Sample-Oriented Task-Driven Visualizations: Allowing Users to Make Better, More Confident Decisions

Nivan Ferreira
New York University
Six MetroTech Center
Brooklyn, NY 11201 USA
nivan.ferreira@nyu.edu

Danyel Fisher
Microsoft Research
1 Microsoft Way
Redmond, WA 98052 USA
danyelf@microsoft.com

Arnd Christian König
Microsoft Research-XCG
1 Microsoft Way
Redmond, WA 98052 USA
chrisko@microsoft.com

ABSTRACT

We often use datasets that reflect samples, but many visualization tools treat data as full populations. Uncertain visualizations are good at representing data distributions emerging from samples, but are more limited in allowing users to carry out decision tasks. This is because tasks that are simple on a traditional chart (e.g. "compare two bars") become a complex probabilistic task on a chart with uncertainty. We present guidelines for creating visual annotations for solving tasks with uncertainty, and an implementation that addresses five core tasks on a bar chart. A preliminary user study shows promising results: that users have a justified confidence in their answers with our system.

Author Keywords

Incremental visualization; uncertainty visualization; user study; boxplot; error bars.

ACM Classification Keywords

H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

INTRODUCTION

The goal of data analysis is, in general, to describe attributes of a population based on quantifiable properties. Yet we often interact with *samples* of data, rather than the full population. Sometimes, samples are employed because processing the entire data set places unacceptable overhead on storage or computing [8, 11]. More often, only a subset of a much larger real-life distribution is available: because the data is a sample by its very nature, such as the results from a survey, or because the instrumentation to obtain the data can only capture a small subset of the data universe [17], such as when only a subset of nodes in a data center run potentially expensive telemetry instrumentation. Despite the ubiquity of samples in data analysis, far too many visualization tools neglect the fact that the data is a sample.

CHI 2014, April 26 - May 01 2014, Toronto, ON, Canada.

ACM 978-1-4503-2473-1/14/04…$15.00.
http://dx.doi.org/10.1145/2556288.2557131

We suspect there to be several reasons for this neglect. Many users are unaware of the importance of seeing their data as a sample. While it is common to generate boxplots to show error bars, and to run statistical tests, these usually are prepared only at the end of an analysis process. Many analysts simply explore their data based on the sample available, looking at averages or sums without taking into account uncertainty. Including statistics and uncertainty in an analysis can add a great deal of complexity to the process and slow it down, but data analysts prioritize rapid iteration for exploration.

Even for knowledgeable users, reasoning in the presence of probabilities and uncertainty can be very challenging [3]. In order to think about samples properly, users need to interpret all questions and conclusions about the data in a probabilistic manner: "is A greater than B?" changes to "what are the chances that A is greater than B?" Even with the aid of specialized visualizations, this task can still be very hard, as Micallef *et al* showed in their work on visualizing Bayesian probability [15].

Part of the challenge is that showing an uncertain value does not necessarily help *reason about* uncertain values. Many visualizations have been adapted for showing uncertainty, ranging from error bars to more exotic tools [21]. These visualizations often focus on specifically showing uncertainty ranges [18]. However, there are many tasks that we understand how to accomplish on non-uncertain charts [1, 2], such as comparing bars to each other, or finding the largest and smallest values; these uncertain visualizations do not directly support them. While it is easy to compare the heights of two bars, it can be difficult to compute the probability of a nearly-overlapping set of uncertainty regions. Previous work has shown that even experts trained in statistics make mistakes when interpreting confidence intervals [6, 7]. All of this suggests the need for a better integration of statistical techniques and interactive visual interfaces to enable data analysts to understand the meaning of sampled data.

In this paper, we take a first step in this direction: we investigate how to adapt the data analysis process to respect samples. In order to do so, we modify analysis tools to allow users to carry out tasks based on quantified uncertainty.

More precisely, we design visual encodings and interactions with the goal of allowing data analysts not only to identify the presence and magnitude of uncertainty, but to carry out common data exploration tasks. We discuss the design space for such visualizations and describe our approach.

We focus on two common visualizations used in exploratory data analysis, bar charts and ranked lists. For each of these, we identify common tasks that are performed on these charts in exploratory data analysis. Users can interact with these charts with task-specific queries; these are shown as annotations and overlays [13] that allow users to carry out these tasks easily and rapidly. Finally, we perform a preliminary user study to assess how our visualizations compare to standard approaches, and to establish whether users are better able to carry out these tasks with uncertain data. We find that our annotations help users to be more confident in their analyses.

BACKGROUND AND RELATED LITERATURE

We discuss common visual analysis tools, including those that do not currently handle uncertainty. Various tools have been suggested that visualize uncertainty; we compare these tools to our approach. Last, we discuss the idea of 'task-driven' visualization.

Visual Data Analysis Ignores Uncertainty

Major exploratory visualization tools available today—such as Tableau, Spotfire, and Microsoft Excel—do not have a built in concept of samples or uncertainty. Rather, they treat the data presented within the system as the whole population, and so present any numbers computed from the data—sample sums and averages, for example—as precise. However, as Kandel *et al* note [12], data analysts often deal with samples or selections of data.

Statistical software, such as SPSS and SAS, do have a more sophisticated concept that the data introduced is a sample, and draw their visualizations with error bars and confidence intervals as appropriate. However, these visualizations are usually produced in the process of running an explicit statistical test; by the time this test has been run, the user usually knows what questions they wish to investigate. This is highly effective for hypothesis-testing, but less useful when the user wishes to explore their data.

There is an opportunity, then, to provide lightweight data exploration techniques combined with statistical sampling.

Visualization Techniques that Handle Uncertainty

It can be difficult for users to reason in the presence of probabilistic data: Tversky and Kahanen [21] show that people make incorrect decisions when presented with probabilistic choices. It is possible to make more accurate decisions about data analysis when provided with confidence intervals and sample size information [6]. Unfortunately, the classic visual representations of uncertainty—such as drawing confidence intervals or error bars—do not directly map to statistical precision.

