What Users Don't Expect about Exploratory Data Analysis on Approximate Query Processing Systems

Dominik Moritz
University of Washington
domoritz@cs.washington.edu

Danyel Fisher
Microsoft Research
danyelf@microsoft.com

ABSTRACT

Pangloss implements "Optimistic Visualization", a method that gives analysts confidence to use approximate results for exploratory data analysis. In this paper, we outline how analysts' experience with an approximate visualization system did not match their intuitions. These observations have implications for the design of future data exploration systems that expose uncertainty. We also describe requirements for approximate query engines to enable the next generation of exploratory visualization systems.

CCS CONCEPTS

• **Human-centered computing** → **Graphical user interfaces**;
• **Information systems** → *Database management system engines*;

KEYWORDS

approximate query processing, user experience, visualization

ACM Reference format:
Dominik Moritz and Danyel Fisher. 2017. What Users Don't Expect about Exploratory Data Analysis on Approximate Query Processing Systems. In *Proceedings of HILDA'17, Chicago, IL, USA, May 14, 2017,* 4 pages.
DOI: http://dx.doi.org/10.1145/3077257.3077258

1 INTRODUCTION

Approximate Query Processing (AQP) allows users to exchange precision for query speed on large datasets and complex queries: that is, it returns rapid, uncertain answers to aggregate queries. This is invaluable for exploratory visualization systems, which struggle to deliver results quickly when the data is sufficiently large that precise database queries take too long.

These approximate results, however, can be less intuitive for users to understand than the precise results that are more traditionally returned by databases and visualized by analysis tools. Users may be misled into making decisions based on incomplete or incorrect information. Moreover, for many aggregation queries it is difficult or impossible to find a close approximation or guarantee that errors are bounded.

In Pangloss [5], we approach challenges with approximate queries from a user experience perspective. Pangloss is a two-phase big data system: it allows users to explore their data through fast, approximate queries; users can then request precise responses with slow queries over the full data. In the first phase, the engine that drives Pangloss, called "Sample+Seek" [3], returns approximate results in interactive time with an overall uncertainty level.

This is a new experience for users. Pangloss requires users to work with a new uncertainty model, with two-round queries, and to directly face the implications of uncertainty. We see all of these as important and valuable changes in a world that increasingly embraces AQP; however, they can be surprising for users. These user stories will allow us to begin to design for interacting with new AQP systems.

In this paper, we summarize some of our insights with designing the user experiences for Pangloss, and what we learned from analysts using our system. The more detailed conference paper [5] outlines the broader user experience and design of Pangloss. In this paper, we describe some general issues and challenges we saw with users exploring uncertain data; in addition, we look at ways that their use can help us shape the design of future AQP system.

