Practical advice for analysis of large, complex data sets

October 31, 2016

By PATRICK RILEY

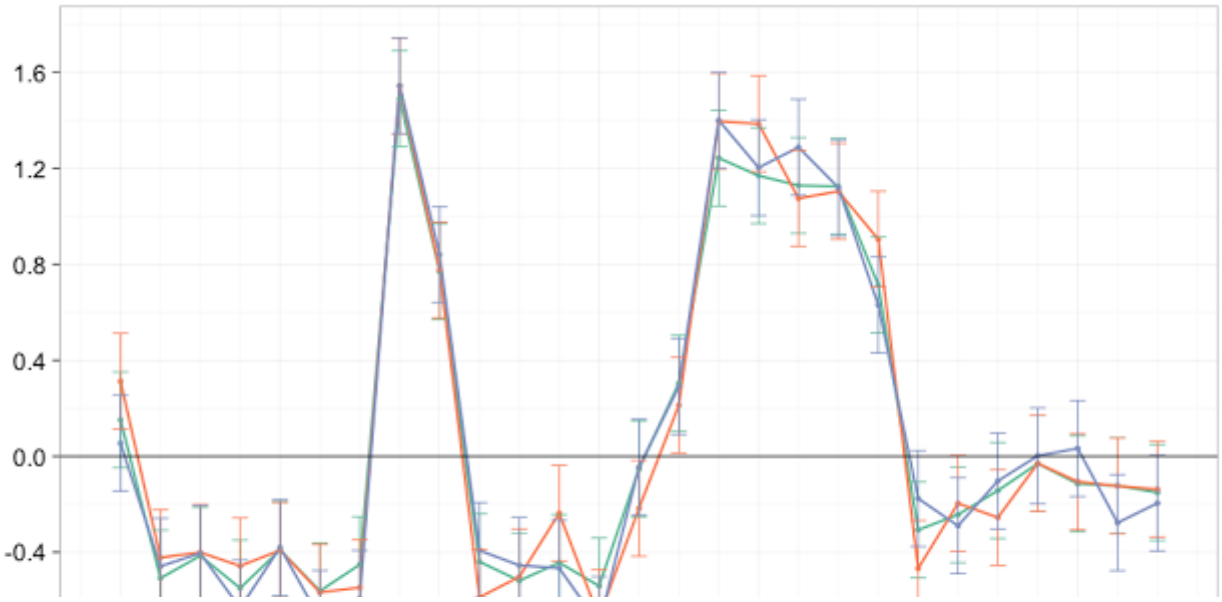
For a number of years, I led the data science team for Google Search logs. We were often asked to make sense of confusing results, measure new phenomena from logged behavior, validate analyses done by others, and interpret metrics of user behavior. Some people seemed to be naturally good at doing this kind of high quality data analysis. These engineers and analysts were often described as “careful” and “methodical”. But what do those adjectives actually mean? What actions earn you these labels?

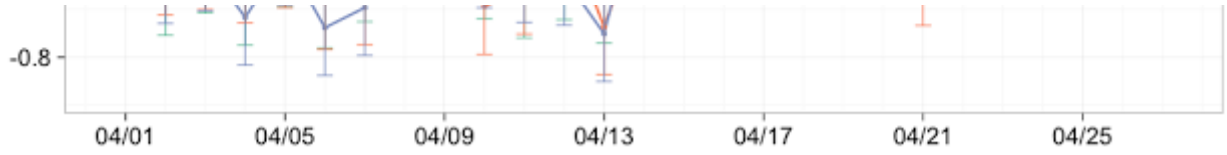
To answer those questions, I put together a document shared Google-wide which I optimistically and simply titled “Good Data Analysis.” To my surprise, this document has been read more than anything else I’ve done at Google over the last eleven years. Even four years after the last major update, I find that there are multiple Googlers with the document open any time I check.

Why has this document resonated with so many people over time? I think the main reason is that it’s full of specific actions to take, not just abstract ideals. I’ve seen many engineers and analysts pick up these habits and do high quality work with them. I'd like to share the contents of that document in this blog post.

The advice is organized into three general areas:

- *Technical*: Ideas and techniques for how to manipulate and examine your data.
- *Process*: Recommendations on how you approach your data, what questions to ask, and what things to check.
- *Social*: How to work with others and communicate about your data and insights.





Technical

Look at your distributions

While we typically use summary metrics (means, median, standard deviation, etc.) to communicate about distributions, you should usually be looking at a much richer representation of the distribution. Something like histograms, CDFs, Q-Q plots, etc. will allow you to see if there are important interesting features of the data such as multi-modal behavior or a significant class of outliers that you need to decide how to summarize.

