Avoiding Drill-down Fallacies with *VisPilot*: Assisted Exploration of Data Subsets

Doris Jung-Lin Lee†, Himel Dev†, Huizi Hu†, Hazem Elmeleegy★, Aditya Parameswaran†
†University of Illinois, Urbana-Champaign, ★Google, Inc.
jlee782|hdev3|huizihu2|adityagp@illinois.edu,elmeleegy@google.com

ABSTRACT

As datasets continue to grow in size and complexity, exploring multi-dimensional datasets remain challenging for analysts. A common operation during this exploration is drill-down—understanding the behavior of data subsets by progressively adding filters. While widely used, in the absence of careful attention towards confounding factors, drill-downs could lead to inductive fallacies. Specifically, an analyst may end up being "deceived" into thinking that a deviation in trend is attributable to a local change, when in fact it is a more general phenomenon; we term this the *drill-down fallacy*. One way to avoid falling prey to drill-down fallacies is to exhaustively explore all potential drill-down paths, which quickly becomes infeasible on complex datasets with many attributes. We present VisPilot, an accelerated visual data exploration tool that guides analysts through the key insights in a dataset, while avoiding drill-down fallacies. Our user study results show that VisPilot helps analysts discover interesting visualizations, understand attribute importance, and predict unseen visualizations better than other summarization baselines.

ACM Reference Format:
Doris Jung-Lin Lee†, Himel Dev†, Huizi Hu†, Hazem Elmeleegy★, Aditya Parameswaran†. 2019. Avoiding Drill-down Fallacies with *VisPilot*: Assisted Exploration of Data Subsets. In *IUI Conference on Intelligent User Interfaces Proceedings (IUI 2019), March 17–20, 2019, Los Angeles, USA.* ACM, New York, NY, USA, 13 pages. **https://doi.org/10.1145/XXXXXX.XXXXXX**

1 INTRODUCTION

Visual data exploration is the *de facto* first step in understanding multi-dimensional datasets. This exploration enables analysts to identify trends and patterns, generate and verify hypotheses, and detect outliers and anomalies. However, as datasets grow in size and complexity, visual data exploration becomes challenging. In particular, to understand how a global pattern came about, an analyst may need to explore different subsets of the data to see whether the same or different pattern manifests itself in these subsets. Unfortunately, manually generating and examining each visualization in this space of data subsets (which grows exponentially in number of attributes) presents a major bottleneck during exploration.

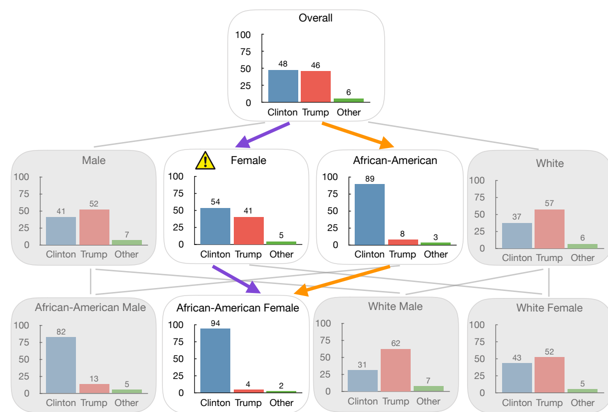

Figure 1: Example data subset lattice from the 2016 US election dataset illustrating the drill-down fallacy along the purple path as opposed to the informative orange path.

One way of navigating this combinatorial space is to perform *drill-downs* on the space—a *lattice*—of data subsets. For example, a campaign manager who is interested in understanding voting patterns across different demographics (say, race, gender, or social class) using the 2016 US election exit polls [1] may first generate a bar chart for the entire population, where the x-axis shows the election candidates and the y-axis shows the percentage of votes for each of these candidates. In Figure 1, the visualization at the top of the lattice corresponds to the overall population. The analyst may then use their intuition to drill down to specific demographics of interest, say gender-based demographics, by generating bar charts for female voters by following the purple path, as shown in the second visualization at the second row of Figure 1, and then to the visualization corresponding to `African-American Female` voters in the third row.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-1-4503-5970-2/19/05...$15.00
https://doi.org/10.1145/XXXXXX.XXXXXX

Challenges with Manual Drill-down. There are three challenges associated with manual drill downs:

First, it is often not clear which attributes to drill-down on. Analysts may use their intuition to select the drill-down attribute, but such arbitrary exploration may lead to large portions of the lattice being *unexplored*—leading to missed insights.

Second, a path taken by analysts in an uninformed manner may lead to visualizations that are *not very surprising or insightful*. For example, an analyst may end up wasting effort by drilling down from the `African-American` visualization to the `African-American Female` one in Figure 1, since the two distributions are similar and therefore not very surprising.

Third, an analyst may encounter a ***drill-down fallacy***—a new class of errors in reasoning we identify—where incomplete insights result from potentially confounding factors not explored along a drill-down path. As shown in Figure 1, an analyst can arrive at the `African-American Female` visualization via the purple or the orange drill-down path. An analyst who followed the purple path may be surprised at how drastically the `African-American Female` voting behavior differs from that of `Female`. However, this behavior is not surprising if the analyst had gone down the orange path that we saw earlier, where the proper reference (i.e., the distribution for `African-American`) explains the vote distribution for `African-American Female`. In other words, even though the vote distribution for `African-American Female` is very different from that of `Female`, the phenomenon can be explained by a more general "root cause" attributed to the voting behavior for the `African-American` community as a whole. Attributing an overly specific cause to an effect, while ignoring the actual, more general cause, not only leads to less interpretable explanations for the observed visualizations, but can also lead to erroneous decision-making. For example, for the campaign manager, this could lead to incorrect allocation of campaign funds. To prevent analysts from fall prey to such drill-down fallacies—consisting of misleadingly "surprising" local deviations in trend during drill-down (`Female` → `African-American Female`)—it is important to preserve the proper parent reference (`African-American`) to contextualize the behavior of the visualization of interest (`African-American Female`). One approach to avoid this fallacy is to exhaustively explore all potential drill-down paths. Unfortunately, this approach does not scale.

While there have been a number of statistical reasoning fallacies that have been identified in visual analytics, including Simpson's paradox [6, 14], multiple comparisons [39], and selection bias [12], to the best of our knowledge, our paper is the first to identify the drill-down fallacy, a common fallacy that appears during manual data exploration. There have been efforts to develop visualization recommendation systems [24, 34] that assist or accelerate the process of visual data exploration [7, 20–22, 32, 34, 37], none of these systems have provided a conclusive solution to the problem of aiding drill-downs to explore data subsets, while avoiding drill-down fallacies. We discuss related work in detail in Section 7.