2 BRIEF OVERVIEW OF PANGLOSS

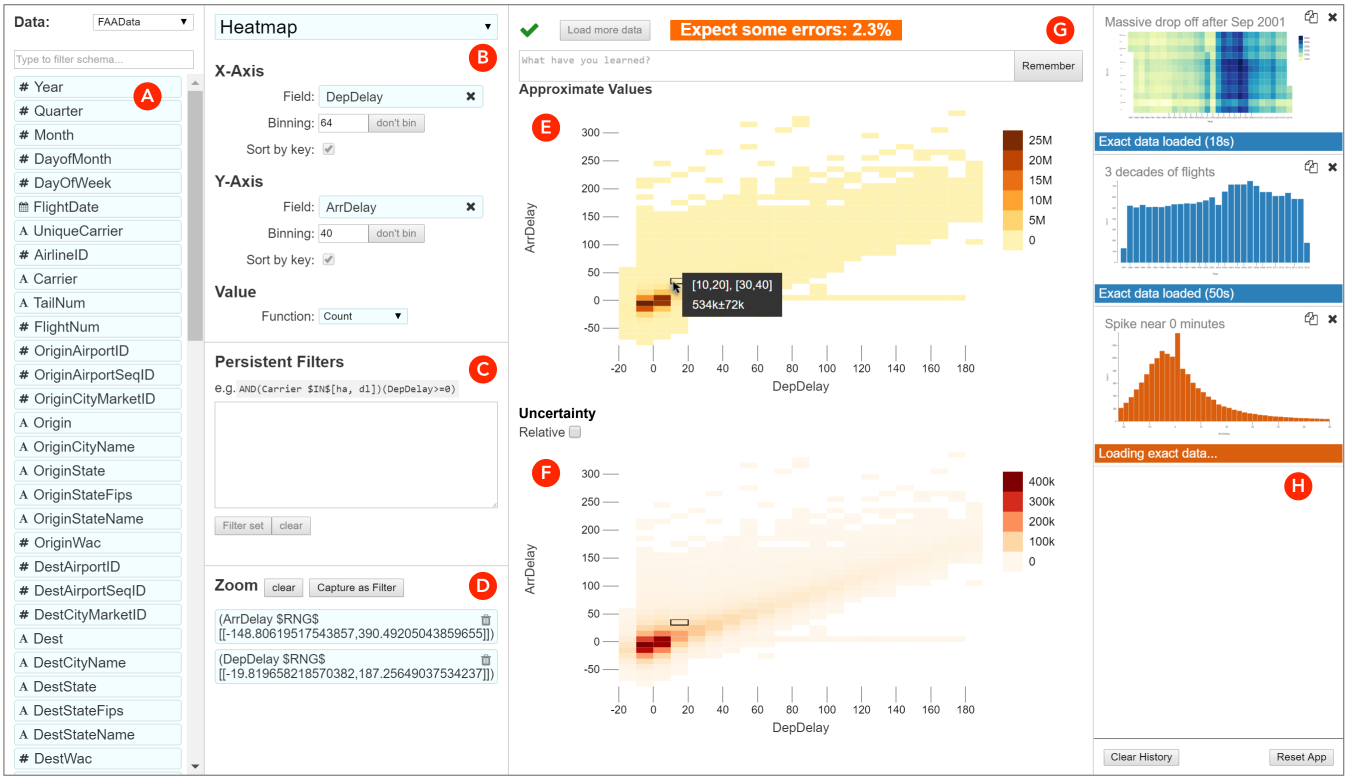
Pangloss, as shown in Figure 1, is an exploratory visualization system, similar to familiar tools such as Tableau. Users can create 1D or 2D histograms and bar charts by dragging and dropping fields from the schema onto the chart specification forms. The system initially shows approximate visualizations; the analyst may request that Pangloss compute the precise query result asynchronously by pressing the "remember" button. When the precise data is computed, the analyst can see the precise data to confirm or challenge their observation. This is the main idea behind Optimistic Visualization. It gives analysts the confidence that they will know if the approximation was significantly different from the precise result.

3 UNCERTAINTY IN PANGLOSS

Unlike many other systems, Pangloss uses **distribution uncertainty**. Distribution uncertainty is different from familiar confidence intervals: it is a metric of uncertainty across all groups in the result. It is defined as the expected distance (e.g., sum of distances, Euclidean, etc.) between the normalized distributions of the approximate answer and the precise one. Using a distribution uncertainty captures the fact that uncertainties in different groups are not independent: the system instead claims that collectively, the errors are not far. One implication of this is that the uncertainty implied by confidence intervals is always higher or equal to the distribution uncertainty.

HILDA'17, Chicago, IL, USA

978-1-4503-5029-7/17/05…$15.00
DOI: http://dx.doi.org/10.1145/3077257.3077258



Data: FAAData

X-Axis
Field: DepDelay
Binning: 64
Sort by key: ☑

Y-Axis
Field: ArrDelay
Binning: 40
Sort by key: ☑

Value
Function: Count

Persistent Filters
e.g. `AND(Carrier $IN$[ha, dl])(DepDelay>=0)`

Zoom
(ArrDelay RNG [[-148.80619517543857,390.49205043859655]])
(DepDelay RNG [[-19.819658218570382,187.25649037534237]])

Expect some errors: 2.3%

[10,20], [30,40]
534k±72k

Uncertainty
Relative □

Massive drop off after Sep 2001 — Exact data loaded (18s)
3 decades of flights — Exact data loaded (50s)
Spike near 0 minutes — Loading exact data...

Figure 1: The Pangloss UI, exploring a flight delay dataset, with a list of draggable fields (A), which can be placed on the view specification (B), filters (C), and zoom specification (D) to describe the view. The view shows both a visualization of the approximate data (E) and the uncertainty (F). Users may press the “remember” button (G) to store the view in the history (H) and request an offline compute of the precise result. Two precise results are ready (in blue), while a third is loading (in orange).

