Avoiding the Drill-down Fallacy with Storyboard: Assisted and Accelerated Data Exploration Through Data Subsets

ACM Reference Format:
. 2019. Avoiding the Drill-down Fallacy with Storyboard: Assisted and Accelerated Data Exploration Through Data Subsets. In *IUI Conference on Intelligent User Interfaces Proceedings (IUI 2019), March 17–20, 2019, Los Angeles, USA.* ACM, New York, NY, USA, 12 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 identify patterns that merit further investigation, an analyst may need to explore different subsets of the data to determine when and where certain patterns occur. Manually generating and examining each visualization in this space of data subsets (which grows exponentially in number of attributes) presents a major bottleneck.

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

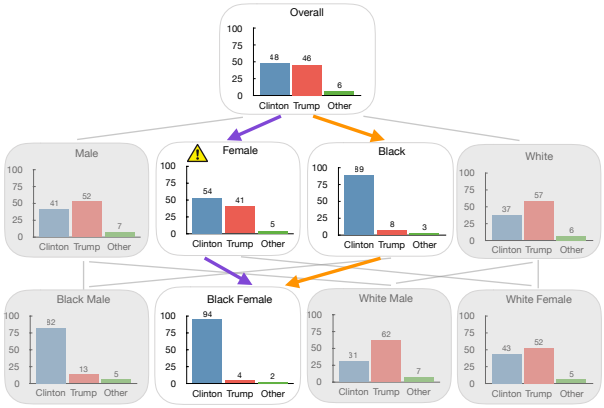

Figure 1: Example data subset lattice from the 2016 US election dataset illustrating the drill-down fallacy along the purple path as opposed to the informative orange path.

charts for female voters, as shown in the second visualization at the second row of Figure 1.

There are three challenges associated with performing manual drill downs in this manner. First, it is often not clear which attributes to drill-down on. Analysts may use their intuition for choosing the drill-down attribute, but such arbitrary exploration may lead to large portions of the lattice being unexplored. Second, an uninformed path taken by analysts 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 `Black` visualization to the `Black Female` one in Figure 1, since the two distributions are similar and therefore not very surprising. Last but most importantly, an analyst may encounter what we are calling the "drill-down fallacy". As shown in Figure 1, an analyst can arrive at the `Black Female` visualization by either going through the purple or the orange drill-down path. An analyst who followed the purple path may be surprised at how drastically the `Black Female` voting behavior differs from that of the `Female`. This behavior is no longer surprising if the analyst had gone down the unsurprising orange path that we saw earlier, where the proper reference (i.e., the vote distribution for `Black`) explains the vote distribution for `Black Female`. In other words, even though the vote distribution for `Black Female` is very different from that of `Female`, the phenomenon can be explained by a more general "root cause"

IUI 2019, March 17–20, 2019, Los Angeles, USA

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

attributed to the voting behavior for the `Black` community. Attributing an overspecific cause to an effect, while ignoring the actual cause, not only leads to less interpretable explanations for the observed visualizations, but can also be detrimental to decision-making. For example, for the campaign manager, this could lead to a misallocation of campaign funds.

The aforementioned example demonstrates the *drill-down fallacy*—incomplete insights that result from potentially confounding factors not explored along a drill-down path. In particular, while performing drill-downs on randomly selected paths, analysts may find a "local difference" in trends, without being aware of the more "general phenomenon" that could explain the trend of interest. Without the proper parent reference visualization that explains the behavior of the visualization of interest, analysts are at risk of falling prey to the drill-down fallacy. A naive solution to avoid this fallacy is to explore all potential drill-down paths. Unfortunately, this approach does not scale with the increasing number of factors in the drill-down path.

In this paper, we present a visual data exploration tool, titled STORYBOARD, that addresses the three aforementioned challenges of exploration through three principles: (i) **Safety** (i.e., ensure that proper informative references are present to avoid drill-down fallacies), (ii) **Saliency** (i.e., identify interesting visualizations that convey new information or insights), and (iii) **Summarization** (i.e., succinctly convey the key insights present in a dataset). To facilitate safety, we develop a notion of *informativeness*—the capability of a reference 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 summarization, we embrace a *collective* measure of visualization utility by recommending a connected network of visualizations that collectively offer informative insights. Based on these three principles, STORYBOARD automatically identifies a network of visualizations that succinctly conveys the key informative insights in a dataset. Our user study results demonstrate that STORYBOARD can guide analysts toward meaningful insights for a variety of tasks. Our contributions include:

- Identifying and characterizing the notion of "drill-down fallacy", a common fallacy that have not yet been studied extensively in the past.
- Introducing the novel concept of *informativeness* that helps users identify meaningful insights that arise from something *actually interesting* about the data (instead of confounding variables),
- Designing a system, STORYBOARD, that automatically identifies a network of visualizations that succinctly conveys the key informative insights in a dataset,
- Demonstrating the efficacy of our system through a user study evaluation on how well users can retrieve interesting visualizations, judge the importance of attributes, and predict unseen visualizations, against two other summarization baselines.

