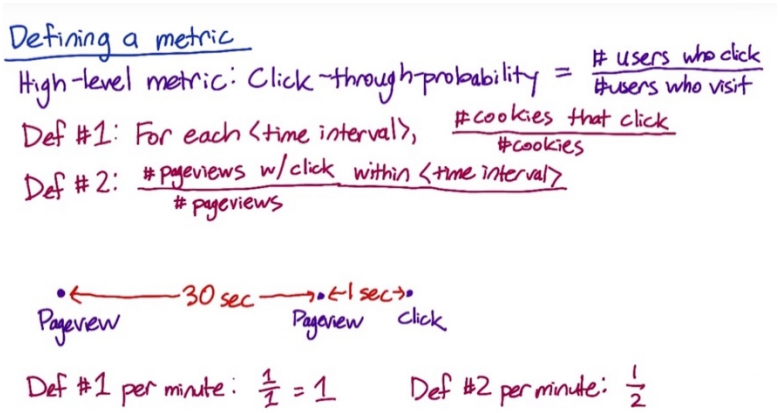
Need to worry about a bunch of other detailed things that come up. For example:
If you have a page load but no click. Then a day later, the same cookie comes back, load the page, waits 15 mins and then clicks – do you consider them all being associated with the same record?

You may want to plot your data over the course of a day, look at evening, weekday effects. Look at the minute or the hour, where things are happening. What happens if someone's page load and click are around midnight and fall under two days, do you consider them as same day event or separately?
You should also consider the technology that used when you're actually capturing the events. As an example, JavaScript pings are one of the most common ways of capturing clicks. The issue is that certain browsers don't implement JavaScript at all. And other browsers will have different failure rates on that JavaScript ping back. And so, what happen is that as you look at different browsers or different platforms, you'll actually get different click-through rates as the technology used to gather the clicks are different. You have to really work with your engineering team to understand all of those nuances, understand when you have a real difference, versus the difference through the underlying technology.

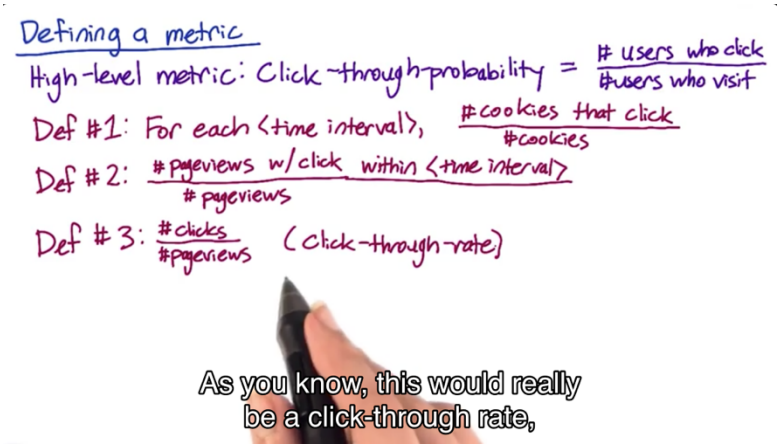
Metric definitions



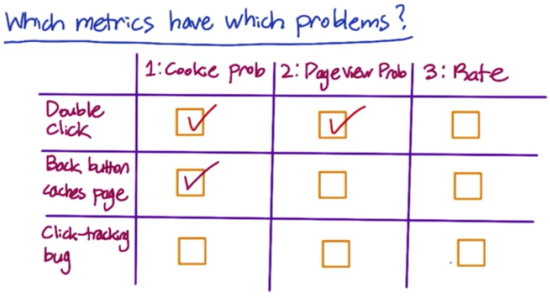
To use this definition, we need some way of determining whether 2 events are from the same user. Let's say we use cookies. Next, if the same user or cookie visits the page once and then comes back a week or two later, do we really only want to count that once? Usually, you'll want to count those visits separately, which means you'll also need to choose a time period. Do you only count one-page view per user each minute, hour, day, or what? So, one fully specified definition would be that for each time interval, you take the number of cookies that clicked during that time interval divided by the number of cookies that interacted with the page at all during that time interval.



An alternative definition would be to remove the idea of a unique user and instead create a unique ID for each page view. When a user clicks, record the ID of the corresponding parent page view. Then you could define the click-through-probability as the number of page views that eventually result in a click within the time interval, divided by the number of page views. This data capture is usually easier then recording cookies and grouping by cookies.
If a user refreshes the page within the given time period, Def#1 and Def#2 would give different results, e.g. if the user refresh after 30 sec



An even simpler definition would be to count the total number of clicks and divided by the total number of page views. This would be a click-through-rate.



Often, Def#1 and Def#2 will be almost indistinguishable if you choose the same relatively short time interval. So you might want to go with Def#2, since it's easier to compute.

Filtering and Segmenting

External factors to consider:

You often see, sort of abuse on your site such as spam or fraud, and you want to try to filter that out

e.g. if you have a competitor, who's looking for your site clicking on absolutely everything, and you may not want to use that data in your experiment.

e.g. you may even have some malicious trying to mess up your metric

e.g. if you get blog coverage for your experiment, you could potentially get a lot of traffic that coming to look at the experiment need to at least flag and identify these issues, and eventually filter them out

Internal factors to consider:

Some changes only impact a subset of your traffic. For example, maybe you didn't want to internationalize your change and so it only impacts English Traffic. Or maybe it only impacts the mobile app version and not the web version. If you don't want to dilute your result, you need to filter only the affected traffic and then increases the power and sensitivity of your experiment.

The goal of filtering is to de-bias or dull-bias the data. So, you should be careful you don't introduce bias into your data.

e.g. if you have a metric that can only be measured on logged-in users, you might actually be biased in your data because there's a bunch of noncommittal or newer users trying to use the site who maybe haven't created an account yet.

e.g. if you want to filter out some especially long or weird sessions of user behavior, you should check and make sure it's not actual your website, your metric or even your logging that's causing these sessions to come up.

How to tell if the data is biased or not?

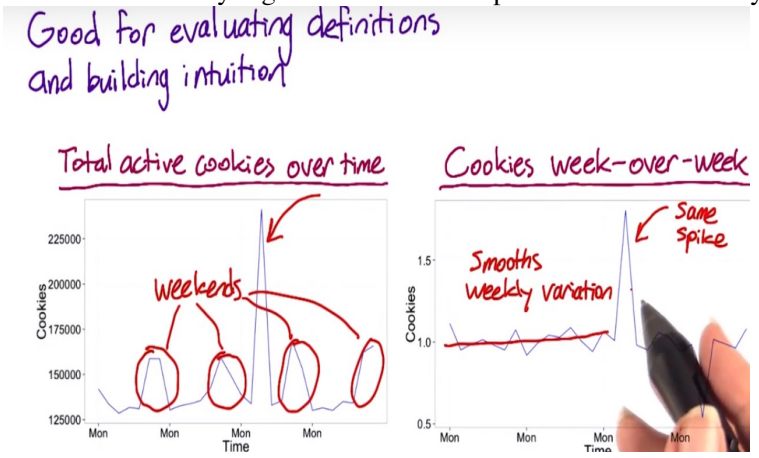
Realistically, in most cases, you're computing a baseline value for your metric.

One way is to slice your data. You compute the metric on a bunch of disjoint segments (e.g. country, language, platform). And then when you look at the filter, what you want to see is whether or not you're moving traffic disproportionally from one of these slices or not.

If it is, it makes sense because let's say all of your spam coming from a certain country. But if you're actually moving disproportionately from one of these slices, it may be an indicator that you're actually biasing your results further.

Look at Day over Day or Week over Week traffic pattern changes to identify things that are unusual, such as spam or fraud.

Looking at different segments of your data can be useful for evaluating metric definition as you can look at how the different definitions varies by segments. This can help build intuition about your data and system.



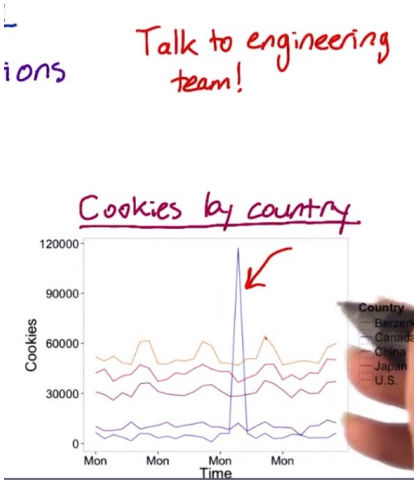
One way I can verify whether the spike is odd is by looking at what's called a week-over-week plot. That is, I'll divide each data point by the corresponding data point from a week ago. As you can see, that tends to smooth out the weekly variation. I can see looking at this plot that it would stand out if one of these total cookies in the left plot is abnormally high given what day of the week it was.

Since we see the same spike in the right plot, that makes it clear that this spike is not due to weekly variation. The coming corresponding drop a week later happens because we divided that data point by the spike a week earlier.

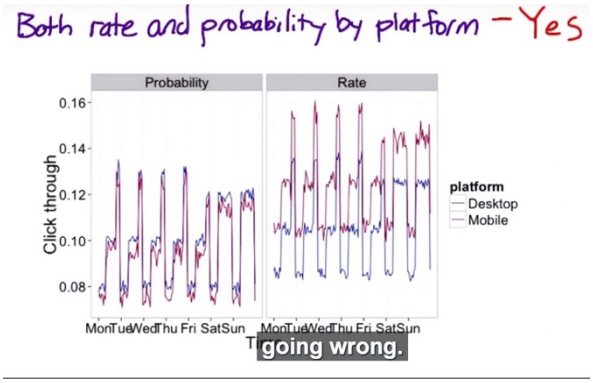
Another thing to look at is year-over-year data. If there's an annual conference or something causing the spike, it will disappear.

However, in this plot, the same spike is still here, meaning it's probably not due to a yearly variation. You can also see that the weekly variation is back, since the day of week is not quite matched up to the day of week from a year ago.

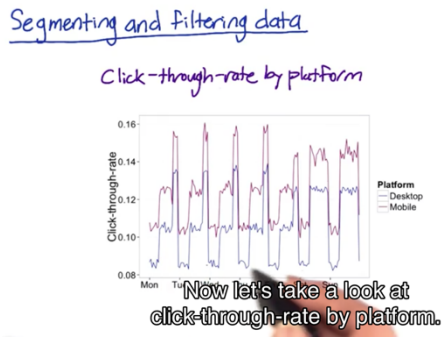
Now the question is, if we can pin down what's causing this spike, since it doesn't seem to be caused by either weekly or yearly variation. One way we can figure it is by looking at different segments of our population to see if one segment is causing the spike.



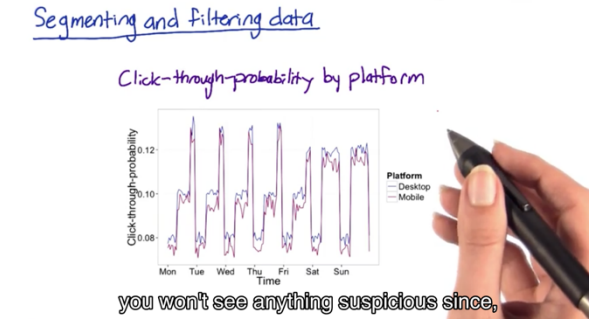
So, let's trying looking at how this metric varies by country. We don't see the spike in most countries but we do see it in Berzerkistan, so that one country was causing the entire spike.
At this point, it's a good idea to talk to the engineering team, and maybe they'll be able to figure out if this spike is in fact caused by only a small number of rogue IP addresses. And this is pretty likely to be spam, or a row grow bot, or some competitor trying to get information.



Now suppose that you suspect there is an issue with JavaScript click tracking. Specifically, you're worried that JavaScript is counting each click event twice on mobile but not on desktop. This graph of both rate and probability by platform can tell you whether the problem exists. The click-through-probability is slightly lower on mobile but the click-through-rate is significantly higher. This point is pretty clearly to some sort of instrumentation issue. Only this graph really made it crystal clear that there was a problem.



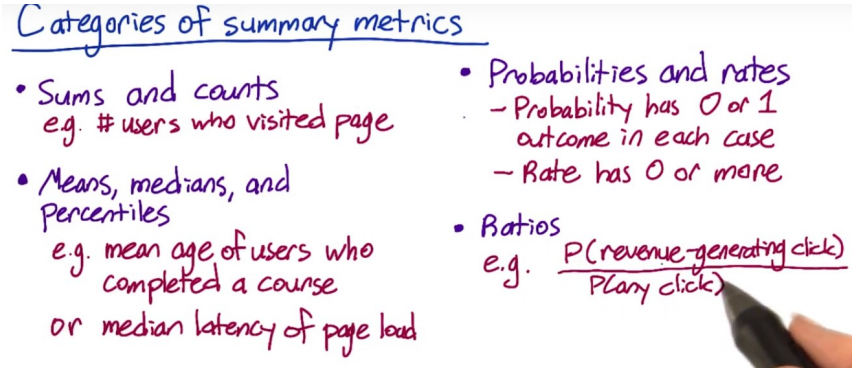
The click-through-rate by platform graph is suspicious, because it shows the click-through-rate being higher on mobile than on desktop. But users will have different behavior on desktop than on mobile so, it's still not really clear that this is a problem with the JavaScript tracking and not just a difference in user behavior. It's hard to draw a conclusion.