4 UNDERSTANDING APPROXIMATION ERROR

In order to think about how users interact with error, we refer to **approximation error** as the difference between the estimate and the true value. Uncertainty estimates are most effective when approximation errors are small. Ideally, the estimate is close to the true value: approximation errors are below the perceptual threshold, and so users will draw only correct conclusions from the approximation. In contrast, when approximation errors are large, analysts might draw incorrect conclusions from the data and would probably prefer to use precise results.

Of course, we cannot know the approximation error until after the precise answer has been computed. Uncertainty is meant to be a predictor of the approximation error: we hope that high uncertainty would help see cases where approximation error is likely to be high. However, in a recent study, Agarwal et al. examined logs of 70,000 approximate queries from Facebook and found a large fraction had error estimates that were too wide or too narrow [1]. Figure 2 shows the relationship between uncertainty and approximation error. When the uncertainty is high, analysts cannot make confident decisions, and so the approximation is less useful.

The most dangerous area is the lower right of this chart, where the true error is large but the analyst expected small errors because the uncertainty was low. Optimism pays off in this area: running a two-round query can assure analysts that they have a good result.

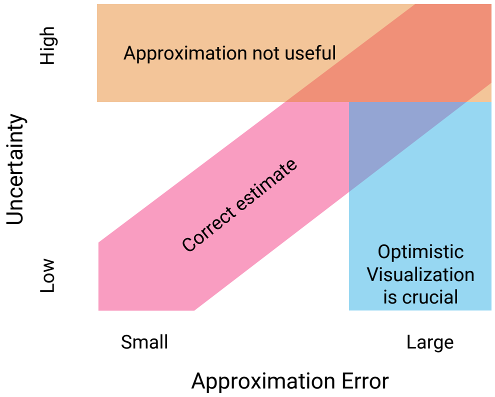


Figure 2: Relationship between uncertainty (estimated approximation error) and approximation error (true error of the approximation). When the uncertainty is high, analysts cannot draw useful conclusions from the results. When approximation error is larger than expected, approximation failed and analysts need to be informed.

5 EXPERIENCES WITH AQP

In designing Pangloss and the subsequent case studies we encountered unexpected hurdles that we have not seen discussed in the literature. We tested our system at Microsoft with five data analysts who analyzed a large dataset about flights and three teams that brought in their own data; we report on their experience with the system in some detail [5]. Here, we discuss issues that came up in those conversations not related directly to the Pangloss UI.

5.1 Representation Challenges

Representing Distribution Uncertainty. We do not know of other systems that visualize distribution uncertainty; we resorted to showing it as a number above the charts (Figure 1 above E). Other possible representations might include simulating possible results that are compliant with these bounds, showing them to users laid out in space or over time as hypothetical outcome plots (HOPs) [4]. There is an opportunity for research to continue to explore visualizations or interactions with visualizations that consider global uncertainty measures.

Uncertainty for Heatmaps. Pangloss computes a local uncertainty for each data item, although this is a very conservative estimate, far wider than the distribution uncertainty. For bar charts, Pangloss draws the confidence intervals directly on the bar chart. As there is no standard way to show confidence intervals for heatmaps, Pangloss instead displays a second parallel heatmap that shows the uncertainty (Figure 1 F). Our users struggled to connect values and uncertainty although we provided tooltips and highlighted corresponding cells: a separate chart for uncertainty was too easy to ignore. General visualization techniques to show values and uncertainty in heatmaps such as Layercake [2] might help, but this clearly is an area that requires additional exploration.

Visualizing differences. Just as we need to continue to develop visualizations to cue uncertainty, Pangloss also requires difference visualizations, so that analysts can judge whether there was a significant difference between the approximation and the precise result that changes their observation.

The problem of comparing increasingly accurate results for the same query also comes up in progressive visualization systems. However, it has not been addressed with explicit difference visualization. Instead, progression is implicitly visualized with animation.

Access to samples. Visualizations in any Big Data system show aggregations because showing all rows in infeasible. AQP systems don't provide access to row level data because aggregates are computed from samples. Our participants nonetheless asked to see examples of items in a group. A complete row of the data provides context across all dimensions when a chart is typically limited to two or three dimensions. Future AQP systems could return individual samples that match queries alongside aggregate values.