Consider the outliers

You should look at the outliers in your data. They can be canaries in the coal mine for more fundamental problems with your analysis. It's fine to exclude them from your data or to lump them together into an "Unusual" category, but you should make sure you know why data ended up in that category. For example, looking at the queries with the lowest click-through rate (CTR) may reveal clicks on elements in the user interface that you are failing to count. Looking at queries with the highest CTR may reveal clicks you should not be counting. On the other hand, some outliers you will never be able to explain so you need to be careful in how much time you devote this.

Report noise/confidence

First and foremost, we must be aware that randomness exists and will fool us. If you aren't careful, you will find patterns in the noise. Every estimator that you produce should have a notion of your confidence in this estimate attached to it. Sometimes this will be more formal and precise (through techniques such as confidence intervals or credible intervals for estimators, and p-values or Bayes factors for conclusions) and other times you will be more loose. For example if a colleague asks you how many queries about frogs we get on Mondays, you might do a quick analysis looking and a couple of Mondays and report "usually something between 10 and 12 million" (not real numbers).

Look at examples

Anytime you are producing new analysis code, you need to look at examples of the underlying data and how your code is interpreting those examples. It's nearly impossible to produce working analysis code of any complexity without this. Your analysis is removing lots of features from the underlying data to produce useful summaries. By looking at the full complexity of individual examples, you can gain confidence that

useful summaries. By looking at the full complexity of individual examples, you can gain confidence that your summarization is reasonable.

You should be doing stratified sampling to look at a good sample across the distribution of values so you are not too focussed on the most common cases.

For example, if you are computing Time to Click, make sure you look at examples throughout your distribution, especially the extremes. If you don't have the right tools/visualization to look at your data, you need to work on those first.

Slice your data

Slicing means to separate your data into subgroups and look at the values of your metrics in those subgroups separately. In analysis of web traffic, we commonly slice along dimensions like mobile vs. desktop, browser, locale, etc. If the underlying phenomenon is likely to work differently across subgroups, you must slice the data to see if it is. Even if you do not expect a slice to matter, looking at a few slices for internal consistency gives you greater confidence that you are measuring the right thing. In some cases, a particular slice may have bad data, a broken experience, or in some way be fundamentally different.

Anytime you are slicing your data to compare two groups (like experiment/control, but even time A vs. time B comparisons), you need to be aware of mix shifts. A mix shift is when the amount of data in a slice is different across the groups you are comparing. Simpson's paradox and other confusions can result. Generally, if the relative amount of data in a slice is the same across your two groups, you can safely make a comparison.

Consider practical significance

With a large volume of data, it can be tempting to focus solely on statistical significance or to hone in on the details of every bit of data. But you need to ask yourself, "Even if it is true that value X is 0.1% more than value Y, does it matter?" This can be especially important if you are unable to understand/categorize part of your data. If you are unable to make sense of some user agents strings in our logs, whether it's 0.1% of 10% makes a big difference in how much you should investigate those cases.

On the flip side, you sometimes have a small volume of data. Many changes will not look statistically significant but that is different than claiming it is "neutral". You must ask yourself "How likely is it that there is still a practically significant change"?

Check for consistency over time

One particular slicing you should almost always employ is to slice by units of time (we often use days, but

other units may be useful also). This is because many disturbances to underlying data happen as our systems evolve over time. Typically the initial version of a feature or the initial data collection will be checked carefully, but it is not uncommon for something to break along the way.

Just because a particular day or set of days is an outlier does not mean you should discard it. Use the data as a hook to find a causal reason for that day being different before you discard it.

The other benefit of looking at day over day data is it gives you a sense of the variation in the data that would eventually lead to confidence intervals or claims of statistical significance. This should not generally replace rigorous confidence interval calculation, but often with large changes you can see they will be statistically significant just from the day-over-day graphs.

Process

Separate Validation, Description, and Evaluation

I think about about exploratory data analysis as having 3 interrelated stages:

1. *Validation or Initial Data Analysis*: Do I believe data is self-consistent, that the data was collected correctly, and that data represents what I think it does? This often goes under the name of "sanity checking". For example, if manual testing of a feature was done, can I look at the logs of that manual testing? For a feature launched on mobile devices, do my logs claim the feature exists on desktops?
2. *Description*: What's the objective interpretation of this data? For example, "Users do fewer queries with 7 words in them?", "The time page load to click (given there was a click) is larger by 1%", and "A smaller percentage of users go to the next page of results."
3. *Evaluation*: Given the description, does the data tell us that something good is happening for the user, for Google, for the world? For example, "Users find results faster" or "The quality of the clicks is higher."

By separating these phases, you can more easily reach agreement with others. Description should be things that everyone can agree on from the data. Evaluation is likely to have much more debate because you imbuing meaning and value to the data. If you do not separate Description and Evaluation, you are much more likely to only see the interpretation of the data that you are hoping to see. Further, Evaluation tends to be much harder because establishing the normative value of a metric, typically through rigorous comparisons with other features and metrics, takes significant investment.