In the click-through-probability by platform plot, you won't see anything suspicious, since if JavaScript does send a duplicate ping. The click-through probability will eliminate that, collapsing it into one. So here we see probability is pretty similar between desktop and mobile.

Now we've gone over some techniques for getting to a high-level concept for a metric and then translating that into a specific data measurement, and also evaluating possible filters. And then we're going to summarize all the individual events of our direct data measurements (e.g. page view, click measure if latency, etc.) into a single summary metric.

The summarization is actually part of the metric definition. But there's a whole bunch of other cases where you actually have a choice of summary metric. The primary situation that occurs is when your per event measurement is itself a number. And this is something like the load time of a video, or how many terms are in a query, or what the position of the first click on the page is. When you have a number like that, you have a whole set of metric s to choose from.



Four broad categories of summary statistics
(1) Sums and Counts – e.g. how many cookies visit the website
(2) Distributional metrics - e.g. means, median, 25th percentiles, 90th percentile.
(3) Rates or probabilities: probability: 0 or 1, rate: 0 or more (e.g. the user clicks 2 times or 4 times)
(4) Ratio. These are more general than rates, which are 2 counts divided by each other. Business metrics often make sense as ratios. They can compute a whole range of different business models, but it's very hard to categorize

Mean: 1.3, median: 0.9. This is an example of an exponential distribution. If you're thinking about something like, how many users really get information from this page, you might want to use something besides the median or the mean. Maybe the 75th percentile or the 90th percentile.
In addition to these possible metrics, you could take this one step further. Let's say you find out, in some UER studies that, it would take the average person at least 3 seconds to read most of the content on the page. Based on this, you could use as your metric the percent of users who spend at least 3 seconds. In this case, that would be about 11% of the data points that stay about three seconds or longer.

How to choose between these different options?
You're going to establish a few characteristics for your metric.

- The first one is going to be the sensitivity and robustness. You want your metric to be sensitive enough, in order to detect a change when you're testing.
- The second that you're going to characterize is what the distribution of your metric looks like, and that's going to help you choose, you can do a retrospective analysis, and to compute a histogram. On the x-axis, you have all the different values for your metric. So, for example, you're going to have all the different values for load time on the x-axis. The y-axis is going to be the frequency. So how often individual events have that particular load time. If the distribution is a very normal shape, then a mean or median's going to make a lot of sense. As it becomes more one sided, or lopsided, you might want to go more for a 25th, or a 75th, or a 90th percentile.

Sensitivity and robustness
Metric should pick up the changes you care about (sensitivity), and does not pick up the changes that you do not care (robustness).
For example, on our latency example where we're looking at the load time of a video, you may use neither mean or median.

- Mean is sensitive to outliers. So, if in your data you see a lot of cases of really long load times, maybe due to something going on in the users machine, or a bad network connection, then you want to maybe not choose the mean, because the mean is going to be pretty heavily influenced by those types of observations. And so that's called not being robust.
- Median tends to be much more robust to that type of behavior, but if you only affect a fraction of your users, even if it's a fairly large fraction, like 20% with a change, you might not see the median move at all. So the median is robust.

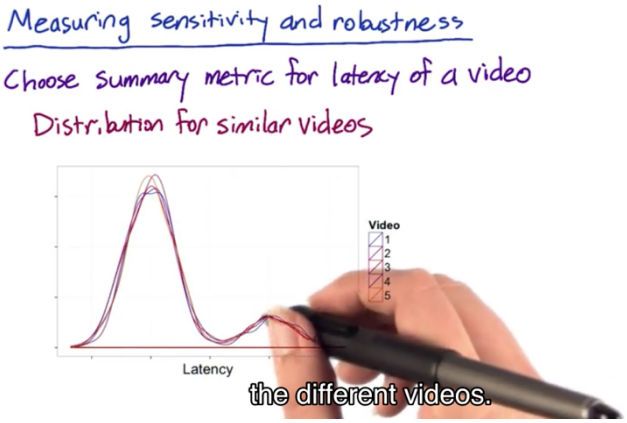
But in this case, you might want to actually consider using some other statistics, such as the 90th or the 99th percentile, and see how those change as well.

How would you measure the sensitivity and robustness?
A. Experiment

#1. Run experiment or use experiments already have
e.g. latency – increase the quality of video (increase the quality of the video to increase the load time for users), and see if the metric responds to that. We should be able to tell if they're actually moving in a way that intuitively makes sense.
#2. A/A experiment to determine if it's too sensitive.
That's an experiment where you don't change anything. You just compare people who saw the same thing to each other. See if your metric picks up any spurious differences between the two. Make sure that you're not going to be calling things significant that maybe don't really mean anything.

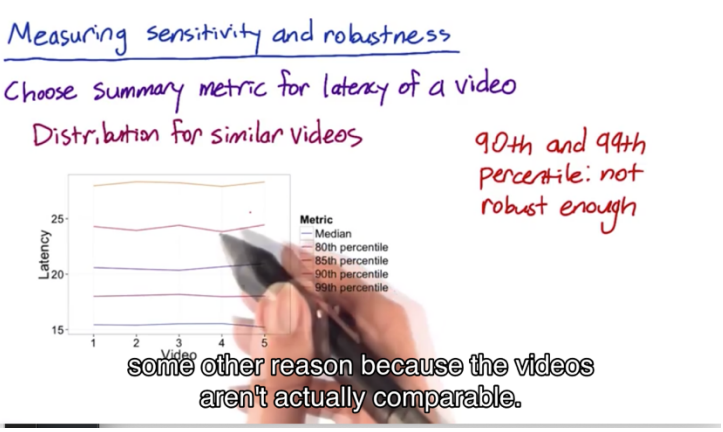
How to measure the sensitivity and robustness of some different metrics:
For example: video latency

A. Retrospective analysis
Look back at the changes on your website, and see if the metrics you are interested in move in conjunction with these changes. Or can look at the history of the metrics and see if there is anything that causes these changes.



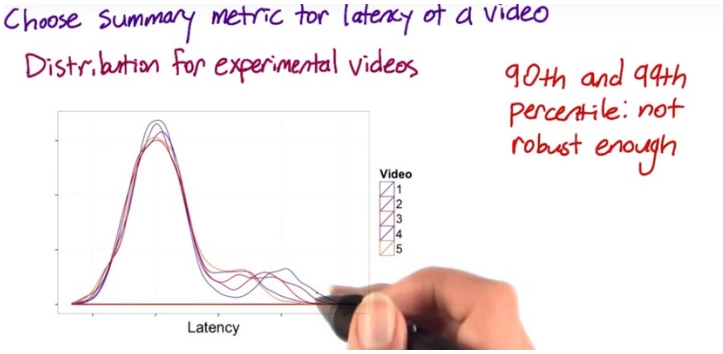
Let's do the retrospective analysis first by segmenting the data by different videos. In other words, look at the distribution of load times per video.
You can see two peaks here, a fairly long load time, and then more people with a shorter load time. This could happen if you had people with different types of Internet access, a slower Internet access and a faster one.

Now, in order to characterize the sensitivity and robustness of different summary metrics, I can see how they vary across videos.



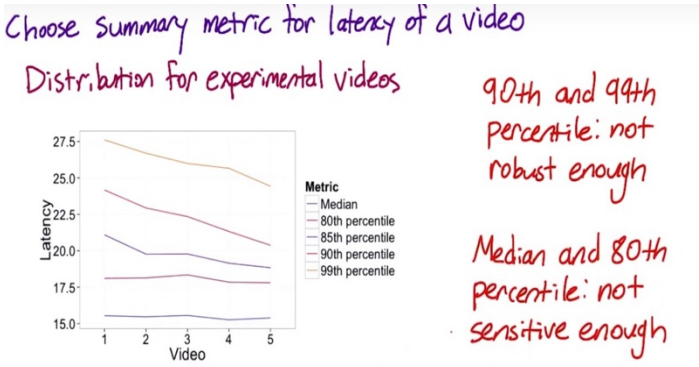
So, here I've plotted a few different summary metrics by video. In theory, since these videos are all comparable, there should not be too much difference between the different videos for a good metric.
Here, you can see that the median, the 80th and the 85th percentile don't move around too much. They're pretty good. But the 90th and the 99th percentile are zigzagging around a bit. This is a good indication that the 90th and 99th percentile are not robust enough as summary metrics, since they're moving around quite a bit, even for videos that are pretty comparable.
Of course, you have to be careful. Maybe these metrics are moving around for some other reason because the videos aren't actually comparable. For example, maybe the videos are at different resolutions or have a different encoding scheme.

B. experiment
You can look at actual experiments, preferably experiments you've already run to save yourself some effort, but you can also run new experiments.
For example: if we had experiments that changed the resolution. That should impact the latency, and if it doesn't, then our metric isn't sensitive enough.



Video one has the highest resolution, which means that it should have the highest load time. And in fact, you do see that video one is off to the right a bit more. You can also see that the people who already have the slow Internet connection are a lot more affected by the resolution than the people with the faster connection type.

Now let's also look at the same summary metrics for these experimental videos.



What we should see is the latency going down as we increase the video number, that is, we have a lower resolution. For the 85th, 90th and 99th percentile, we do see that, but for the median and the 80th percentile, they don't really seem to be moving. This is a good indication that the median and the 80th percentile are not sensitive enough. They're not showing a change when we do make a change that we care about.

So in this case, the 85th percentile might be a good choice.

How are we going to compute the difference between the experiment and control?
What we have is, we have a value for your experiment, and you have a value for your control. But we have to actually decide, how are you going to compute the comparison between the experiment and the control?

- Absolute difference:
  The simplest way is just to take the difference. And, if you're just getting started with experiments, or you're building up your knowledge of a whole bunch of different metrics, that's probably the way to go.
- % change advantage:
  But if you're running lots of experiments, you may want to consider computing the relative change, as opposed to the absolute change. In other words, the percent change. Now, the main advantage of computing the percent change is that you only have to choose one practical significance boundary, to get stability over time. Now, the main situations that I really see this being applicable are basically with regards to seasonality, and as your system is changing over time.

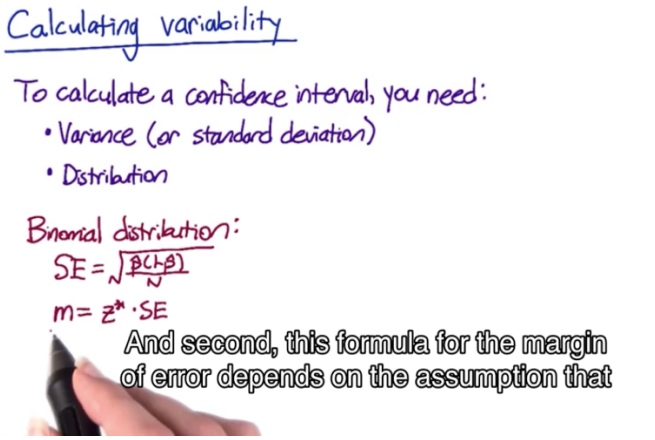
e.g. for a shopping site, in June, most people are on vacation, they're not shopping a lot. So, you have fewer users, you probably have a lower click-through rate. Whereas, in December, you've got loads of users, and a much higher click-through rate. If you have the same practical significance boundary, and across the same times, you can basically have the same comparison.
e.g. if you're actually running lots of experiments, and your system is actually changing over time, your metrics are probably changing over time as well. Again, if you're using the relative difference, you can stick with one practical significance boundary as opposed to having to change it as your system changes.

The main disadvantage is really variability. Ratios, such as relative difference, are not always as well behaved as absolute differences. So, if you're just starting out with this, or if you have some metrics you don't understand that well, it's often good to start with the absolute difference, and then work your way up.

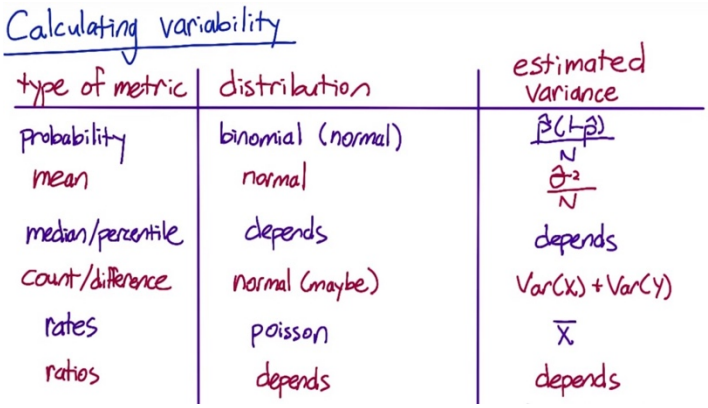
Variability
We're going to need a really more rigorous statistical definition of variability, so that we can use it to look at sizing the experiment, and to actually analyze the confidence intervals, and draw conclusions.
We also want to check that the practical significance we choose is realistic for our metric – if we have a metric that varies a lot under normal circumstances, that may not work because the practical significance is just not feasible for the metric.