2 PROBLEM DESCRIPTION

In this section, we first describe how analysts manually explore the space of data subsets using drill-downs. We then introduce the three design principles aimed at addressing the current challenges of manual exploration, and automatically guide analysts to the key informative insights.

The Challenge of Manual Drill-Down Exploration

During visual data exploration, an analyst may need to explore different subsets of the data, which 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, 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 to access data subsets at lower levels by adding one filter at a time and visualize the measures of interest for each data subset. Further, as analysts perform drill-downs, they use the most recent visualization in the drill-down path (known as the 'parent') as a reference to establish what they expect to see in the new visualization (known as the 'child'). For example in Figure 1, the visualizations `Female` and `Black` are the *parents* of the `Black Female` visualization, explored along the purple and orange path respectively.

As exemplified by the purple path in Figure 1, during drill-downs analysts may be misguided by improper references that exhibits high deviation locally, in particular when other potential parents (i.e., parents not explored in the drill-down path) that could explain the more general phenomenon are overlooked. We refer to this misinterpretation as the *drill-down fallacy*, since the fallacy arises from the inductive nature of the drill-down operation.

Design Objectives for Informative Exploration

Our goal is to help analysts discover the key informative insights in a dataset, while avoiding drill-down fallacies. We argue in favor of three essential principles for finding such insights—namely the three S's: *safety*, *saliency*, and *summarization*. We adopt these principles to develop a visual exploration tool that automatically selects visualizations that collectively convey the key informative insights of a multidimensional dataset.

Safety. To prevent the drill-down fallacy, we concentrate on *safety*—using informative references for discovering insights. We identify informative references in a drill-down context by modeling the *informativeness* of an observed parent in characterizing the child visualization. An observed parent is *informative* if its data distribution closely follows the child visualization's data distribution, since the parent serves as a proper reference that helps analysts form an accurate mental picture of what to expect from the child visualization. Specifically, we formulate the informativeness of an observed parent V_i^j for a visualization V_i as the similarity between their data distributions measured using a distance function D. The distance $D(V_i, V_i^j)$ is computed based on the probability distributions represented by each visualization (in this case, a vector of bar values). For example, based on the Figure 1 example, the Euclidean distance between `Female` (V_i^j=[54,41,5]) and `Black Female` (V_i=[94,4,2]) is 54.57. .The most informative parents V_i^* for visualization V_i are the ones whose data distributions are most similar to V_i.

$$V_i^* = \{V_i^j : \underset{V_i^j}{argmin}\, D(V_i, V_i^j)\} \quad (1)$$

We regard a parent visualization as informative if its distance from the child visualization falls within a threshold θ% compared to the most informative parent, set as default as 90% and adjustable by user if needed. We regard a parent visualization as informative if its distance from the child visualization falls within a threshold θ% compared to the most informative parent:

$$V_i^{*,\theta} = \{V_i^j : \frac{D(V_i, V_i^*)}{D(V_i, V_i^j)} \geq \theta\} \quad (2)$$

For example in Figure 1, while both `Black` and `Female` visualizations are considered parents of the `Black Female` visualization, only the `Black` visualization is considered an informative parent of the `Black Female` population, for any values of $\theta \geq 11\%$ via the Euclidean distance metric. Note that our proposed system can work with other common distance metrics such as Kullback-Leibler Divergence and Earth Mover's distance [**?**]. Without loss of generality, we chose to use Euclidean distance metric for the remainder of our paper.

Saliency. To discover insights, we emphasize *saliency*— identifying interesting 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 information or unexpected insights. The notion of such interestingness have been explored in past work [**? ? ?**], particularly through the usage of distance-based metrics, where a large distance from some reference visualization indicates that the selected visualization is interesting. However, unlike past work, we concentrate on *informative interestingness*, where the goal is to 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* [**?**] and is known to be NP-Complete [**?**]. We design several approximate algorithms to solve this problem efficiently.

3 STORYBOARD SYSTEM

In this section, we present our system, STORYBOARD, 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.