Even experts have difficulty using confidence intervals for tasks beyond reading confidence levels. For example, a common rule of thumb suggests that two distributions are distinct if their 95% confidence intervals just barely overlap. Yet, as Belia *et al* [3] point out, this corresponds to a t-test value of a $p < 0.006$—the correct interval allows much more overlap. Cummings and Finch [7] further note that most researchers misuse confidence intervals; they discuss "rules of eye" for reading and comparing confidence intervals on printed bar charts. While their suggestions are effective, they require training, and are limited to comparing pairs of independent bars.

While it may be complex, representing uncertainty can help users understand the risk and value of making decisions with data [14]. For example, long-running computations on modern "big data" systems can be expensive; Fisher *et al* [8] show that analysts can use uncertainty ranges, in the form of confidence intervals on bar charts, to help decide when to terminate an incremental computation.

The idea of visualization techniques that can handle uncertainty is a popular one in the visualization field. Skeels *et al* [16] provide a taxonomy of sources of uncertainty; in this paper, we refer specifically to quantitative uncertainty derived from examining samples of a population. Olston and Mackinlay [18] suggest a number of different visualizations for quantitative uncertainty, but do not carry out a user study.

Three recent user studies [5, 19, 23] examined ways that users understand uncertainty representations. All three studies examine only the tasks of identifying the most certain (or uncertain) values, and do not ask about the underlying data.

Annotating Visualizations to Address Tasks

Beyond identifying the *existence* of uncertainty, we also want users to be able to carry out basic tasks with charts. To identify what those tasks should be, we turn to Amar *et al* [1, 2], who identify ten different tasks that can be carried out with basic charts. Their tasks include comparing values to other, discovering the minimum value of a set of data points, and even adding several points together. All of these tasks are very quick operations on a standard bar chart without uncertainty: comparing two bars, for example, is as easy as deciding which one is higher.

To make chart-reading tasks easier, Kong and Agrawala [13] suggest using overlays to help users accomplish specific tasks on pie charts, bar charts, and line charts. Their overlays are optimized for presentation; they are useful to highlight a specific data point in a chart. In contrast, our approach allows users to read information that would have been very difficult to extract.

UNCERTAIN VISUALIZATIONS FROM SAMPLED DATA

Quantitatively uncertain data can come from many different sources [16]. In this paper, we focus on computations based

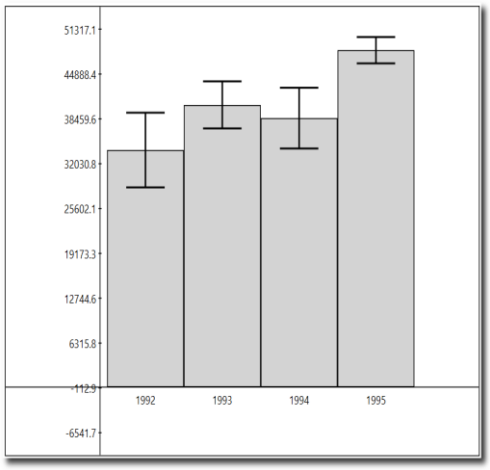


Figure 1: A bar chart with 95% confidence intervals representing the mean value over a dataset. Note the overlapping regions in 1992-1994.

on samples; however, many of these techniques could be applied more broadly. We use aggregates because they are common in exploratory data analysis: a core operation in understanding a dataset is examining the filtered and grouped average, sum, and count of a column. Indeed, visualization tools like Tableau are based largely around carrying out these aggregate operations against different groupings.

In sample-based analyses, we carry out approximate versions of these queries: we estimate the expected average, sum, or count of a dataset based on the sample, and infer a distribution on this *expected* value. Hellerstein *et al* provide a simple overview of how to use the Central Limit Theorem [11] to estimate error bounds based on these estimators.

As a result, the aggregate value and confidence interval represent a *distribution* of possible values. One use for this is in incremental analysis [8, 11], in which the system sees cumulative samples from a large dataset, and generates converging estimates of the final value. The distribution for each value represents the possible values once all of the data has been seen. For example, consider the bar chart shown in Figure 1. This chart is based on a sample from a large dataset of sales by year. The 95% confidence intervals mean that we expect—with probability 0.95—the mean value for sales in 1992 to be somewhere between 27,000 and 39,000.

In this scenario, the analyst's task is to extract information from the probability distributions modeled from the sample. Amar *et al* [1, 2] collect a series of different tasks that are commonly performed during the exploratory data analysis process. Their list includes low-level tasks like *retrieve value*, *find extrema (minimum and maximum)*, *sort values*, and *compare values*. In a representation without uncertainty, such as an ordinary bar chart, these tasks have direct interpretations: to find the minimum value in the bar chart, for example, the users simply finds the shortest (or most negative) bar.

However, when comparing representations of probability distributions, it may not be so simple to extract this information. Instead of comparing fixed values, the user needs to perform statistical inferences based on the given distributions [7]. Furthermore, a change in mindset is required: instead of asking whether or not a particular fact is true, the analysts can only estimate the likelihood of a fact being true or not.

For example, for the extreme value tasks, the question changes to be "what aggregates are *likely* to be the maximum or minimum?" These cannot be read directly off of a set of bars with uncertain ranges: a user would need to estimate how much uncertainty is represented by error bars, and how likely that makes a maximum or minimum measure. In Figure 1, we can be quite confident that 1995 represents the highest aggregate value; but while it is *likely* that 1992 is the lowest, there are several other possibilities, too. Several different bars might have overlapping confidence intervals, and so the correct answer might not be a single value, but a distribution.

