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??

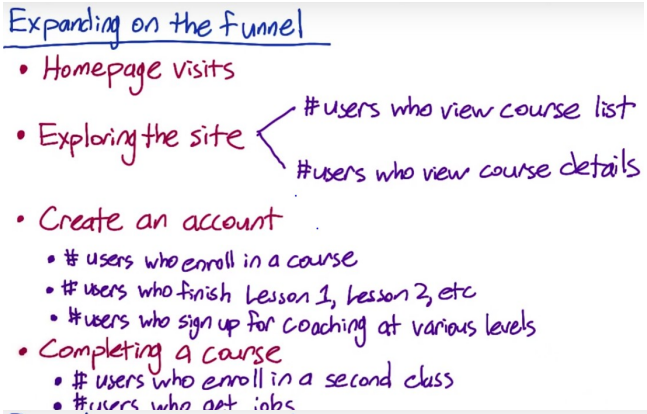
Depend on company culture and how comfortable people are with the data. The leader may be more comfortable with a whole suite of metrics where they can see things move.

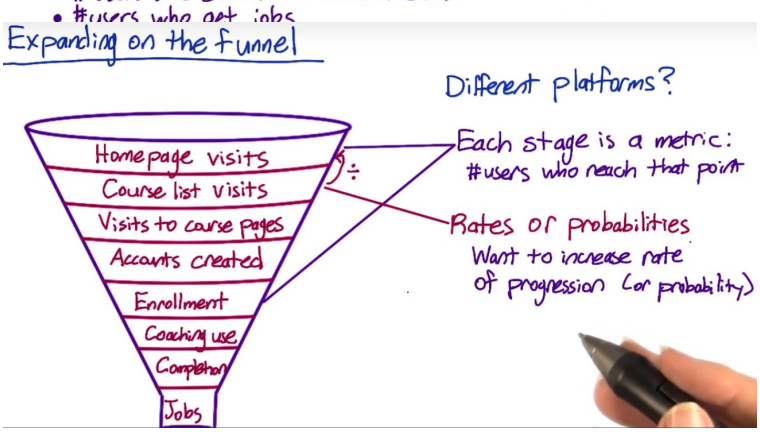
For PR purposes, external reporting, you may have to settle on a single overall objective. In large companies, people may want different teams to move towards the same goal, therefore in that case might want a single metric.

Do not suggest using composite metric because:

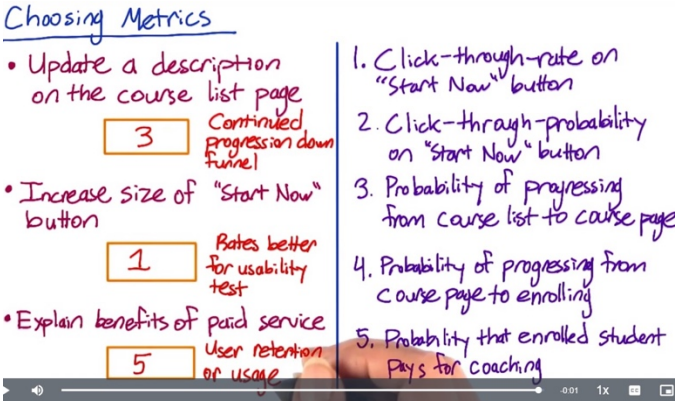
1. hard to define and agree on a definition
2. can run into problem if you over-optimize looking into one thing without looking at others
3. when the metric moves, people will come in and ask why it moves, and you have to go back and check individual metric anyway

Refine the customer funnel

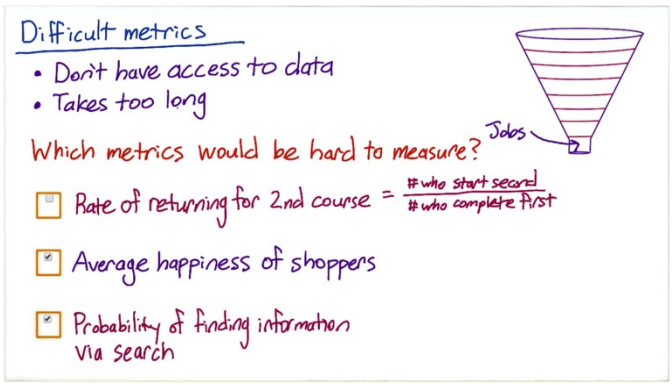




Might also care about whether or not a customer ever gets to a certain step – binary variable.
Probability – unique user that progress across the funnel.