5.2 Multiple Rounds of Queries

When building a visualization in Pangloss, a user may issue many different queries: every visual manipulation of the Pangloss UI – selecting data fields, zooming the visualization, and filtering and brushing the data – correspond to queries issued against the server. Sample+Seek's high responsiveness returns query results in interactive time. A user may then issue a more-precise slower query. In our discussion of user experience, we found user issues and opportunities for optimizations both with the multiple queries in an interactive session, and the precise query.

Even in a highly responsive system like Sample+Seek, the design of the user interface can affect the query load sent to the engine. In this section, we outline several challenges for both the back-end and the user-interface for interactive analysis systems that issue multiple rounds of queries.

Optimizing Sessions. Consecutive queries tend to be similar because users refine their questions. However, current AQP engines do not have a notion of an exploration session and thus have to recompute the samples for each query. New engines could improve performance by preloading or reusing previous samples or results.

SQL for Histograms. Traditionally, SQL systems have emphasized optimizing low-level queries. Future AQP systems are likely to find it valuable to optimize the high-level query workloads from exploratory analysis, which might include pushing higher-level queries to the server. To support Pangloss, we added binning to Sample+Seek's API. Separate queries for the data range and the aggregation in an AQP system may use different samples. Generally, the assumption that we can compose a more complex query of many small queries does not hold in an AQP system.

Performance in Sparse Areas. As a user zooms into a more detailed area, the filters cause the data to be sparse. Even Sample+Seek becomes somewhat less interactive when it is hard to collect enough samples. Pangloss is built with a specific time budget: a query will never return in more than a particular number of milliseconds, although it may return an inexact distribution approximation. There are opportunities to explore fine-grained tradeoffs around collecting more samples and truncating queries. A user experience might even offer the user more accurate results, which requires restartable queries or progressive loading.

Changing Domains. Sample+Seek chooses samples to ensure the distribution uncertainty falls under a threshold – which means that when the user changes the domain, so do the set of samples. The samples will be picked only from the new domain. Therefore, filtering a value can lead to other estimates changing –indeed, it means that other elements in the domain can appear! To understand this phenomenon, consider a long-tailed domain. Samples were more likely to come from the head, and so some items in the tail were missed. Therefore, if the head of the distribution is filtered out, more samples will come from the tail.

Changing Values. When the user zooms into a Pangloss chart, the domain of the data changes; which issues a new query. Each new query has new samples to maintain local error bounds, which means that the estimated values for each group may change. This leads to multiple simultaneous changes that can be confusing to users; a change of the scale and a change of the value. Visual consistency across multiple visualizations is a growing concern in the visualization community; it recurs here with approximation.

In one case, a user filtering through his dataset found an example where a more-filtered measure had a larger count than the less-filtered count: it was like learning that there are more male bankers than there are bankers. While this is clearly a false conclusion, estimation artifacts can lead to these errors.

5.3 Cuing User Memory

Sample+Seek can compute the answer to a query twice: once as an approximation, once as a final precise result. The user might derive an insight from the first, and confirm it on the second some time later. This turns out to be a challenge for users to remember. Pangloss assists this by comparing the two rounds.

Aligning data domains. We can only directly compare two query results if the data domain matches. However, a more precise result may have additional groups or the order of groups may change. Thus, to create a difference visualization, we need a comprehensive set of visual cues for these kinds of qualitative changes.

In binned charts the intersection of domains of different progressions may be empty if the bounds change. In Pangloss we overcome this issue by computing the binning offset and bucket size from the range of values of the approximate result and reuse the same parameters for subsequent binnings.

Importance of annotations. Participants of our study repeatedly praised the option to take notes on the observations when they remembered a view. It helped them to remember what exactly was the thing the wanted to check. In the future, we should develop a vocabulary of annotations – both textual and visual – to help users express what they care about in a visualization. These annotations can also help the software to know whether a more targeted query is appropriate and whether the changes in a precise result are relevant.

5.4 Semantics of Selection

Filter In and Out. Filtering is a very different operation in a situation with multiple rounds of queries. In most applications, "filter out" is semantically equivalent to "filter in", and users choose merely on the number of clicks or the convenience of selection. In a sample-based system, if I see a chart like Figure 3 with groups A through F, and filter out A & B, I might suddenly learn about "G", too! Thus, filters must be carefully written to be appropriately restrictive: do I want "everything except A-B" (which includes G), or do I want "only C-D-E-F"?