- For nice normal data, like demographic data, you have counts or probabilities, then usually, you can do the confidence interval, theoretically.
- If you move on to using ratios or percentiles, like the 90th percentile, or if your data, like our latency data, is pretty lumpy, then you probably want to actually compute the variability empirically,

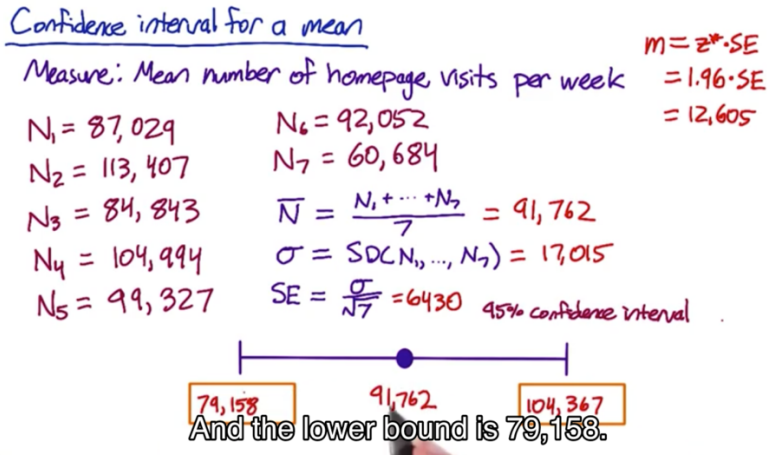
We've looked at a lot of the distributions you might see in your data and used that information to choose a specific summary metric or to understand the sensitivity and robustness of your summary metric. But now, it's time to add a level of rigor.



In order to calculate a confidence interval for your metric, you'll need to know the variance (standard deviation) of your metric and its distribution.
In binomial distribution, we used the fact that this was a binomial distribution in two ways. First, we use the fact that this was a binomial distribution to get this formula for the standard error. And second, this formula for the margin of error depends on the assumption that this is a normal distribution. The binomial approaches a normal distribution as N gets larger.



1. Probability metric: assume a binomial distribution, which approximates a normal for a large enough sample size.
2. Mean: by the central limit theorem, your metric will follow a normal distribution if the sample size is large enough.
3. Median/other percentile: if the underlying data is normal and the sample is large, then the median will be approximately normal. If the underlying data is not normal, then the median might not be normal either. You'd need to make an assumption about how the underlying data was distributed.
4. Count/the difference between two counts: For things like demographic data, it will always be normally distributed.
5. Rate: have more unusual distributions, Rates tend to follow a poisson distribution and the variance of the poisson distribution is actually equal to the mean.
   For your experiment though, you would be interested in the difference between two rates. That is you would need to estimate the variance of the difference between two poisson distributions. Unlike for the normally distributed data, these difference in rates aren't likely to be either poisson or normal in distribution.
6. General ratios: e.g. you might want to use the ratio of click-through probabilities in your experiment and control group instead of the difference. The distribution and estimated variance of a ratio will depend on the distribution for the numerator and the denominator.



Non-parametric methods: analyze the data without making assumption on what the distribution is
e.g. sign test – run A/B experiment for 20 days. 15 days, experiment has higher measurement than control. You can use the binomial to calculate how likely it is to occur if there is no difference.

- Downside of doing this is that it doesn't help estimate the size of the effect. That is, you can't say, you know, I'm confident this is at least 2% change in my metric.
- Upside: it's pretty easy to do, and you can do it under a lot of different circumstances. So if you wanted to launch any positive change in your experiment, then you could figure out whether there was one using a sign test.

After you actually computing the variance empirically, from the sample data, you have 2 choices by looking at the summary statistics distribution:

- If it is nice and normal, use the normal distribution and normal confidence interval with the variance you estimated empirically
- Otherwise calculate the nonparametric confidence interval

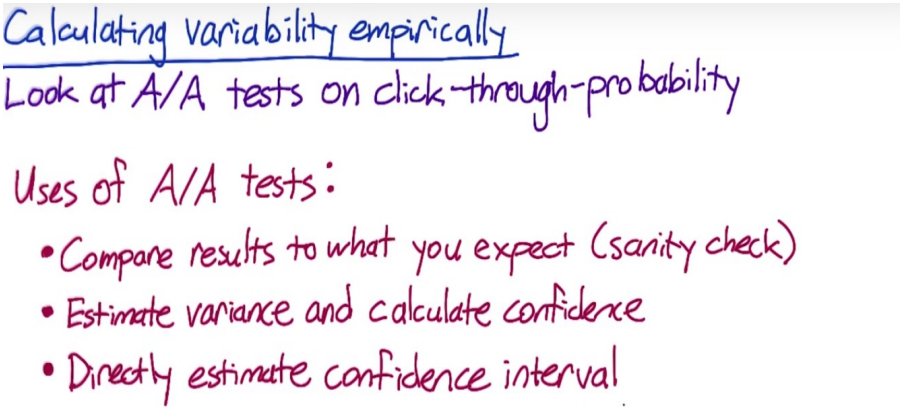
Empirical Variances:
For more complicated metrics, you might have to estimate the variance empirically than analytically
Other reasons to use empirical methods is that you're making assumptions for the underlying data distribution when computing the variance of a metric. This might work for simple metrics, but not for complicated metrics. And even for simple metrics, the variances could be under-estimated (refer to Lesson #5) by using analytical method.

Use A/A test to estimate the empirical variance of the metrics.
What you have is a control, A against another control A, and so there's actually no change in what the users are seeing. What that means that any differences that you measure are due to the underlying variability, maybe of your system, of the user population, what users are doing, all of those types of things.
If you see a lot of variability in a metric in an A/A test, it might be too sensitive to use in experiment.
So you can kind of pin down the variability with these A/A tests.

At Google we started with ten, then we moved to twenty. Now, we literally run hundreds of A versus A tests at a whole bunch of different sizes.
One of the biggest benefits of running a lot of different A versus A tests is because if your experiment system is itself complicated, it's actually very good test of your system. So, for example, is your randomization function truly random? Do you have any other issues with regards to bias or weird population effects?

The key rule of thumb to keep in mind is that the standard deviation is going to be proportional to the square root of the number of samples.

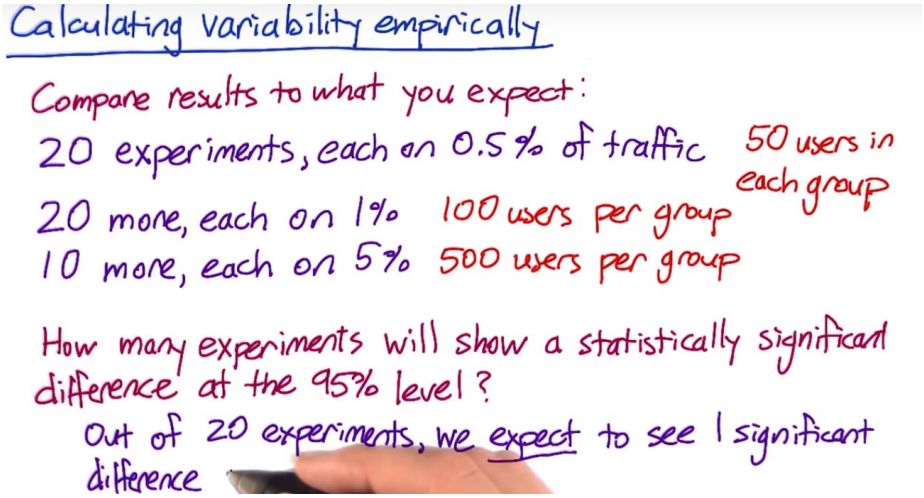
Now, in reality, what we have is a whole gradation of different methods:
If you're starting out and you're running your first experiment using a relatively simple metric, do the analytical estimate of your ants.
→
As you're starting to push towards more complicated metrics or you're running more and more features through, at that point, you might want to consider at least doing the bootstrap.
→
Now if your bootstrap estimate is agreeing with your analytical estimate, you can probably move on and you don't have to worry about it.
But if your bootstrap estimate isn't agreeing with your analytical, at that point you may want to consider running a lot of A versus A tests and really digging into understanding what's going on.



Since we've already done the analytics calculation for click-through-probability, we'll be able to compare the empirical results to the analytic results.
But you can also use A/A tests in cases where you weren't able to do an analytic calculation.

Use of A/A test:

- Firstly, if you already have an analytical calculation of confidence interval, you can check your A/A test results to see if you're getting what you expect. This functions as a kind of sanity check. If the results you get is not in line with expectation, this indicates that something is wrong with your calculations. Maybe you made an invalid assumption about the distribution of your data.
- Second, if you are willing to make an assumption about the distribution of your metric, but you weren't able to estimate the variance analytically, you can estimate the variance empirically, and then use your assumption about the distribution to calculate the confidence interval the same way we did before.
- Third, if you don't want to make any assumptions about your data, you can directly estimate a confidence interval from the results of the A/A tests.



(1) Example 1: Sanity Checking:

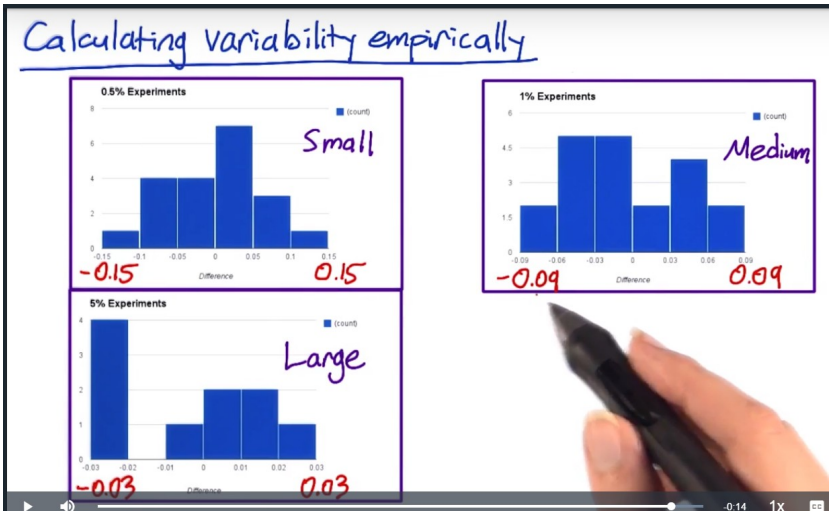
Now, let's say that this Google spreadsheet shows the actual results of the experiment. This column shows the click-through-probability measured for group one and this column shows what was measured for group two. If I scroll down, I can also see the results for the 1% experiments here and the 5% experiments at the bottom.

Based on the empirical data and the confidence interval derived from analytical approach, only one significant difference in 50 users experiment, and 0 in the experiments with 100 and 500 users. This is in line with expectation.

A	B	C	D	E	F	G	H
	Group 1	Group 2	diff	p_pool	SE_pool	d_cutoff	Significant?
				(B3+C3)/2	SQRT(E3*(1-E3)*(1/50+1/50))	F3*1.96	
0.5 % Experiments	0.1	0.04	0.06	0.07	0.05102940329	0.1000176304	
Standard Deviation	0.1	0.1	0	0.1	0.06	0.1176	
0.05921415194	0.04	0.12	-0.08	0.08	0.05425863987	0.1063469341	
	0.14	0.08	0.06	0.11	0.06257795139	0.1226527847	
	0	0.1	-0.1	0.05	0.04358898944	0.08543441929	YES
	0.08	0.16	-0.08	0.12	0.06499230724	0.1273849222	
	0.18	0.12	0.06	0.15	0.07141428429	0.1399719972	
	0.08	0.2	-0.12	0.14	0.06939740629	0.1360189163	
	0.08	0.08	0	0.08	0.05425863987	0.1063469341	
	0.12	0.16	-0.04	0.14	0.06939740629	0.1360189163	
	0.06	0.06	0	0.06	0.04749736835	0.09309484196	
	0.08	0.12	-0.04	0.1	0.06	0.1176	

Another thing we can check about the A/A tests is whether the differences follow the distribution we expect. We can derive this from the column which contains the difference between the two groups for each experiment.