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

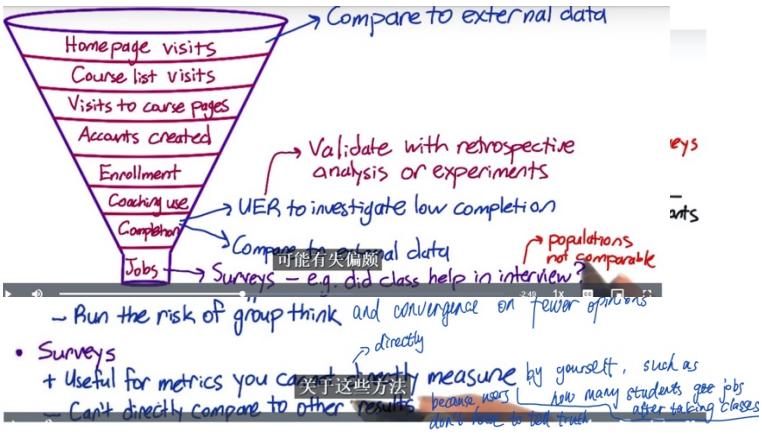
Doesn't have data

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

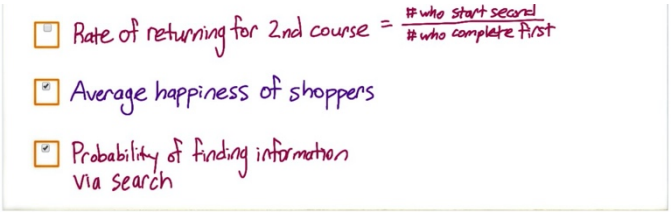


UER for a particular course with low completion:
Watch the students complete the lesson – understand where to click? Can they find all the info on the screen? Do they follow the order? How are they interacting with the coach? Waiting for video to load – latency? Link for the additional materials used?

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:

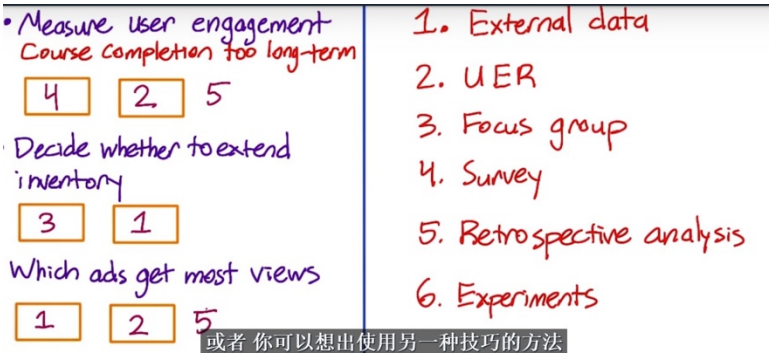


Rate of returning to 2nd course: Follow up with a survey to find out the reason to return to 2nd course. If something measurable that predicts the return, can use it as proxy.
Average happiness of shoppers: Find things correlated – survey at the end of purchase, or UER.

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.

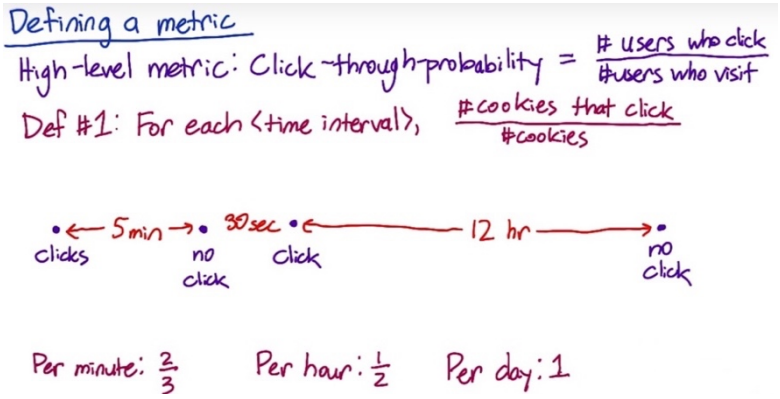
Build intuition about your metric, your data, and your system. A good data analyst should be able to understand what changes in your data and metric, your system can produce? You need to first decide, given the

