Trust, but Verify: Optimistic Visualizations of Approximate Queries for Exploring Big Data

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

Danyel Fisher
Microsoft Research
danyelf@microsoft.com

Bolin Ding, Chi Wang
DMX, Microsoft Research
bolind@microsoft.com,
chiw@microsoft.com

ABSTRACT
Analysts need interactive speed for exploratory analysis, but big data systems are often slow. With sampling, data systems can produce approximate answers fast enough for exploratory visualization, at the cost of accuracy and trust. We propose *optimistic visualization*, which approaches these issues from a user experience perspective. This method lets analysts explore approximate results interactively, and provides a way to detect and recover from errors later. *Pangloss* implements these ideas. We discuss design issues raised by optimistic visualization systems. We test this concept with five expert visualizers in a laboratory study and three case studies at Microsoft. Analysts reported that they felt more confident in their results, and used optimistic visualization to check that their preliminary results were correct.

ACM Classification Keywords
H.5.2. Information Interfaces and Presentation: UI; H.2.4. Database Management: Systems

Author Keywords
Data visualization; exploratory analysis; optimistic visualization; approximation; uncertainty

INTRODUCTION
Data analysts want to be able to derive insights from increasingly large datasets. Exploratory visualization, however, runs into an obstacle where the scale of the data is sufficiently large that a screen cannot render each point, and where database queries would take a long time to return. By sampling the dataset, though, we can create a visualization with approximate values in interactive time. This is known as *Approximate Query Processing* (AQP).

There are several well-known challenges with approximations. The most critical of these is trust: approximate values can be, by their nature, possibly incorrect. In an exploratory visualization, an analyst might see dozens of visualizations that are accurate 95% of the time. Can an analyst trust an approximation with a business-critical decision?

CHI 2017, May 06 - 11, 2017, Denver, CO, USA

ISBN 978-1-4503-4655-9/17/05…$15.00

DOI: http://dx.doi.org/10.1145/3025453.3025456

In this paper, rather than addressing the problems with AQP from an algorithmic or systems perspective, we formulate them as user experience problems. What user experience would enable analysts to gain the benefits of approximate queries, while still being able to trust the results?

We propose an approach which we call *optimistic visualization*. Optimistic visualization produces approximate results quickly, and computes precise results in the background. The analyst can make observations on the approximation, and later check them against the precise results.

We call this approach "optimistic" because the analyst expects the approximation to be very close to the precise value; in those rare cases when there is a significant difference between the approximate and precise results, the analyst can decide which parts of the exploration have to be redone. Optimism provides a way to detect and recover from errors, and so increases confidence in working with samples. The technique can be combined with other approximation techniques such as confidence intervals and online aggregation.

We present *Pangloss*, an optimistic visualization tool based on AQP. With it, analysts can rapidly explore very large multi-dimensional datasets by grouping, aggregating, and filtering. Working in a sample-based system affects the user experience of visual data exploration. We describe the design decisions that went into the system. We validated our decisions by running a user study with five participants exploring a 170 million row dataset about flight delays; we then deployed our prototype system to three data scientists using their own data.

BACKGROUND AND RELATED WORK
The concept of optimistic visualization builds on past research on data exploration, AQP, and uncertainty visualization. We first discuss the importance of rapid iteration in exploratory data analysis, and then the ways in which it changes when working with very large datasets. Last, we focus specifically on sample-based queries and visualization.

Exploratory Visualization
Exploratory data analysis (EDA), a term coined by Tukey [38], is broadly understood as a process of examining multi-dimensional data by looking at the distributions and correlations of fields. As Card et al. [5] note, this process prizes iteration and speed of exploration; an analyst might look at dozens or hundreds of graphs as they get to know the data.

The process of moving through these dimensions is iterative. An analyst begins with a broad question, and creates views that address some part of it. This view can inform a more-specific question, leading them to create another view to address that question [42, 35, 34]. These increasingly-specific questions require analysts to change representations, to filter the data by zooming or filtering views, and to choose new fields to explore. Some of these views will contain interesting insights; others will be dead ends with less value. When the analyst has sufficiently addressed the broad question and follow up questions, they often continue with a new broad question and chains of specific follow up questions.

Visualization tools—whether point-and-click, such as Tableau (an extension of Polaris [37]) and PowerBI, or programming, like Matplotlib—support this process with tools that allow users to rapidly specify and refine their visualizations.

Each step in this process involves generating *observations* of the data. Optimistic visualization helps analysts confirm (or challenge) their observations. An observation is a single fact about the data; it is the unit of knowledge that allows an analyst to move on to the next step of their analysis [43]. For example, when examining a dataset of flight data, an observation might be "Delta is the airline with the most flights in the dataset." It is a more modest unit than the insights that the research community has focused on as the outcome of the analysis process. An insight can bring in external context and the results of a number of queries; an example might be "the biggest airlines have trouble with congestion near the holidays, while smaller airlines do not" [43, 28].

The visualization system must be fast enough to enable iteration. Liu and Heer show that analysts lose effectiveness when a result takes more than 500 ms to return [24]; Norman argues that when a computer operation takes more than a second, users lose their flow of thought [27].

Big Data Visualization

These requirements for responsiveness become urgent when dealing with large datasets. As Fisher [12] and Godfrey et al. [14] outline, when dataset sizes exceed even a few million records, analysts run into two fundamental issues: visual scalability and data processing scalability.

It is impractical to display every element of a large dataset: the number of records may far exceed the available pixels. For example, drawing raw data in a scatterplot without aggregation leads to overplotting—drawing many points in the same place—and visual clutter. The data can be grouped by a dimension, however, and a single aggregate measure computed for each group. The simplest such aggregate visualization is a bar chart, in which each bar represents the aggregated value of a group. Elmqvist [9] outlines visualizations that work with aggregate data; Wickham proposes the bin-summarize-smooth framework [40] as a general strategy for visualizing big data.

Data retrieval and processing are the other major bottleneck. Handling very large datasets can be comparatively slow. Analysts sometimes resort to an offline process: formulating and submitting a query, waiting for a result, and then formulating a follow up question. This is not only frustrating but requires analysts to carefully design their queries to be worth the wait and the resources.

There are three major technologies to achieve more-responsive queries. Data cubes precompute and store partially-aggregated results; at query time, the system can assemble these partial answers quickly [23, 25, 15]. Unfortunately, these cubes require a designer to select the fields to optimize. Second, the query can be spread across many computers, which assemble an answer [4, 32]. In these distributed *Online Analytical Processing* (OLAP) systems [6], though, network latencies can last into the seconds. The third major approach is to sample the dataset.

Approximate Query Processing

Optimistic visualization is based on the technique of Approximate Query Processing (AQP). In AQP, the tool uses a representative subset, or sample, of the data; the goal is to look at less data more quickly. Tools can estimate the true value of an aggregation function based on that sample. As a simple example, we can approximate the sum of a set of values by computing the sum of 10% of the values and then estimating the true sum to be ten times the aggregate value of the sample. This value is an estimate, and carries some uncertainty, which can be expressed as error bounds. Those bounds widen with the variance of the data, and narrow with the square root of the size of the sample.

Some tools create a sample of the data before the user begins their analysis. In these systems, the precision of the approximation greatly diminishes as the analyst filters away more records. For example, every record in a census can help the approximation for "average age", but far fewer will be helpful for "average age of unemployed men living in Aspen, Colorado". Choosing a sample that is large enough to lead to statistically meaningful results while maintaining interactive response times is important.