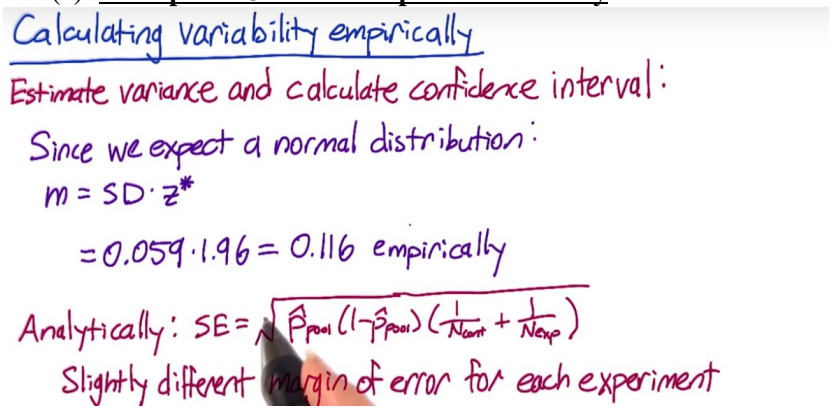
One thing to check is whether the differences are following a normal distribution as we expect.



For the smallest experiments, the distribution looks fairly normal, but for the other two it doesn't. However, I'd say that this is probably due to the fact that we didn't run that many experiments.

Another thing that these plots show is that the distribution is getting tighter as the sample size increases, which is in line with what we expected.

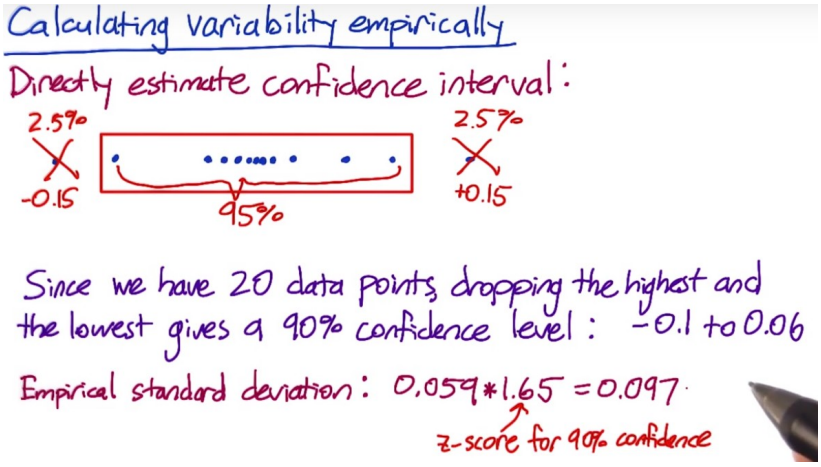
(2) Example 2: Calculate empirical variability



If we weren't able to calculate it analytically. I'll actually compute the standard deviation instead, since that's the direct analog of standard error. And, I do that by taking the standard deviation of each of the twenty differences from the smallest experiments. And, for that size of experiment, I get that the standard deviation of the difference is 0.059.
Now, since we expect that our metric follows roughly a normal distribution, we can compute the margin of error as the standard deviation we just calculated times the Z-square of our confidence level.

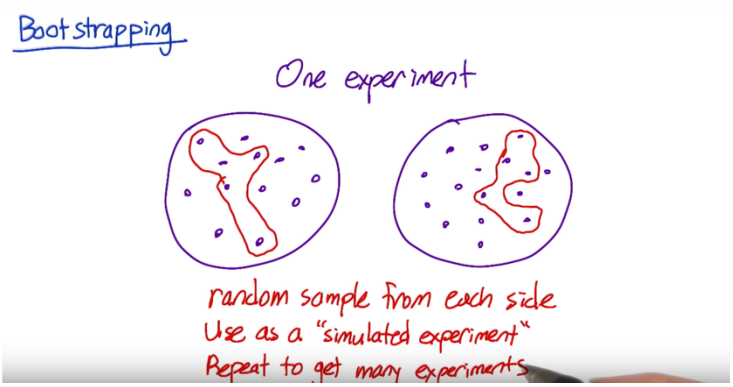
If you didn't know beforehand whether to expect your metric to follow a normal distribution, then you might look at histogram to see whether the metric look like it followed a normal distribution.
Remember, if we had done this analytically, the standard error depends on the pooled probability, which will be different for each experiment. That means we actually would have gotten a slightly different margin of error for each experiment. Whereas, empirically, we calculated one margin of error across all the experiments.

(3) Example 3: Directly estimate the confidence interval



The way to do this is take all your differences and put them in order. Then if you want a 95% confidence interval, select a box that includes only 95% of the values. That is discard 2.5% of your values on each side.
Then, the range of your remaining data points will give you a 95% confidence level.
Since we have 20 data points in our smallest experiment group, dropping the highest and the lowest difference give us a 90% confidence interval. Recall that the empirical standard deviation I calculated a minute ago was 0.059. Now, if I multiply that by 1.65 which is the z-score for a 90% confidence level, then that comes to about 0.097. So, if the true difference were 0 that would give us a confidence interval of negative 0.097 to positive 0.097. So, these two methods give a sort of close answer, but it's not that close. The main reason for this is that we only have 20 data points. You'd probably want to run more A/A tests to actually trust this confidence interval

Bootstrapping:



If you don't have enough traffic to run a lot of A/A tests – run one A/A test – although it is just one experiment, it is calculated from a lot of individual data points (individual clicks and page views). Take random sample of data points from each side of the experiment, and calculate the click-through probability based on that random samples as if it was a full experimental group. Record the difference in click through probability, and use that as a simulated experiment. Repeat this process multiple times, record the results, and use them as if they were from actual experiment.

Different metrics might have different variabilities. For some metrics, their variability is so high and it's not practical to use them in the experiment even if the metric makes a lot of business or product sense. To calculate the variability, we need to understand the distribution of the underlying data, and do the calculation by using analytical or empirical techniques.

Lessons learned:
For a lot of the analysts that I've worked with at Google, we spend the majority of our time actually coming up with, validating, and choosing metrics to actually use in evaluation. As opposed to evaluating the experiments themselves.

(1) Definitions and data capture

Just being able to standardize the definition was really important towards just being able to start the conversation at a whole different level.

- Click-through-rate: how hard can it be to calculate click-through rate? I mean, really.
- It's clicks divided by impressions or page views. Well this is the problem. Talking about impressions or page views? The first page of the search results, or all the next pages? Are you doing it in the US only or globally? Are you removing spam or are you not removing spam?
- Latency: when you say how long does it take the page to load, are you talking about when the first byte loads or when the last byte loads?

Need to agree on what metric you are using.

(2) Sensitivity and robustness

- Latency: latency tends to be really lumpy. And you look at the mean, and it doesn't move at all. And part of the reason is that you have users who have very different connection speeds. So you have a bunch of people who have super-fast speeds. You have people who have slow speeds. You have people who have some kind of problem on their computer, maybe they have an old browser.
  And so these signals that you're getting cause these sort of lumpiness in the distribution. You should think about do I have to use a higher percentile? Because I can't get the mean to move at all. One change effects the people who have the fast connections, and one change effects the people who have the slow connections, and I can't get any sort of central measure.
  So we spent a lot of time with latency, looking for the right higher percentile metric, that we could actually get to move, when we knew we'd done something that was positive for the latency experience.
- Search – tasks per user per day. A very stable metric. Does not change much with the experiments. What time period makes most sense? Per day or per week? Does your metric have a big weekly variability? If so, 28-day makes more sense than 30 days.

(3) Variability

- Good to start with analytical characterization of variability, and in some cases might be sufficient, But, if nothing else, it means that you have to look at the distribution, and start to get a feel for your data. Which is really important as part of this process. In some cases, like where you're using counts, or probabilities, or averages, your data is fairly nice, it may be sufficient. Or, it may give you a good sense of how to size your experiment. So, at least, you can tell if you're in the ballpark.
- It turns out that for some metrics, it was actually easier to compute it empirically as opposed to analytically, like revenue per query. And, once we're just computing that empirically, then we were like, well, we may as well try it for all the other metrics.
- The necessity of sanity check/invariability

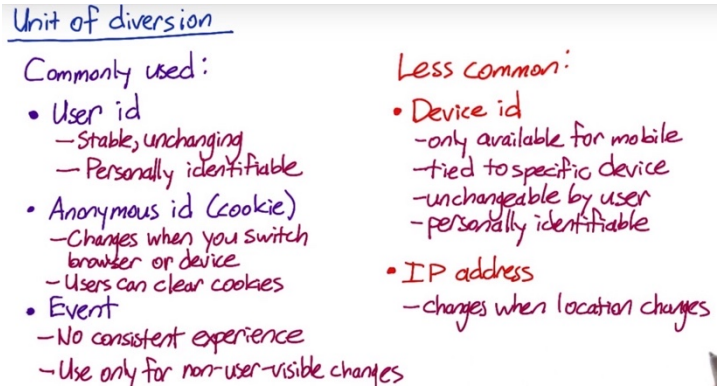
e.g. number of search results to show. Should keep latency as invariant as opposed to evaluation metric.

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.

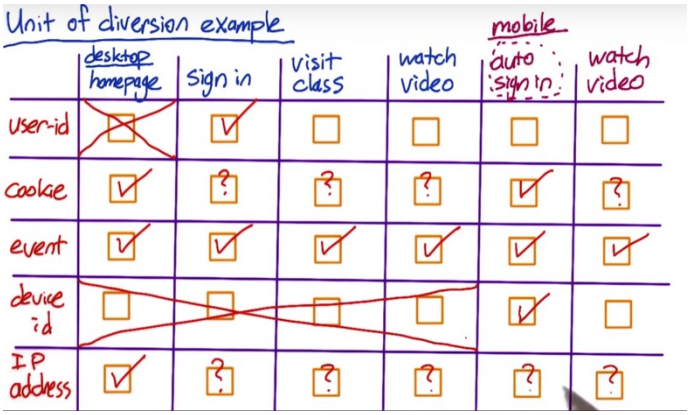
3. Event-based diversion:

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.

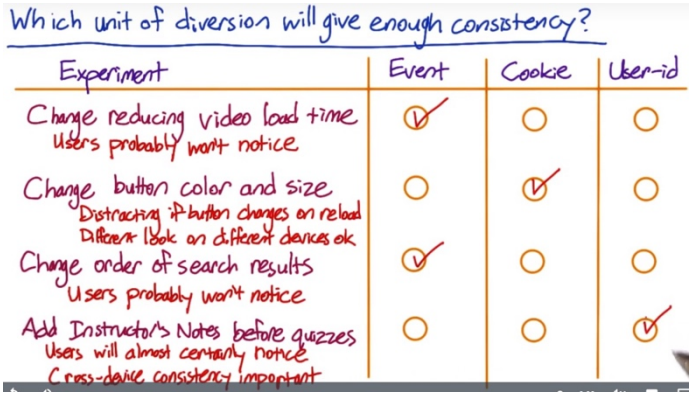
2. User Visibility:

- 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.

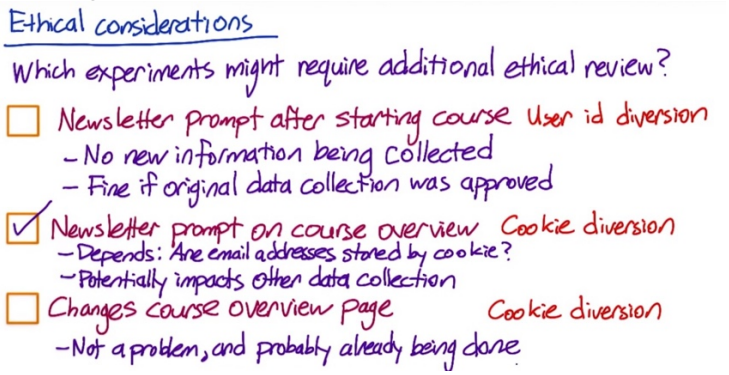
IP-based diversion
IP base diversion is not very useful generally speaking. You don't get the consistency because user's IP address could randomly change depending on the provider nor do you get the clean randomization that you get from event-based diversion.
There's a whole host of changes where IP 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

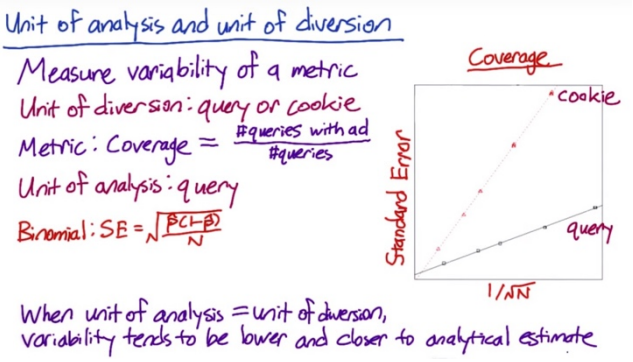
Unit of analysis is whatever the denominator of your metric is. And when your unit of diversion is the same as your unit of analysis, the analytically computed variability is likely to be very close to the empirically computed variability.