VisPilot with Safety, Saliency, and Succinctness. We present a visual data exploration tool, titled VisPilot, that addresses the three aforementioned challenges of exploration by espousing three principles: (i) **Safety** (i.e., ensure that proper references are present to avoid drill-down fallacies), (ii) **Saliency** (i.e., identify interesting visualizations that convey new information or insights), and (iii) **Succinctness** (i.e., convey only the key insights in the dataset). To facilitate safety, we develop a notion of *informativeness*—the capability of a reference parent visualization to explain the visualization of interest. To facilitate saliency, we characterize the notion of *interestingness*—the difference between a visualization and its informative reference in terms of underlying data distribution. Finally, to facilitate succinctness, we embrace a collective measure of visualization utility by recommending a *compact* connected network of visualizations. Based on these three principles, VisPilot *automatically identifies a **compact** network of **informative** and **interesting** visualizations that collectively convey the key insights in a dataset*. Our user study results demonstrate that VisPilot can help analysts gain a better understanding of the dataset and help them accomplish a variety of tasks. Our contributions include:

- Identifying the notion of a *drill-down fallacy*;
- Introducing the concept of *informativeness* that helps identify insights that arise from something that holds in the data (as opposed to confounding local phenomena);
- Extending the concept of informativeness to a measure to quantify the benefit of a network of visualizations;
- Designing VisPilot, which efficiently and automatically identifies a network of visualizations conveying the key insights in a dataset; and
- Demonstrating the efficacy of VisPilot through a user study evaluation on how well users can retrieve interesting visualizations, judge the importance of attributes, and predict unseen visualizations, against two baselines.

2 PROBLEM FORMULATION

In this section, we first describe how analysts manually explore the space of data subsets. We then introduce three design principles for a system that can automatically guide analysts to the key insights.

Manual Exploration: Approach and Challenges

During visual data exploration, an analyst may need to explore different subsets of the data that together form a combinatorial *lattice*. Figure 1 shows a partial lattice for the 2016

US election dataset. The lattice contains the overall visualization with no filter at the first level, all visualizations with a single filter at the second level (such as `Female`), all visualizations with two filters at third level, and so on. Analysts explore such a combinatorial lattice from top to bottom, by generating and examining visualizations with increasing levels of specificity. In particular, analysts perform *drill-downs* [13] to access data subsets at lower levels by adding one filter at a time (such as adding `African-American` to `Female` along the purple path) and visualize their measures of interest for each data subset—in this case the percentage of votes for each candidate. Further, as analysts perform drill-downs, they use the most recent visualization in the drill-down path—the *parent*—as a *reference* to establish what they expect to see in the next visualization in the path—the *child*. In Figure 1, the visualizations `Female` and `African-American` are the *parents* of the `African-American Female` visualization, explored along the purple and orange path respectively.

As we saw in the purple path in Figure 1, while performing drill-downs, analysts may detect a local deviation (we will formalize these and other notions subsequently) between a parent and a child to be significant. For example, they may be surprised by the fact that the `Female` and `African-American Female` visualizations are very different from each other, and may find this to be a novel insight. However, this deviation is a result of `Female` not being an *informative* parent or reference for `African-American Female`—instead, it is a *deceptive* reference. Here, a different parent, `African-American`, is the most informative parent or reference of `African-American Female` because it is the parent that exhibits the least deviation relative to `African-American Female`. Here, the `African-American Female` visualization is not really all that surprising given the `African-American` visualization. We refer to this phenomenon of being deceived by a local difference or deviation relative to a deceptive reference as an instance of the *drill-down fallacy*. One way to avoid such fallacies is to ensure that one or more informative parents are present for each visualization so that analysts can contextualize the visualization accurately. While this fallacy is applicable to any chart type that can be described as a probability distribution over data (e.g., pie charts, heatmaps), we will limit our discussion to bar charts for brevity.

Design Principles for Informative Exploration

Our goal is to help analysts discover the key insights in a dataset, while avoiding drill-down fallacies. We outline three essential principles for finding such insights—the three S's: *safety*, *saliency*, and *succinctness*, and progressively layer these principles to formalize a measure of utility for a network of visualizations. We adopt these principles to develop a visual exploration tool that automatically generates a network of visualizations conveying the key insights in a multidimensional dataset.

Safety. To prevent drill-down fallacies, we ensure *safety*—by making sure that informative parents are present to accurately contextualize visualizations. A parent is said to be *informative* if its data distribution closely follows the child visualization's data distribution, since the presence of the parent allows the analysts to form a accurate mental model of what to expect from the child visualization. We compute the informativeness of the j^{th} parent V_i^j for a visualization V_i as the similarity between their data distributions measured using a distance function D. For bar charts, the data distribution refers to the height of bars assigned to the categories labeled by the x-axis, suitably normalized. Accordingly, the computed distance $D(V_i, V_i^j)$ refers to the sum of the distances between the normalized heights of bars across different categories. Quantifying deviation using distances between normalized versions of visualizations in this manner is not new—we leverage prior work for this [10, 25, 32, 34]. The specific distance measure D is not important; while we use the Euclidean metric, we can easily work with other common distance metrics such as Kullback-Leibler Divergence and Earth Mover's distance [34]. The most informative parent $V_i^\dagger$ for a visualization V_i is the one whose data distribution is most similar to V_i.

$$V_i^\dagger = \underset{V_i^j}{argmin}\, D(V_i, V_i^j) \tag{1}$$

Instead of insisting that the most informative parent is always present to contextualize a given child visualization, we relax our requirement somewhat: we don't need *the most* informative parent to be present, just *an* informative parent. We define a parent to be informative (denoted V_i^*) if its distance from the child falls within a threshold θ% of the most informative parent—the default is set to 90% and adjustable by the user.

Saliency. Simply ensuring that informative parents are present is insufficient; we also want to emphasize *saliency* by identifying visualizations that convey new information. In general, a visualization is deemed to be *interesting* if its underlying data distribution differs from that of its parents, and thus offers new unexpected information or insight. Such distance-based notions of interestingness have been explored in past work [9, 19, 34], where a large distance from some reference visualization indicates that the selected visualization is interesting. We deviate from this prior work in two ways: first, we concentrate on *informative* interestingness, where the interestingness of a child visualization is only defined in the presence of informative parent references. Second, we weigh the interestingness by the proportion of the population captured by the child visualization. (That is,)Aditya: got till here. However, unlike past work, we concentrate on *informative interestingness*, where we identify interesting visualizations in presence of informative references. Specifically, to model the interestingness of a visualization V_i in the context of its informative parent V_i^*, we characterize the deviation between their data distributions using $D(V_i, V_i^*)$. Notice that our informativeness objective minimizes the distance between parent and child, whereas interestingness objective maximizes the resultant minimum distance. Accordingly, our overall objective function uses a maximin function of distance to capture informative insights. To incorporate the effect of subpopulation size into our objective function, we multiply the distance $D(V_i, V_i^*)$ between an informative parent V_i^* and a child visualization V_i by the ratio of their sizes $U(V_i) = \frac{|V_i|}{|V_i^*|} \cdot D(V_i, V_i^*)$.[1]

Summarization. To succinctly convey insights, we concentrate on *summarization*—identifying a group of visualizations that collectively contain informative insights. Since our aim is to identify a unified narrative, instead of discrete insights, we enforce that any selected visualization must have at least one of its informative parents present in the dashboard. Specifically, we identify a set of k connected visualizations that collectively maximize the sum of the proposed utility $U(V_i)$ across each selected visualization, V_i, and thus succinctly convey informative insights, more formally stated as follows:

PROBLEM. *Given a dataset and user-provided X, Y attributes, select k visualizations from the lattice of data subsets $\mathcal{L}$ to be included in the dashboard, such that:*
(i) one of the selected visualization is the overall visualization, corresponding to the entire dataset with no filter;
(ii) for each visualization except for the overall, at least one of its informative parents is present in the k visualizations;
(iii) the k selected visualizations maximize the total utility $\sum_{V_i \in \mathcal{L}} U(V_i)$ as defined above.

