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 multidimensional data analysis baselines.

CCS CONCEPTS

• **Human-centered computing** → **Visual analytics**; **Information visualization**; **User models**; **User studies**.

KEYWORDS

exploratory data analysis, visualization recommendation, drill-down data analysis

IUI '19, March 17–20, 2019, Marina del Rey, CA, USA

ACM ISBN 978-1-4503-6272-6/19/03…$15.00
https://doi.org/10.1145/3301275.3302307

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 the number of attributes) presents a major bottleneck during exploration.

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

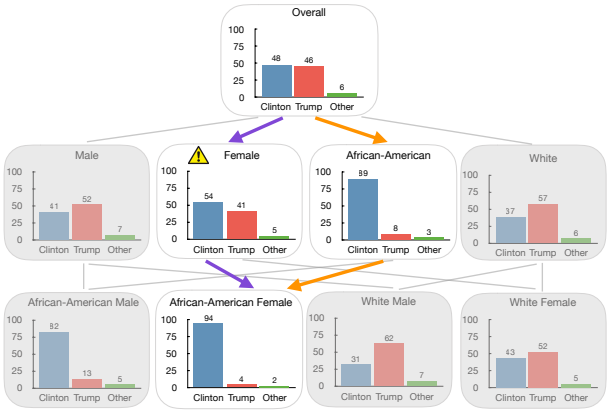

Figure 1: Example data subset lattice from the 2016 US election dataset illustrating the drill-down fallacy along the purple path as opposed to the informative orange path.

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.

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 falling 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, 13], multiple comparisons [43], and selection bias [11], 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 [23, 37] that assist or accelerate the process of visual data exploration [7, 20–22, 33, 37, 41], 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* [12] 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.

The "3S" Design Principles

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 multi-dimensional 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 analyst to form an 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 a novel idea—we leverage prior work for this [9, 24, 33, 37]. 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 [37]. 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 [8, 19, 37], 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 with respect to informative parent references. Second, we weigh the interestingness by the proportion of the population captured by the child visualization. (That is, when a deviation is manifested in a larger population, it is deemed to be more significant and therefore more interesting.) Thus, we define the utility of a visualization V_i, $U(V_i)$ as follows:

$$U(V_i) = \begin{cases} \frac{|V_i|}{|V_i^*|} \cdot D(V_i, V_i^*) & \text{if } V_i^* \text{ is present} \\ -\infty & \text{otherwise} \end{cases}$$

That is, the utility or interestingness of a visualization is the distance between the visualization and its informative parent, if present[1]. 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. Notice that the objective U has a minimax form [39], in that informativeness aims to minimize the distance between parent and child, while interestingness aims to maximize the resulting minimum distance. For convenience, we define $U(V_0)$, where V_0 is the overall visualization, to be 1, which is the maximum value that the expression $\frac{|V_i|}{|V_i^*|} \cdot D(V_i, V_i^*)$ can take, ensuring that the overall visualization is always valuable to include.

Succinctness. We cannot possibly display all of the visualizations in the lattice of data subsets: this lattice scales exponentially in the number of attributes. Instead, we aim for *succinctness*, where we only select a subset S of size $|S| = k$ from all the visualizations. We define the utility of S as follows:

$$U(S) = \sum_{V_i \in S} U(V_i)$$

In this subset, for every visualization except for the overall visualization, one of its informative parents must be present (otherwise $U = -\infty$). Thus, this subset ends up being a connected network (a sub-graph of the overall lattice) rooted at the overall visualization, ensuring that for each visualization, there is an informative parent available for context. We can now formally define our problem statement.

Problem. *Given a dataset and user-provided X, Y attributes, select a subset S of $|S| = k$ visualizations from the lattice of data subsets $\mathcal{L}$, such that $U(S)$ is maximized.*

Thanks to how we have defined U, S will include the overall visualization, corresponding to the entire dataset with no filter. And, for each visualization in S except the overall one, at least one of its informative parents will be present in S. This network of visualizations S can be displayed on a dashboard. Since the edges between non-informative parents to children are not pertinent to the solution, we can remove those edges from the lattice, leaving only the edges from the informative parents to the children. Then, we are left with an arbitrary graph, from which we need to select a rooted subgraph of size k, with greatest utility U.