Sampling can be integrated directly into databases [30]; other systems build on different sampling and estimation methods [2, 8, 22]. These systems pick a sample and compute a result along with estimated error bounds; the analyst may choose either a maximum amount of time that the query runs, or desired error bounds. Interactive systems tend to use time bounds to get a best-effort approximation within that time bound.

Progressive Visualization with Online Aggregation

Rather than forcing the user to settle for a fixed-size sample, or wait for the system to reach a fixed level of precision, Online Aggregation (OLA) picks ever-growing samples and displays results to the user; when the analyst determines the visualization has tight-enough bounds, they can end the process. OLA computes aggregations and confidence intervals and returns them to the user. Hellerstein et al. [19] first suggested the idea; it has been adopted by the visualization community as a "progressive analytics" approach [13, 10, 36, 33, 39].

Optimistic visualization can be seen as an asynchronous form of progressive sampling; it places the updates in the background, allowing the analyst to continue their analysis without

watching for updates. The CONTROL project [18] noted that progressiveness adds costs; e.g. "ripple joins" [16] require multiple passes over the data, and so may take longer to reach a precise result. Pangloss can use existing, highly-optimized database systems for the precise results.

Visualizing Approximations

There are a number of techniques for communicating approximate query results [31]. Commonly used methods are confidence intervals [26], visualizing distributions [41], or visualizing possible instances of the underlying statistical model [20]. Researchers are less certain of ways to visualize uncertainty on some visualization types such as heatmaps.

Even with the help of visualization, users struggle to correctly interpret uncertainty and can draw incorrect conclusions [21, 7]. In a visual data exploration where an analyst creates tens or hundreds of visualizations, the "rare" cases when the true values are outside the estimated bounds become likely. Worse, many confidence estimates are inaccurate: 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].

Optimistic visualization ensures that precise results will eventually be available so that the analyst can discover places where estimates were unavailable, hard to understand, or far from the true value.

OPTIMISTIC DATA VISUALIZATION

We address the challenges of sample-based visualization with *optimistic visualization*. In an optimistic visualization system, an analyst begins by constructing a fast, approximate query and seeing results instantly. The analyst may choose to *remember* that query, in which case it will run in the background. While the query is running, the analyst continues their exploration. The system allows the analyst to know when the query is complete; at which point the analyst can verify their observations. The system shows the error between the approximate and precise views.

In most cases, the final result validates their earlier work. Should a past approximation turn out to be inaccurate, however, the analyst must then re-evaluate how much exploration must be redone. There are a number of edge cases for approximation: it is difficult to choose good confidence intervals on some functions, such as percentile measures; some datasets have high enough variances that confidence intervals are extremely wide; and some approximations turn out to be incorrect.

Even in these riskier scenarios, optimistic visualization allows the analyst to feel certain that their final results will confirm whether their assumptions were justified. Without optimism the analyst can only rely on the *uncertainty*—the estimated error of the approximation—to make a decision. The uncertainty should be a good predictor of the *approximation error*; the true error of the approximation. However, only the precise results can confirm this; analysts have to know when the approximation error was significantly larger than the uncertainty.

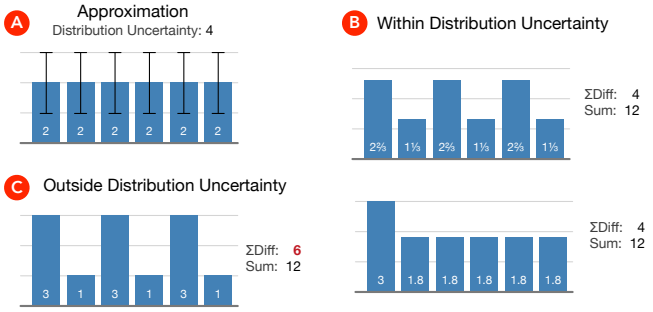

Figure 1. The uncertainty implied by the confidence intervals from (A) is larger than the distribution uncertainty of 4 (here for illustration defined as the sum of absolute, unnormalized differences). (B) are possible instances that stay within both the confidence intervals and the distribution uncertainty. The values for all groups in (C) are within the confidence intervals but the distribution is off by 6. The sum of the values is always 12.

In progressive visualization systems [13, 33, 39] analysts watch progressive updates as more data arrives. During the waiting period, though, values continuously change [11]. Analysts can be distracted by these *dancing bars*; they can also incorrectly assess trends as the confidence intervals converge. Progressive systems require accurate communication of uncertainties so analysts can decide when to stop. However, some forms of uncertainty are difficult to visualize, and true errors can be outside the estimated bounds. Optimistic visualization defers the confirmation and moves the computation of the precise result into the background; analyst can continue their exploration sooner.

An optimistic visualization is most effective when the complete query takes long enough to get in the way of interactivity, but shorter than an analysis session—it is highly effective when a query takes several minutes to return.

THE PANGLOSS SYSTEM

We implemented optimistic visualization in the Pangloss system. Pangloss is a web based UI that queries Sample+Seek [8], an AQP system. Because Sample+Seek has some unique features, we discuss it in some detail before presenting our interface.

Sample+Seek for Approximate Query Processing

Sample+Seek [8] is designed to be highly responsive for aggregate queries on a single table. It incrementally loads more records into the sample until either the uncertainty bound is lower than a predefined threshold or until a timeout. Instead of uniformly sampling records—as many other AQP system do—Sample+Seek uses *measure-biased* sampling, a method that biases sampling according to the aggregation measure. Measure-biased sampling has a tremendous advantage over uniform sampling: fewer samples are necessary for the same accuracy in a visualization. This sampling method has been developed to optimize the *distribution uncertainty*. It is a metric of the uncertainty across all groups in the result, and 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. For example, the