This problem of finding a connected subgraph in the lattice that has the maximum total edge utility is known as the *maximum-weight connected subgraph problem* [4] and is known to be NP-Complete [29]. We design several approximate algorithms to solve this problem efficiently.

3 STORYBOARD SYSTEM

In this section, we present our system, VISPILOT, by first providing a high-level overview of the underlying algorithms, and then describing the user interaction mechanisms.

Lattice Traversal Algorithm

We discuss the algorithm used for traversing the lattice to select the k-connected maximum-weighted subgraph. The objective of the traversal algorithm is to find the connected subgraph in the lattice that has the maximum combined edge utility. The frontier greedy algorithm first compiles a list of

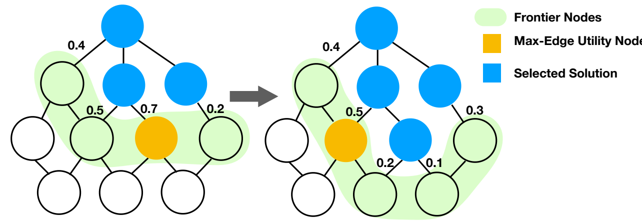

Figure 2: Example illustrating how the frontier greedy algorithm incrementally build up the solution by selecting the maximal-edge utility node from the frontier at every step. On the left, given the three existing nodes in the solution (blue), we select the node with the highest edge utility (yellow) amongst the frontier nodes (green). On the right, the newly added node results in an updated frontier and a maximum-edge utility node is selected amongst them.

candidate nodes known as the *frontier* nodes, which encompasses all nodes that are connected to the existing subgraph solution. As long as the informative parents of frontier nodes are already present in the solution, the frontier nodes can be appended to the current solution without violating requirement (ii) in the problem formulation, enforcing the presence of informative parent for every selected visualization. To obtain the frontier nodes, the algorithm scans and adds all children of leaf nodes of the current dashboard as part of the frontier. As illustrated in Figure 2, at each step, our algorithm greedily picks the node with the maximum edge utility amongst the eligible frontier nodes to add to the current solution, and updates the frontier accordingly.

Algorithm 1 Frontier Greedy Algorithm

```
procedure PICKVISUALIZATIONS(k,lattice)
    dashboard ← { V_overall }
    while |dashboard| < k do
        frontier ← getFrontier(dashboard,lattice)
        maxNode ← getMaxUtilityNode(frontier)
        dashboard ← dashboard ∪ {maxNode}
    return dashboard
```

User Interaction

Given the selected visualizations, we render them in a dashboard, where users can inspect the visualizations through panning and zooming with navigation buttons, mouse clicks, and key bindings. Users can also select the x and y axes of interest, aggregation function, and set the number of visualizations (k) to generate a dashboard. As shown in Figure 3a, the analyst first generates a 7-visualization dashboard for the Police Stop dataset [30]. The dataset contains records of vehicle and pedestrian stops from law enforcement departments in Connecticut, dated from 2013 to 2015. In this case, the analyst is interested in the percentages of police stops (Y)

[1] If multiple informative parents, $V_i^{*,\theta}$, are selected for a given visualization, V_i, then $U(V_i)$ is defined in terms of the most informative *selected* parent.

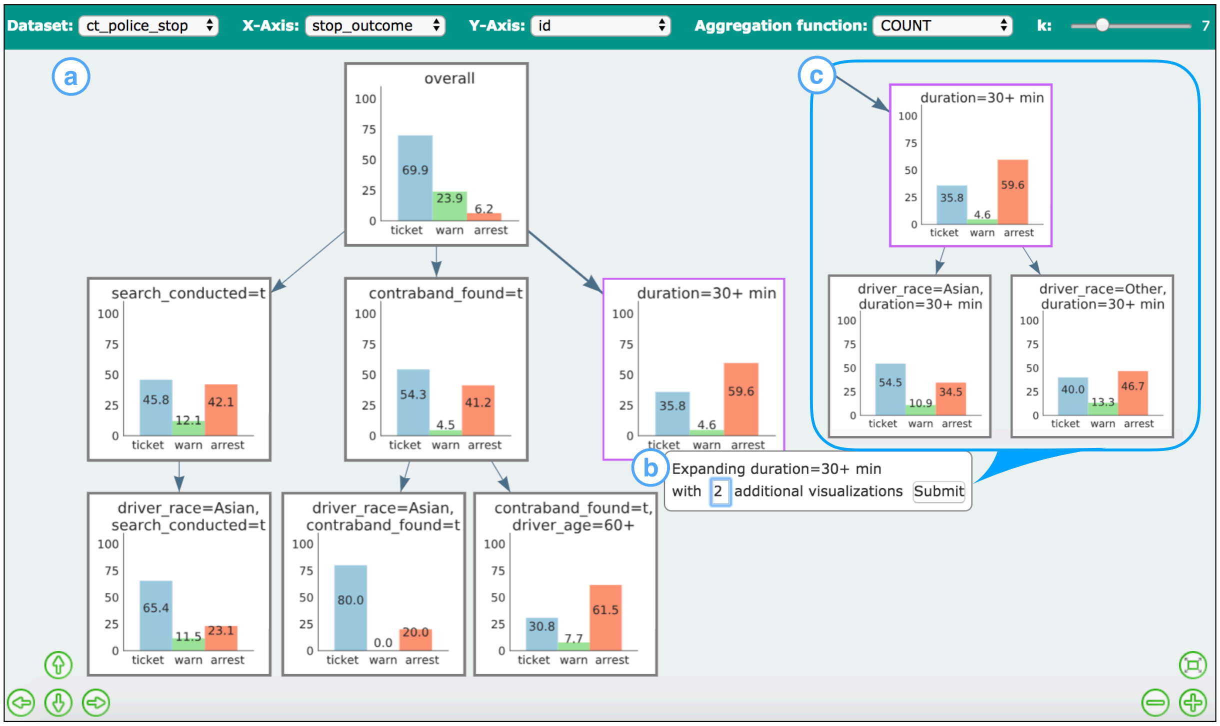

Figure 3: a) Overview of the VISPILOT interface for the Police Stop dataset. Users can select x, y axes, and aggregation function via the dropdown menu, to define the visualization space of interest, as well as adjusting dashboard parameters, such as the number of visualizations to show in the dashboard (k) via the sliders. b) User clicks on the duration=30+min visualization to request 2 additional visualizations. c) A preview of the added portion of the resulting dashboard is shown.

that led to different outcomes (X), such as ticket, warning, or arrest.

After browsing through visualizations in the dashboard, users may be interested in getting more information about a specific visualization. VISPILOT allows users perform additional drill-downs by requesting a new dashboard centered on a chosen visualization of interest as the new starting point (or equivalently, the root of the lattice) for analysis. The analyst learns that for drivers who had contraband found in the vehicle, the arrest rate for those who are 60 and over is surprisingly higher than usual, whereas for Asian drivers the arrest rate is lower. Say the analyst is now interested in learning more about the other factor that contributes to the high arrest rate: duration=30+min. In Figure 3b, she can click on the corresponding visualization to request for additional visualizations. Upon seeing the updated dashboard in Figure 3c, she learns that any visualization that involves the duration=30+min filter is likely to result in high ticketing and arrest rates. This implies that if a police stop lasts more than 30 minutes, the outcome would more or less be the same, independent of other factors such as the driver's race or age. To generate the expanded dashboard, VISPILOT uses the same models and algorithms as before, except the root node is now set as the selected visualization, rather than the overall visualization. This node expansion capability is motivated by the idea of *iterative view refinement* common to other visual analytics systems, which is essential for users to iterate on and explore different hypotheses [17, 37].