For arbitrary distance metrics D, we can show that our problem is NP-Hard via a reduction from the AND-graph prerequisite problem, which has been shown to be NP-Hard in [29]. Similar to our problem, in the AND-graph prerequisite problem, all prerequisites need to be taken before a node can be selected. Both the scoring function in the prerequisite problem and the distance-based utility in our problem are non-negative and independent of selection order. Furthermore, both the AND-graph in the prerequisite problem and our lattice have only directed edges across neighboring levels (i.e. only edge connections from n→n+1). Based on our informativeness criteria, some directed edges may not be present in the graph. These pruned edges are equivalent to the AND-graph case where the score is zero. For a distance metric D, two nodes are discernable if the distance between them is zero. Hence, we prove by bijection that the two problems are equivalent and thereby proving that our problem is NP-hard. Next, we design an approximate algorithm to solve this problem.

3 VISPILOT: OUR SOLUTION

We present our system, VisPilot, by first providing a high-level overview of the underlying algorithm, and then describing the user interaction mechanisms.

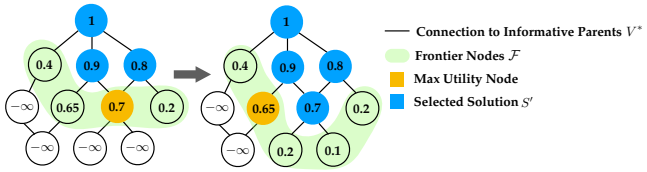

Figure 2: Example illustrating how the frontier greedy algorithm incrementally builds up the solution by selecting the node or visualization that leads to the highest gain in utility from the frontier at every step. Starting from a pruned lattice comprising only connections to informative parents (left) and three nodes in the existing solution (blue), we select the node with the highest utility gain (yellow) amongst the frontier nodes (green). The contribution to the utility of a node/visualization is depicted as the number within the node. On the right, the newly added node results in an updated frontier and the node leading to the highest utility gain is selected among them.

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

Lattice Traversal Algorithm

For a given dataset and user-selected X and Y axes, we first enumerate all possible attribute-value combinations (i.e., filters) to construct the lattice upfront. Like we described in the previous section, we retain only the edges that correspond to informative parents. Then, we traverse this pruned lattice to select the connected subgraph S of k visualizations (or equivalently, nodes in the lattice) that maximizes the utility U. Our algorithm for traversing the lattice, titled *frontier-greedy*, is inspired by the notion of "externals" in Parameswaran et al. [29]. The algorithm incrementally grows a subgraph S' until k nodes are selected. Throughout, the algorithm maintains a set of *frontier* nodes $\mathcal{F}$—nodes that are connected to the existing subgraph solution S' but have not yet been added. The frontier nodes includes all of the children of the nodes in S'. Given that our pruned lattice only retains edges between children and their informative parents, all frontier nodes are guaranteed to have an informative parent in the the existing solution and can be added to S without violating informativeness. At each iteration, the algorithm adds the node from the frontier nodes that leads to the greatest increase in the utility of S': i.e., the node V_n such that $U(S' \cup \{V_n\})$ is the largest. Figure 2 displays how the algorithm maintains the list of frontier nodes (in green), and the current S' (in blue), adding the node that leads to the greatest increase in utility (in yellow). Algorithm 1 provides the pseudocode.