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.

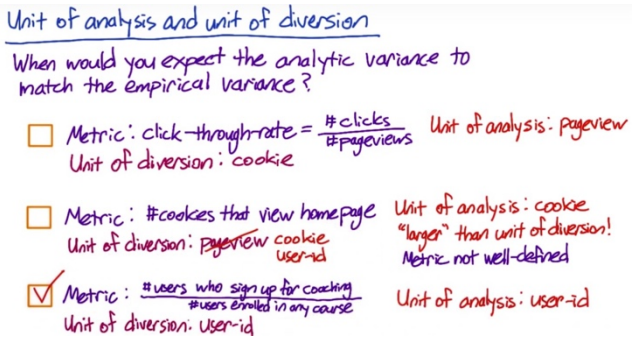


An experiment done by Google:

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.

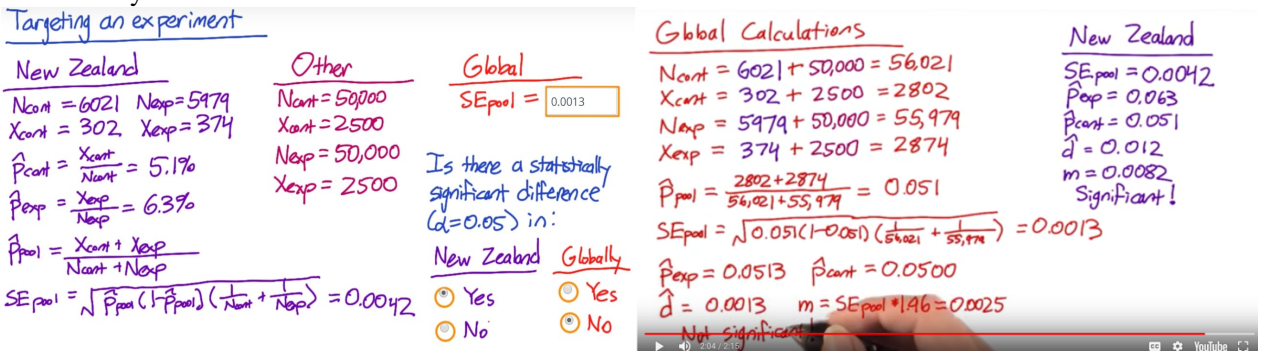
What you need to do to decide your targeting population:

- 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

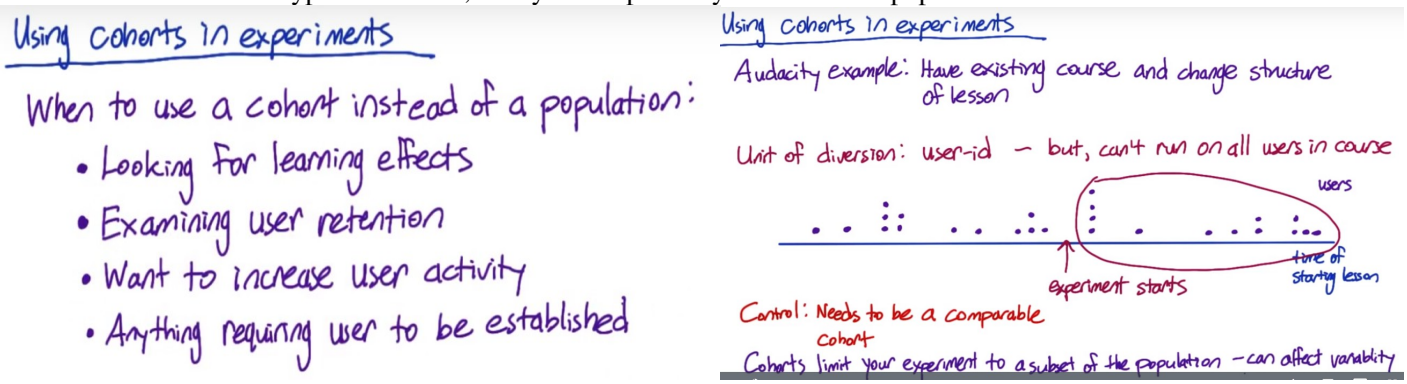
practice, your data will actually be a mix of different populations almost every time. When you filter, you're going to get a smaller but also more uniform population. Which means that for the same number of data points, the variability of the filtered data is likely to be lower.



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.

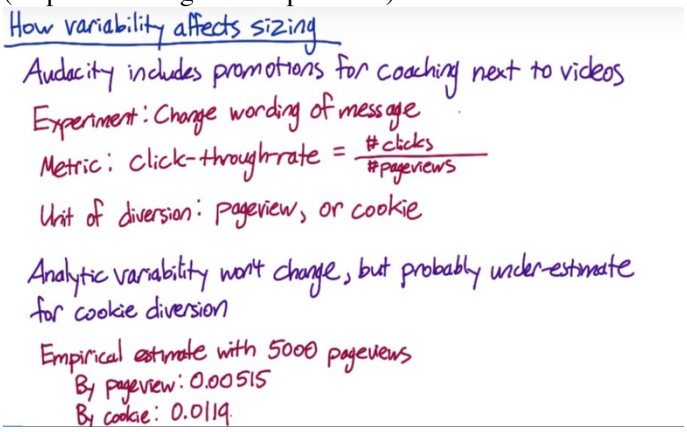
So, instead, Audacity will need to split this cohort into an experiment cohort and a control cohort, so that they all have the same timing of when they started the lesson.

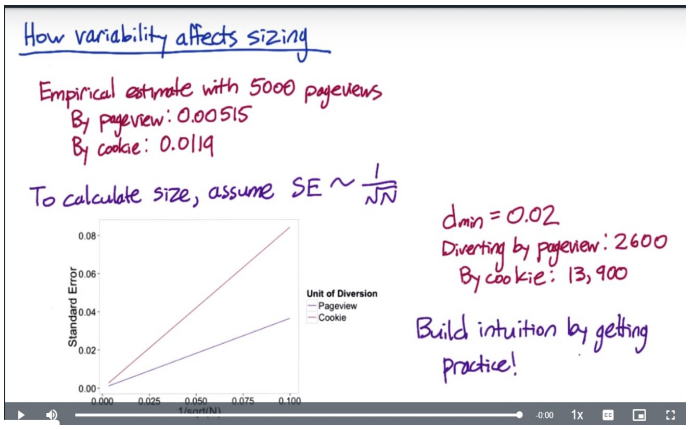
Sizing:

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.

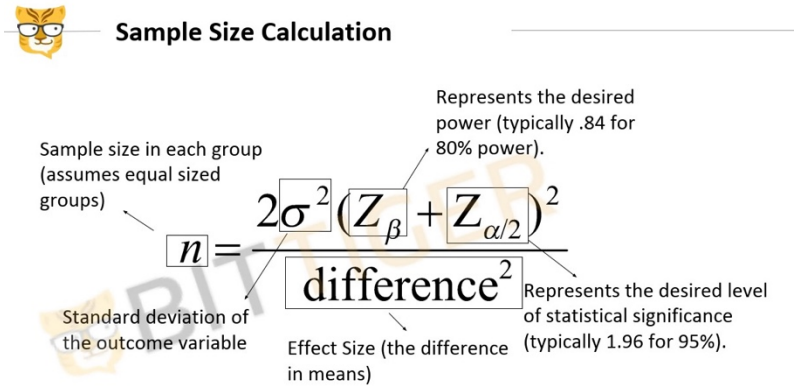
Your choice of metric, unit of diversion, choice of population – all these can affect the variability of the metric. So you want to take all the stuff into account and then start to determine and determine the size. You're going to have to figure out whether what you plan to do is realistic given how 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)





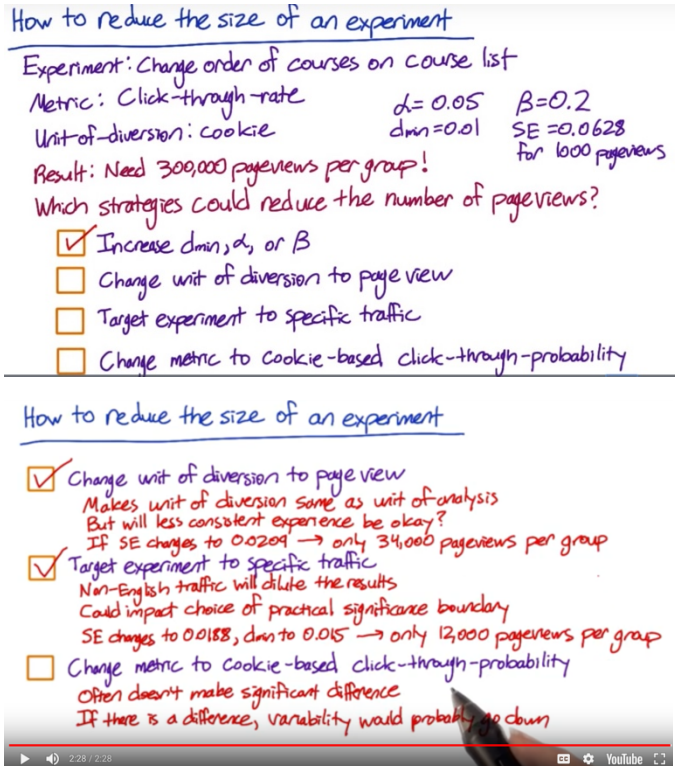
If you calculate the variability analytically, it won't change between the two units of diversion but for the cookie-based diversion, the analytic estimate is likely to be an under-estimate. So Audacity does an empirical estimate of the variability with 5,000 page views in each group.
In order to calculate 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

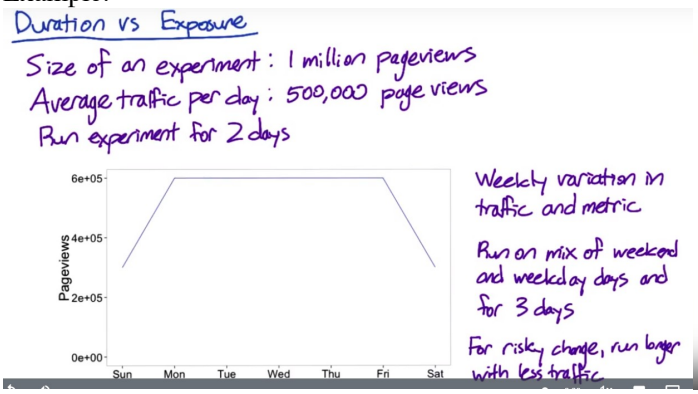
- First of all, what's the duration of the experiment that I want to run?
- Secondly, when do I want to run the experiment? Is back to school a good time to run it? What about holidays? Is it going to overlap something that's important?
- Third you have to think about what fraction of your traffic you're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

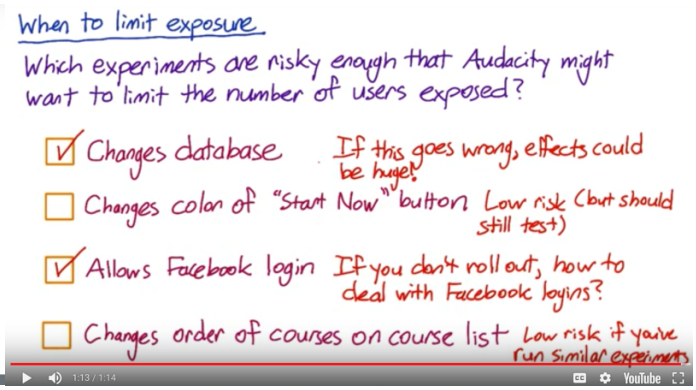
Those are all interrelated as they get you to the ideal size but you need to think about them a little bit separately.
For the 3rd question, what we're really asking is on any given day what proportion or what percentage of the cookies are you sending to your experiment and your control? Let's say we're 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:

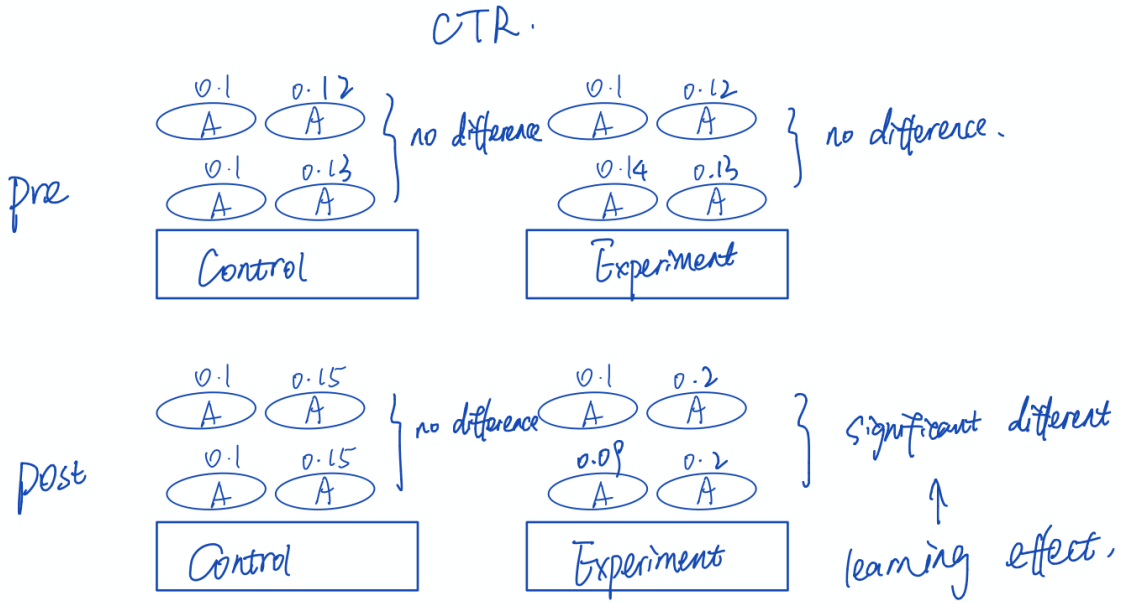
1. Change aversion, where when users first see a change they're like, what is this? I don't like anything.
2. Novelty effect which is the exact opposite. Oh, this is a new thing.