4 EVALUATION STUDY METHODS

In this section, we describe the methodology and results for a user study we have conducted for evaluating the utility of VISPILOT. To assess the efficacy of VISPILOT across various exploratory analysis goals, we evaluate how VISPILOT dashboards enable participants to gain *interesting*, *informative*, *summarized* insights (corresponding to the three aforementioned design objectives—saliency, summarization, and safety) compared to baseline approaches.

Participants and Conditions

We recruited 18 participants (10 Male; 8 Female) with prior experience in working with data. Participants included undergraduate and graduate students, researchers, and data scientists, with 1 to 14 years of data analysis experience (average = 5.61). No participants reported prior experience in working with the two datasets used in the study (described below). Participants were randomly assigned two of the three types of dashboards with k=10 visualizations generated by

following conditions. The specific dashboards for each dataset and condition is shown in the technical report for reference.
VisPilot: The dashboards for this condition are generated by the aforementioned frontier greedy algorithm and displayed in a hierarchical layout (as seen in Figure 3). In order to establish a fair comparison with the two other conditions, we deactivated the interactive node expansion capabilities.
BFS (short for breadth-first search): Starting from the visualization of the entire population, k visualizations are selected level-wise, traversing down the subset lattice, adding the visualizations at the first level with 1-filter combination one at a time, proceeding with the 2-, 3-, and so on, until k visualizations have been added to the dashboard. This baseline is designed to simulate a dashboard generated by a meticulous analyst who exhaustively inspects all possible visualizations (i.e., filter combinations) from the top-down. The chosen visualizations are displayed in a 5x2 table in the traversed order.
Cluster: K-Means clustering is performed on the data distributions of all possible visualizations for the dataset. This results in k clusters covering all visualizations of the dataset, corresponding to k, the number of visualizations to be shown in the dashboard. For each representative cluster, we select the visualization with the least number of filter conditions for interpretability[2] and display them in a 5x2 table layout. This baseline is designed to showcase a diverse set of pattern distributions within the dataset.

Each participant was assigned two different conditions on two different datasets. The ordering of each condition was randomized to prevent confounding learning effects. The study began with a 5-minute tutorial using dashboards generated from the Titanic dataset [2] for each condition. To prevent bias across conditions, participants were not provided an explanation of how the dashboards were generated and why the visualizations were arranged in a particular way. Then, participants proceeded onto the aforementioned Police Stop dataset. The attributes in the dataset include driver gender, age, race, stop time of day, stop outcome, whether a search was conducted, and whether contraband was found. We generated dashboards of bar chart visualizations with x-axis as the stop outcome (i.e., whether the police stop resulted in a ticket, warning, or arrest) and y-axis as the percentage of police stops that led to each outcome.

The second dataset in the study is the Autism dataset [11], which includes the result of autism spectrum disorder screening for 704 adults. The attributes in the dataset are binary responses to 10 diagnostic questions that are part of the screening process. This dataset serves as a data-agnostic condition, since there was no descriptions of the questions or answer labels provided to the user. We generate dashboard visualizations based on whether the participant is diagnosed with autism or not.

Study Procedure

Participants were given some time to read through a worksheet containing descriptions of the data attributes. Then, they were given an attention check question where they were given a verbal description of the visualization filter and asked about the distributions for the corresponding visualization in the dashboard. After understanding the dataset and chart schema, participants were asked to accomplish the following tasks in the prescribed order below:
Interestingness Label: Participants were asked to talk aloud as they interpreted the visualizations in the dashboard and labelled each visualization as either interesting, not interesting, or leave it as unselected. This subjective task measures how *interesting* the selected visualizations were to participants (RQ1).
Attribute Ranking: Participants were given a sheet of paper with all the attributes listed and asked to rank the attributes in order of importance in contributing to a particular outcome (e.g., factors leading to an arrest or autism diagnosis). Participants were allowed to assign equal ranks to more than one attribute or skip attributes that they were unable to infer importance for. Attribute ranking tasks are common in feature selection and other data science tasks. The goal of this task was to measure how well participants *summarized* key insights in the dataset based on whether they understood the relative importance of each attribute in contributing towards an outcome (RQ2).
Informative Prediction: Participants were given a separate worksheet and asked to sketch an estimate for a visualization that is not present in the dashboard. For every condition, the visualization to be estimated contained 2 filter combinations, with exactly one parent present in the given dashboard. After making the prediction, participants were shown the actual data distribution and asked to rate on a Likert scale of 10 how surprising the result was (where 1 is not surprising and 10 is very surprising). The prediction task measured how accurate participants are at predicting an unseen visualization, estimating how well they understood key *informative* insights that influences other distributions from the dataset (RQ3).

We repeated the same study procedure described above for the Autism dataset. At the end of the study, we asked two open-ended questions regarding the insights that participants have learned and what they like or dislike about each dashboard. On average, the study lasted around 48 minutes.

[2] Since the clusters cover all visualizations in the dataset and the overall visualization has the minimum number of filter across all visualization, the overall visualization is guaranteed to be picked as one of the displayed visualizations.

5 STUDY RESULTS

RQ1: How *informative* are the visualizations in the dashboard at providing users with an accurate understanding of unseen child visualizations?

As discussed in Section 2, an accurate and informative understanding of the "root cause" that led to the child distribution can help prevent users from falling prey to the drill-down fallacy. To this end, the prediction task serves as a proxy for evaluating how accurately analysts understood the distributions present in various drill-down paths. In particular, we can get a sense of how *informative* the dashboards were by examining how accurately participants could use visualizations present in the dashboard to predict an unseen visualization.

The accuracy of participants' predictions was measured by the Euclidean distance between the predicted distributions and ground truth data distributions. As shown in Figure 4 (left), predictions made using VISPILOT (highlighted in red) were closer to the actual distribution than compared to the baselines, as indicated by the smaller Euclidean distances. Figure 4 (right) also shows that VISPILOT participants were able to more accurately reason about the expected properties of unseen data subsets, since they rated the resulting visualizations to be less surprising. CLUSTER may have performed better in the Police dataset than it did in the Autism dataset due to the same reason as in the attribute ranking task, where more univariate visualizations happened to be selected.

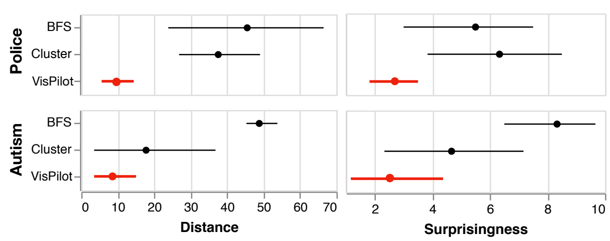

Figure 4: Left: Euclidean distance between predicted and ground truth. In general, predictions made using VISPILOT are closer to ground truth. Right: Surprisingness rating reported by users after seeing the actual visualizations on a Likert scale of 10. VISPILOT participants had a more accurate mental model of the unseen visualization and therefore reported less surprise than compared to the baseline.