Algorithm 1 Frontier Greedy Algorithm

```
1:  procedure PickVisualizations(k, L)
2:      S' ← {V_0} /* adding the overall node */
3:      while |S'| < k do
4:          F ← getFrontier(S', L)
5:          bestUtility ← −∞
6:          for V_i ∈ F do
7:              if U(S' ∪ {V_i}) >bestUtility then
8:                  maxNode ← V_i
9:                  bestUtility ← U(S' ∪ {V_i})
10:         S' ← S'∪ {maxNode}
        return S'
```

User Interaction

Given the visualizations in S', we can render these visualizations 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. Figure 3 displays VisPilot in action on the Police stop dataset [31]. 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) that led to different outcomes (X), such as ticket, warning, or arrest. As shown in Figure 3a, the analyst may begin by generating a 7-visualization dashboard. They would learn that if a search is conducted (`search_conducted=t`), then the probability of being arrested increases from 6.2% to 42.1%. However, the probability goes down to 23.1% if the driver is Asian (`driver_race=Asian, search_conducted=t`). When examining these visualizations, the analyst can be confident that any deviations are both informative and interesting: that is, the informative parents are present for each child, making the takeaways more significant. Moreover, the analyst may learn that for drivers who had contraband found in the vehicle (`contraband_found=t`), the arrest rate for those who are 60 and over is surprisingly higher than usual, whereas for Asian drivers the arrest rate is lower.

After browsing through visualizations in the dashboard, the analyst may be interested in getting more information about a specific visualization. VisPilot allows analysts to 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. Say the analyst is now interested in learning more about the other factor that contributes to high arrest rates: a long stop with `duration=30+min`. In Figure 3b, they can click on the corresponding visualization to request additional visualizations. Upon seeing the updated dashboard in Figure 3c, they learn 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 selected visualization is set as the the overall visualization V_0 at the root node of the new lattice. This node expansion capability is motivated by the idea of *iterative view refinement* common in other visual analytics systems, which is essential for users to iterate on and explore different hypotheses [17, 41].

4 EVALUATION STUDY METHODS

In this section, we describe the methodology for a user study we conducted for evaluating the usefulness of VisPilot for various exploratory analysis tasks. We aim to evaluate whether VisPilot's "3S" design principles enables analysts to effortlessly identify insights in comparison with conventional approaches for multidimensional data exploration.