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

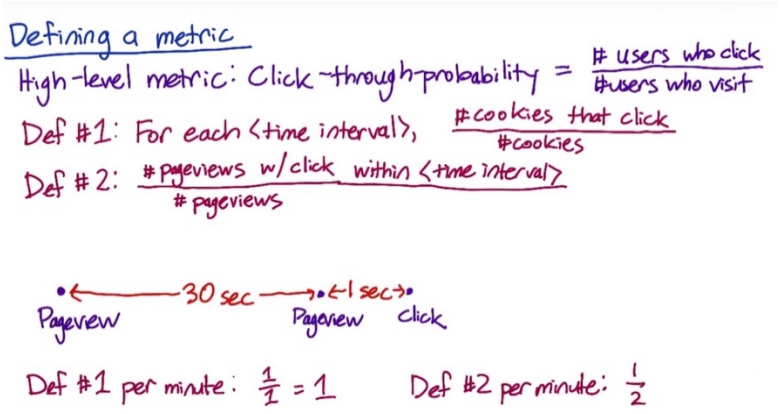
- Total number of clicks/ total number of views
- A more nuanced version for probability: there's something called a cookie which is an anonymous identifier for a user. What we can do instead is say, did a cookie visit the site, and then, given that a cookie visited, did they click or not?

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

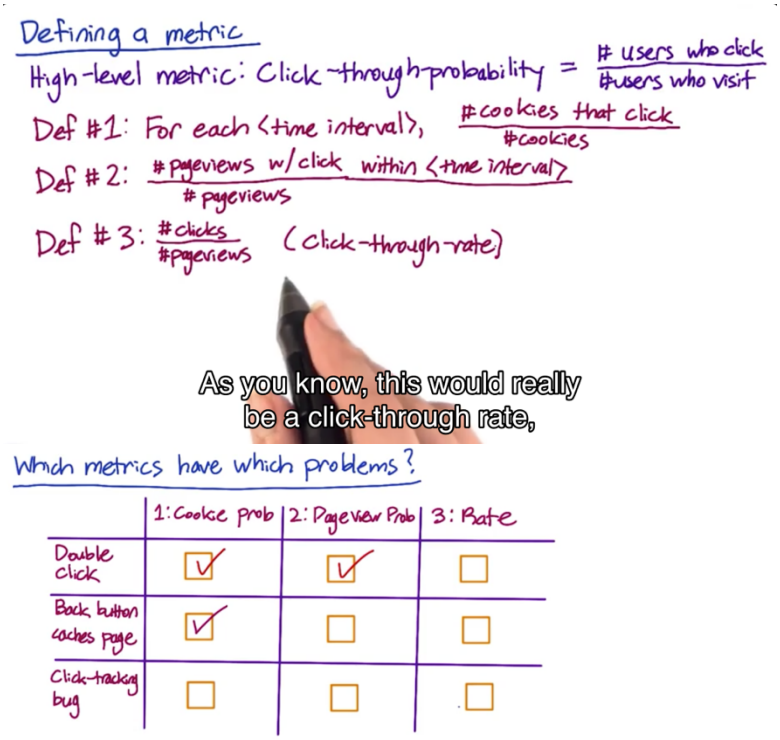


To use this definition, we need some way of determining whether 2 events are from the same user. Let's say we use cookies. Next, if the same user or cookie visits the page once and then comes back a week or two later, do we really only want to count that once? Usually, you'll want to count those visits 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:

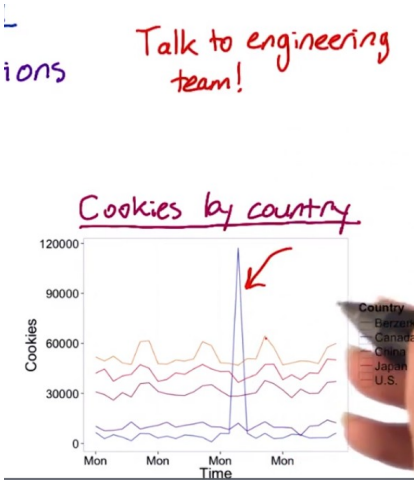
You often see, sort of abuse on your site such as spam or fraud, and you want to try to filter that out
e.g. if you have a competitor, who's looking for your site clicking on absolutely everything, and you may not want to use that data in your experiment.
e.g. you may even have some malicious trying 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.