The visualizations that we discuss in upcoming sections (and shown in Figures 2 and 3) are designed to allow users to answer these questions directly and visually, rather than by making mathematical inferences.

THE VISUAL ANALYSIS ENVIRONMENT

To begin our design, we selected two core data visualizations: the *bar chart* and the *ranked list*. Bar charts, of course, are ubiquitous; they are a core of every visualization toolkit, and are used to represent many sorts of data. Ranked lists are used to represent sorted elements, and often show just the top few bars of a broad histogram. For example, when exploring search logs with millions of entries, a researcher might wish to see the top 10 most-frequent queries. These lists, truncated to the top values, are particularly relevant when the number of distinct results is too high to be shown on a single chart.

Ranked lists are particularly interesting because they can be unstable in an incremental analysis environment. As an incremental system processes increasing amounts of data, its estimate for the top few items can change, sometimes radically. As more data arrives, the top few items gradually stabilize; one at a time, additional items would also stay in place. Gratzl *et al* [10] present a visual treatment for showing how a ranked list changes across different attributes; their mechanism does not address uncertain rankings.

Uncertain ranked lists can be seen as having a partial order: we are certain that some items will be greater than others, but may be uncertain about other pairwise relationships. Soliman and Ilyas [20] provide a mathematical basis for rapidly evaluating rankings as a partial order; they do not present a user interface for interacting with rankings.

Other visualizations, such as line charts, scatterplots, and parallel coordinates, might also be interesting to examine; we leave those for future work.

Tasks for Visual Analysis

Our goal was to design a visual data analysis environment containing summaries for bar charts and ranked lists that supported sample based analysis. We selected some particularly relevant tasks from Amar *et al* [1, 2]. For the **bar chart**, we support *compare pair of bars; find extrema; compare values to a constant;* and *compare to a range*. Amar *et al* also suggest the task *sort values*. For the **ranked list**, we selected two tasks based on sorting a list: *identify which item is likely to fall at a given rank,* and *identify which items are likely to fall between a given pair of rankings***.** This latter task includes identifying all objects that fall in the top 3, but also every item ranked between 10 and 20.

Computational Framework

It can be challenging to compute the statistical tests required to compare distributions. If we assume independent normal distributions, the simplest operations—such as comparing a distribution with a constant, or comparing two distributions—can be computed using standard techniques such as t-tests. However, there is no simple closed form for many other distributions and tasks.

To address this problem, we have constructed a two-phase computational framework that applies to all of the visualizations. The first phase is an uncertainty quantification phase, in which we estimate the probability distribution from the aggregate we are interested in. As a heuristic, we use the Central Limit Theorem to estimate confidence intervals based on the count, standard deviation, and running average of items we have seen so far. We create one distribution for each aggregate on the chart; we will later interpret these distributions as bars with confidence intervals.

In the second phase, we use these distributions to compute probabilities using a Monte-Carlo approach. (This method is adapted from a technique in the statistical simulation community [9]). We represent each task by a corresponding non-probabilistic predicate (that is, an expression that has a true or false value) that refers to samples. For example, the task 'is the value of the distribution D1 likely to be greater than D2' corresponds to the predicate 'a sample from D1 is greater than a sample from D2.'

From each distribution, we repeatedly draw samples and evaluate the predicate against the samples. We repeat this process a large number of times—in this paper, 10,000 times. We approximate the probability of an event as the fraction of those iterations in which the predicate is true. Table 1 shows an example of this process for two normal distributions D1 and D2 and the predicate D1 > D2. In the simplified example, we take six samples; the predicate is evaluated on each.

Although this approach computes only approximate probabilities, it is able to compute general predicates for any probability distributions, with the only requirements that we can draw samples from the distributions and can assume the distributions are independent. While many iterations are needed for precision, given the speed of computing systems, we find in practice that this computation can be done interactively.

Table 1: Evaluating the probability of D1 > D2, where D1~$\mathcal{N}(5, 9)$ and D2~$\mathcal{N}(4, 16)$, from on random samples (S1..S6). The resulting approximation is p (D1>D2) ≈ 4/6.

	S1	*S2*	*S3*	*S4*	*S5*	*S6*
D1	2.92	7.92	4.38	4.16	12.1	5.15
D2	5.16	2.26	0.69	3.77	3.43	7.23
D1>D2	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE

THE DESIGN OF SAMPLE-BASED VISUALIZATIONS

Our goal is to assist data analysts in making decisions about uncertain data. We expect those analysts to be at least familiar with bar charts with confidence intervals, and so our design extends existing familiar visual representations. Our system should allow them to carry out the tasks listed above.

Design Goals

After reviewing literature in visualization and interface design, we settled on these design goals:

Easy to Interpret: Uncertainty is already a complex concept for users to interpret; our visualizations should add minimal additional complexity. One useful test is whether the visualization converges to a simple form when all the data has arrived.

Consistency across Task: One elegant aspect of the classic bar chart is that users can carry out multiple tasks with it. While we may not be able to maintain precisely the same visualization for different uncertain tasks, we would like a user to be able to change between tasks without losing context on the dataset.

Spatial Stability across Sample Size: In the case of incremental analysis [8, 11], where samples grow larger over time, the visualizations should be change as little as possible. In particular, it should be possible to smoothly animate between the data at two successive time intervals: changes in the visualization should be proportionate to the size of the change in the data. This reduces display changes that would distract the user for only minor data updates.

Minimize Visual Noise: We would like to ensure that the visualization is not confusing. If the base data is displayed as a bar chart, showing a second bar chart of probabilities is likely to be more confusing than a different visual representation.

To fulfill these criteria, we apply interactive annotations [13] to the base visualizations. The annotations will show the results of task-based queries against the dataset. We select particular annotations that we believe will minimize confusion.