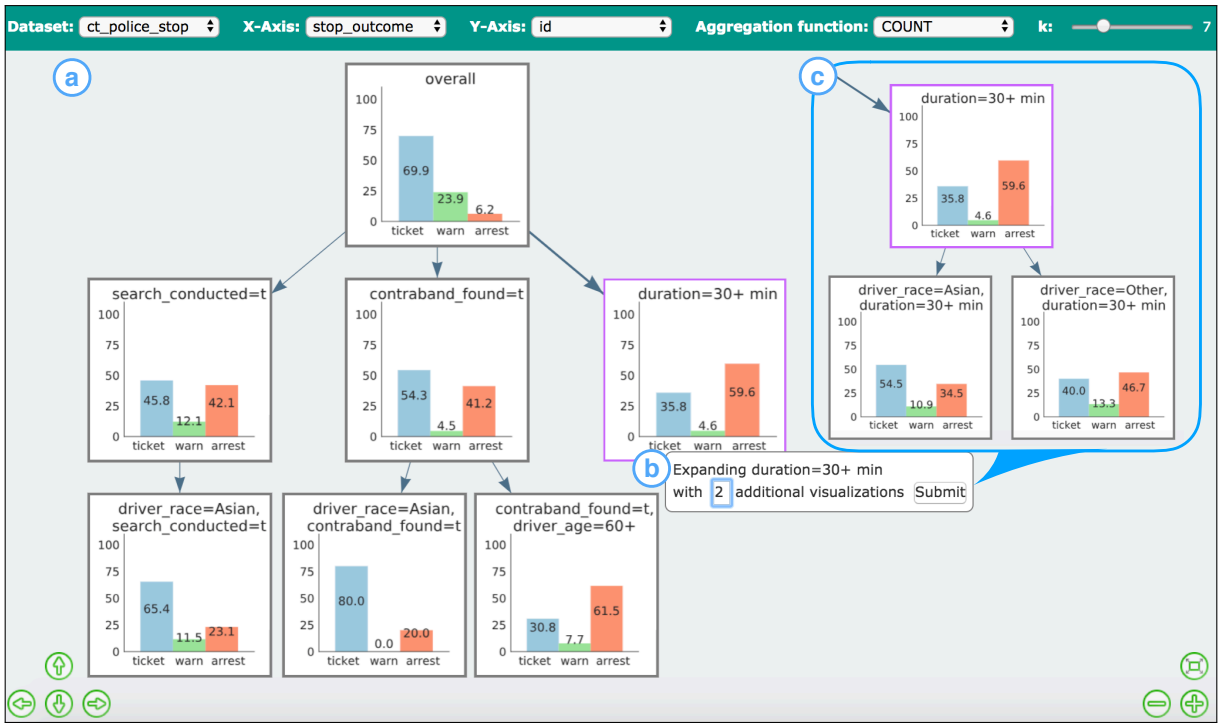

Figure 3: a) Overview of the VisPilot interface for the Police Stop dataset. Users can select x, y axes, and aggregation function via the dropdown menu, to define the visualization space of interest, as well as adjusting dashboard parameters, such as the number of visualizations to show in the dashboard (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.

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 – 14 years of data analysis experience (average: 5.61). This can include, but are not limited to, browsing and reading data, data cleaning and wrangling, data visualization and model building. The inclusion criteria is assessed based on a self-reporting basis in the pre-study survey. 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 via the following conditions.

VisPilot: The dashboards for this condition are generated by the aforementioned frontier greedy algorithm and displayed in a hierarchical layout as in Figure 3. 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 overall 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, and then visualizations with 2-filter combinations, and so on, until k visualizations have been added. This baseline is designed to simulate a dashboard generated by a meticulous analyst who exhaustively inspects all visualizations (i.e., filter combinations) from the top down. These visualizations are then displayed in a 5×2 table.

Cluster: In this condition, k-means clustering is first performed on the data distributions of all of the visualizations in the lattice. This results in k clusters that cover the rest of the visualizations. For each cluster, we select the visualization with the least number of filter conditions as the cluster representative for interpretability[2] and display them in a 5×2 table layout. This baseline is designed to showcase a diverse set of distributions within the dataset.

Dataset Descriptions. Each participant was assigned two different conditions on two different datasets (Police Stop and Autism, described below). 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.

[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.

Figure 4: Specific dashboards for each dataset and condition used in the study.

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.

The first dataset in the study was 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 was the Autism dataset [10], describing the results of autism spectrum disorder screening for 704 adults. The attributes in the dataset are binary responses to 10 diagnostic questions as 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 our study participants. We generated dashboard visualizations based on the percentage of adults that were diagnosed with autism.

Study Procedure

After the tutorial, for each dataset, participants were given some time to read through a worksheet containing the descriptions of the data attributes. Then, they were given an attention check question where they were provided a verbal description of the visualization filter (i.e., data subset) and asked about the corresponding visualization in the dashboard. After understanding the dataset and chart schema, participants were asked to accomplish various tasks. Since VisPilot was developed based on a joint utility objective, it is impossible to design tasks that evaluate each of the "3S" principles individually. Instead, our tasks were selected to measure the overall efficacy and usefulness of the dashboards in

helping a participant understand and become aware of different aspects of and insights within a dataset during drill-down analysis. These different aspects of dataset understanding can be roughly illustrated via Figure 2, from insights gained from *individual* displayed visualizations (blue selected nodes), to predicting behavior of *related* visualizations (green related nodes), to understanding *overall* attribute importance (entire lattice, a mix of green, blue, and unselected white nodes).

Labeling (Individual Assessment): Participants were asked to talk aloud as they interpreted the visualizations in the dashboard and label each one as interesting or not interesting, or leave it unselected. This subjective task measures how interesting *individual* selected visualizations were to participants.