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.



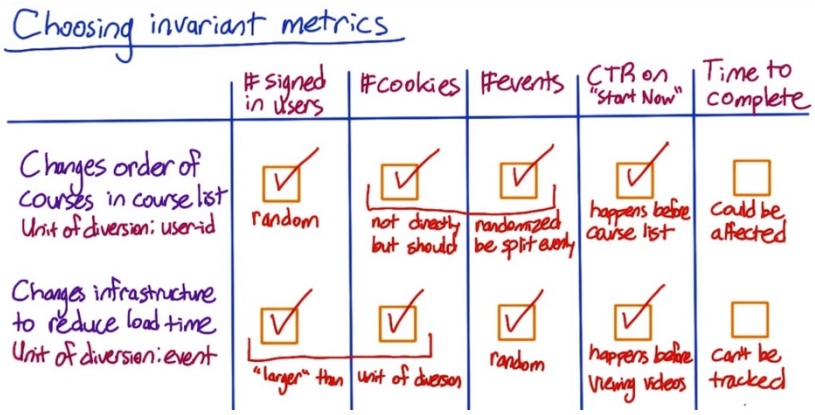
Analyzing results

Sanity Check: check whether there're things going wrong by looking at invariant metrics
Examples of things can go wrong:

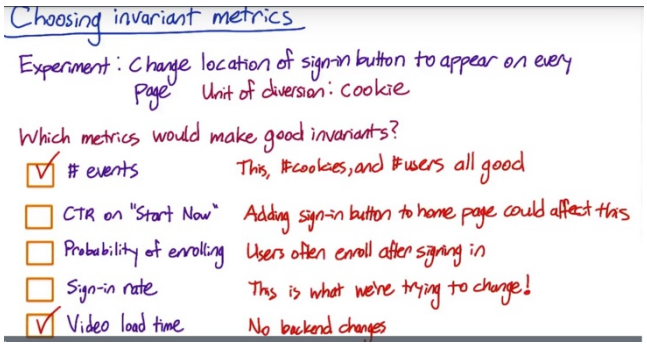
- Unit of diversion – experiment and control should be comparable
- Set up filters consistently between experiment and control
- Is data capture set up accurately capturing the events you are looking for?

Use invariants to do sanity check – two types:
(1) population sizing metrics based on unit of diversion: experiment population and control population should be comparable.
(2) other invariants: the metrics that shouldn't change in your experiment. We can check where the invariant metric falls under the overall process. E.g. if the feature affects the steps from #4, then the metrics associated with steps before #4 can be used as invariant metrics.

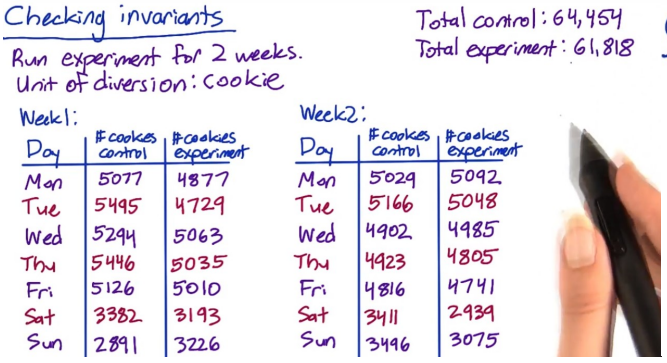
Case #1.
Since user id is used as unit of diversion, which is randomly assigned between 2 groups. Cookies and events might not be exactly the same between control and experiment, but should not vary too much unless users visit the pages significantly differently between the two group. CTR happens before course list. Time to complete might be affected as students might start with easier course based on the new order. Maybe putting easier courses first causes more users to start with easier courses, and then they finish them faster.

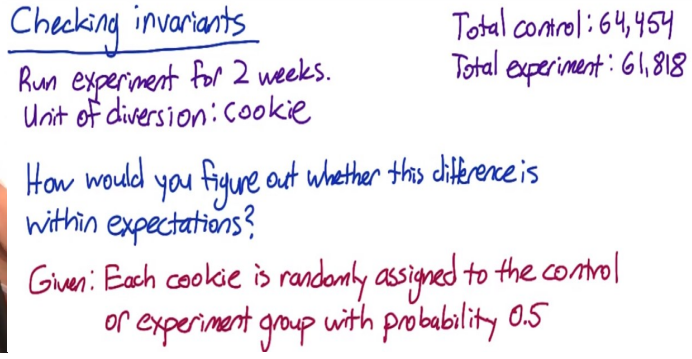


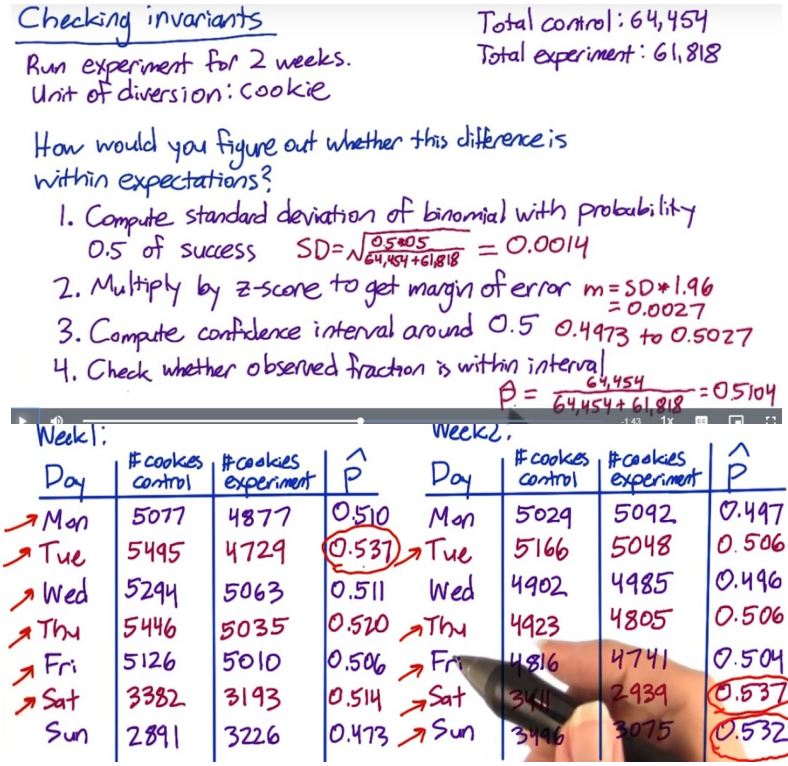
Case #2.
Signed in users and cookies are both larger than the unit of diversion in the sense that one user or one cookie could correspond to multiple events. So, since the events are being randomly assigned, the number of signed in users and cookies shouldn't be different between the two groups either.
The time to get through a class can't be tracked if you're using event-based diversion. Since by the time the user gets through a course, they could have been assigned to both the experiment and the control group multiple times. Even if you could track this, it wouldn't be a good invariant, since load time could affect how long it takes to complete a class.



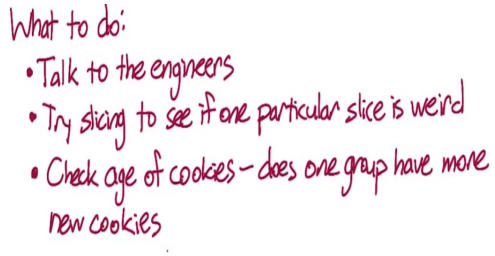
Cookies are being explicitly randomized over. User IDs are typically larger than cookies, in the sense that one user ID can correspond to multiple cookies. So user IDs should be evenly split as well. And it's more likely that the events could end up unevenly split, but it's not something you're expecting. And it would be good to catch that if it does happen.







In step 1, the "standard deviation" is the standard deviation of the sampling distribution for the proportion, or standard error. The abbreviation SE should be used in computations instead of SD.
Use p=0.5 to calculate the confidence interval. The observed fraction of control group is greater than the upper bound of CI, so there is something wrong with the setup. Do day-by-day analysis:

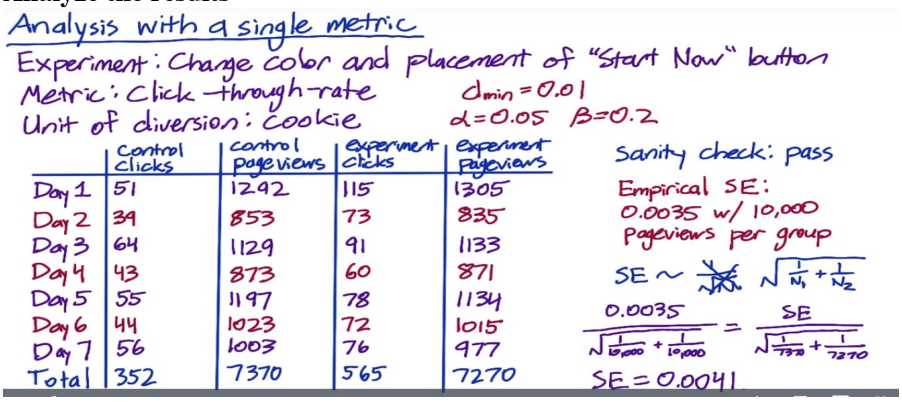


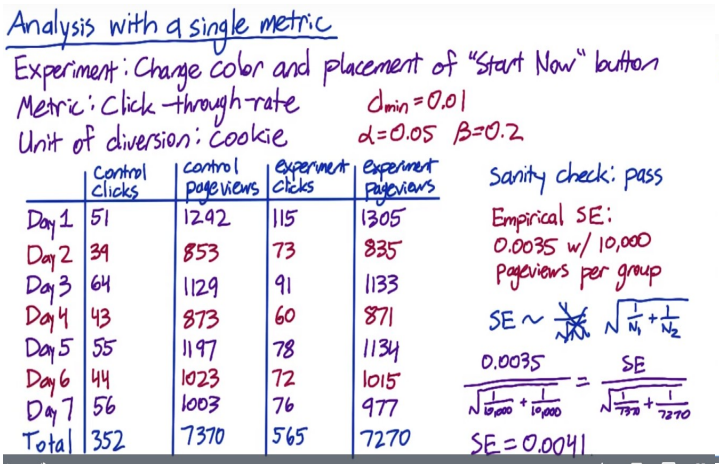
What to do if you find issues during the sanity check
(1) Issues to check with the engineering team:
- Experiment infrastructure
- Unit of diversion

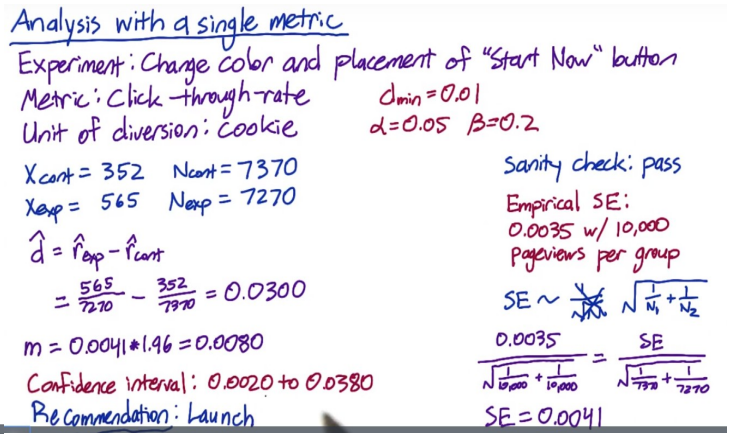
(2) Retrospective Analysis
Recreate the experiment diversion from the data capture to understand if there is something endemic to what you are trying to do that might cause the situation.
(3) Pre and Post period
If observe changes for invariant metrics on post period, check if similar changes exist on pre period. If so, there could be problems with the experiment infrastructure, setup, etc. If the changes is only observed on the post period, it means the issue is associated with the experiment itself such as data capture.