Visual Annotations

In this section, we outline the five different task-based annotations that we have created. Each annotation corresponds to a task or group of closely-related tasks. In our prototype interface, a user can select from these annotations; the display adapts appropriately.

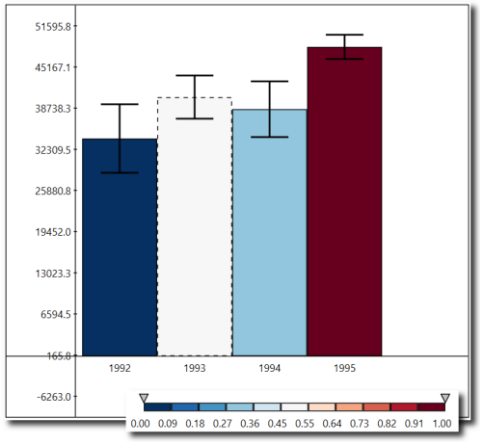
Compare Bars to Each Other

The Compare Bars tool is used to directly compare the distributions in the plot. The user selects one of the distributions; the system compares all the distributions against the selected one. Each bar is colored by the probability that its distribution is larger than the selected bar. A divergent color scale ranges from 0% likely—that is, "is definitely smaller"—to 100%, "definitely larger." At the center, we use white coloring to represent "unknown". This tool is illustrated in Figure 2(a).

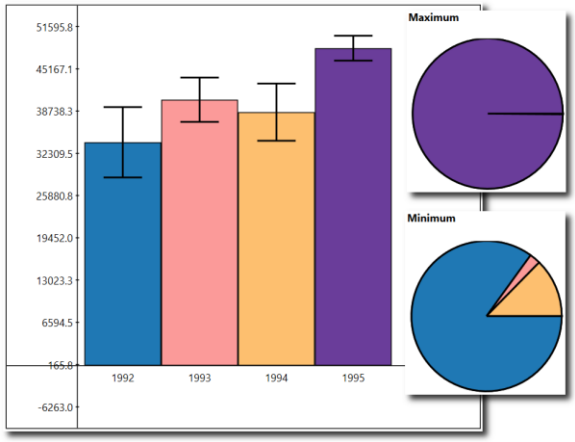
Identify Minimum and Maximum

The Extrema tool is used to quantify the probability that any bar would be either the maximum or minimum among all the distributions. We compute the probability that each bar represents the minimum; separately, we compute the probability it represents the maximum. The total probability across all bars must equal 100%, and so we map the data to a pair of pie charts. Pie charts avoid the confusion of presenting a second, different bar chart.

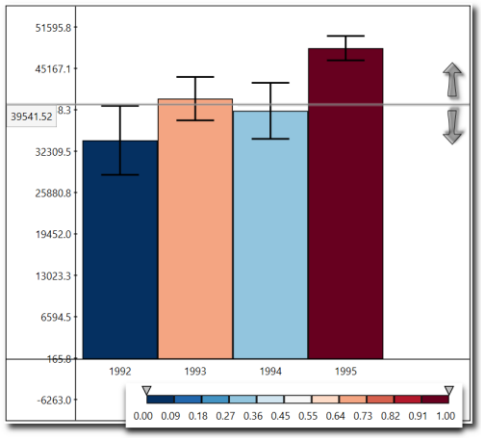
A qualitative color mapping is used to identify bars and the regions in the pie charts. We note that this color map would not scale to large numbers of bars. In those cases, we could consider coloring only bars that are candidates for the top position. When even that is infeasible, the ranked list visualization, below, is a better choice. This tool is illustrated in Figure 2(b).



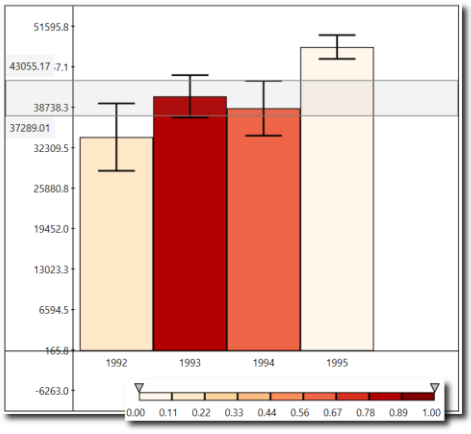
(a) Comparing bars to each other. We compare the white bar to the others; dark blue means "certainly below", while dark red means "certainly above."



(b) Identify minimum and maximum: the pie charts show the probability that any given bar could be the maximum or minimum value.



(c) Compare each bar to a fixed value. The user can move the line.



(d) Compare each bar to a range. Dark colors mean "likely to be inside the range", light ones mean "outside the range."

Figure 2. Four of the tasks and their visual representations. All data is the same as in Figure 1.

Compare to Constant

This annotation enables users to compare a given value to the probability distributions represented by the error bars. Users drag a horizontal line representing a constant value; the probability that the distribution is larger than this constant value is mapped as a color to the corresponding bar. As with the bin comparison, a divergent color scale is used to represent the full space from "definitely lower" to "definitely higher". The tool is illustrated in Figure 2(c).

Compare to Range

The Range tool is similar to comparing to a constant. It is used to evaluate the probability of a distribution's value falling within a range. Users can drag and scale a horizontal strip. The probability that the distribution represented by the error bar is contained in the region is mapped as a color to the corresponding bar. Unlike the comparison tools, which map to a divergent color scheme, this uses a single-ended palette; it only tests whether the value is likely to be inside or outside the range. This tool is illustrated in Figure 2(d).

Find Items at Given Rank

The Ranked List tool is used for ranking probability distributions. Without uncertainty, a ranked list has a straightforward presentation. Therefore, to maintain the visual analogy, the visual representation resembles a list. Each line of the list is a single rank; the line is populated by the set of items that have some probability of having that rank. The height, width, and color of each rectangle are mapped to the probability of that ranking. Very unlikely results, therefore, shrink to nothing; likely results take up almost all the space. The bars are sorted in a stable order, and so are easier to find between levels. We use the single-ended color scale to highlight regions of certainty (see Figure 3(d)).