Prediction (Related Assessment): 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 (1: not surprising and 10: very surprising). This task measured how well participants inferred the behavior of *related*, unobserved visualizations based on a limited set of selected dashboard visualizations.

Ranking (Overall Assessment): 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 many data science use-cases, such as feature selection and key driver analysis. Since all dashboards were equal in size, our goal was to check whether this size limitation came at the cost of *overall* dataset understanding. Thus, the goal of this task was to study participant's overall dataset understanding by measuring how well participants judged the relative importance of each attribute.

At the end of the study, we asked two open-ended questions regarding the insights gained by participants and what they liked or disliked about each dashboard. On average, the study lasted around 48 minutes.

5 STUDY RESULTS

We introduce the study findings for each task starting from the narrowest scope of *individual* visualizations to the widest scope of *overall* dataset understanding.

RQ1: How are *individual* selected visualizations in the dashboard perceived subjectively by the users?

Using click-stream data logged from the user study, we recorded whether a participant labeled each visualization in the dashboard as interesting, not interesting, or left the visualization unselected. Table 1 summarizes the 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 this task is inherently subjective, with many possible reasons why a participant may have marked a visualization as interesting, this result is indicative of the fact that the selected visualizations were deemed to be relevant by users. We will drill into the possible reasons why in the next section.

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

Table 1: 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.

RQ2: How well do dashboard visualizations provide users with an accurate understanding of *related* visualizations?

As discussed in Section 2, contextualizing visualizations correctly with informative references can help prevent users from falling prey to drill-down fallacies. To this end, the prediction task aims to assess whether users can employ visualizations in the dashboard to correctly predict unseen ones. Indeed, if the dashboard is constructed well, one would expect that visualizations that are not very surprising relative to their informative parents would be excluded from the dashboard (i.e., their deviation from their informative parents is not large).

The accuracy of participants' predictions is measured using the Euclidean distance between their predicted distributions and ground truth data distributions. As shown in Figure 5 (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 5 (right) also shows that VisPilot participants were able to more accurately reason about the expected properties of unseen data subsets (or visualizations), since they rated the resulting visualizations to be less surprising. Cluster may have performed better for the Police dataset than

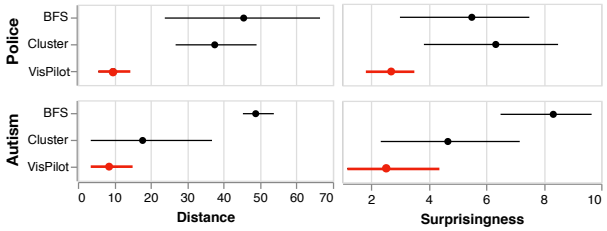

Figure 5: Left: Euclidean distance between predicted and ground truth. In general, predictions made using VisPilot are closer to ground truth. Right: Surprisingness rating reported by users after seeing the actual visualizations on a Likert scale of 10. VisPilot participants had a more accurate mental model of the unseen visualization and therefore reported less surprise than compared to the baselines.

it did for the Autism one, for the same reason as in the attribute ranking task, where more univariate visualizations happened to be selected.

We also compute the variance of participants' predictions across the same condition. In this case, low variance implies that there is consistency or agreement between the predictions of participants who consumed the same dashboard, 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 prior knowledge or guessing to inform their predictions. These trends can be observed in both Figure 5 and in more detail in Figure 6, where the prediction variance amongst participants who used VisPilot is generally lower than the variance for the baselines. Overall, VisPilot provides participants with a more accurate and consistent model of *related* visualizations.

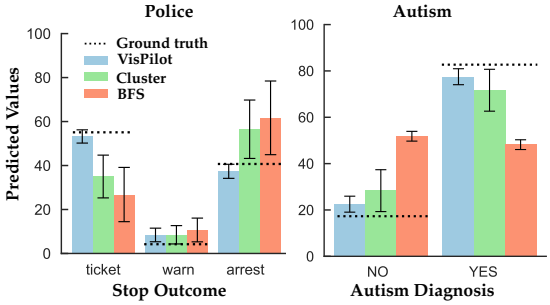

Figure 6: Mean and variance of predicted values. Predictions based on VisPilot exhibit lower variance (error bars) and closer proximity to the ground truth values (dotted).