The most common thing – data capture. Maybe the changes trigger rarely, and you capture it correctly under the experiment but not the control.
Experiment setup – didn't set up filter correctly between control and experiment.
More rarely could be system issue such as cookie reset (need to dig deeper and find out with engineering team)
Learning effect may take time. If the issues are observed at the beginning of the experiment, might not be learning effect.

Analyze the results



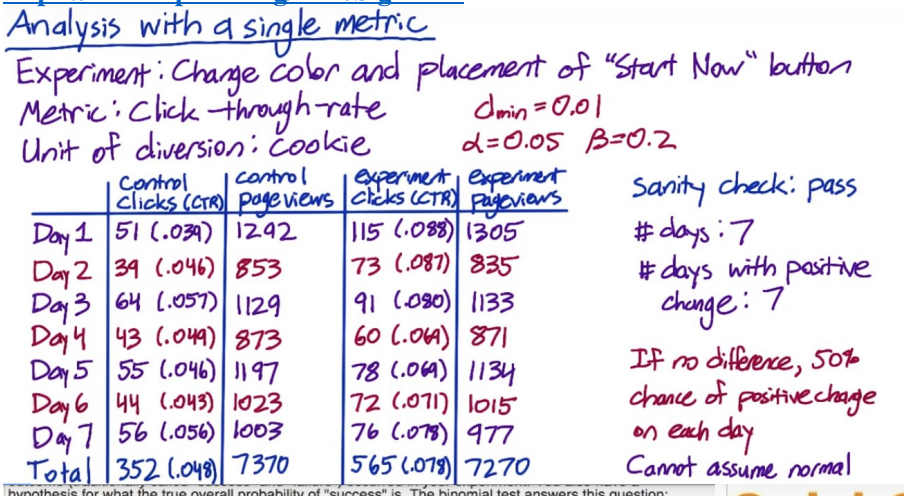


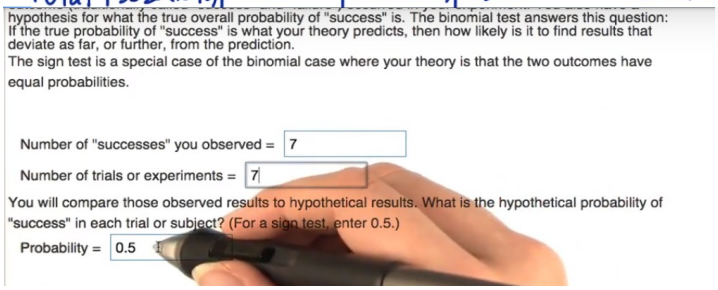


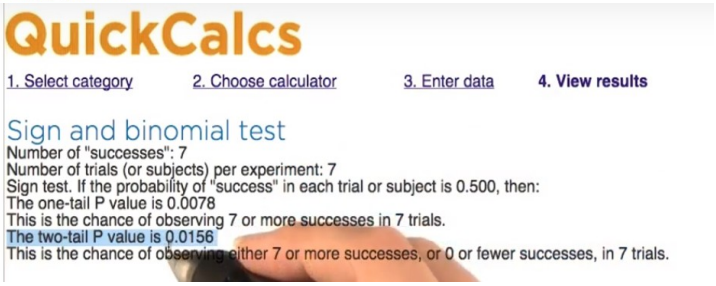
Unlike click-through probability, click-through rate more follows Poisson distribution and the analytical variance is harder to estimate compared to Binomial. Need to analyze it empirically. Typo: 0.022 – 0.038. Practical significance is out of the CI, which is 0.01, meaning the difference is significant to be captured, and the change should be launched.

Sign Test:

https://en.wikipedia.org/wiki/Sign_test

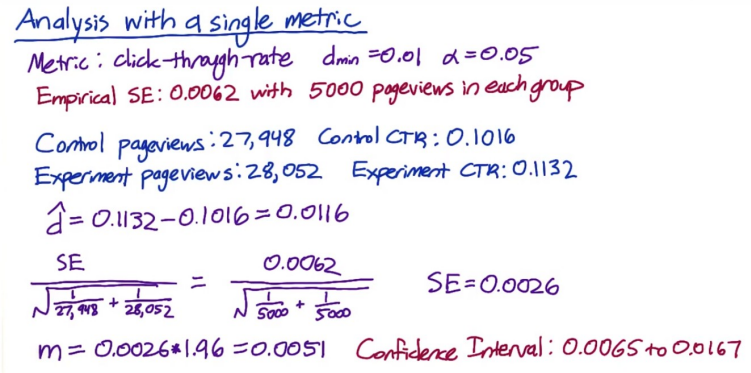






p-value for the sign test < 0.05

Another example:



The CI does not include 0 – indicating that the difference is at 95% level significantly different from 0. But 0.01 is included, indicating that cannot be 95% confident that the change is greater than 0.01 – i.e. the size effect we care about.

Sign test:

Two-tail p-value is 0.424, which is not significant at 0.05 level.

Analysis with a single metric

Metric: click-through-rate $d_{min}=0.01$ $\alpha=0.05$

Days where CTR is higher in experiment: 9/14

Week 1

	Control CTR	Control #pageviews	Experiment CTR	Experiment #pageviews
Mon	0.097	2029	0.091	1971
✓Tue	0.100	1991	0.104	2009
Wed	0.103	1951	0.100	2049
Thu	0.109	1985	0.087	2015
Fri	0.107	1973	0.094	2027
✓Sat	0.092	2021	0.147	1979
✓Sun	0.110	2041	0.142	1959

Week 2

	Control CTR	Control #pageviews	Experiment CTR	Experiment #pageviews
✓Mon	0.094	1980	0.107	2020
✓Tue	0.105	1951	0.110	2049
Wed	0.106	1988	0.103	2012
✓Thu	0.097	1977	0.101	2023
✓Fri	0.097	2019	0.101	1981
✓Sat	0.110	2035	0.151	1965
✓Sun	0.096	2007	0.150	1993

QuickCalcs

1. Select category 2. Choose calculator 3. Enter data **4. View results**

Sign and binomial test

Number of "successes": 9
Number of trials (or subjects) per experiment: 14
Sign test. If the probability of "success" in each trial or subject is 0.500, then:
The one-tail P value is 0.2120
This is the chance of observing 9 or more successes in 14 trials.
The two-tail P value is 0.4240
This is the chance of observing either 9 or more successes, or 5 or fewer successes, in 14 trials.

Hypothesis on the effect size showed statistically significant results, but the sign test didn't. Why?

1. The sign test has lower power than the effect size test, which is frequently the case for nonparametric tests. That's the price you pay for not making any assumptions. So this isn't necessarily a red flag, but it's worth digging deeper and figure out what's going on.
2. In the above example, weekend click-through rates are much higher than weekdays. Size effect on weekends are a lot higher. Weekdays don't have a statistically significant difference but weekends have.

Based on this, I would recommend not launching the experiment at this point. Instead, I would dig deeper into why the change didn't affect weekday visitors. Once I understood that, I might have an idea for how to iterate on the change to help it affect more of the users.

If not, then I'd talk to the decision makers about whether a change of this magnitude on weekend traffic is worth launching.

Simpson Paradox: different subgroups in the data, within each group the results are stable, but when aggregated the mix of the subgroups drive the results.

Simpson's paradox

	Men applied	Women applied	Men accepted	Women accepted
Department A	825	108	512 (62%)	89 (82%)
Department B	417	375	137 (33%)	132 (35%)
Total	1242	483	649 (52%)	221 (46%)

Women's acceptance rate is higher than men in both departments, but the overall acceptance rate is lower. This is because most of the women applied to department B, whose acceptance rate is lower than A.

Simpson's paradox

	N_{cont}	X_{cont} (CTR)	N_{exp}	X_{exp} (CTR)
New Users	150,000	30,000 (0.2)	75,000	18,750 (025)
Experienced Users	100,000	1,000 (0.01)	175,000	3,500 (0.02)
Total	250,000	31,000 (0.124)	250,000	22,250 (0.089)

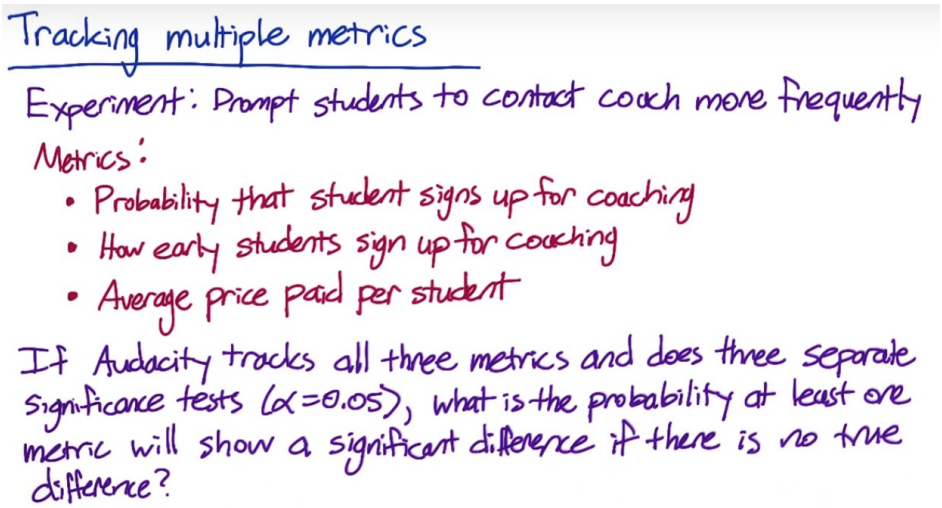
Goal: click-through-rate is higher in experiment group for both new and experienced users, but overall click-through-rate is lower in the experiment group

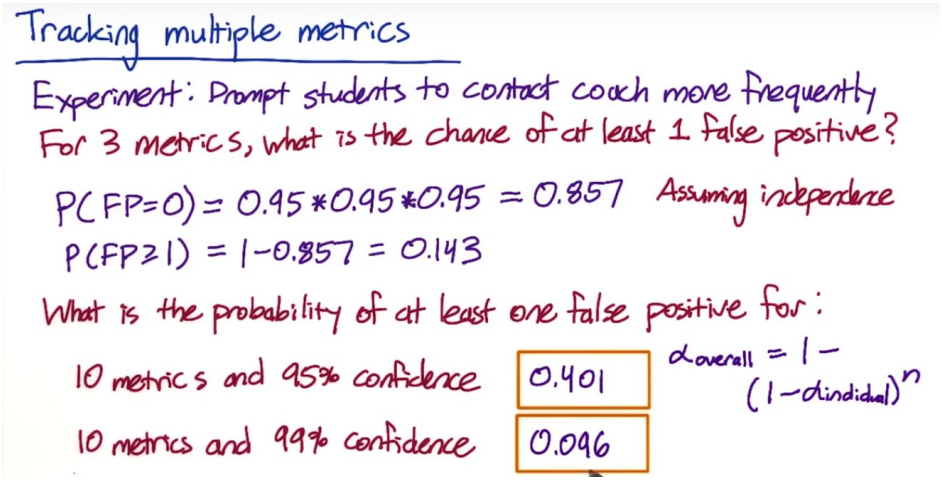
CTR for experiment group is higher than control for both new users and experienced users. But for total users, control group CTR is higher. This is because control group has more new users, which has higher CTR and experienced users.
Problems:

1. Why are there more page views from new users in the control group than in the experiment group? If the assignment to the control in the experiment group is random, then shouldn't the new users be evenly split between the control and experiment? And same for the experienced users.
   It should be. So this problem within the experiment setup largely affects the result. It's a good idea to make sure the number of page views is the same in the experiment group and the control group as a sanity check. Checking that breakdown across different slices could also be a good sanity check.
2. However, it's also possible to get skewed numbers like this, even if your setup is correct, if your change, or experiment, affects new users and experienced users differently.
   Suppose you're diverting based on user ID and the change makes new users generate fewer page views, for example, they refresh the page less, and experienced users generate more page views. That explains why there are more page views in the experiment group for experienced users and more page views in the control group for new users.