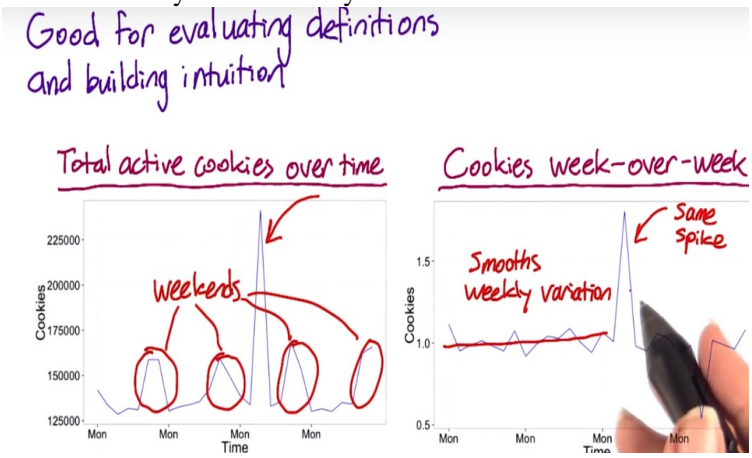
The goal of filtering is to de-bias or dull-bias the data. So, you should be careful you don't introduce bias into your data.
e.g. if you have a metric that can only be measured on logged-in users, you might actually be biased in your data because there's a bunch of noncommittal or newer users trying 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.

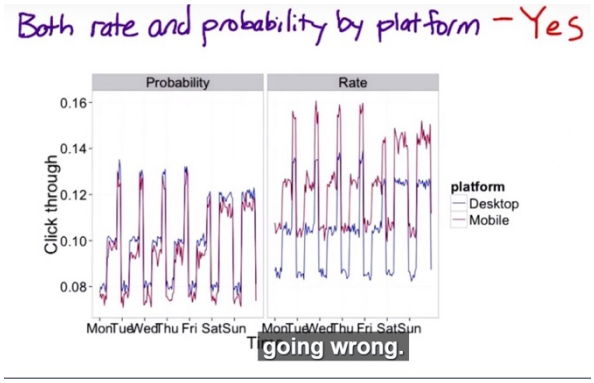


One way I can verify whether the spike is odd is by looking at what's called a week-over-week plot. That is, I'll divide each data point by the corresponding data point from a week ago. As you can see, that tends to smooth out the weekly variation. I can see looking at this plot that it would stand out 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.

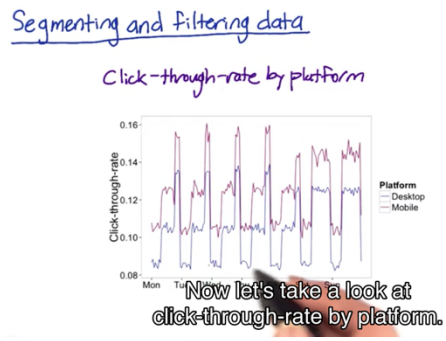
Another thing to look at is year-over-year data. If there's an annual conference or something causing the spike, it will disappear.
However, in this plot, the same spike is still here, meaning it's probably not due to a yearly variation. You can also see that the weekly variation is back, since the day 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.

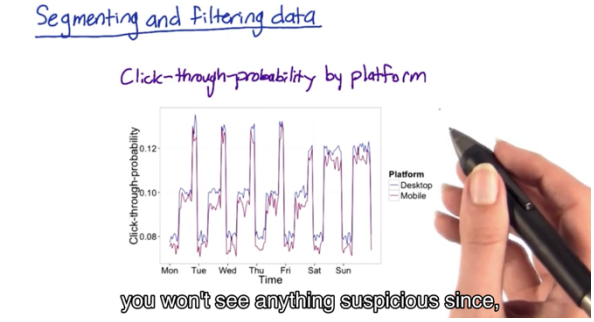
So, let's trying looking at how this metric varies by country. We don't see the spike in most countries but we do see it in Berzerkistan, so that one country was causing the entire spike.
At this point, it's a good idea to talk to the engineering team, and maybe they'll be able to figure out if this 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.

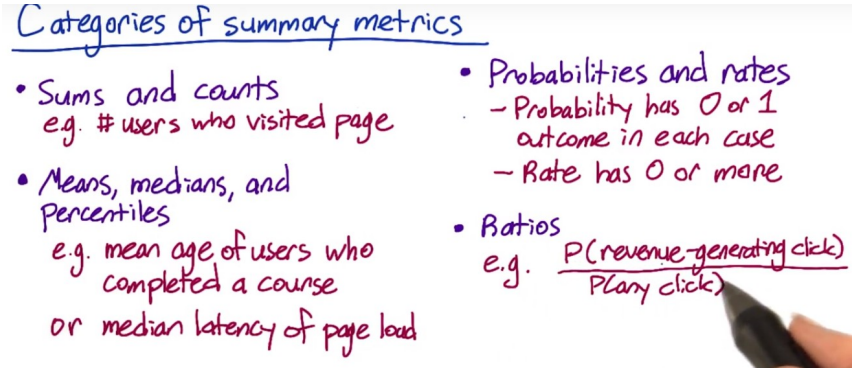


The click-through-rate by platform graph is suspicious, because it shows the click-through-rate being higher on mobile than on desktop. But users will have different behavior on desktop than on mobile so, it's still not really clear that this is a problem with the JavaScript tracking and not just a difference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