RQ3: How well does the dashboard convey information regarding the *overall* dataset schema?

We use the common task of judging the relative importance of attributes as an indicator of the participants' overall understanding. To determine ground truth attribute importance, we computed the Cramer's V statistics between attributes to be ranked and the attributes of interest. Cramer's V is commonly used 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 that participants chose not to rank. We compute the F-scores and average precision (AP) at k relative to the ground truth for various values of k. Table 2 reports the average across participants in each condition, after picking the best performing k value for each participant based on F-score and AP respectively. Both measures capture how accurately participants were able to identify 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 2: Best AP and F-scores for the attribute ranking task.

For this task, we expected BFS to have an inherent advantage, since BFS dashboards consist of all univariate distributions, providing more high-level, "global" information regarding each attribute. However, both VisPilot and Cluster (which contained more "local" information) performed better than BFS. The problem with BFS is that given a limited dashboard budget of $k = 10$ visualizations that could be displayed, not all univariate distributions were shown.For the Police dataset, it happened to select several important attributes (related to contraband and search) to display in the first 10 visualizations. However, for Autism, only visualizations corresponding to binary diagnostic questions 1-4 fit in the dashboard. 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 consistency across different datasets, due to the fact that exhaustive exploration can only lead to limited understanding of the data.

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 [14]. For Autism, Cluster happened to pick the majority of visualizations (8/10) as univariate distributions that exhibited high-skew and diversity, leading to more informed inference of 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,

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

making it difficult to interpret the visualization without an appropriate context to compare against.

Overall, 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 *overall* dataset understanding regarding attribute importance using VisPilot, with only a few targeted visualizations that tell the "entire story". This is without VisPilot being explicitly optimized for the ranking task.

6 DISCUSSION OF STUDY RESULTS

To further understand how participants made use of the recommended visualizations during their analysis, we analyzed the user study transcripts 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 expectation or mental model for unseen visualizations. We refer to the visualizations used for such purposes as *contextual references*. The appropriate choice of a contextual reference (such as an informative parent) is necessary to ensure 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 compared against to inform unseen visualizations. In particular, we thematically encoded the participants' use of contextual references based on their verbal explanations for justifying their prediction task responses. As shown in Table 3, we find that participants make more comparisons in total using VisPilot than Cluster and BFS.

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.

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,age=21-30` (in the order of most to least similar to V_i):

(1) **Parent** : Comparison against a visualization with one filter removed (e.g., `gender=F`)
(2) **Sibling** : Comparison against a visualization that shares the same parent. In other words, the filtered attributes are the same, but one filter has a different value. (e.g., `gender=F,age=60+`)
(3) **Relative** : Comparison against a visualization that shares some common ancestor (excluding overall), but not necessarily the same parent. These visualizations share at least one common filter, but with more than one filter or filter value being different. (e.g., `gender=F,age=60+,race=White`)
(4) **Overall** : Comparison against the distribution that describes the overall population (no filters applied).

Studying the participants' use of contextual references reveals inherent challenges that arise from using the BFS and Cluster dashboards. For Cluster, participants mainly compared against relatives and overall visualizations. Since Cluster optimizes the diversity of distributions amongst the selected visualizations, these 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 visualization 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.

Furthermore, improper comparisons against contextual references often make it difficult to interpret displayed visualizations. In particular, when visualizations composed of multiple filter conditions were shown in Cluster dashboards, 25% of the 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"

	VisPilot	Cluster	BFS
Difficulty Interpreting Visualizations	0	3	1
Misjudged Significance of Population Size	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.

contextual references to compare against. In contrast, the linear ordering of BFS and hierarchical ordering of VisPilot were natural and interpretable for participants.