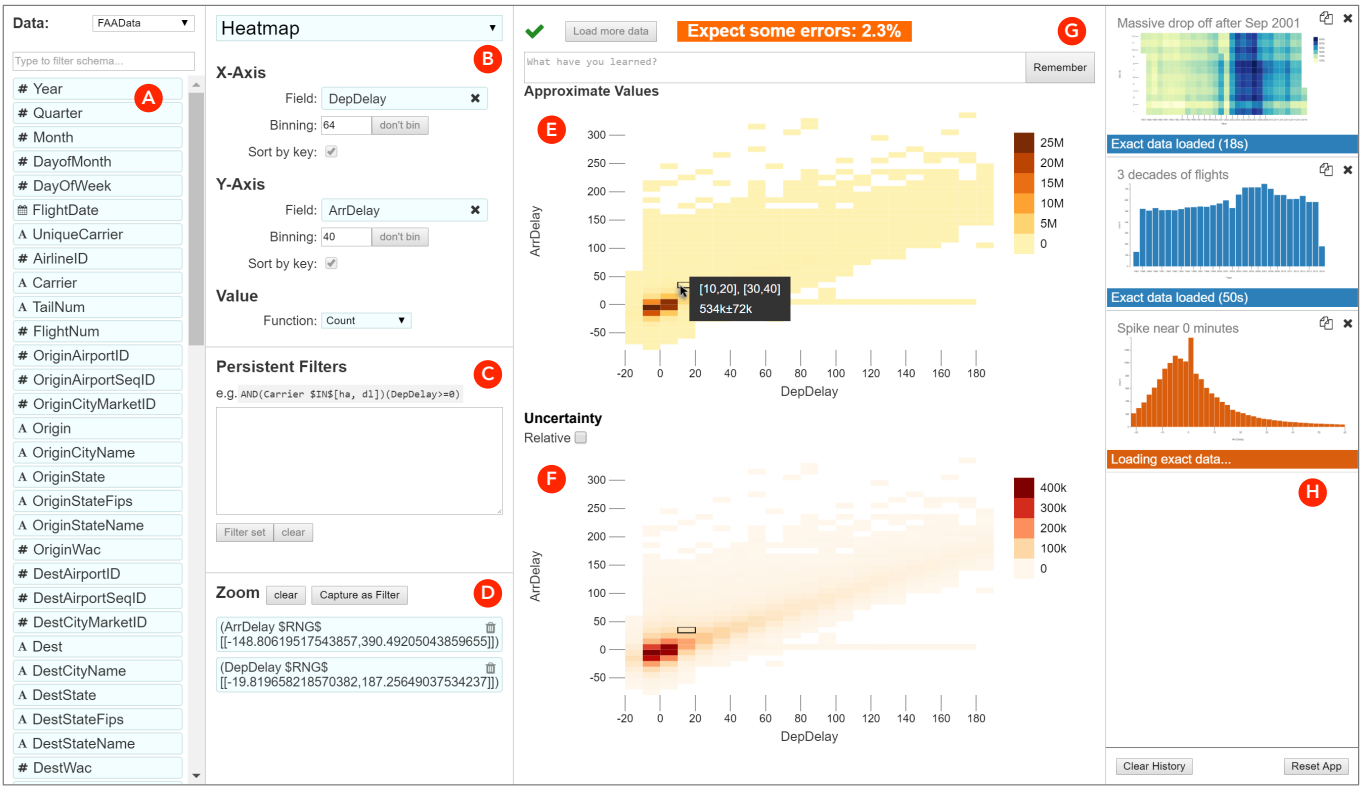

Figure 2. The Pangloss UI, exploring a flight delay dataset, with a list of fields (A), chart specification forms (B), a textfield for filters (C), zoom specification (D), approximate visualization (E), visualization of the uncertainty (F), field for annotations and "remember" button (G), and a list of views in the history (H). Two precise results are ready, while a third is loading.

Euclidean distance between the normalized distribution answer $x = \langle 0.39, 0.61 \rangle$ and the approximation $\hat{x} = \langle 0.40, 0.60 \rangle$ is $||x - \hat{x}||_2 = \sqrt{0.01^2 + 0.01^2} = 0.014$.

More general, the distribution uncertainty with Euclidean distance is [8]:

$$\sqrt{\sum_{\text{group } i} \left(\frac{\text{group } i\text{'s value}}{\text{total group value}} - \frac{\text{estimated group } i\text{'s value}}{\text{total estimated group value}} \right)^2}$$

Distribution uncertainty is different from familiar per-group confidence intervals [18, 13, 20]. Rather than seeing each group as having its own confidence interval, the distribution uncertainty is a total amount by which the whole visualization is likely to be imprecise. Distribution uncertainty recognizes that uncertainties are not independent. As Figure 1 illustrates, one group might be off by a lot, or many groups might be off by just a little.

Besides the distribution uncertainty, we also compute a confidence interval for each group. However, the sum of these per-group uncertainties are worst-case estimates, and can be significantly higher than the overall distribution uncertainty. When we visualize confidence intervals, the possible range of uncertainty they imply is far greater than the distribution uncertainty (Figure 1). For the analyst, knowing the overall distribution uncertainty means that they do not have to expect the worst case for all groups.

Sample+Seek supports aggregate measures such as count, sum, and average. Queries can have multiple group-by dimensions, of either categorical values or binned numerical values. Queries can also filter the data, based on boolean predicates. When predicates are selective, AQP systems need to look at many records before they find records that match the filters. Sample+Seek maintains indices that help rapidly identify matching records. Each query uses a different sample because we limit the time per query.

In the algorithm a sample is always a strict subset of all rows. Even if we scan all rows, we can only get within a fixed factor of the precise answer. This prevents us from providing users with progressively improving results.

In the original Sample+Seek [8], data samples were kept in memory; Pangloss uses a modified version that supports larger datasets and reduces the memory footprint by keeping the dataset as a randomly shuffled file on disk. Our version of Sample+Seek is able to respond within 100 ms with acceptable levels of approximation error.

Designing the Pangloss UI

Sample-based visualizations require a different user experience than more traditional visualization systems. In this section, we discuss the Pangloss user interface, highlighting design decisions that accommodate the unusual aspects of AQP and optimistic visualization. The interface is implemented as a web application, using the React framework and D3 [3].

The interface, shown in Figure 2, uses interaction paradigms well-known in visualization tools such as Tableau and PowerBI. On the left is a searchable schema (A) with fields that can be dropped to the chart specification (B). Below the chart specification form are fields for filtering (C) and showing the current zoom predicates (D). The largest area of the screen is taken up by the view (E) and its uncertainty (F). Above the view is a textbox for observations and a button to *remember* the current view (G), which computes the precise result for this view. Remembered views are listed in the history on the right; they are drawn as orange while they are still loading, and turn blue when precise result is available (H).

Approximate Visualizations

Pangloss supports two core visualization types: bar charts and heatmaps. The bar chart shows aggregated measures grouped by values; it can also be used as a histogram by binning numeric fields. The heatmap, a generalization of density plot, allows users to see the interaction between two dimensions; aggregate values are encoded using a color scale. Each dimension of the heatmap can be binned. Other visualizations can be implemented in this system; in theory, any aggregation-oriented visualization [9] can be accommodated.

Because many queries have long tails, Pangloss limits the number of bars or cells that can be shown to a top k and shows a warning if groups are hidden (Figure 3-A). When dealing with samples, the values further down the tail are based on fewer samples, and so are likely to be very uncertain; we chose not to add tools to scroll over to the tail of the distribution. Of course, an analyst can filter the highest values, working their way down the distribution.

The distribution uncertainty of the approximation is displayed above the main view (Figure 3-B). The system computes confidence intervals for each group; however, these per-group intervals are worst case estimates (Figure 1). 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 uncertainty (Figure 7, right). In all visualizations, tooltips show the group name or bin bounds, the approximate value, and the uncertainty (Figure 4).

Zooming with Samples

Pangloss supports zooming and filtering, like most visualization systems. In in-memory visualization tools, zooming and focusing only change the domain of the dimensions and measures; the group categories stay constant. With samples, every zoom focus interaction changes the predicate, forcing a new query to run on the AQP system. Consequently, the aggregate value and the uncertainty can change.

As we noted above, the groups themselves can change as the user adds filters: collecting a new sample might mean that numerical ranges might expand and new groups might appear. The semantics of filtering, then, call for a design decision. An analyst cannot know whether there are more groups to be seen until they filter. If the analyst filters ten categories down to three, do we interpret that as a *negative* filter, removing seven,

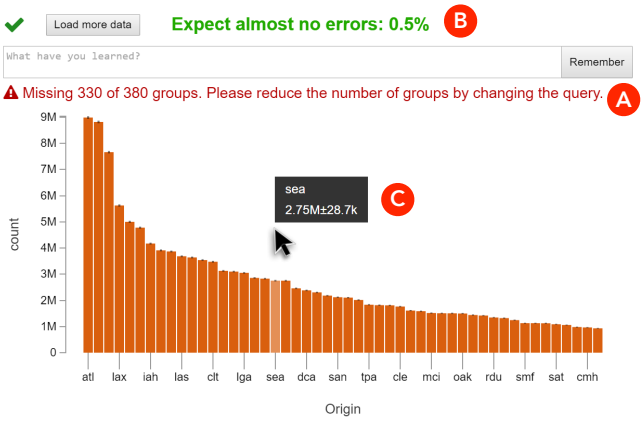

Figure 3. A bar chart for a result with a long tail. The view warns that it only shows the top groups (A). Above the chart is the distribution uncertainty (B). Tooltips in bar charts are shown for the area above the bar (C).