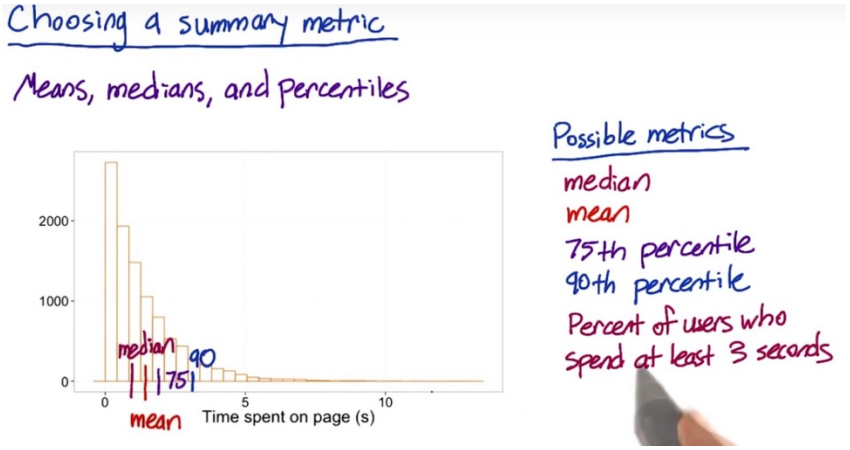


In the click-through-probability by platform plot, you won't see anything suspicious, since if JavaScript does send a duplicate ping. The click-through probability will eliminate that, collapsing it into one. So here we see probability is pretty similar between desktop and mobile.

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.

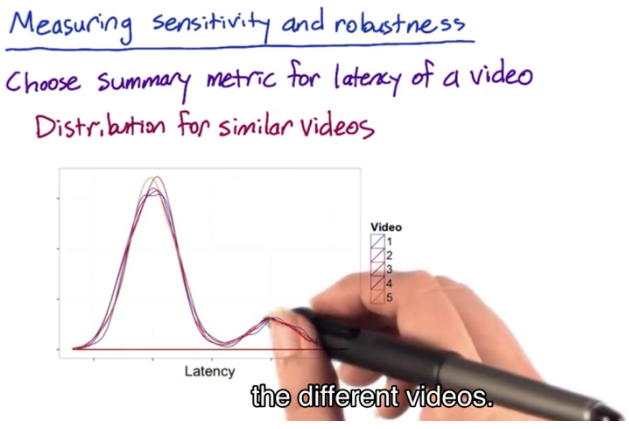
- Mean is sensitive to outliers. So, if in your data you see a lot of cases of really long load times, maybe due to something going on in the users machine, or a bad network connection, then you want to maybe not choose the mean, because the mean is going to be pretty heavily influenced by those types 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
<u>#1. Run experiment or use experiments already have</u>
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.
<u>#2. A/A experiment to determine if it's too sensitive.</u>
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.

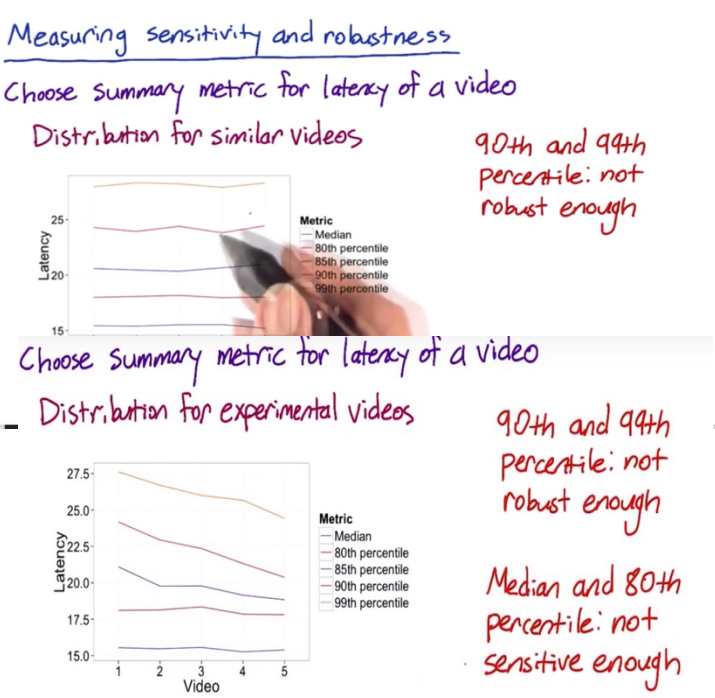
How to measure the sensitivity and robustness of some different metrics:
<u>For example: video latency</u>