These stages do not progress linearly. As you explore the data, you may jump back and forth between the stages, but at any time you should be clear what stage you are in.

Confirm expt/data collection setup

Before looking at any data, make sure you understand the experiment and data collection setup. Communicating precisely between the experimentalist and the analyst is a big challenge. If you can look at experiment protocols or configurations directly, you should do it. Otherwise, write down your own understanding of the setup and make sure the people responsible for generating the data agree that it's correct.

You may spot unusual or bad configurations or population restrictions (such as valid data only for a particular browser). Anything notable here may help you build and verify theories later. Some things to consider:

- If it's a features of a product, try it out yourself. If you can't, at least look through screenshots/descriptions of behavior.
- Look for anything unusual about the time range the experiment ran over (holidays, big launches, etc.)

Check vital signs

Before actually answering the question you are interested in (e.g. "Did users use my awesome new feature?") you need to check for a lot of other things that may not be related to what you are interested in but may be useful in later analysis or indicate problems in the data. Did the number of users change? Did the right number of affected queries show up in all my subgroups? Did error rates changes? Just as your doctor always checks your height, weight, and blood pressure when you go on, check your data vital signs to potential catch big problems.
This is one important part of the "Validation" stage.

Standard first, custom second

This is a variant of checking for what shouldn't change. Especially when looking at new features and new data, it's tempting to jump right into the metrics that are novel or special for this new feature. But you should always look at standard metrics first, even if you expect them to change. For example, when adding a brand new UI feature to the search page, you should make sure you understand the impact on standard metrics like clicks on results before diving into the special metrics about this new UI feature. You do this because standard metrics are much better validated and more likely to be correct. If your new, custom metrics don't make sense with your standard metrics, your new, custom metrics are likely wrong.

Measure twice, or more

Especially if you are trying to capture a new phenomenon, try to measure the same underlying thing in multiple ways. Then, check to see if these multiple measurements are consistent. By using multiple measurements, you can identify bugs in measurement or logging code, unexpected features of the underlying data, or filtering steps that are important. It's even better if you can use different data sources

for the measurements.

Check for reproducibility

Both slicing and consistency over time are particular examples of checking for reproducibility. If a phenomenon is important and meaningful, you should see it across different user populations and time. But reproducibility means more than this as well. If you are building models of the data, you want those models to be stable across small perturbations in the underlying data. Using different time ranges or random sub-samples of your data will tell you how reliable/reproducible this model is. If it is not reproducible, you are probably not capturing something fundamental about the underlying process that produced this data.

Check for consistency with past measurements

Often you will be calculating a metric that is similar to things that have been counted in the past. You should compare your metrics to metrics reported in the past, even if these measurements are on different user populations. For example, if you are looking at measuring search volume on a special population and you measure a much larger number than the commonly accepted number, then you need to investigate. Your number may be right on this population, but now you have to do more work to validate this. Are you measuring the same thing? Is there a rational reason to believe these populations are different? You do not need to get exact agreement, but you should be in the same ballpark. If you are not, assume that you are wrong until you can fully convince yourself. Most surprising data will turn out to be a error, not a fabulous new insight.
New metrics should be applied to old data/features first

If you gather completely new data and try to learn something new, you won't know if you got it right. When you gather a new kind of data, you should first apply this data to a known feature or data. For example, if you have a new metric for user satisfaction, you should make sure it tells you your best features help satisfaction. Doing this provides validation for when you then go to learn something new.

Make hypotheses and look for evidence

Typically, exploratory data analysis for a complex problem is iterative. You will discover anomalies, trends, or other features of the data. Naturally, you will make hypotheses to explain this data. It's essential that you don't just make a hypothesis and proclaim it to be true. Look for evidence (inside or outside the data) to confirm/deny this theory. For example, If you believe an anomaly is due to the launch of some other feature or a holiday in Katmandu, make sure that the population the feature launched to is the only one affected by the anomaly. Alternatively, make sure that the magnitude of the change is consistent with the expectations of the launch.

Good data analysis will have a story to tell. To make sure it's the right story, you need to tell the story to yourself, predict what else you should see in the data if that hypothesis is true, then look for evidence that it's wrong. One way of doing this is to ask yourself, "What experiments would I run that would validate/invalidate the story I am telling?" Even if you don't/can't do these experiments, it may give you ideas on how to validate with the data that you do have.

The good news is that these hypotheses and possible experiments may lead to new lines of inquiry that transcend trying to learn about any particular feature or data. You then enter the realm of understanding not just this data, but deriving new metrics and techniques for all kinds of future analyses.