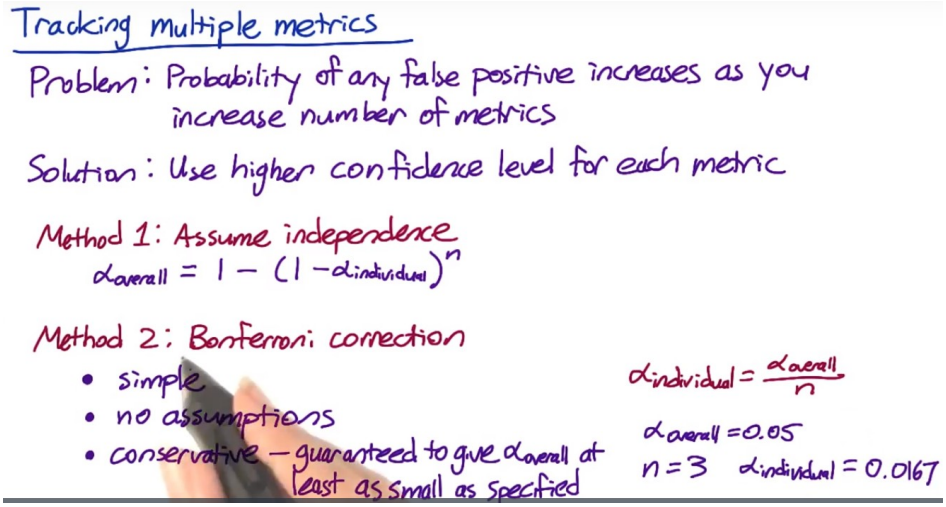
Therefore, although for each subgroup it seems that CTR has improved, the overall CTR was not improved and cannot say the experiment is successful. Whether it's a faulty experiment set up, or something where your change affects new and experienced users differently, you won't be able to make a valid conclusion until you understand what's going on.

Multiple Metrics
As you test more metrics, it becomes more likely that one of them will show a statistically significant result by chance. So if you're testing 20 metrics, and you have a 95% confidence level. You would expect to see one case at least that time where you got a result that says it's significant but it's only concurring by chance.
So this is a problem, but you're not sunk because it shouldn't be repeatable. That is if you did the same experiment on another day or you divide or just slices or you did some bootstrap analysis, you wouldn't see the same metric showing up as significant differences every time, it should occur randomly.

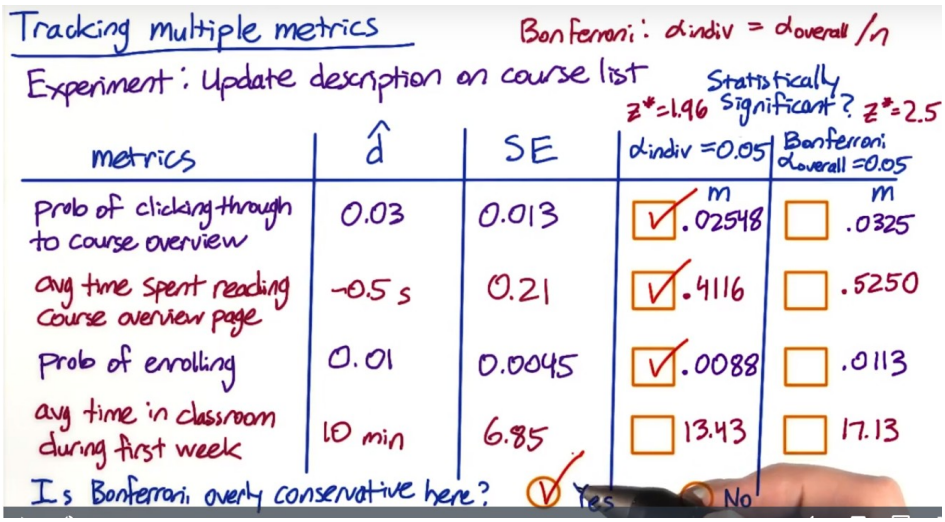




I was assuming that the metrics were independent. In fact, this isn't true here. These three metrics are all related and more likely to move together. So 14.3% is an overestimate of the probability of a false positive. But assuming independence is an easy way to get a conservative estimate.



Method 1: set up an overall alpha and use it to calculate each individual alpha.
Method 2: often will be tracking metrics that are correlated and all tend to move at the same time, in which case this method is too conservative – this results in less significant difference, and launch less experiments.



In this case, the Bonferroni method is probably too conservative. If the course description was an improvement, then it makes sense that it could cause more than one of these metrics to move and they're probably more likely to move together.

Analyze the multiple metrics
Are all the related metrics moving in the same direction – e.g. click-through rate and click through probability.
Revenue per thousand queries is composed of click through rate and cost per click

Stay time on the page vs. clicks on the page – people might spend more time on clicking than staying. Need to better understand how people reacts to the changes.

Overall evaluation criteria (OEC) should be established based on an understanding of what your company is doing and what the problems are. It should balance long-term and short-term benefits. Business analysis is needed to make the decision. Once you have some candidates of OEC, you can run a few experiments to see how they steer you (whether in the right direction).

Change in metric and not others:

- Maybe you know for small changes, a change in one metric and not others might be fine.
- But for big changes, this may indicate something is wrong. Depends on your understanding of the changes itself.

Different impact across slices:

- Again need to understand the changes. Is there a bug? Have you seen this in other experiments? Is this because of different users (like or do not like the change)
- e.g. bolding works better in English/German than Chinese/Japanese. May consider using color than bolding for Chinese/Japanese.

Whether to launch an experiment or not??
1. Statistically and practically significant to justify the change?
2. Do you understand what the change can do to user experience?
3. Is it worth the investment?

Ramp up AB test

Maybe start with 1% of the traffic and divert to experiment and increase that until the feature is fully launched.
Also remove all filters to test the change on all users to understand if there is any incidental impact to unaffected users that you didn't test in your original experiment.

Gotcha: the effect might flatten out as you experiment the change – effects are not repeatable even they are statistically significant.

- Seasonality such as school season, holiday, etc.

Holdback – launch the experiment to everyone except for a small holdback who don't get the change, and you continue to compare them to the control. You will see a reverse of the impact in your experiment, and you can track that over time until you are confident that your results are repeatable. This can help track lots of seasonal or event-driven impacts.
Other things that cause the disappearing launch effect?

- Novelty effect or change aversion: as users discover or change their adoption of your change, their behavior can change and measured effect can change – can do cohort analysis.
- Pre- and post- period analysis in combination of cohort analysis to understand learning effect – i.e. how users adapt to the changes over time.

Lessons learned:
(1) Always make sure your experiment setup is correct
(2) In addition to statistical significance, you are making business decision. E.g. what if it improves for 30% and neutral for the rest? Or what if it improves for 70% but makes it worse for the 30% left? Want to launch as is or fine-tune it first
(3) overall business analysis – what's the engineering cost of maintaining the change? Are there customer support or sales issue? What's the opportunity cost? These are judgment calls which your recommendation should be based on.
(4) As noted earlier, test for all users for the incidental impact.

Four principles of AB Test

First Principle: Risk

First, in the study, *what risk is the participant undertaking*? The main threshold is whether the risk exceeds that of "minimal risk". Minimal risk is defined as the probability and magnitude of harm that a participant would encounter in normal daily life. The harm considered encompasses physical, psychological and emotional, social, and economic concerns. If the risk exceeds minimal risk, then informed consent is required. We'll discuss informed consent further below.

In most, but not all, online experiments, it can certainly be debated as to whether any of the experiments lead to anything beyond minimal risk. What risk is a participant going to be exposed to if we change the ranking of courses on an educational site, or if we change the UI on an online game?

Exceptions would certainly be any websites or applications that are health or financial related. In the Facebook experiment, for example, it can be debated as to whether participants were really being exposed to anything beyond minimal risk: all items shown were going to be in their feed anyway, it's only a question of whether removing some of the posts led to increased risk.

Second Principle: Benefits

Next, *what benefits might result from the study*? Even if the risk is minimal, how might the results help? In most online A/B testing, the benefits are around improving the product. In other social sciences, it is about understanding the human condition in ways that might help, for example in education and development. In medicine, the risks are often higher but the benefits are often around improved health outcomes.

It is important to be able to state what the benefit would be from completing the study.

Third Principle: Alternatives

Third, *what other choices do participants have*? For example, if you are testing out changes to a search engine, participants always have the choice to use another search engine. The main issue is that the fewer alternatives that participants have, the more issue that there is around coercion and whether participants really have a choice in whether to participate or not, and how that balances against the risks and benefits.

For example, in medical clinical trials testing out new drugs for cancer, given that the other main choice that most participants face is death, the risk allowable for participants, given informed consent, is quite high.

In online experiments, the issues to consider are what the other alternative services that a user might have, and what the switching costs might be, in terms of time, money, information, etc.

Fourth Principle: Data Sensitivity

Finally, *what data is being collected, and what is the expectation of privacy and confidentiality*? This last question is quite nuanced, encompassing numerous questions:

- Do participants understand what data is being collected about them?
- What harm would befall them should that data be made public?
- Would they expect that data to be considered private and confidential?

For example, if participants are being observed in a public setting (e.g., a football stadium), there is really no expectation of privacy. If the study is on existing public data, then there is also no expectation of further confidentiality.

If, however, new data is being gathered, then the questions come down to:

- What data is being gathered? How sensitive is it? Does it include financial and health data?
- Can the data being gathered be tied to the individual, i.e., is it considered personally identifiable?
- How is the data being handled, with what security? What level of confidentiality can participants expect?
- What harm would befall the individual should the data become public, where the harm would encompass health, psychological / emotional, social, and financial concerns?

For example, often times, collected data from observed "public" behavior, surveys, and interviews, if the data were not personally identifiable, would be considered exempt from IRB review (reference: NSF FAQ below).

To summarize, there are really three main issues with data collection with regards to experiments:

- For new data being collected and stored, how sensitive is the data and what are the internal safeguards for handling that data? E.g., what access controls are there, how are breaches to that security caught and managed, etc.?
- Then, for that data, how will it be used and how will participants' data be protected? How are participants guaranteed that their data, which was collected for use in the study, will not be used for some other purpose? This becomes more important as the sensitivity of the data increases.
- Finally, what data may be published more broadly, and does that introduce any additional risk to the participants?

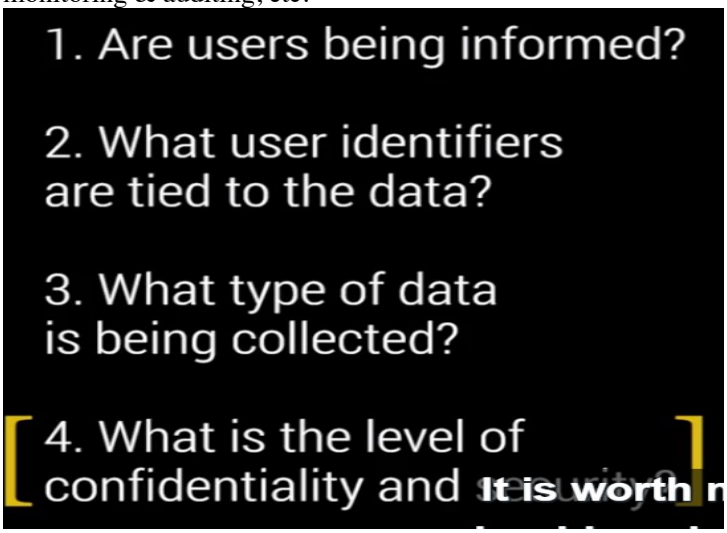
Difference between pseudonymous and anonymous data

One question that frequently gets asked is what the difference is between identified, pseudonymous, and anonymous data is.
Identified data means that data is stored and collected with personally identifiable information. This can be names, IDs such as a social security number or driver's license ID, phone numbers, etc. HIPAA is a common standard, and that standard has 18 identifiers (see the Safe Harbor method) that it considers personally identifiable. Device id, such as a smartphone's device id, are considered personally identifiable in many instances.
Anonymous data means that data is stored and collected without any personally identifiable information. This data can be considered **pseudonymous** if it is stored with a randomly generated id such as a cookie that gets assigned on some event, such as the first time that a user goes to an app or website and does not have such an id stored.
In most cases, anonymous data still has time-stamps -- which is one of the HIPAA 18 identifiers. Why? Well, we need to distinguish between anonymous data and anonymized data. **Anonymized data** is identified or anonymous data that has been looked at and guaranteed in some way that the re-identification risk is low to non-existent, i.e., that given the data, it would be hard to impossible for someone to be able to figure out which individual this data refers to. Often times, this guarantee is done statistically, and looks at how many individuals would fall into every possible bucket (i.e., combination of values).
What this means is that anonymous data may still have high re-identification risk (see AOL example).

So, if we go back to the data being gathered, collected, stored, and used in the experiment, the questions are:

- How sensitive is the data?
- What is the re-identification risk of individuals from the data?

As the sensitivity and the risk increases, then the level of data protection must increase: confidentiality, access control, security, monitoring & auditing, etc.



2. personal information tied to the data?
3. financial, health data? Or other less important data? individual level or group level?
4. secure access?