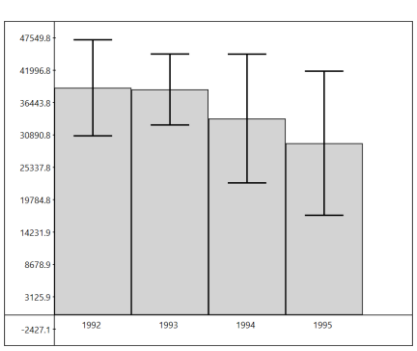
Unlike the other annotations discussed here, this view can also be used in a standalone setting, without being displayed next to a bar chart. This is particularly useful when the number of distributions being ranked is large. This tool is illustrated in Figure 3(b).

Find Items within Ranks

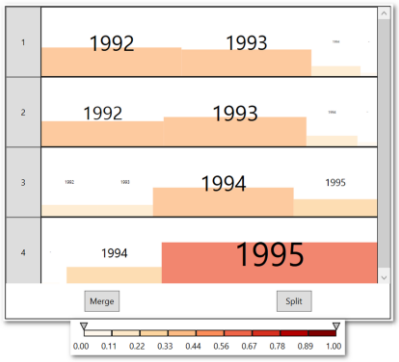
The Ranked List tool is also used to find what items fall within a range of ranks. This would allow a user to learn the set of items that are likely to fall in the top five—without regard for individual rank. That set might be very large when sample sizes are small and uncertainty ranges are high. A user can select the rows to be merged and click the "merge" button. At that point, the system displays the probability that the bars will fall within the range (Figure 3(c)).

Design Discussion

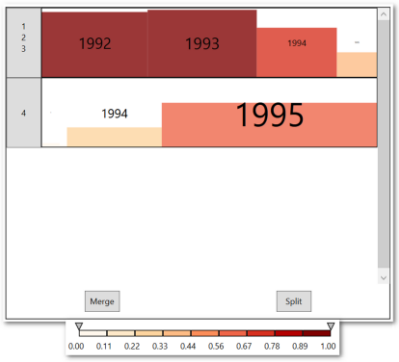
These visual representations share a number of design concepts and themes. In a standard bar chart, these tasks can



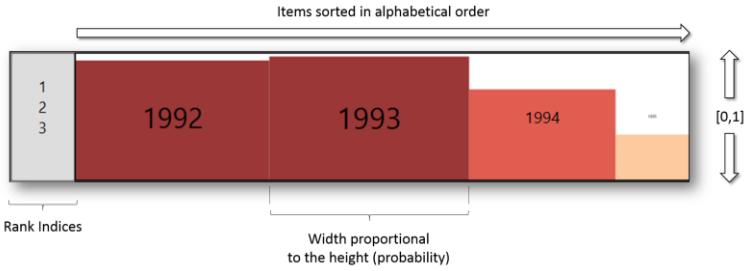
(a) Standard confidence interval bars for a dataset.



(b) The Ranked List visualization corresponding to the bar chart in (a)



(c) Ranked List tool after using the merge operation to compute top-3 probabilities.



(d) Ranked List tool row schematic. Height, width, and color are proportional to the probability that this item will fall in this bin. It is nearly certain that 1992 and 1993 will fall in the first three items; 1994 and 1995 divide the rest.

Figure 3: The Ranked List tool shows the probability of rank orders.

largely be addressed at a glance; in a probabilistic scenario, it requires more work.

All the interactions are lightweight: users need only select the tool, and choose the relevant value. With these simple mechanisms, users can interactively perform complex queries in the data. While "compare bar to bar" and "compare bar to bin" can be visually approximated [7], the other tasks simply cannot be done visually.

Our design process considered several alternative visualizations for these tasks. For example, we considered having matrix-like visualizations to compare each bin against the others. While this would reduce the amount of interaction needed, it would massively increase the complexity of the visualization.

The Sort tool has a more complex design compared to the others, although it is conceptually still very simple. It is basically a list, in which every row represents all the possible values of that row. The redundant mapping—probability maps to height, width, and color—is meant to address three distinct problems. By mapping to *width*, very small bars fall off the chart. By mapping to *height*, a user can easily read across to find high bars: comparing lengths is much harder. Finally, colors help to highlight regions of the list where the rank is certain.

All the color scales were obtained from ColorBrewer [4].

EVALUATION

We conducted an initial user study in order to evaluate the effectiveness of our design. In particular, we wanted to confirm that our techniques were learnable, interpretable, and potentially valuable. Both qualitative and quantitative feedback would help assess whether these annotations would enable users to make better decisions with greater confidence under uncertainty. Because current charting techniques often neglect confidence intervals, it would be important to allow users to compare our annotations to both plain bar charts, and to charts that had traditional confidence intervals.

Our working hypotheses are that users with our system will be (H1) *more accurate* in their answers to these questions, and be (H2) *more confident* in their answers. We do not expect them to be *faster* to respond, as our method requires additional interaction.

Study Design

Our study was designed to explore a broad space of possibilities in order to understand the use of each of our annotations. We ask about five different question types: *compare-to-constant*, *compare-to-bar*, *find-minimum*, *find-maximum*, and *top-k*.

Our study design compares three visual conditions. In the first condition, the user can see only a basic bar chart with neither error bars nor annotations. In the second, we present a bar chart with confidence intervals. In the third, users begin with confidence intervals, but may also turn on the annotations using a menu. The study apparatus is shown in Figure 5. In all conditions, users can see the amount of data that this question represents.