Figure 4. Left, tooltips for approximations show the group, the value, and the associated uncertainty. Right, tooltips for precise results show how much the estimate was off.

or a *positive* one, limiting to just those three? The difference is that "removing seven" might discover more groups (Figure 5).

To ensure that analysts never lose information, Pangloss treats categorical filters as negative by default; analysts can explicitly select positive filters if they need. Similarly, with numerical data, the domain can change (Figure 9); we add an inequality constraint (as opposed to a range predicate) when the user brushes all the way to the end of an axis.

Functions and Transformations with Samples

In most visualization systems, the user can carry out transformations and functions on the data. For example, it is easy to add a logarithmic transformation over a visualization, or to compute average by dividing sum by count. Distribution uncertainty, however, is a global measure across a view. As such, the distribution uncertainty for a value will be very different from the log of that same measure value; they will be based on different numbers of samples. Just as zooms must be computed as separate computed queries, so too must be transforms and functions.

Many existing visualization systems and techniques build on assumptions that are not true when samples are used to approximate results. With samples we cannot assume that the result of an aggregation query does not miss groups. Consequently, if we calculate the average of a measure as the ratio of the sum and the count, we can only do so for the common groups in the two samples. We cannot, however, compute the combined distribution uncertainty.

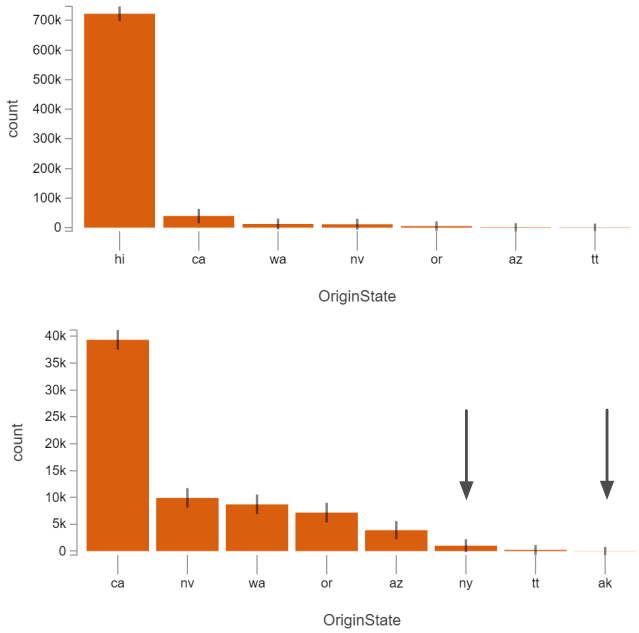

Figure 5. Above, an approximate histogram of origin states for Hawaiian Airlines flights. Below, if we filter out Hawaii, a new query runs on the AQP system and the approximation also shows that New York and Alaska (arrows) are also origin states. Because the new predicate is more selective, the uncertainty decreases for all groups.

Remembering Views and History

The heart of optimistic visualization is the process of selecting a view and re-running its query to get a precise result. In Pangloss, we call this *remembering* the view. We wish to support analysts being able to look back at a past view, and verify the observation they made with it.

One important design decision is which views should be remembered. We considered verifying every past view that Pangloss produces, modeled after Graphical Histories [17]. There are several disadvantages to keeping this complete history. First, we expect users to review the precise views; it would be overwhelming to review every view. Second, precise queries are computationally expensive; issuing hundreds of them can overwhelm back-end data systems. As such, we want to encourage users to remember only views that are relevant for observations.

In Pangloss, we decided to make this an explicit process. The "Remember" button (Figure 2-G) stores the view in the history and runs the precise query in the background. We would like to encourage analysts to track the observations; we support it by allowing them to add a small textual annotation describing the remembered view.

The entries in the history (Figure 2-H) change when a precise result is available: we render approximate views in orange shades, and precise views in blue. Views in the history are immutable, but users can revise their annotations to note new information. In addition, users can make a mutable (and approximate) copy of the view if they wish to modify it.

Queries for precise answers are sent to a commercial SQLServer database. On current hardware precise queries return within 30 seconds to a few minutes for datasets of around 50 GB to 1 TB.

Visualizing Approximation Error

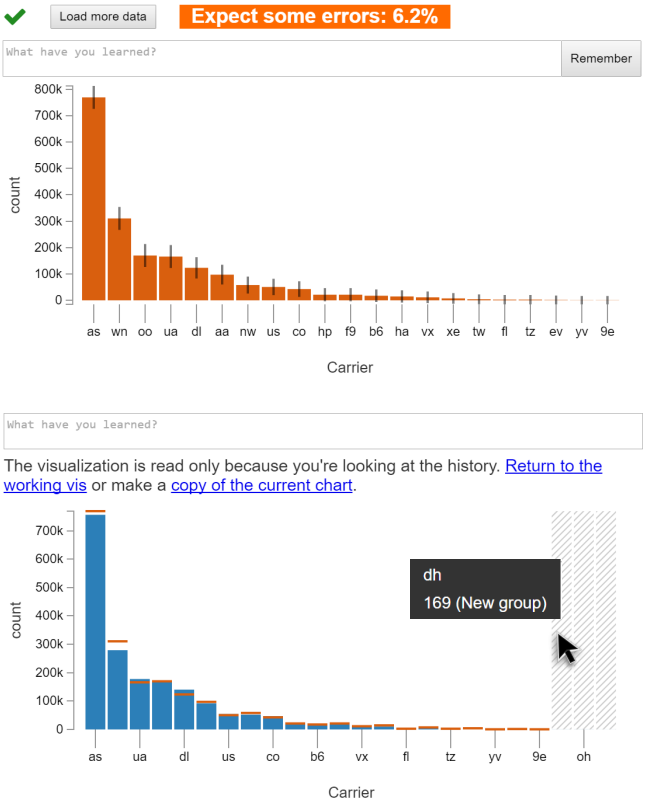

Figure 6. Above, approximate bar chart for count of flights out of Washington state, grouped by airline, with per-group confidence intervals and distribution uncertainty. Below, the precise result for the same chart shows the values as blue bars, the approximate result as orange lines, and highlights airlines that were missing from the approximation (e.g., Independence Air (dh) with 169 flights).

Once the precise query has completed, an analyst must be able to verify whether their observation during the exploration was justified. Several types of changes might occur between the approximate and the precise views. First, the values of some groups can change; if the chart is sorted, this also means that bars might be in a new order. Second, some groups that were not in the chart before might have been added. Last, binned data might change its range.

We wish to support the analyst in comparing the approximate and precise views.