For BFS, most comparisons were based on the overall visualization and siblings. Due to the sequential level-wise picking approach, the overall visualization corresponded to the immediate parent of all of the dashboard visualizations generated by BFS (all of which are univariate distributions for $k = 10$), 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, as observed in the Autism prediction task. In contrast, for VisPilot, almost all users compared against the overall one 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.)

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 were assigned dashboard conditions with 5×2 table layouts (i.e., BFS and Cluster conditions) sketched and explained how they would like the layout of the visualizations to be done. These 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.

Since we did not inform participants about how the dashboards were generated, it was surprising to see that some participants presumed that certain dashboards were handpicked 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 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 dashboards similar to ones that were 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

[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.

by VisPilot as 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 felt that the dashboard was intentionally 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 lead 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 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 our current goal was to simply provide an informative dashboard and evaluate its utility, the present version of VisPilot is limited in its interactivity and the extent of free-form data exploration it supports. This result also points to how VisPilot could serve as a helpful assistant alongside other conventional visualization tools, such as Tableau. Outside the context of the user study, it is essential to explain how VisPilot selects the visualizations in an easy and interpretable manner to establish a sense of the summarization objectives 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 comparisons against baselines stemming from conventional approaches for multidimensional data exploration. While we understand how the VisPilot study condition may confound the hierarchical layout with the algorithmic choice of visualizations, our intention for the baseline was to simulate how analysts generate a large number of visualizations individually, typically arranged in a table grid layout, rather than using a hierarchical layout. Further evaluation comparing how different hierarchically-displayed visualization selection algorithms assist users in drill-down exploration is a direction of future work.

7 OTHER RELATED WORK

Our work draws from past research in multidimensional data exploration and fallacies in visual analytics and is distinguished from existing work on decision tree visualizations.

Guided Exploration of Multidimensional Data. Given a dataset, tools such as Tableau support automatic generation of visualizations based on graphical presentation rules [25, 41]. A more recent body of work automatically selects visualizations based on statistical measures, such as scagnostics and deviation. For discovering interesting conditional structures in scatterplots, Anand et al. [4] apply randomized permutation tests to select partitioning variables that reveal interesting small multiples using scagnostics [35, 40]. For recommending visualizations for assessing data quality, Kandel et al. [21] uses mutual-information as a distance metric for recommending views that highlight anomalies, as well as related views that explains the value distribution of the anomalous views. Vartak et al. [36, 37] uses deviation to recommend visualization attributes that highlight differences in two populations, while Siddiqui et al. [33] and Macke et al. [24] employ deviation to find similar visualizations. Qetch [26] and ShapeSearch [34] craft more sophisticated deviation measures to identify visualizations of interest. Our work extends these deviation-based measures to formulate user expectation. However, unlike existing work, we concentrate on informativeness rather than the exhaustive enumeration of the entire space, 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, 11, 38, 43]. Wall et al. [38] present 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 paradoxes by comparing statistical trends in the aggregate data to those in the disaggregated subgroups. Zgraggen et al. [43] present a method to detect the presence of the multiple comparisons problem in visual analysis. This paper, on the other hand, focuses on a novel type of fallacy that occurs during drill-down exploration that has not been addressed by past work.

Decision Tree Visualization

The popularity of decision trees in a variety of classification tasks have led to the development of visualizations that make these models more interpretable [5, 16, 30]. These visualizations often contain a visual representation of the rules as paths connecting the decision nodes, illustrating the proportion of sample along different paths, as well as statistics regarding the prediction accuracy at every node. Though our dashboards visually look similar to decision trees, the underlying objectives are different for the two methods. During tree construction, a decision tree algorithm aims to improve the classification accuracy of a target variable, typically by minimizing the entropy of distribution from parent node to