We wished to select a scenario that would be closely resemble the ways that users might really deal with this system. Thus, we wanted queries that a user might realistically run, at a reasonable scale, and based on realistic

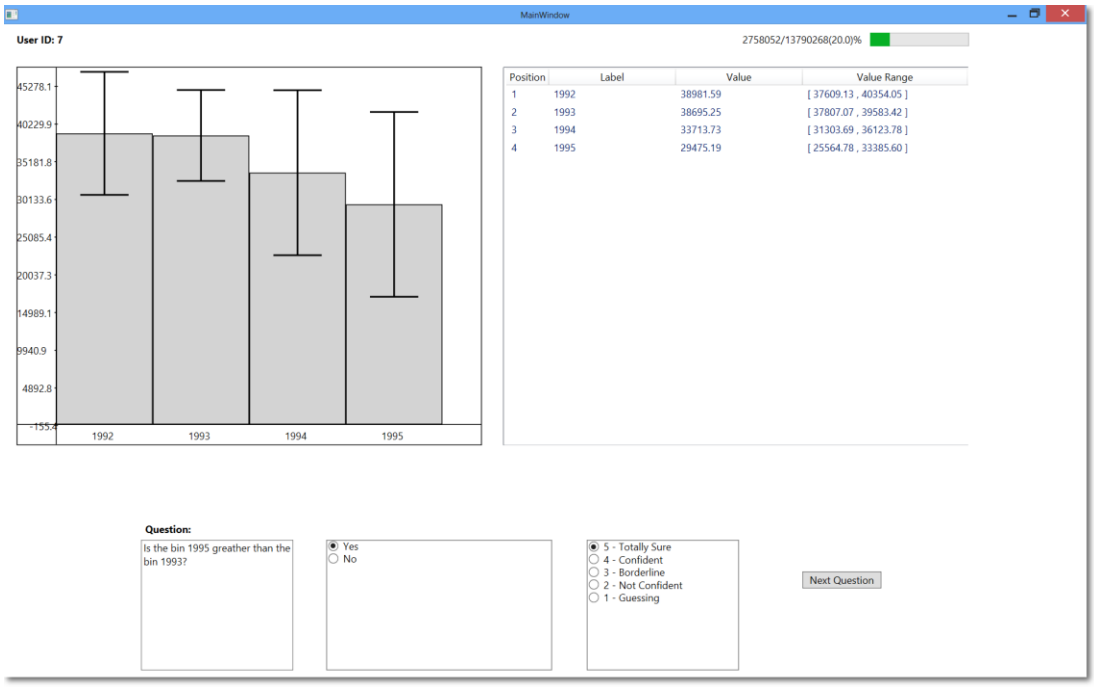


Figure 5: The study apparatus. This user is being asked a question in the error bar condition. The bar at top right shows that this question is based on 20% of the data.

data. We selected TPC-H[1], a standard decision support benchmark designed to test performance of very large databases with realistic characteristics. To generate realistic data, we generated skewed data (with a Zipfian skew factor of z=1, using a standard tool[2]). Part of TPC-H is a series of testing queries with many sample parameters. Different parameters to the query produce different results. We selected one query, Q13, which produces a bar chart of four or five bars. The raw Q13 data table carries 13 million rows.

To simulate an analysis scenario, we randomly sampled the TPC-H tables at five different fractions, from 10% of the data through 50% of the data. Because the Q13 query is very restrictive, each bar only represented a couple of dozen or hundred (and not several million) data points.

A single question, then, is a combination of a question type (see Figure 6), a visual condition (PLAIN, ERROR BARS, or ENHANCED), a sample size, and a parameter to the question.

Our study uses a repeated-measures design. Each user answered 75 questions in random order. We balanced within users by question type, and randomly assigned the other values. Questions were roughly balanced: no user answered fewer than 19 questions in any condition, nor more than 30.

Is the bin 1995 larger than 47000? (True/False)
Is the bin 1994 greater than the bin 1995? (True/False)
Which bar is most likely to be the minimum? (Choice of four)
What are the most probable top 3 items? (Choice of four)

Figure 6: Sample questions from the user study illustrate the tasks: compare to value, compare bars, find extrema, and ranked list.

We also wanted to understand how *certain* users were about their answers: we expected the system to make more of a difference in marginal cases where confidence intervals were broad; when confidence intervals are narrow, certainty is less interesting. Users rated confidence on a five-point Likert scale from "completely uncertain" to "completely certain."

For each question, user selected an answer, self-rated their certainty in that answer, and then pressed "next question." We logged the answer, their confidence in the answer, and the time it took to answer. After the experiment users were presented with a questionnaire that to assess their overall user experience.

Participants

As described earlier, our techniques are designed to enhance traditional confidence intervals for data analysts with at least basic training in statistics. While our annotations might also be valuable to non-experts, we wanted to understand the value they provided over confidence intervals.

For this preliminary study, we recruited seven participants. All were male graduate students in computer science; all were generally familiar with reading charts and interacting with data. All had at least basic statistical training, have some familiarity with confidence intervals and error bars, and had used analytics systems.

RESULTS

Comments and Feedback from Users

During the training before the study, all of our subjects learned the system and visualizations quickly and reported that they felt comfortable using them. Users had no difficulty understanding the purpose for the enhancements.

After the study, we debriefed the users. Our users understood all of the annotations. User 2, for example, had avoided dealing with confidence intervals before, as he found them difficult; using our system, he said, "It is good that I don't need to do much thinking." Users were least happy with the sort tool; several complained that it was too complex to use easily. While it was designed to be a variant on a traditional list, it may have added too much material.

We wanted to better understand how users made decisions about their confidence in a visualization. In the baseline PLAIN condition, users had very few cues to guess how broad the confidence intervals were; several reported that they eyeballed their confidence by looking at the progress bar in the top right: they felt more confident with larger dataset sizes, and less confident with smaller ones.