Brushing. Brushing gestures, similarly, become ambiguous when the data might change. In Figure 3, selecting B-D might mean any of the following things: "Bars B-D", "Any bar that isn't A, E, or F", "Any value between 4 and 8", and "Any value under 10 and over 2." When the precise query comes in, the system must have an interpretation that is consistent.

This semantic confusion will also occur in any system where the data can change over time.

We had to develop new semantics for filters to describe these actions. In our Pangloss implementation, filters are exclusive. When users brush over continuous domains, we reserve the left axis of a chart as a lower bound if a user brushes all the way to the end; otherwise, we assume that a user only wants to filter the region they specify.

Choosing Filters Users also found it challenging to choose filters which would support their conclusions. For example, a user might generate a chart of airports by number of flights, and want to confirm which are the top three busiest airports. Intuitively, users would filter down to those top three items, keeping only them in view, and issue the precise query. Unfortunately, those items might not be "top three" anymore.

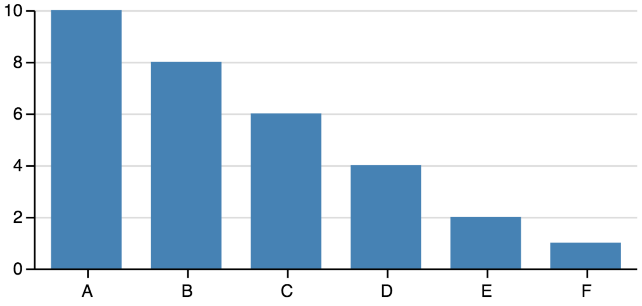


Figure 3: A bar chart with six groups. Without approximation "B, C, and D" is the same as "not A, E, and F". When the data is the result of a query over samples, the two statements are not necessarily equivalent.

6 CONCLUSIONS

Approximate query processing systems could soon become a tool that data analysts use when exploring massive datasets. Trading off a bit of accuracy against a massive performance boost is promising but many open challenges remain: current UI tools make assumptions that do not hold for approximations and data analysts are not used to working with uncertain data. Similarly, existing AQP engines are not designed for exploratory data analysis.

Researchers need to continue to investigate the user experience issues that analysts will face when AQP systems become widely available. We need a systematic evaluations of uncertainty visualizations for complex visualizations and new uncertainty models. Next generation AQP engines can better support exploratory visualization systems if they support binning, access to samples, and feedback. New engines can increase performance by utilizing patterns in exploration sessions and limits of human perception; maybe even in streaming data contexts where the data is not constant and samples cannot be computed offline.

Working on these challenges requires close collaboration of user researchers with visualization designers, database and statistical experts. These collaborations will be invaluable for the development of tools that best benefit the users of future AQP systems.

REFERENCES

[1] Sameer Agarwal, Henry Milner, Ariel Kleiner, Ameet Talwalkar, Michael Jordan, Samuel Madden, Barzan Mozafari, and Ion Stoica. 2014. Knowing when You'Re Wrong: Building Fast and Reliable Approximate Query Processing Systems. In *Proc. SIGMOD '14*. ACM. DOI: http://dx.doi.org/10/f3tvrz

[2] Michael Correll, Adam L Bailey, Alper Sarikaya, David H O'Connor, and Michael Gleicher. 2015. LayerCake: a tool for the visual comparison of viral deep sequencing data. *Bioinformatics* (2015). DOI: http://dx.doi.org/bzj7

[3] Bolin Ding, Silu Huang, Surajit Chaudhuri, Kaushik Chakrabarti, and Chi Wang. 2016. Sample + Seek: Approximating Aggregates with Distribution Precision Guarantee. In *Proc. SIGMOD '16*. ACM. DOI: http://dx.doi.org/10/f3tvr2

[4] Jessica Hullman, Paul Resnick, and Eytan Adar. 2015. Hypothetical Outcome Plots Outperform Error Bars and Violin Plots for Inferences about Reliability of Variable Ordering *(PloS one)*. DOI: http://dx.doi.org/10/f3tvsd

[5] Dominik Moritz, Danyel Fisher, Bolin Ding, and Chi Wang. 2017. Trust, but Verify: Optimistic Visualizations of Approximate Queries for Exploring Big Data. In *Proc. CHI '17*. ACM. DOI: http://dx.doi.org/10.1145/3025453.3025456