Lattice Generation: Our system supports two variants of traversal based on the lattice generation procedure—offline variants that first generate the complete lattice and then work

[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.

towards identifying the solution with maximum combined-edge utility, and online variants that incrementally generate the lattice and simultaneously identify the solution. The offline variants are appropriate for datasets with a small number of low-cardinality attributes, where we can generate the entire lattice in a reasonable time; whereas the online variants are appropriate for datasets with many high-cardinality attributes, where we need to incrementally generate a partial lattice. To prevent the danger of visualizations with small population size, users can also select an *iceberg condition* (δ) to adjust the extent of pruning on visualizations whose sizes fall below a certain percentage of the overall population size. [2]

Lattice Traversal: We first describe the offline version of the algorithm before outlining the modification required for the online variant of the algorithm. Given a lattice that has been materialized offline, The objective of the traversal algorithm

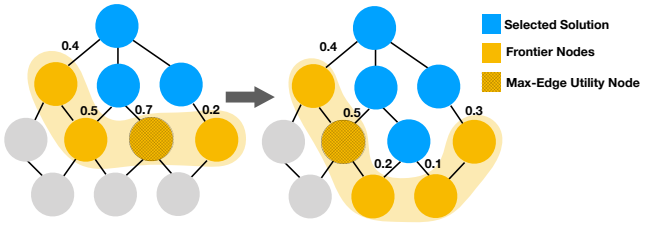

Figure 2: Example illustrating how the frontier greedy algorithm incrementally build up the solution by selecting the maximal-edge utility node from the frontier at every step. On the left, given the three existing nodes in the solution (blue), we select the node with the highest edge utility (hashed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

is to find the connected subgraph in the lattice that has the maximum combined edge utility. Here, we discuss the *frontier greedy* algorithm which is used for generating the dashboards for our user study and defer the details of other algorithms that we have developed to the technical report. The frontier greedy algorithm first compiles a list of 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. In the online version, it additionally checks for each child whether its informative parent is present in the current dashboard. 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.

[2]The terminology is used in the discussion of iceberg cubes in OLAP literature [**?**].

Algorithm 1 Frontier Greedy Algorithm

```
1: procedure PICKVISUALIZATIONS(k,lattice)
2:     dashboard ← { V_overall }
3:     while |dashboard| < k do
4:         frontier ← getFrontier(dashboard,lattice)
5:         maxNode ← getMaxUtilityNode(frontier)
6:         dashboard ← dashboard ∪ {maxNode}
       return dashboard
```

User Interaction

Given the selected visualizations, we render them in a dashboard, where users can inspect the visualization dashboard 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 optional system parameter settings to generate a dashboard. As shown in Figure 3a, the analyst would start with a 7-visualization dashboard on the Police Stop dataset [**?**]. 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 that led to different outcomes, 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. STORYBOARD allows users to request 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 the 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 and 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, STORYBOARD 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*

Figure 3: a) Overview of the STORYBOARD 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.

refinement in other visual analytics systems, which is essential for users to iterate on and explore different hypotheses [**? ?**].

4 USER STUDY EVALUATION

In this section, we describe the methodology and results for a between-subject user study we have conducted for evaluating the utility of STORYBOARD. To assess the efficacy of STORYBOARD across various exploratory analysis goals, we focus on addressing the following research questions:

- RQ1: How *interesting* are the visualizations in the dashboard perceived subjectively by the users?
- RQ2: How well does the dashboard *summarize* the relative importance of different attributes within a given dataset?
- RQ3: How *informative* are the visualizations in the dashboard at providing users with an accurate understanding of unseen child visualizations?

These research questions roughly correspond to the three S's in our objective—saliency, summarization, and safety.

Methods

We recruited 18 participants 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). There were 8 female and 10 male participants. 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.

STORYBOARD: The dashboards for this condition are generated by the aforementioned frontier greedy algorithm and displayed in a hierarchical layout (as seen in Figure 3). To ensure the informativeness of the generated dashboards, we selected a more stringent θ=90% criteria to generate the dashboards for our user study. In order to establish a fair comparison with the two other conditions, we deactivated iceberg pruning (by setting δ=0) and 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 and display them in a 5x2 table layout. 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. 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 [**?**] 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 [**?**], 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.

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:

Retrieval: Participants were asked to talk aloud as they interpreted the visualizations in the dashboard and mark each visualization as either interesting, not interesting, or leave it as unselected. This task was intended to measure 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 understood the relative importance of each attribute in contributing towards an outcome (RQ2).

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.

Quantitative Results

Retrieval (RQ1): Using the click-stream data logged from the user study, we recorded whether a participant marked a visualization in the dashboard as interesting, not interesting, or left the visualization unselected. Table 1 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 STORYBOARD had more visualizations that they found interesting compared to the BFS and CLUSTER condition. This result indicates that STORYBOARD's *saliency* objective was able to select visualizations that were perceived as interesting to the users.