In the annotated condition, in contrast, users had several different cues to judge confidence. Indeed, user 4 complained that in the annotated condition, he had "too many things to consider:" sample size, error bars and annotations. Another user said he did not feel confident in any answer when the sample size was small. This is an interesting misperception: in theory, the sample size should not matter at all to the analysis. Confidence intervals should provide at least as much information as the progress bar would have; our annotations should override confidence intervals. Users still attempted to juggle all three.

Quantitative Results

Because accuracy and confidence are on ordered, categorical data, we carried out non-parametric Kruskal-Wallis chi-squared test to compare accuracy and confidence across conditions.

Overall, our users were very accurate, getting 84% of all questions right. There was no difference in overall accuracy between the three conditions, and so H1 was not supported ($\chi^2 = 2.2968$, $df = 2$, $p = 0.3171$). We see, however, that users

[1] http://www.tpc.org/tpch

[2] Program for TPC-H Generation with Skew: ftp://ftp.research.microsoft.com/users/viveknar/TPCDSkew

made fewer mistakes with larger samples—virtually no one got questions wrong with the larger sample set, but many did get them wrong with small samples. Figure 7 looks at accuracy by sample size across the three conditions.

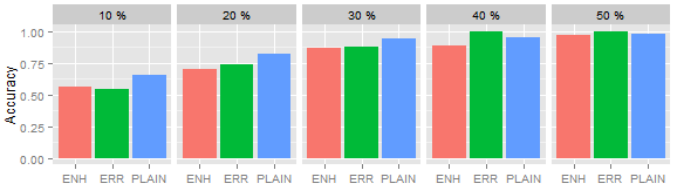


Figure 7: Average accuracy by sample size. Across all conditions, users are more accurate with access to more data.

We now turn to confidence. As Figure 8 suggests, users in the ENHANCED condition largely felt more confident in their results than the other users. H2 was supported (χ^2 = 32.9335, $df = 2$, $p << 0.001$).

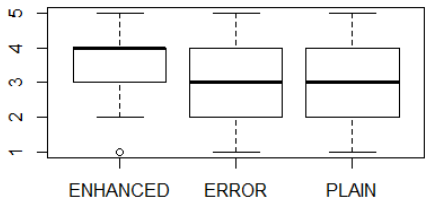


Figure 8: Confidence by condition, across all sample sizes and tasks. Users in the ENHANCED condition were more confident in their answers.

We wanted to understand the interaction between confidence and accuracy—we wanted to ensure we did not deliver confidence without accuracy. However, we do not expect our system to deliver accuracy at all levels: we expect our system to provide justified confidence. That is, a user using our system should be confident when they are right, and conversely feel unsure when they do not have sufficient information.

To explore this idea, in Figure 9, we bucket confidence into three categories. In the PLAIN condition, users maintain approximately the same level of confidence: in other words, being right and being confident are unrelated. In contrast, in the ENHANCED condition, the highly-confident users were very likely to be right; the less-confident users were comparatively more likely to be wrong. Not only that, but from the test for H2, we know that users are more likely to be confident with our system. We believe this is good preliminary evidence that our visualization helps encourage justified confidence.

DISCUSSION & FUTURE WORK

Our annotations did not increase raw accuracy. Instead, we have suggested that they increase what we call "justified confidence." To pursue this further, though, we would need more ambiguous questions: as is reflected by the high accuracy rates, a number of the questions were too easy for users. In future tests of user interaction with uncertainty, it may be worth looking at techniques that would generate questions with more ambiguity.

We have shown how these annotations could be applied to a bar chart with error bars; however, our design principles are very general: almost any aggregate chart type could presumably be adapted to show task annotations. Indeed, we suspect that more complex charts would benefit even more from our techniques.

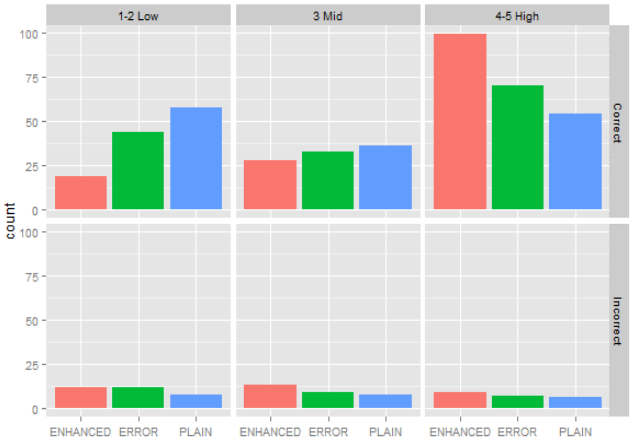


Figure 9: In the three conditions, tallies of confidence against accuracy. Trials in the 'Enhanced' condition with high confidence were more likely to be correct than in the 'Plain' condition.

Similarly, the Monte-Carlo framework that we outline is highly adaptable to other tasks. It could be incorporated into a variety of tasks beyond those in this paper. For example, multiple range tools could be combined to test the likelihood of being within a disjoint union of ranges.

We are currently incorporating the system discussed in this paper within a progressive data processing framework; we hope to make interacting with uncertainty and samples an everyday part of its users' experiences.

CONCLUSION

Many data systems use sampled data, either for progressive computation or because sample data is the available or affordable subset. Drawing confidence intervals can help as a static view, but cannot help users handle more sophisticated queries against their visualizations data.

Tasks involving probability and confidence intervals have been shown to be difficult, even for experts. Past work has looked mainly at interpreting *whether* a given point was uncertain, and *how* uncertain it is. In this work, we have expanded that to look at techniques that will allow users to make use of that uncertainty—to predict when one value is likely to be higher than another, or to look at the ranked sequence of values. These techniques allow users to directly read the answers to these tasks off of the chart, analogously to the way that non-probabilistic data can be read directly off a bar chart without confidence intervals.