child node [32]. In contrast, our method aims to deliver informative insights, by maximizing the informative deviation between parent and child nodes. Consequently, the generated outcomes are different for the two methods—a decision tree well explains the general rules (e.g., if stop duration is more than 30 minutes, the driver has 60% probability of being arrested), whereas our method well explains the exceptions (e.g., if a stop duration is more than 30 minutes and the driver's race is Asian, the probability of arrest goes down to 35%). Note that the general rule is useful for predicting the stop outcome for an unlabeled test datapoint (classification), whereas the exception is useful for realizing when the general rule no longer holds (insight). The latter insight may not be discovered by a decision tree as it does not directly improve classification accuracy. Another key difference between the two methods is *coverage*—a decision tree covers the entire dataset (consistent with its classification goal), whereas our method highlights only the interesting regions of a dataset (consistent with its insight goal).

8 CONCLUSION

Common analytics tasks, such as causal inference, feature selection, and outlier detection require studying data distributions at different levels of data granularity [4, 15, 18, 42]. 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 convey key distributions within a dataset. Our user study demonstrates that VisPilot can guide participants toward more informed decisions for retrieving interesting visualizations, judging the relative importance of attributes, and predicting unseen visualizations than compared to two other 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.

Acknowledgments. We thank the anonymous reviewers for their valuable feedback. We acknowledge support from grants IIS-1513407, IIS-1633755, IIS-1652750, and IIS-1733878 awarded by the National Science Foundation, and funds from Microsoft, 3M, Adobe, Toyota Research Institute, Google, and the Siebel Energy Institute. The content is solely the responsibility of the authors and does not necessarily represent the official views of the funding agencies and organizations.

REFERENCES

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

[2] 2017. Titanic: Machine Learning from Disaster. Kaggle. **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] 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**

[5] Mihael Ankerst, Christian Elsen, Martin Ester, and Hans-Peter Kriegel. 1999. Visual Classification: An Interactive Approach to Decision Tree Construction. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '99)*. ACM, New York, NY, USA, 392–396. **doi.acm.org/10.1145/312129.312298**

[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] 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**

[9] 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. **https://doi.org/10.1145/2882903.2915249**

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

[11] 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. **https://doi.org/10.1145/2856767.2856779**

[12] 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**

[13] 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).

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

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

[16] Jeremy Hermann and Mike Del Balso. 2017. Meet Michelangelo: Uber's Machine Learning Platform. http://eng.uber.com/michelangelo/

[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. https://doi.org/10.1145/2213836.2213931

[23] 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).

[24] 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. https://doi.org/10.14778/3231751.3231753

[25] 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

[26] Miro Mannino and Azza Abouzeid. 2018. Qetch: Time Series Querying with Expressive Sketches. In *SIGMOD Conference.*

[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. https://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] Terence Parr and Prince Grover. 2018. How to visualize decision trees. explained.ai/decision-tree-viz/

[31] 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. http://openpolicing.stanford.edu/data/

[32] J. R. Quinlan. 1986. Induction of Decision Trees. *Mach. Learn.* 1, 1 (March 1986), 81–106. dx.doi.org/10.1023/A:1022643204877

[33] 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. https://doi.org/10.14778/3025111.3025126

[34] Tarique Siddiqui, Zesheng Wang, Paul Luh, Karrie Karahalios, and Aditya G. Parameswaran. 2018. ShapeSearch: A Flexible and Efficient System for Shape-based Exploration of Trendlines. *CoRR* abs/1811.07977 (2018). arXiv:1811.07977 http://arxiv.org/abs/1811.07977

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

[36] Manasi Vartak, Samuel Madden, Aditya G. Parameswaran, and Neoklis Polyzotis. 2014. SEEDB: Automatically Generating Query Visualizations. *PVLDB* 7, 13 (2014), 1581–1584. https://doi.org/10.14778/2733004.2733035

[37] 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

[38] 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).

[39] Wikipedia contributors. 2018. Minimax — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Minimax&oldid=866945016 [Online; accessed 30-December-2018].

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

[41] 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

[42] 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

[43] 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