What should happen when a bar is added, or the relative order of sorted bars changes? There are advantages to both maintaining *stability*, by keeping the layout of the approximate visualization with revised values, or *precision* by showing the precise view. We settled on the latter, because our major goal

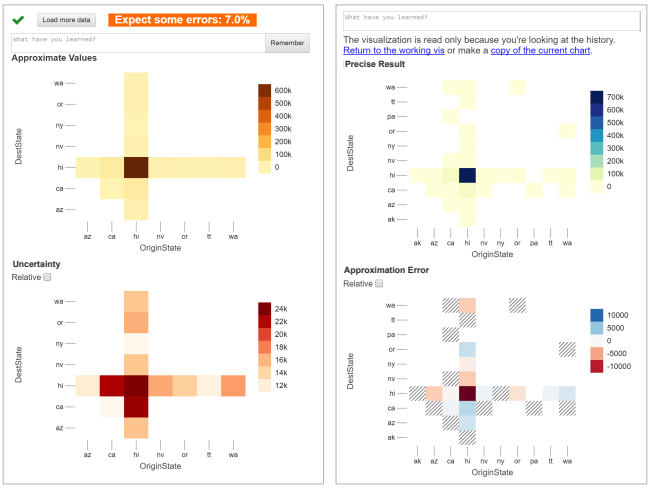

Figure 7. Approximate (left) and precise (right) heatmaps, examining origin and destination states for Hawaiian Airlines. (Left) The approximate view shows the estimated count above and the uncertainty below. (Right) The precise view shows the count above and the approximation error below. Both lower images toggle to show relative or absolute differences.

is encouraging users to interpret the view they see. The final visualization emphasizes the precise values.

Visualizing the difference along with the true values in the same chart poses challenges similar to uncertainty visualizations; we use similar methods. Pangloss superimposes the approximate value on the bars as orange lines, as in Figure 6. This makes changes in order very visible, as the orange lines no longer decrease monotonically. When a new group appears, we highlight it with a gray striped background (Figure 8, right). It is worth noting that a histogram's range might change between sample and final. Therefore, in the precise query, we fix the bin width and the offset so that the precise histogram always aligns with the approximate one (Figure 9).

For heatmaps, one chart shows the true value, while the approximation error is shown in a separate chart (Figure 7). The analyst can toggle between seeing absolute error, which shows the difference between the estimate and the actual value, and relative error, measured as a percentage. We found that each is useful for different circumstances: relative error is good when cells have similar values; on the other hand, a small amount of error can still be thousands of percentage points from a small cell value. Consistent with how new bars are highlighted, any new cells have a diagonally striped pattern (Figure 8, left).

USER STUDIES

There are several motivating concepts behind Pangloss that we wished to validate. First, are analysts comfortable with incomplete or inaccurate results, and are they willing to use them to explore approximate data? Progressive visualization systems [13, 33, 39] allow analysts to linger at one view until it reaches a level of accuracy; Pangloss does not. Second, optimistic visualization expects users to proceed with their exploration even without precise results; we wanted to know whether they would do so. Last, we wanted to know how users

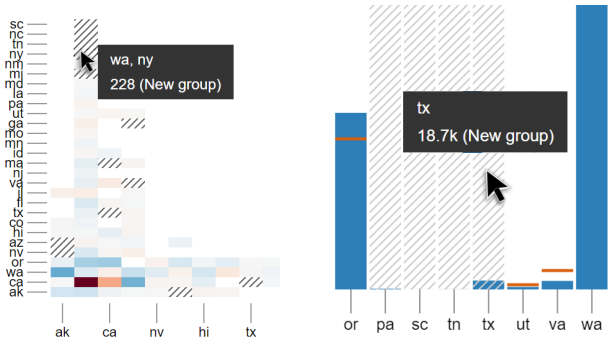

Figure 8. To draw attention to new groups, the corresponding cells in heatmaps (left) and bars (right) are highlighted with a stripe pattern.

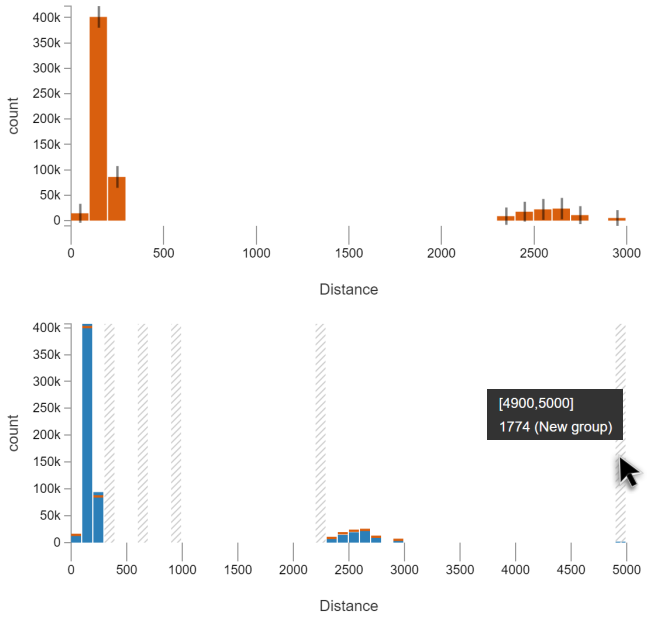

Figure 9. Approximate histogram at the top shows that Hawaiian Airlines has short range (inter state) and long range (island to mainland) flights. The precise histogram below shows additional flights around 5000 miles. The range has changed but bins are still aligned.

would interact with precise results, and whether checking those results would interfere with their flow.

We believed these questions would be best addressed by collecting rich stories of users interacting with the system; we modeled our user study after Fisher et al. [13]. We carried out two studies. We first wished to establish that Pangloss works as a data analytics toolkit that can enable users to come up with usable insights. We chose a single dataset and recruited data analysts from within Microsoft. We encouraged them to explore the data, providing them with guiding questions. The shared dataset allows us to factor out feedback specific to the data.

We wanted to also validate that this system works for real-world situations. We solicited data scientists from Microsoft, and asked them to share a current dataset with us; we loaded it into Pangloss. We then invited them to explore the system and interviewed them about their experience.

Flight Delay Study

Our first study used the "BTS Flight Delays dataset" [29]. The full dataset is 70 GB, and contains records on about 170 million commercial domestic flights from the last three decades within the United States. Each record represents one flight, with its arrival and departure time and location, as well any delays and reroutes in flight. There are a total of 109 fields in the raw data; for the user study, we removed some of the sparser fields, to result in 78 fields. Pangloss is able to maintain the 100 ms response time for histograms and bar charts at a 1%–2% uncertainty level; heatmaps had higher uncertainty of 2%–5%. Views with averages and highly selective predicates have much larger errors. In contrast, a full SQL query on the dataset runs in one minute.

We recruited five data scientists for the study. All participants were associated with a product or consulting team, including IT Support, software development, and post-deployment monitoring. Each of them was familiar with creating visualizations in R, PowerBI, Tableau, or Excel. We call them P1 through P5 below.

Three of the sessions were carried out in person; two others, with analysts located further away, were carried out by video-sharing session. At Microsoft, it is not unusual to meet via video-sharing; all users were familiar with the technology. Sessions lasted an hour. We started sessions with a tutorial, which guided subjects through a series of training questions to help them learn the system. We then invited them to explore the data using the tool; we gave some introductory questions but invited them to pursue questions that caught their curiosity for half an hour. At the end of the time, if the users had not reviewed precise results, we encouraged them to do so. Near the end of the hour, we asked our subjects to discuss advantages and tradeoffs of Pangloss.

We encouraged users to think aloud; all sessions were voice- and screen-recorded. Remote users used the tool in their web browser, so that they did not suffer from video-sharing lag.

Flight Delay Study Results

During the study, most users were resistant, at first, to explicitly recording their observations: we needed to remind almost all users to record observations they were making. After the first few, however, it became more habitual for them to record their observations over time. Only one user refused to use the "remember" function.