Our experiment suggests that enhancing bar charts with task-specific annotations may indeed help users make decisions about samples. While we did not show in this context that users would be more accurate, we *did* show that they would be more confident in their accurate responses (and, conversely, would know when not to be confident.) This seems a desirable trait in a system based on partial data: we would like analysts to be able to make decisions about when to terminate expensive and slow queries.

The current reliance on variations of the box plot is insufficient for real data fluency—we need to broaden our tools for visualizing uncertainty, not only of individual levels, but of complex operations on data.

ACKNOWLEDGEMENTS

Our thanks to the MSR Big Sky team, who are applying these concepts, and the participants of our study. The first author was partially supported by the National Science Foundation grant MRI-1229185.

REFERENCES

1. R. Amar and J. Stasko. A knowledge task-based framework for design and evaluation of information visualizations. *IEEE Symp. on Information Visualization, 2004. INFOVIS 2004.* (pp. 143-150).
2. R. Amar, J. Eagan, J. Stasko. Low-level components of analytic activity in information visualization. *IEEE Symp. on Information Visualization, 2005. INFOVIS 2005.* (pp. 111-117).
3. S. Belia, F. Fidler, J. Williams, G. Cumming. Researchers misunderstand confidence intervals and standard error bars. *Psychological methods*, *10*(4), 389-396, 2005.
4. C. Bewer., G. W. Hatchard and Mark A. Harrower, 2003, ColorBrewer in Print: A Catalog of Color Schemes for Maps, Cartography and Geographic Information Science 30(1): 5-32.
5. N. Boukhelifa, A. Bezerianos, T. Isenberg, J. D. Fekete. Evaluating Sketchiness as a Visual Variable for the Depiction of Qualitative Uncertainty. *IEEE Trans. on Vis. and Comp. Graphics*, 18(12), 2769–2778, 2012.
6. G. Cumming. Understanding the new statistics: Effect sizes, confidence intervals, and meta-analysis. New York, Routledge, 2012.
7. G. Cumming, S. Finch. Inference by eye: Confidence intervals and how to read pictures of data. *American Psychologist*, 60(2), 170–18, 2005.
8. D. Fisher, I. Popov, S. M. Drucker, and mc schraefel. Trust Me, I'm Partially Right: Incremental Visualization Lets Analysts Explore Large Datasets Faster. *ACM Conf. on Human Factors in Comp. Systems*. CHI 2012. (pp. 1673-1682).
9. D. Goldsman, B. Nelson, and B. Schmeiser. Methods for Selecting the Best System. *Proceedings of the 1991 Winter Simulation Conf.* 177-186.
10. S. Gratzl, A. Lex, N. Gehlenborg. LineUp: Visual Analysis of Multi-Attribute Rankings. *IEEE Trans. on Vis. and Comp. Graphics* 2013
11. J. Hellerstein, R. Avnur, A. Chou, C. Olston, V. Raman, T. Roth, C. Hidber, P. Haas. Interactive Data Analysis with CONTROL. *IEEE Computer*, 32(8), 51-59, 1999.
12. S. Kandel, A. Paepcke, J. M. Hellerstein, J. Heer. Enterprise data analysis and visualization: An interview study. *IEEE Trans. on Vis. and Comp. Graphics*, *18*(12), 2917-2926.
13. N. Kong, M. Agrawala. Graphical Overlays: Using Layered Elements to Aid Chart Reading. *IEEE Trans. on Vis. and Comp. Graphics,* 18(12), 2631-2638.
14. A. M. MacEachren, A. Robinson, S. Hopper, S. Gardner, R. Murray, M. Gahegan, E. Hetzler. Visualizing geospatial information uncertainty: What we know and what we need to know. *Cartography and Geographic Information Science*, *32*(3), 139-160.
15. L. Micallef, P. Dragicevic, J. D. Fekete. Assessing the Effect of Visualizations on Bayesian Reasoning through Crowdsourcing. *IEEE Trans. on Vis. and Comp. Graphics*, 18.12 (2012): 2536-2545.
16. M. Skeels, B. Lee, G. Smith, and G. Robertson. Revealing Uncertainty for Information Visualization. *In Proc. of the Working Conf. on Advanced Visual Interfaces*. ACM, New York, NY, USA. 2008, 376-379.
17. F. Olken, D. Rotem. Random sampling from database files: a survey. In *Proc. of the 5th Int'l Conf. on Statistical and Scientific Database Management (SSDBM'1990)*, Zbigniew Michalewicz (Ed.). Springer-Verlag, London, UK, 92-111. 1990.
18. C. Olston, J. Mackinlay. Visualizing data with bounded uncertainty. *IEEE Symp. on Information Visualization, 2002. INFOVIS 2002.* (pp. 37-40).
19. J. Sanyal, S. Zhang, G. Bhattacharya, P. Amburn, and R. Moorhead. A User Study to Compare Four Uncertainty Visualization Methods for 1D and 2D Datasets. *IEEE Trans. on Vis. and Comp. Graphics* 15(6), 1209-1218.
20. M. A. Soliman, I. F. Ilyas. Ranking with uncertain scores. Data Engineering, 2009. ICDE'09. IEEE 25th International Conference on. IEEE, 2009.
21. A. Tversky, D. Kahneman. Judgment under Uncertainty: Heuristics and Biases. *Science*, 185 (1974). 1124-1131.
22. H. Wickham, L. Stryjewski. 40 Years of Boxplots. Technical Report from http://vita.had.co.nz/. 2012.
23. T. Zuk, S. Carpendale. Visualization of Uncertainty and Reasoning. In *Proceedings of the 8th Int'l Symp. on Smart Graphics (SG '07)*. Springer-Verlag, Berlin, Heidelberg. 2007