We also compute the variance of participants' predictions across the same condition. In this case, low variance implies that any participant who reads the dashboard is able to provide consistent predictions, whereas high variance implies that the dashboard did not convey a clear data-driven story that could guide participants' predictions. So instead, participants had to rely on different priors or guessing to form their prediction. These trends can be observed in both Figure 4 and in more detail in Figure 5, where the prediction variance amongst participants who used VISPILOT is generally lower than the variance from the baselines.

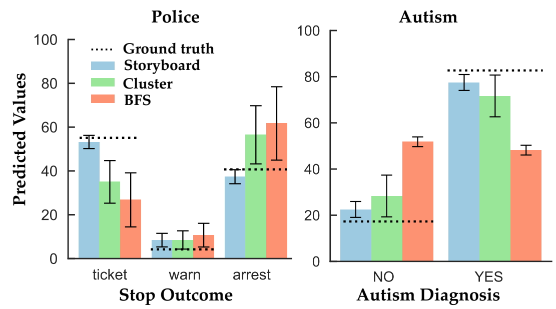

Figure 5: Mean and variance of predicted values. Predictions based on VISPILOT exhibit lower variance (error bars) and closer proximity to the ground truth values (dotted).

RQ2: How well does the dashboard *summarize* the relative importance of different attributes for a given dataset?

For our study, we use the common analytical task of judging the relative importance of an attribute as an indicator of how well users are able to summarize key insights in a dataset based on the dashboard visualizations. To determine the attribute importance for a dataset, we computed the Cramer's V statistics between attributes to be ranked and the attributes of interest. Cramer's V is a common measure for determining the strength of association between categorical attributes [27]. We deem an attribute as important if it has one of the top-three[3] Cramer's V scores amongst all attributes of the dataset. For the list of rankings provided by each participant, we first remove attributes where participants chose not to rank. Then we obtain the ground truth ranking based on the Cramer's V statistics for the ranked attributes. We compute the F-scores and average precision (AP) at k across a list of different k values (from 1 up to the number of ranked attributes, with k values corresponding to attributes ranked as ties deduplicated). Table 1 summarizes the average across users in each condition, after picking the best performing k value for each user based on F-score and AP respectively. Both measures effectively capture how accurately participants were able to retrieve the three most important attributes for each dataset.

	Police		Autism	
Metric	F	AP	F	AP
VISPILOT	0.750	0.867	0.723	0.600
CLUSTER	0.739	0.691	0.725	0.665
BFS	0.739	0.592	0.222	0.200

Table 1: Best AP and F-scores for the attribute ranking task.

[3]This relevancy cutoff is visually-determined via the elbow method to indicate which rank the Cramer's V score drops off significantly.

Even though BFS has inherent advantage for this task since BFS dashboards consist of all univariate distributions, which provides more high-level information regarding each attribute, both VisPilot and Cluster (which contained more 'local' information) performed better than BFS for both datasets. The problem with BFS is that given a limited budget of visualizations that could be displayed, not all univariate distributions can be exhaustively shown. For the Police dataset, it happened to select several of the important attributes (related to contraband and search) to display in the first 10 visualizations. However, with a budget of k=10, only visualizations regarding binary diagnostic questions 1-4 fit in the dashboard for the Autism dataset. So the poor ranking behavior comes from the fact that the BFS generated dashboard failed to display the three most important attributes (questions 5, 6 and 9) given the limited budget. This demonstrates BFS's lack of providing a guarantee especially when exhaustive exploration has a limit (e.g., time or attention of analyst).

We see that VisPilot performs better than Cluster for the Police dataset and closely follows Cluster for the Autism dataset. It is not entirely surprising that Cluster did well, since it is a well-established method for summarizing high-dimensional data [15]. For the Autism dataset, Cluster happened to pick the majority of visualizations (8/10) as univariate distributions that exhibited high-skew and diversity, leading to more informed inference on attribute importance. Since clustering seeks visualizations that exhibit diversity in the shape of the data distributions, it could potentially result in visualizations with many filter combinations. For the police dataset, 6 out of 10 visualizations had more than 2 filters, making it difficult for analysts to interpret the visualization without appropriate context to compare against.

Both BFS and Cluster do not provide consistent guarantees for highlighting important visualizations across different datasets. In general, our results indicate that participants gain a better understanding of attribute importance using VisPilot, with only a few targeted visualizations that tells the "whole story". Note that this is without VisPilot being explicitly optimized for the ranking task.

RQ3: How *interesting* are the visualizations in the dashboard perceived subjectively by the users?

Using the click-stream data logged from the user study, we recorded whether a participant labelled a visualization in the dashboard as interesting, not interesting, or left the visualization unselected. Table 2 summarizes counts of visualizations marked as interesting or not interesting aggregated across conditions. We also normalize the interestingness count by the total number of selected visualizations to account for variations in how some participants select more visualizations than others. The results indicate that participants who used VisPilot saw more visualizations that they found interesting compared to the BFS and Cluster conditions. While the labelling task is inherently subjective with varied reasons why a participant may have marked a visualization as interesting, this result is some indication that VisPilot's *saliency* objective was able to select visualizations that were interesting to the users.

Condition	VisPilot	BFS	Cluster
Interesting	66	61	51
Not Interesting	10	20	22
Interesting (Normalized)	0.87	0.75	0.7

Table 2: Total counts of visualizations marked as interesting or not interesting across the different conditions. VisPilot leads to more visualizations marked as interesting and fewer visualizations marked as uninteresting.

6 DISCUSSION OF STUDY RESULTS

To further understand how participants made use of the recommended visualizations in their drill-down analysis, we analyzed the user study transcriptions through an open coding process [28] by two of the authors. For each task in our study, we assigned a binary-valued code to indicate whether or not a participant engaged in a particular action or thought process. Table 4 highlights results from thematic coding discussed in this section. We will use the notation [Participant.DatasetAlgorithm] to refer to a participant engaging with a dashboard created by an algorithm={1,2,3}={VisPilot, Cluster, BFS} on a dataset ={A,B}={Police, Autism}.

The Choice of Contextual References

As discussed earlier, analysts often make use of related visualizations to form their expectations for unseen visualization. We refer to visualizations used for such purposes as *contextual references*. The choices of a proper contextual references (ideally as the informative parent) is essential for ensuring the *safety* of insights derived through drill-downs. To understand how "safe" the dashboards generated from each condition were, we examined the visualizations that participants utilized and compared against to form their expectations for unseen visualizations. In particular, we thematically encoded participants' use of contextual references based on their verbal explanations for justifying their prediction task responses. As shown in Table 3, in general, we find that participants make more comparisons in total using VisPilot than compared to Cluster and BFS.

Participants can (and often do) make comparisons against more than one type of contextual references to obtain their prediction. We uncovered four main classes of contextual references, described below using the example visualization V_i=`gender=F, race=White, age=21-30` (in the order of most to least similar to V_i) and illustrated graphically in Figure 6: Doris: modified this section with purple colored filter attributes-values indicating the ones corresponding to V_i and orange as ones that are different.

(1) **Parent** : Comparison against a visualization with one filter criterion removed (e.g., `gender=F, race=White`)
(2) **Siblings** : Comparison against a visualization that shares the same parent. In other words, the filter types are the same, but with one criterion changed to inherit a different value. (e.g., `gender=F, race=White, age=60+`)
(3) **Relatives** : Comparison against a visualization that shares some common ancestor (excluding overall), but not necessarily the same parent. In other words, these visualizations share at least one common filter type, but with more than one criterion that inherits a different value. (e.g., `gender=F, race=White, age=60+, search conducted=T`)
(4) **Overall** : Comparison against the distribution that describes the overall population (no filters applied).