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.



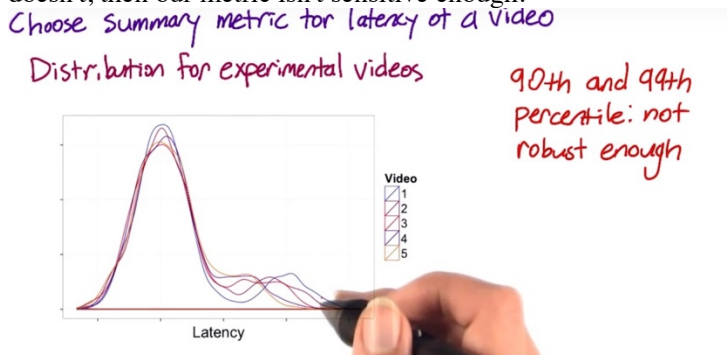
Let's do the retrospective analysis first by segmenting the data by different videos. In other words, look at the distribution of load times per video.
You can see two peaks here, a fairly long load time, and then more people with a shorter load time. This could happen if you had people with different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

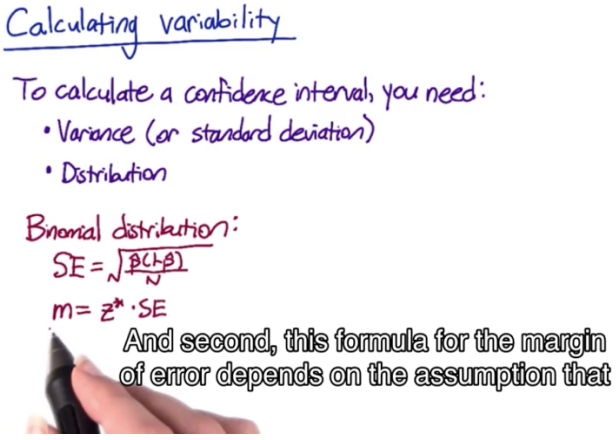
e.g. for a shopping site, in June, most people are on vacation, they're not shopping a lot. So, you have fewer users, you probably have a lower click-through rate. Whereas, in December, you've got loads of users, and a much higher click-through rate. If you have the same practical significance boundary, 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.



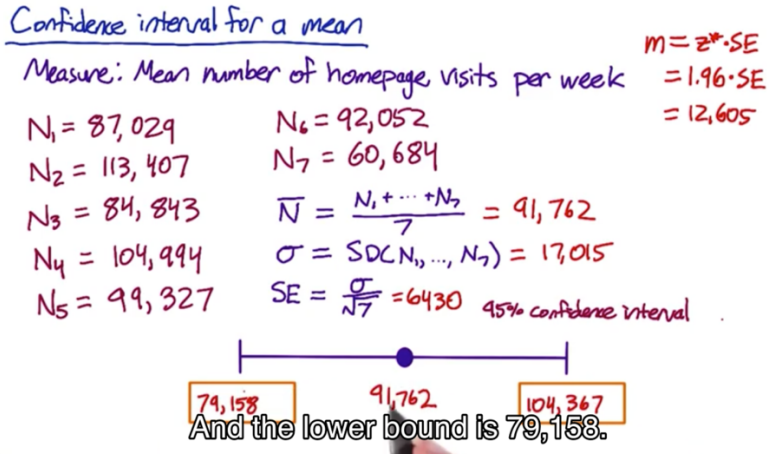
In order to calculate a confidence interval for your metric, you'll need to know the variance (standard deviation) of your metric and its distribution.
In binomial distribution, we used the fact that this was a binomial distribution in two ways. First, we use the fact that this was a binomial distribution 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.