We had feared that some users would be unwilling to explore their data, knowing that their initial results were inaccurate; perhaps they would wait to ensure their earlier investigation was valid. We saw this only once; P2, mid-analysis, paused to wait for his remembered view to complete. A moment later, impatient, he resumed his flow, and continued to "remember" things. By the end of the study, four of our users were regularly "remembering" visualizations. Users "remembered" 4-7 views during their half hour.

Some users found opportunities to check in on their history during the study. P4 said, "I was thinking what to do next—and I saw that it had loaded, so I went back and checked it . . . [the passive update is] very nice for not interrupting your workflow."

Interacting with Big Data: All users had dealt with slow queries in big data systems, and appreciated the speed of Pangloss. As P4 said, "[with a competitor] I was willing to wait 70-80 seconds. It wasn't ideally interactive, but it meant I was looking at all the data." All the users commented on the responsiveness of the system.

Our users were also familiar with sampling; for example, P4 had used sampling for other projects: "We can't look at all the sensor data at once—it wouldn't work. So we sample."

Waiting for Precise Results: We wondered whether analysts would find the precise version of the visualization useful: after all, they had already seen the approximate version. P1 said, "From my perspective, [uncertainty] almost passed me by. Nothing that I saw after-the-fact fundamentally changed any conclusions."

Despite that, this made Pangloss feel safer to him: P1 said he checked whether the precise results "are fundamentally different in a way that warrants my attention—if the top 5 are different, that's important." P5 used precise results to feel more confident in the approximations: "[The precise view] is an enabler. [Sometimes] you have a doubt whether sampling has made it better or worse, but seeing something right away at first glimpse is really great."

Our participants most often decided to remember very uncertain results: the large distribution uncertainty and wide confidence intervals cued them to "remember" their findings and request a more precise results. In contrast, they chose to trust more certain results.

The data scientists also talked about the importance of presenting precise data to their teams. While they might be able to use sample data with approximation, P3 said, "I know some information gets lost when I use samples . . . If I want to give my boss specific numbers, I don't want to use samples." P2 said, "full and complete data always makes the most sense."

We wondered whether going back to check the results of queries would disrupt their workflow. P4 felt that the color change to cue the arrival of precise results was not disruptive, and "the ability to keep working . . . and know that you will get a complete visualization is very handy."

Limitations: Users did bump into the limitations of Pangloss, which shows specific visualizations of aggregate data. P5 said "When I'm using R, it's like I'm on a mountaintop, I can go anywhere I want; when I'm using your system, there is a path that I need to follow." Similarly, P3 wanted to see lower-level data: "you want to go down to the sample level to see which samples are causing this pattern."

Case Studies

Our second study emphasized real-world use of datasets. We recruited data analysts from an internal list of data scientists. We looked specifically for users who had datasets over 10 gigabytes with structured tabular data, and selected three candidates. We ingested their data into Sample+Seek, and then

scheduled meetings with the groups: we met with David in person, while we met remotely with Madhu and Faraz. [1]

We followed a similar protocol to the Flight Delay study: each session started off with a short demonstration and tutorial. When subjects made observations out loud, we encouraged them to "remember" the views. At the end of the experiment, if they had not reviewed some of the remembered values, we encouraged them to click through those observations, and to evaluate whether their observations had changed. These case study sessions ran between an hour and a half and two hours.

Case Study 1: David and Software Crashes

David works on the telemetry team for a family of software products. His team is responsible for helping developers identify which features are causing problems for their users. They do so by examining telemetry across multiple builds of their software, both beta and released, categorizing the circumstances under which software fails.

David's dataset consists of activities that users are carrying out in these products with error information. Their data collection gathers 100 TB/day of raw timestamped event data. This is too much for their system (or for Pangloss) to analyze; instead, his team pre-aggregates this data into summaries, which average around 200 MB/day. These summaries consist of activities, hierarchically categorized by product and feature; broken down by minute of the day; it stores the number of times that users attempted to use the feature, and how many of them succeeded.

His team's visualization technology cannot view more than a gigabyte of data, representing a week of data. This can miss out on deployment problems that emerge over longer ranges of time, and makes it hard to compare between builds of the software. Pangloss was able to load three months' data.

David is used to working with slow queries: as he began to navigate his data, he excitedly said that "instantaneous visualizations are great!" He freely jumped between different slices of the data, admiring "the power of being able to pivot so quickly." David applied multiple filters to the data, limiting it by date, code branch, and application.

At first, he did not see the utility in remembering his queries: "going back and looking at the more precise data would be valuable if I were drawing any real conclusions—if I was going to send an email to somebody. But in a lot of these cases, if I didn't see anything too exciting in the approximate number, I'd be OK with that." We insisted he remember his first few queries to get precise results; by partway through the session, he did so on his own. When we went back through his results at the end, he reflected "Now that I've been sitting here for an hour, after I go back, it makes a lot of sense [to have these annotations], but as I was doing it, I was thinking, 'I want to move on, I want to move on."

Like some of the users in the flight delay study, he began to think about the value of precise data: "A lot of what we do gets used in ship rooms and standups; ship decisions are made on those numbers. Those meetings cost thousands of dollars a minute. You need super-precise data for them."

During the study, David ran into a surprising result: one day had an aberrantly low value for a particular data series. He went off wanting to delve into it more: "I'm going to go over this with my team and send them some screenshots. I want to find out what happened on 8/8 with this stream."

David had been limited by the amount of data they collected; Pangloss allowed him and his team to broaden their view.

Case Study 2: Madhu and Search Terms

Madhu works on the advertising platform for a search engine. His team is responsible for trying to predict trends in searches and keywords. They analyze usage data from the search engine, looking for terms and concepts that are gaining in popularity. Madhu wanted to look at trends and changes in the popularity of these topics across different countries. His dataset consisted of topics, with categories, countries, and timestamps. As in David's case, the dataset he gave us was pre-aggregated: each of these categories was then labeled with the number of impressions—that is, the number of people who searched for terms that matched this category—and the number of people who clicked through on those searches. Madhu gave us 994 million rows of data, covering eight months of usage.

Madhu was excited that Pangloss could let him get to know the shape of his full dataset. In the past, he had not felt like he could explore his data: queries took too long, and so he would focus only on specific questions that he needed to answer.

Madhu searched carefully for patterns: he wanted to find ways that the data changed in regular and systematic ways. He spent most of his time in the heatmap, looking at a dozen or more keywords at a time. He did manage to find trends in the data, including a weekly pattern in one keyword, and another that spiked over a month. He found these results useful enough that he later asked whether he could send us a new, less aggregated dataset, and asked for a follow-up appointment with his team, in the hope that more of his colleagues had an opportunity to understand how the data they worked with operated.

Case Study 3: Faraz and Social Computing

Faraz is a data scientist who works with a Twitter "firehose" feed. One of his projects is to assess the credibility of Twitter users. His team expects that Twitter users can be distinguished by their followers lists, and by the keywords they use. His dataset looks at Twitter users, their hashtags, and the people who they follow; his statistical algorithms also label users who are likely to be spammers.

Faraz looked at the top k charts of the most commonly-used tags, finding terms like "brexit" and "rio2016" were popular; he contrasted this list to keywords tweeted by persons who were labeled as likely to be spammers. Looking at the precise view, he asked to look further down, at tail queries.