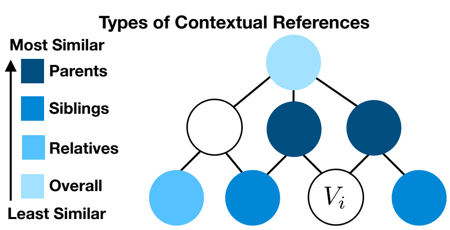

Figure 6: Different types of contextual references for a given visualization of interest V_i. The degree of similarity with respect to V_i is denoted by darkest to lightest color hue.

Studying participants' use of contextual references reveals inherent challenges that arise from using the dashboards generated by BFS and CLUSTER. For CLUSTER, participants mainly compared against relatives and overall visualizations. Since CLUSTER optimizes the diversity of shape distributions amongst the visualizations, the selected visualizations had up to 4 filters and were disconnected from each other. For this reason, in many cases participants could only rely on relatives and the overall visualizations as contextual references. For example, P4.A2 pointed at a 4-filter visualization with extreme values (100% for warning; 0% for arrest and ticket) and indicated how "*a lot of [the visualizations] are far too specific. This is not very helpful. You can't really hypothesize that all people are [sic] going to be warned, because it is such a specific category, it might just be one person*". He further explained how he "*would not want to see the intersections [(visualizations with many filters)] at first and would want to see all the bases [(univariate summaries)] then dig in from there.*" The lack of informative contextual references in the CLUSTER dashboard is also reflected in how analysts exhibited high variance and deviation in their prediction responses. Note that the prediction task was chosen so that exactly one parent must be present in the dashboard, so these results do not point to the absence of parent visualizations in these dashboard, but rather indicate how participants made use of the information presented in the dashboards to form their prediction.

Furthermore, improper comparisons against contextual references often make it difficult for analysts to interpret displayed visualizations. In particular, when visualizations composed of multiple filter conditions were shown in CLUSTER dashboards, 25% of participants had trouble making sense of the meaning of a filter for at least one of the datasets (e.g., understanding that `gender=F AND age=60+` corresponds to female drivers with ages larger than 60 years old) at some point during the study. In contrast, as shown in Table 4, this confusion only happened once for BFS and none for VISPILOT. This is due to the fact that CLUSTER dashboards seemed random to the users, making it challenging to find 'close' contextual references to compare against and form an accurate mental model, whereas the linear ordering of BFS and hierarchical ordering of VISPILOT were natural and interpretable for participants.

Algorithm	Parent	Sibling	Relative	Overall	Total
VISPILOT	12	8	0	11	31
CLUSTER	4	0	7	8	19
BFS	0	5	1	8	14

Table 3: Out of 12 participants, the number of participants who made use of each contextual reference across the two datasets. Participant behavior shows a similar trend in individual datasets. VISPILOT participants made more comparisons in general and against parents compared to the baselines.

For BFS, most comparisons were based on the overall and siblings. Due to the sequential level-wise picking approach, the overall visualization for BFS dashboards corresponded to the immediate parent, so they are not explicitly recorded as a parent. While the overall and sibling comparisons can be informative, the incomplete comparisons, due to the limited number of first-level visualizations displayed, can result in flawed reasoning, likewise observed in the aforementioned Autism prediction task. In contrast, for VISPILOT, almost all users compared against the overall and parents, while some also exploited sibling comparisons to make weaker guesses for less-frequently observed attributes (e.g., using a 2-filter sibling visualization involving `driver_age` to infer another 2-filter visualization involving `driver_age` with a different parent.)

	VisPilot	Cluster	BFS
Difficulty with Interpreting Visualizations	0	3	1
Misjudged Significance of Potential Small-Size Population	0	4	1
Interpretable "Human-like" Dashboard	5	1	0
Number of Insights (Police)	11	8	9
Number of Insights (Autism)	16	6	11

Table 4: Summary of qualitative insights from thematic coding. We record the total number of insights based on overall dataset findings that were independently discovered by more than two different participants. For each participant, we coded the absence or presence of 7 such insights for the Police dataset and 6 insights for the Autism dataset.

Interpretability of Hierarchical Layouts

In the post-study interviews, participants cited hierarchical layout as a key reason for why they preferred VisPilot recommendations. Even though participants were never explicitly told what the edge connections between the visualizations meant during the study, they were able to interpret the meaning of the dashboards effortlessly through VisPilot's hierarchical layout. For example, P1.A1 stated that "*the hierarchical nature [is] a very natural flow...so when you are comparing, you don't have to be making those comparisons in your head, visually that is very pleasing and easy to follow.*" Likewise, P9 described how VisPilot's hierarchical layout for the Autism dataset was a lot easier to follow than the Police dataset shown in the table layout for Cluster:

> *If I had to look at this dataset in the format of the other one, this would be much more difficult. It was pretty hard for me to tell in the other one how to organize the tree, if there was even a tree to be organized. I like this layout much better, I think this layout allows me to approach it in a more meaningful way. I can decide, what do I think matters more: the overall trend? or the super detailed trends? and I know where to look to start, in the other one, every time I go back to it, I would say, where's the top level, where's the second level? I mentally did this. Like when you asked me that first question, it took much longer to find it, because I literally have to put every chart in a space in my head and that took a lot longer than knowing how to look at it.*

At the end of the study, some participants who saw table layouts sketched and explained how they would like the layout of the visualizations to be done. Participants expressed that they wanted "groupings" or layouts that arranged visualizations with the same attribute together. Other participants advocated for isolating the overall visualization outside of the dashboard table for facilitating easier comparisons. Both of these suggestions provide further motivation for our hierarchical organization of visualizations. Our findings echo prior work on visualization sequences and storytelling [8, 18, 23, 31], which found that analysts prefer visualization sequences structured hierarchically based on shared data properties such as levels of aggregation ordered by increasing levels of summarization.

Since we did not inform participants about how the dashboards were generated, it was surprising to see that some participants presumed that the dashboards were hand-picked by a human analyst and hypothesized what this fictitious analyst's intentions were (e.g., "*It seems like the researcher who created this dashboard was specifically looking at people of Asian descent and people who are 60 or older.*" [P7.A1]). Table 4 shows how 5 out of 12 participants referred to the VisPilot dashboards as if they were generated by a human, whereas there was only 1 participant for Cluster and none for BFS made such remarks[4]. At the end of the study, many were surprised to learn that the VisPilot dashboard was actually picked out by an algorithm, indicating that VisPilot could automatically generate convincing dashboard stories similar to a dashboard that was authored with human intention. The interpretability of VisPilot dashboards may have contributed to the increased number of insights discovered in both datasets compared to the two baselines, as summarized in Table 4.