Calculating variability

type of metric	distribution	estimated variance
probability	binomial (normal)	$\frac{\hat{p}(1-\hat{p})}{N}$
mean	normal	$\frac{\hat{\sigma}^2}{N}$
median/percentile	depends	depends
count/difference	normal (maybe)	$Var(X) + Var(Y)$
rates	poisson	$\bar{X}$
ratios	depends	depends

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

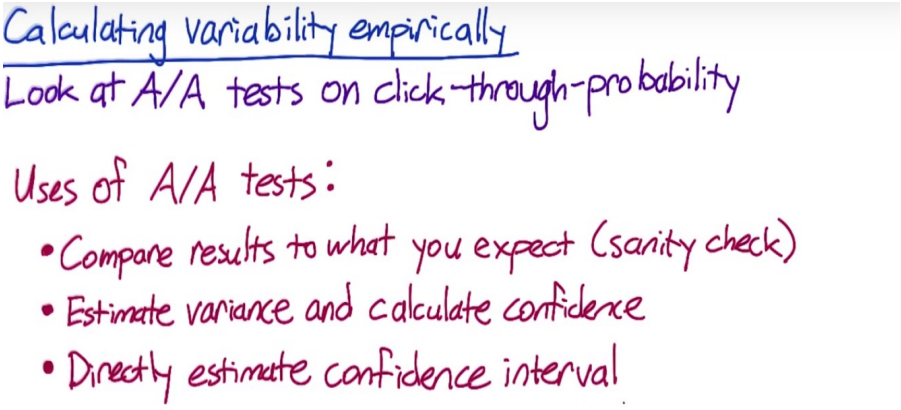
Empirical Variances:
For more complicated metrics, you might have to estimate the variance empirically than analytically
Other reasons to use empirical methods is that you're making assumptions for the underlying data distribution when computing the variance of a metric. This might work for simple metrics, 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.

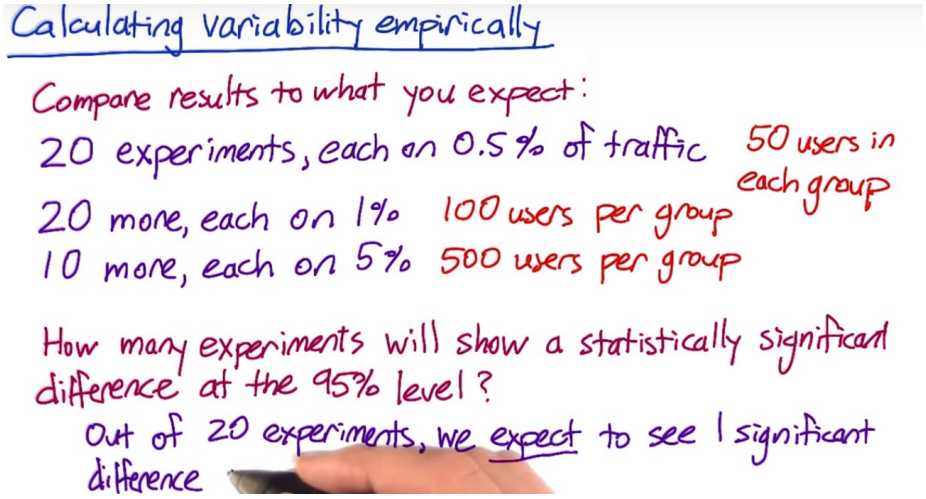
Now, in reality, what we have is a whole gradation of different methods:
If you're starting out and you're running your first experiment using a relatively simple metric, do the analytical estimate of your ants.
→
As you're starting to push towards more complicated metrics or you're running more and 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.

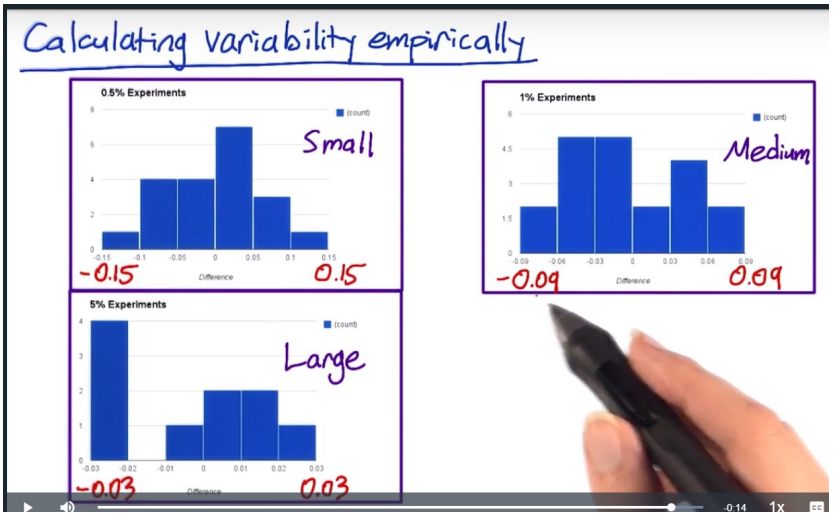


(1) Example 1: Sanity Checking:

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	

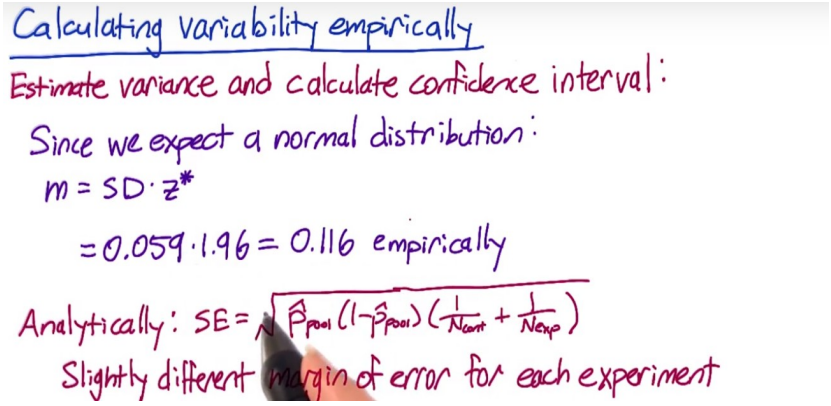
Another thing we can check about the A/A tests is whether the differences follow the distribution we expect. We can derive this from the column which contains the difference between the two groups for each experiment.
One thing to check is whether the differences are following a normal distribution as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

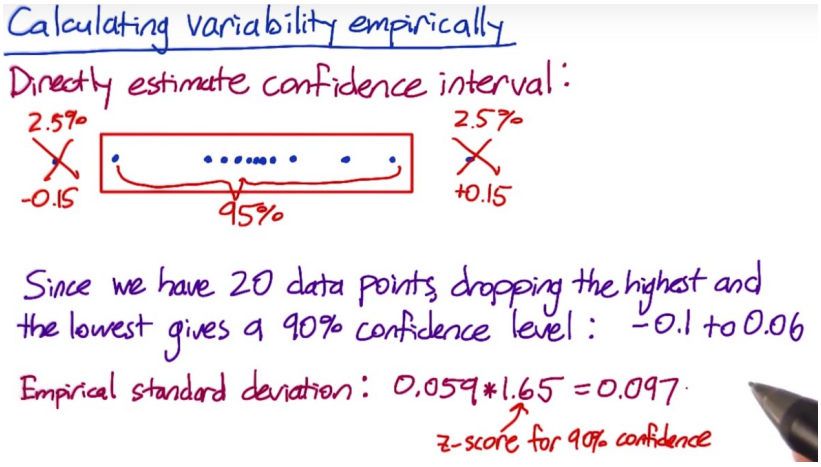
(2) <u>Example 2: Calculate empirical variability</u>



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.

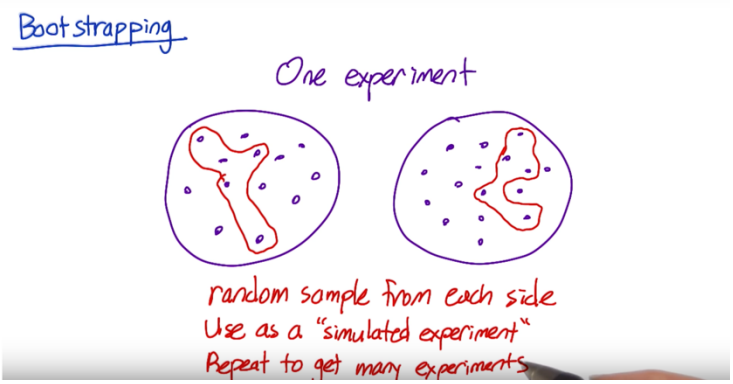
If you didn't know beforehand whether to expect your metric to follow a normal distribution, then you might look at histogram to see whether the metric look like it followed a normal distribution.
Remember, if we had done this analytically, the standard error depends on the pooled probability, which 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) <u>Example 3: Directly estimate the confidence interval</u>



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

<u>Bootstrapping:</u>



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.