Faraz encountered some of the unintuitive aspects of dealing with sample-based analysis: for example, when seeing the top ten keywords, his first impulse was to "remember" just the top keywords. This would not have the desired effect: the full

[1] All names are anonymized.

query might discover new words, but his filtered list would not show them.

Faraz became accustomed to seeing very high uncertainty levels for his high cardinality data. He began to use the approximate query as a draft, less concerned about the answer it showed and more concerned about whether it showed that he had the right query. "I'm doing exploratory data analysis and I don't need full accuracy." When he accidentally formed a view with a bad filter, he quickly noticed the approximate view was incorrect and adjusted the query; after he felt sure he had the right query, he remembered the view.

Pangloss allowed Faraz to explore complex aspects of his data rapidly, and come back to check his results later.

Discussion of User Studies

We were gratified that users were able to use Pangloss to explore their large datasets, and take away actionable and novel conclusions. Users were willing to trust the approximation, and to generate precise results afterward.

Our user studies taught us more about how users see approximate data. Our users see precision broadly: they want both rapid interaction for exploratory phase, and to present precise results to decision makers. Precision is not just a way to verify the approximation, but is an end in itself. In small-data systems, the same tool can fulfill both these roles; here, Pangloss separated those goals.

The process of recording observations during exploratory visualization was unintuitive to all of our users. Most of our users were grateful to have been forced to record their observations, and later found it useful to reconstruct their path. This suggests that Pangloss does not have the right balance of encouraging users to record their observations. There is a broad spectrum of possible approaches–from notebook interfaces that require explicit queries, to systems that automatically recommend visualizations [42]; this design space would reward further exploration.

Users wanted more features from Pangloss: several wanted to be able to see the underlying data: although big data demands aggregations, analysts wanted to see individual records to spot-check their results, and to get a sense of what sat in a bucket. Other users asked for transformations, aggregations, and ways to project the data that Pangloss does not currently support.

CONCLUSIONS

In these conclusions, we note a number of implications of optimistic visualization that emerge from both the user study as well as from our design work.

The concept of optimistic visualization can help users adopt approximate and progressive systems. It is comparatively easy to implement, but the benefits for the users are large. For example, user David used optimism to build confidence in the approximate results after seeing the results of precise queries. Some of our users needed precise data. Under progressive visualization, that means waiting until computation finishes. In Pangloss they could run it in the background. Future work should combine progressive and optimistic, and explore the design space to understand what best benefits users: a system might improve the results for remembered views in the background, allowing the user to check on progress and how the approximation has changed.

Existing visualization tools and techniques make assumptions that do not hold for approximate results. In a sampling environment, more selective predicates can have surprising effects. Additional groups may appear, and both aggregate values and uncertainty levels might change. The same can happen in the transition from approximate to precise results. New visualization systems must be able to handle shifting axes and changing color scales. We will need to develop a vocabulary of visual cues to highlight order changes, new groups (Figure 8), and other qualitative changes.

There are many opportunities to refine the user experience of optimistic visualization for data exploration to address the issues raised during the user studies. One question is how to identify whether a precise view is meaningfully different from the approximation. In Pangloss every precise result is treated the same; it could be valuable to highlight views with significant differences. Whether a change is significant depends on the observation: the observation that A is higher than B is different than observing that $A > 50$; the precise results might invalidate one but not the other.

One interesting question is in deciding what to remember; our users found that the major challenge when using the system. In Pangloss, users must select views explicitly. Pangloss supports an exploratory process; there are cues in the sets of visualizations that the analyst creates to decide which views are worth remembering. With better provenance tracking, we might be able to better decide which observations should be remembered; we might also group visualizations in the history by their broad tasks.

Optimistic visualization comes from a close collaboration between database and visualization researchers. The ideas in this paper have emerged from understanding constraints and opportunities in both areas. We believe that building collaborations between these two fields is critical for improving data analysis going into the future.

This paper contributes the concept of optimistic visualization. We have showed a first implementation of optimism; and discussed ways in which the exploratory data analysis process is different under approximate data. We have shown ways to visualize the difference between the approximate and precise view. Last, we have presented the results of eight users working with an optimistic system, and showed that it can help meet their needs for both speed and precise results.

ACKNOWLEDGMENTS

We thank our participants for their time and the enthusiasm for our prototype. This paper has benefitted from the advice of Jessica Hullman, Arnd Christian König, Steven Drucker, and feedback from the VIBE group at Microsoft Research; and from the comments of the anonymous reviewers.

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 *Proceedings of the International Conference on Management of Data (SIGMOD '14)*. ACM. DOI:http://dx.doi.org/10/f3tvrz
2. Sameer Agarwal, Barzan Mozafari, Aurojit Panda, Henry Milner, Samuel Madden, and Ion Stoica. 2013. BlinkDB: Queries with Bounded Errors and Bounded Response Times on Very Large Data. In *Proceedings of the European Conference on Computer Systems (EuroSys '13)*. ACM. DOI:http://dx.doi.org/10/bwrd
3. Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. 2011. D3 Data-Driven Documents. In *IEEE Transactions on Visualization and Computer Graphics (INFOVIS '11)*. IEEE Educational Activities Department. DOI: http://dx.doi.org/10/b7bhhf
4. Mihai Budiu, Rebecca Isaacs, Derek Murray, Gordon Plotkin, Paul Barham, Samer Al-Kiswany, Yazan Boshmaf, Qingzhou Luo, and Alexandr Andoni. 2015. Interacting with large distributed datasets using Sketch. In *Eurographics Symposium on Parallel Graphics and Visualization*. University of Wisconsin-Madison. DOI: http://dx.doi.org/10/f3tvr9
5. Stuart K. Card, George G. Robertson, and Jock D. Mackinlay. 1991. The Information Visualizer, an Information Workspace. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI '91)*. ACM. DOI:http://dx.doi.org/10/cvtdps
6. Surajit Chaudhuri and Umeshwar Dayal. 1997. An Overview of Data Warehousing and OLAP Technology. In *Proceedings of the International Conference on Management of Data (SIGMOD '97)*. ACM. DOI: http://dx.doi.org/10/bst468
7. Geoff Cumming and Sue Finch. 2005. Inference by eye: confidence intervals and how to read pictures of data. (2005). DOI:http://dx.doi.org/10/c4nsck
8. Bolin Ding, Silu Huang, Surajit Chaudhuri, Kaushik Chakrabarti, and Chi Wang. 2016. Sample + Seek: Approximating Aggregates with Distribution Precision Guarantee. In *Proceedings of the International Conference on Management of Data (SIGMOD '16)*. ACM. DOI:http://dx.doi.org/10/f3tvr2
9. Niklas Elmqvist and Jean-Daniel Fekete. 2010. Hierarchical Aggregation for Information Visualization: Overview, Techniques, and Design Guidelines. In *IEEE Transactions on Visualization and Computer Graphics (INFOVIS '10)*. DOI:http://dx.doi.org/10/db2gxw
10. Jean-Daniel Fekete and Romain Primet. 2016. Progressive Analytics: A Computation Paradigm for Exploratory Data Analysis. (2016). https://arxiv.org/abs/1607.05162
11. Nivan Ferreira, Danyel Fisher, and Arnd Christian König. 2014. Sample-oriented task-driven visualizations: allowing users to make better, more confident decisions. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI '14)*. ACM. DOI: http://dx.doi.org/10/f3tvr6
12. Danyel Fisher. 2016. Big Data Exploration Requires Collaboration Between Visualization and Data Infrastructures. In *Proceedings of the Workshop on Human-In-the-Loop Data Analytics (HILDA '16)*. ACM. DOI:http://dx.doi.org/10/f3tvr7
13. Danyel Fisher, Igor O. Popov, Steven M. Drucker, and Monica M. C. Schraefel. 2012. Trust me, I'm partially right: incremental visualization lets analysts explore large datasets faster. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI '12)*. ACM. DOI:http://dx.doi.org/10/f3tvr5
14. P. Godfrey, J. Gryz, and P. Lasek. 2016. Interactive Visualization of Large Data Sets. In *IEEE Transactions on Knowledge and Data Engineering*. DOI: http://dx.doi.org/10/f3tvr8
15. Jim Gray, Surajit Chaudhuri, Adam Bosworth, Andrew Layman, Don Reichart, Murali Venkatrao, Frank Pellow, and Hamid Pirahesh. 1997. Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals. In *Data Mining Knowledge Discovery*. Kluwer Academic Publishers. DOI: http://dx.doi.org/10/brp273
16. Peter J. Haas and Joseph M. Hellerstein. 1999. Ripple Joins for Online Aggregation. In *Proceedings of the International Conference on Management of Data (SIGMOD '99)*. ACM. DOI:http://dx.doi.org/10/dqgn68
17. Jeffrey Heer, Jock Mackinlay, Chris Stolte, and Maneesh Agrawala. 2008. Graphical Histories for Visualization: Supporting Analysis, Communication, and Evaluation. In *IEEE Transactions on Visualization and Computer Graphics (INFOVIS '08)*. DOI: http://dx.doi.org/10/b5fvmx
18. Joseph M. Hellerstein, Ron Avnur, Andy Chou, Christian Hidber, Chris Olston, Vijayshankar Raman, Tali Roth, and Peter J. Haas. 1999. Interactive Data Analysis: The Control Project. (1999). DOI: http://dx.doi.org/10/bsjgrh
19. Joseph M. Hellerstein, Peter J. Haas, and Helen J. Wang. 1997. Online Aggregation. In *Proceedings of the International Conference on Management of Data (SIGMOD '97)*. ACM. DOI:http://dx.doi.org/10/bbk4fr
20. 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)*. Public Library of Science. DOI: http://dx.doi.org/10/f3tvsd
21. Susan Joslyn and Jared LeClerc. 2013. Decisions with uncertainty: the glass half full. (2013). DOI: http://dx.doi.org/10/f3tvrx