Limitations of VisPilot

As described earlier, since the details of how the dashboards were obtained were not explained to the users during the study, some users expressed that they were initially confused by VisPilot since not all variables were present in the dashboard. Others also found it confusing that the addition of filters did not always correspond to the same variables. For example, P2.A1 criticized how the dashboard was intentionally selected to be biased:

> *I feel like this one, not all the data is here, so we are already telling a story, you are trying to steer the viewer to look at certain things. And the focus seems to be on where the arrest rate is high. You probably could have found other things that led to ticket being high, but you didn't pull those out. You are trying to see if there are other factors that leads to more arrests.*

This sentiment is related to participants' desire to perform their own ad-hoc querying alongside the dashboard to inspect other related visualizations for verifying their hypothesis. For

[4] We encoded this phenomenon by looking at instances where a participant either explicitly referred to a person who picked out the dashboard or implicitly described their intentions through personal pronouns.

example, P7.A1 wanted to inspect all other first-level visualizations for driver's race to assess its influence. P7.A1 expressed that while he had learned many insights from the dashboard, "*the only thing I don't like is I cannot control the types of filter, which is fixed.*" Since the design objective for VISPILOT is to support and evaluate its capability to provide safe summarizations, VISPILOT is limited in its interactivity and the extent of free-formed data exploration it supports. This result also point to how VISPILOT could serve as a helpful assistance alongside other conventional visualization tools, such as Tableau. Outside the context of the user study, it is essential to explain how VISPILOT are picking the visualizations in a easy and interpretable manner to establish a sense of summarization guarantee for the users and help them make better inferences with the dashboard.

Since the goal of our study is to evaluate whether VISPILOT can assist users in drill-down exploration, our preliminary study is limited to comparison against baselines stemming from conventional approaches for multidimensional data exploration . While we understand the limitation of the VISPILOT study condition confounds both the hierarchical layout with the algorithmic choice of visualizations, our intention for the baseline was to simulate how analysts typically generate a large number of visualizations individually arranged in a table grid layout, rather than using a hierarchical layout. Further evaluation comparing how the different hierarchically-displayed visualization selection algorithms assist users in drill-down exploration is a direction of future work.

7 RELATED WORK

Our work draws from, and improves upon, past research in multidimensional data exploration and fallacies in visual analytics.

Guided Exploration of Multidimensional Data

Given a dataset, tools such as Tableau support automatic generation of visualizations based on perceptual graphical presentation rules [26, 37]. A more recent body of work automatically selects visualizations based on statistical measures, such as scagnostics and deviation. Given a scatterplot, Anand et al. [5] applies randomized permutation tests to select partitioning variables that reveals interesting small multiples using scagnostics [33, 36]. Given a bar chart, Vartak et al. [34] finds other interesting bar charts that deviate from the input chart using a deviation-based measure. Our work extends the deviation-based measure to formulate user expectation. However, unlike existing works, we concentrate on informativeness, which enables our system to avoid drill-down fallacies.

Preventing Biases and Statistical Fallacies

Visualizations are powerful representations for discovering trends and patterns in a dataset; however, cognitive biases and statistical fallacies could mislead analysts' interpretation of those patterns [3, 6, 12, 35, 39]. Wall et al. [35] presents six metrics to systematically detect and quantify bias from user interactions in visual analytics. These metrics are based on coverage and distribution, which focus on the assessment of the process by which users sample the data space. Alipourfard et al. [3] presents a statistical method to automatically identify Simpson's paradox by comparing statistical trends in the aggregate data to those in the disaggregated subgroups. Zgraggen et al. [39] presents a method to detect the presence of the multiple comparisons problem in visual analysis. This paper focusses on a novel type of fallacy during drill-down exploration that has not been addressed by past work.

8 CONCLUSION

Common analytics tasks, such as causal inference, feature selection, and outlier detection require studying data distributions at different levels of data granularity [5, 16, 18, 38]. However, without knowing *what* subset of data contains an insightful distribution, manually exploring distributions from all possible data subsets can be tedious and inefficient. Moreover, when examining data subsets by adding one filter at a time, analysts can fall prey to the drill-down fallacy, where they mistakenly attribute the interestingness of a visualization to a "local difference", while overlooking a more general explanation for the root cause of the behavior. To address these issues, we presented VISPILOT, an interactive visualization recommendation system that automatically selects a small set of informative and interesting visualizations to summarize key distributions within a dataset. Our user study demonstrates that VISPILOT can guide analysts towards more informed decisions for retrieving interesting visualizations, judging the relative importance of attributes, and predicting unseen visualizations, than compared to two other summarization baselines. Study participants also find dashboard generated by VISPILOT to be more interpretable and "human-like", leading to more discovered insights. Our work is one of the first automated systems that guides analysts across the space of data subsets by summarizing key insights with safety guarantees—a step towards our grander vision of developing intelligent tools for accelerating and assisting with visual data discovery.

REFERENCES

[1] 2016. Elections 2016 Exit Polls. (2016). http://edition.cnn.com/election/2016/results/exit-polls

[2] 2017. Titanic: Machine Learning from Disaster. Kaggle. (2017). http://www.kaggle.com/c/titanic

[3] Nazanin Alipourfard, Peter G. Fennell, and Kristina Lerman. 2018. Can You Trust the Trend?: Discovering Simpson's Paradoxes in Social Data. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining (WSDM '18)*. ACM, New York, NY, USA, 19–27. doi.acm.org/10.1145/3159652.3159684

[4] Ernst Althaus, Markus Blumenstock, Alexej Disterhoft, Andreas Hildebrandt, and Markus Krupp. 2009. Algorithms for the Maximum Weight Connected k-Induced Subgraph Problem. 5573 (2009), 313–321. dx.doi.org/10.1007/978-3-642-02026-1

[5] Anushka Anand and Justin Talbot. 2015. Automatic Selection of Partitioning Variables for Small Multiple Displays. 2626, c (2015). dx.doi.org/10.1109/TVCG.2015.2467323

[6] Zan Armstrong and Martin Wattenberg. 2014. Visualizing statistical mix effects and simpson's paradox. *IEEE transactions on visualization and computer graphics* 20, 12 (2014), 2132–2141.

[7] Carsten Binnig, Lorenzo De Stefani, Tim Kraska, Eli Upfal, Emanuel Zgraggen, and Zheguang Zhao. 2017. Toward Sustainable Insights, or Why Polygamy is Bad for You. In *CIDR 2017, 8th Biennial Conference on Innovative Data Systems Research, Chaminade, CA, USA, January 8-11, 2017*. cidrdb.org/cidr2017/papers/p56-binnig-cidr17.pdf

[8] Jeremy Boy, Francoise Detienne, and Jean-Daniel Fekete. 2015. Storytelling in Information Visualizations: Does It Engage Users to Explore Data?. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 1449–1458. doi.acm.org/10.1145/2702123.2702452

[9] Michael Correll and Jeffrey Heer. 2016. Surprise! Bayesian Weighting for De-Biasing Thematic Maps. *IEEE Transactions on Visualization and Computer Graphics* 2626, c (2016), 1–1. dx.doi.org/10.1109/TVCG.2016.2598618

[10] Bolin Ding, Silu Huang, Surajit Chaudhuri, Kaushik Chakrabarti, and Chi Wang. 2016. Sample + Seek: Approximating Aggregates with Distribution Precision Guarantee. In *Proceedings of the 2016 International Conference on Management of Data (SIGMOD '16)*. ACM, New York, NY, USA, 679–694. DOI:http://dx.doi.org/10.1145/2882903.2915249

[11] Fadi Fayez Thabtah. 2017. Autism Screening Adult Data Set. UCI Machine Learning Repository. (2017).