Exploratory analysis benefits from end to end iteration

When doing exploratory analysis, you should strive to get as many iterations of the whole analysis as possible. Typically you will have multiple steps of signal gathering, processing, modelling, etc. If you spend too long to get the very first stage of your initial signals perfect you are missing out on opportunities to get more iterations in the same amount of time. Further, when you finally look at your data at the end, you may make discoveries that change your direction. Therefore, your initial focus should not be on perfection but on getting something reasonable all the way through. Leave notes for yourself and acknowledge things like filtering steps and data records that you can't parse/understand, but trying to get rid of all of them is a waste of time at the beginning of exploratory analysis.

Social

Data analysis starts with questions, not data or a technique

There's always a reason that you are doing some analysis. If you take the time to formulate your needs as questions or hypotheses, it will go a long way towards making sure that you are gathering the data you should be gathering and that you are thinking about the possible gaps in the data. Of course, the questions you ask can and should evolve as you look at the data. But analysis without a question will end up aimless.

Further, you have to avoid the trap of finding some favorite technique and then only finding the parts of problems that this technique works on. Again, making sure you are clear what the questions are will help you avoid this.

Acknowledge and count your filtering

Almost every large data analysis starts by filtering the data in various stages. Maybe you want to consider only US users, or web searches, or searches with a result click. Whatever the case, you must

- Acknowledge and clearly specify what filtering you are doing

- Count how much is being filtered at each of your steps

Often the best way to do the latter is to actually compute all your metrics even for the population you are excluding. Then you can look at that data to answer questions like "What fraction of queries did my filtering remove?"

Further, looking at examples of what is filtered is also essential for filtering steps that are novel for your analysis. It's easy to accidentally include some "good" data when you make a simple rule of data to exclude.

Ratios should have clear numerator and denominators

Many interesting metrics are ratios of underlying measures. Unfortunately, there is often ambiguity of what your ratio is. For example, if I say click-through rate of a site on search results, is it:

- "# clicks on site' / '# results for that site'
- '# search result pages with clicks to that site' / '# search result pages with that site shown'

When you communicate results, you must be clear about this. Otherwise your audience (and you!) will have trouble comparing to past results and interpreting a metric correctly.

Educate your consumers

You will often be presenting your analysis and results to people who are not data experts. Part of your job is to educate them on how to interpret and draw conclusions from your data. This runs the gamut from making sure they understand confidence intervals to why certain measurements are unreliable in your domain to what typical effect sizes are for "good" and "bad" changes to understanding population bias effects.

This is especially important when your data has a high risk of being misinterpreted or selectively cited. You are responsible for providing the context and a full picture of the data and not just the number a consumer asked for.

Be both skeptic and champion

As you work with data, you must be both the champion of the insights you are gaining as well as a skeptic. You will hopefully find some interesting phenomena in the data you look at. When you have an interesting phenomenon you should ask both "What other data could I gather to show how awesome this is?" and "What could I find that would invalidate this?". Especially in cases where you are doing analysis for someone who really wants a particular answer (e.g. "My feature is awesome") you are going to have to play the skeptic to avoid making errors.

Share with peers first, external consumers second

A skilled peer reviewer can provide qualitatively different feedback and sanity-checking than the consumers of your data can, especially since consumers generally have an outcome they want to get. Ideally, you will have a peer that knows something about the data you are looking at, but even a peer with just experience looking at data in general is extremely valuable. The previous points suggested some ways to get yourself to do the right kinds of sanity checking and validation. But sharing with a peer is one of the best ways to force yourself to do all these things. Peers are useful at multiple points through the analysis. Early on you can find out about gotchas your peer knows about, suggestions for things to measure, and past research in this area. Near the end, peers are very good at pointing out oddities, inconsistencies, or other confusions.

Expect and accept ignorance and mistakes

There are many limits to what we can learn from data. Nate Silver makes a strong case in The Signal and the Noise that only by admitting the limits of our certainty can we make advances in better prediction. Admitting ignorance is a strength but it is not usually immediately rewarded. It feels bad at the time, but will ultimately earn you respect with colleagues and leaders who are data-wise. It feels even worse when you make a mistake and discover it later (or even too late!), but proactively owning up to your mistakes will translate into credibility. Credibility is the key social value for any data scientist.

Closing thoughts

No short list of advice can be complete even when we break through the barrier of the Top 10 List format (for those of you who weren't counting, there are 24 here). As you apply these ideas to real problems, you'll find the habits and techniques that are most important in your domain, the tools that help you do those analyses quickly and correctly, and the advice you wish were on this list. Make sure you share what you've learned so we can all be better data scientists.

I'd like to thank everyone who provided insight that went into this document, but especially Diane Tang, Rehan Khan, Elizabeth Tucker, Amir Najmi, Hilary Hutchinson, Joel Darnauer, Dale Neal, and Aner Ben-Artzi.

Matt *November 4, 2016 at 7:11 PM*