22. Niranjan Kamat, Prasanth Jayachandran, Karthik Tunga, and Arnab Nandi. 2014. Distributed and interactive cube exploration. In *International Conference on Data Engineering*. IEEE. DOI:http://dx.doi.org/10/f3tvsb
23. Lauro Lins, James T Klosowski, and Carlos Scheidegger. 2013. Nanocubes for real-time exploration of spatiotemporal datasets. In *IEEE Transactions on Visualization and Computer Graphics (INFOVIS '13)*. DOI:http://dx.doi.org/10/bwrc http://nanocubes.net/.
24. Zhicheng Liu and Jeffrey Heer. 2014. The effects of interactive latency on exploratory visual analysis. In *IEEE transactions on visualization and computer graphics (INFOVIS '14)*. DOI:http://dx.doi.org/10/f3tvrw
25. Zhicheng Liu, Biye Jiang, and Jeffrey Heer. 2013. imMens: Real-time Visual Querying of Big Data. In *Proceedings of the Eurographics Conference on Visualization (EuroVis '13)*. DOI: http://dx.doi.org/10/f3tvr4
26. Jerzy Neyman. 1937. Outline of a theory of statistical estimation based on the classical theory of probability. (1937). DOI:http://dx.doi.org/10/b79jvq
27. Jakob Nielsen. 1993. Response times: The 3 important limits. (1993).
28. Chris North. 2006. Toward Measuring Visualization Insight. (2006). DOI:http://dx.doi.org/10/b2758v
29. Bureau of Transportation Statistics. 2016. Airline Delays, Cancellations and Tarmac Times. (2016). http://www.rita.dot.gov/bts/data_and_statistics/by_mode/airline_and_airports/airline_delay.html
30. Frank Olken and Doron Rotem. 1986. Simple Random Sampling from Relational Databases *(VLDB '86)*. http://dl.acm.org/citation.cfm?id=645913.671474
31. Chris Olston and Jock D. Mackinlay. 2002. Visualizing Data with Bounded Uncertainty. In *Proceedings of the IEEE Symposium on Information Visualization (INFOVIS '02)*. http://dl.acm.org/citation.cfm?id=857191.857754
32. Alexandre Perrot, Romain Bourqui, Nicolas Hanusse, Frédéric Lalanne, and David Auber. 2015. Large interactive visualization of density functions on big data infrastructure. In *Symposium on Large Data Analysis and Visualization (LDAV)*. IEEE. DOI: http://dx.doi.org/10/f3tvr3
33. Nicola Pezzotti, Boudewijn Lelieveldt, Laurens van der Maaten, Thomas Hollt, Elmar Eisemann, and Anna Vilanova. 2015. Approximated and User Steerable tSNE for Progressive Visual Analytics. In *IEEE Transactions on Visualization and Computer Graphics (INFOVIS '16)*. IEEE. DOI:http://dx.doi.org/10/f3tvrt
34. Peter Pirolli and Stuart Card. 2005. The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. In *Proceedings of international conference on intelligence analysis*.
35. John Stasko, Carsten Görg, and Zhicheng Liu. 2008. Jigsaw: Supporting Investigative Analysis Through Interactive Visualization. (2008).
36. Charles D Stolper, Adam Perer, and David Gotz. 2014. Progressive visual analytics: User-driven visual exploration of in-progress analytics. (2014). DOI: http://dx.doi.org/10/f3tvrv
37. Chris Stolte, Diane Tang, and Pat Hanrahan. 2002. Polaris: A system for query, analysis, and visualization of multidimensional relational databases. In *IEEE Transactions on Visualization and Computer Graphics (INFOVIS '02)*. IEEE. http://dl.acm.org/citation.cfm?id=857190.857686
38. John W Tukey. 1977. Exploratory data analysis. (1977).
39. Cagatay Turkay, Erdem Kaya, Selim Balcisoy, and Helwig Hauser. 2017. Designing Progressive and Interactive Analytics Processes for High-Dimensional Data Analysis. In *IEEE Transactions on Visualization and Computer Graphics (INFOVIS '16)*. IEEE. DOI: http://dx.doi.org/10/f3tvsc
40. Hadley Wickham. 2013. Bin-summarise-smooth: a framework for visualising large data. (2013).
41. Hadley Wickham and Lisa Stryjewski. 2011. 40 years of boxplots. (2011).
42. K. Wongsuphasawat, D. Moritz, A. Anand, J. Mackinlay, B. Howe, and J. Heer. 2016. Voyager: Exploratory Analysis via Faceted Browsing of Visualization Recommendations. In *IEEE Transactions on Visualization and Computer Graphics (INFOVIS '15)*. DOI:http://dx.doi.org/10/bdsz https://idl.cs.washington.edu/papers/voyager/.
43. Ji Soo Yi, Youn-ah Kang, John T. Stasko, and Julie A. Jacko. 2008. Understanding and Characterizing Insights: How Do People Gain Insights Using Information Visualization?. In *Proceedings of the Workshop on BEyond Time and Errors: Novel evaLuation Methods for Information Visualization (BELIV '08)*. ACM. DOI: http://dx.doi.org/10/c7m5hm