[12] David Gotz, Shun Sun, and Nan Cao. 2016. Adaptive Contextualization: Combating Bias During High-Dimensional Visualization and Data Selection. *Proceedings of the 21st International Conference on Intelligent User Interfaces - IUI '16* (2016), 85–95. DOI: http://dx.doi.org/10.1145/2856767.2856779

[13] 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. *Data Mining and Knowledge Discovery* 1, 1 (01 Mar 1997), 29–53. doi.org/10.1023/A:1009726021843

[14] Yue Guo, Carsten Binnig, Tim Kraska, and T U Darmstadt. 2017. What you see is not what you get ! Detecting Simpson ' s Paradoxes during Data Exploration. *HILDA 2017 - Proceedings of the Workshop on Human-In-the-Loop Data Analytics* (2017).

[15] Jiawei Han. 2005. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[16] Jeffrey Heer and Ben Shneiderman. 2012. Interactive Dynamics for Visual Analysis. *Queue* 10, 2 (2012), 30. dx.doi.org/10.1145/2133416.2146416

[17] Enamul Hoque, Vidya Setlur, Melanie Tory, and Isaac Dykeman. 2017. Applying Pragmatics Principles for Interaction with Visual Analytics. *IEEE Transactions on Visualization and Computer Graphics* c (2017). dx.doi.org/10.1109/TVCG.2017.2744684

[18] Jessica Hullman, Robert Kosara, and Heidi Lam. 2017. Finding a Clear Path: Structuring Strategies for Visualization Sequences. *Comput. Graph. Forum* 36, 3 (June 2017), 365–375. doi.org/10.1111/cgf.13194

[19] Laurent Itti and Pierre Baldi. 2009. Bayesian surprise attracts human attention. *Vision Research* 49, 10 (19 May 2009), 1295–1306. dx.doi.org/10.1016/j.visres.2008.09.007

[20] Manas Joglekar, Hector Garcia-molina, and Aditya Parameswaran. 2015. Smart Drill-Down : A New Data Exploration Operator. *Proceedings of the 41st International Conference on Very Large Data Bases* 8, 12 (2015), 1928–1931.

[21] Sean Kandel, Ravi Parikh, Andreas Paepcke, Joseph Hellerstein, and Jeffrey Heer. 2012. Profiler: Integrated Statistical Analysis and Visualization for Data Quality Assessment. In *Advanced Visual Interfaces*. http://vis.stanford.edu/papers/profiler

[22] Alicia Key, Bill Howe, Daniel Perry, and Cecilia Aragon. 2012. VizDeck. *Proceedings of the 2012 international conference on Management of Data - SIGMOD '12* (2012), 681. DOI:http://dx.doi.org/10.1145/2213836.2213931

[23] Younghoon Kim, Kanit Wongsuphasawat, Jessica Hullman, and Jeffrey Heer. 2017. GraphScape: A Model for Automated Reasoning about Visualization Similarity and Sequencing. *Proc. of ACM CHI 2017* (2017). dx.doi.org/10.1145/3025453.3025866

[24] Doris Jung-Lin Lee and Aditya Parameswaran. 2018. The Case for a Visual Discovery Assistant: A Holistic Solution for Accelerating Visual Data Exploration. *IEEE Bulletin of Technical Committee on Data Engineering* (2018).

[25] Stephen Macke, Yiming Zhang, Silu Huang, and Aditya Parameswaran. 2018. Adaptive Sampling for Rapidly Matching Histograms. *Proc. VLDB Endow.* 11, 10 (June 2018), 1262–1275. DOI:http://dx.doi.org/10.14778/3231751.3231753

[26] Jock D. Mackinlay, Pat Hanrahan, and Chris Stolte. 2007. Show Me: Automatic presentation for visual analysis. *IEEE Transactions on Visualization and Computer Graphics* 13, 6 (2007), 1137–1144. dx.doi.org/10.1109/TVCG.2007.70594

[27] Mary L. McHugh. 2013. The Chi-square test of independence. *Biochemia Medica* 23, 2 (15 Jun 2013), 143–149. ncbi.nlm.nih.gov/pmc/articles/PMC3900058/

[28] Michael J. Muller and Sarah Kuhn. 1993. Participatory Design. *Commun. ACM* 36, 6 (June 1993), 24–28. DOI:http://dx.doi.org/10.1145/153571.255960

[29] Aditya G. Parameswaran, Hector Garcia-Molina, and Jeffrey D. Ullman. 2010. Evaluating, combining and generalizing recommendations with prerequisites. *Proceedings of the 19th ACM international conference on Information and knowledge management - CIKM '10* (2010), 919. dx.doi.org/10.1145/1871437.1871555

[30] E. Pierson, C. Simoiu, J. Overgoor, S. Corbett-Davies, V. Ramachandran, C. Phillips, and S. Goel. 2017. A large-scale analysis of racial disparities in police stops across the United States. (2017). http://openpolicing.stanford.edu/data/

[31] Edward Segel and Jeffrey Heer. 2010. Narrative visualization: Telling stories with data. *IEEE Transactions on Visualization and Computer Graphics* 16, 6 (2010), 1139–1148. dx.doi.org/10.1109/TVCG.2010.179

[32] Tarique Siddiqui, Albert Kim, John Lee, Karrie Karahalios, and Aditya Parameswaran. 2016. Effortless data exploration with zenvisage: an expressive and interactive visual analytics system. *Proceedings of the*

VLDB Endowment 10, 4 (2016), 457–468. DOI:http://dx.doi.org/10.14778/3025111.3025126

[33] Tuan Nhon Dang and Leland Wilkinson. 2014. ScagExplorer: Exploring Scatterplots by Their Scagnostics. *2014 IEEE Pacific Visualization Symposium* (2014), 73–80. DOI:http://dx.doi.org/10.1109/PacificVis.2014.42

[34] Manasi Vartak, Sajjadur Rahman, Samuel Madden, Aditya Parameswaran, and Neoklis Polyzotis. 2015. SeeDB: efficient data-driven visualization recommendations to support visual analytics. *Proceedings of the VLDB Endowment* 8, 13 (2015), 2182–2193. dx.doi.org/10.14778/2831360.2831371

[35] Emily Wall, Leslie M Blaha, Lyndsey Franklin, and Alex Endert. 2017. Warning, Bias May Occur: A Proposed Approach to Detecting Cognitive Bias in Interactive Visual Analytics. *2017 IEEE Conference on Visual Analytics Science and Technology (VAST)* (2017).

[36] Leland Wilkinson, Anushka Anand, and Robert Grossman. 2005. Graph-Theoretic Scagnostics. *IEEE Symposium on Information Visualization (INFOVIS)* (2005).

[37] Kanit Wongsuphasawat, Dominik Moritz, Anushka Anand, Jock Mackinlay, Bill Howe, and Jeffrey Heer. 2016. Voyager: Exploratory Analysis via Faceted Browsing of Visualization Recommendations. *IEEE Transactions on Visualization and Computer Graphics* 22, 1 (2016), 649–658. dx.doi.org/10.1109/TVCG.2015.2467191

[38] Eugene Wu and Samuel Madden. 2013. Scorpion: Explaining Away Outliers in Aggregate Queries. *Proceedings of the VLDB Endowment* 6, 8 (2013), 553–564. dx.doi.org/10.14778/2536354.2536356

[39] Emanuel Zgraggen, Zheguang Zhao, Robert Zeleznik, and Tim Kraska. 2018. Investigating the Effect of the Multiple Comparisons Problem in Visual Analysis. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, Article 479, 12 pages. doi.acm.org/10.1145/3173574.3174053