Condition	STORYBOARD	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. STORYBOARD leads to more visualizations marked as interesting and fewer visualizations marked as uninteresting.

Attribute Ranking (RQ2): 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 [**?**]. We deem an attribute as important if it has one of the top-three Cramer's V scores amongst all attributes of the dataset. This relevancy

cutoff is visually-determined via the elbow method to indicate which rank the Cramer's V score drops off significantly. 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 2 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
STORYBOARD	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.

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 STORYBOARD and CLUSTER (which contained more 'local' information) performed better than BFS for both datasets. The problem with BFS is that given the limited number of visualizations that could be shown on a dashboard, 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 STORYBOARD 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 [**?**]. 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 2-4 filters, making it difficult for analysts to interpret 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 users gain a better understanding of attribute importance using STORYBOARD, with only a few targeted visualizations that tells the 'whole story'. Note that this is without STORYBOARD being explicitly optimized for this ranking purpose.

Prediction (RQ3): 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 STORYBOARD (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 STORYBOARD participants found the resulting visualizations to be less surprising, since they were able to more accurately reason about the expected properties of unseen data subsets. 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.

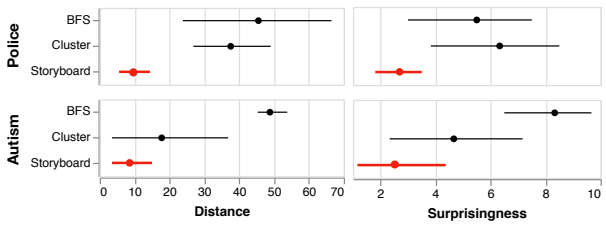

Figure 4: Left: Euclidean distance between predicted and ground truth. In general, predictions made using STORYBOARD are closer to ground truth. Right: Surprisingness rating reported by users after seeing the actual visualizations on a Likert scale of 10. STORYBOARD participants had a more accurate mental 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 relied

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 STORYBOARD is generally lower than the variance from the baselines.

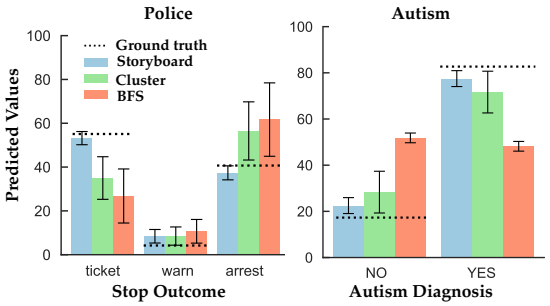

Figure 5: Mean and variance of predicted values. Predictions based on STORYBOARD exhibit lower variance (as indicated by the error bars) and closer proximity to the ground truth values (dotted).

5 DISCUSSION

To understand the usefulness of our recommended visualizations, we analyzed the user study transcriptions through an open coding process 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}={STORYBOARD, 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 types of visualizations that participants utilized and compared against to form their expectations regarding how other unseen visualizations should look like. In particular, we thematically encoded participants' use of contextual references based on the verbal explanations that they provided to justify their prediction task responses. 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 `gender=F, race=White, age=21-30` (in the order of most to least similar) and illustrated graphically in Figure 6:

(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=M, race=White, age=21-30`)
(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).

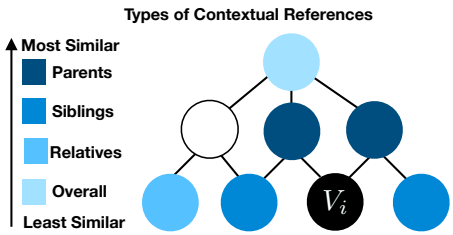

Figure 6: Illustrative example of the different types of contextual reference for a given visualization of interest V_i. The degree of similarity with respect to V_i is denoted by darkest to lightest color hue.

As shown in Table 3, in general, we find that participants make more comparisons in total using STORYBOARD than compared to CLUSTER and BFS. 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 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, the improper comparison against contextual references often makes it difficult for analysts to make sense of the displayed visualizations. In particular, when visualizations composed of multiple filter conditions were shown in CLUSTER dashboards, 25% of participants had trouble interpreting the meaning of the filter for at least one of the datasets. In contrast, as shown in Table 4, this confusion only happened once for BFS and none for STORYBOARD. This is due to the fact that CLUSTER dashboards are seemingly random to the users, making it challenging to find 'close' contextual references to compare against and form an accurate mental model. In contrast, BFS follows a linear ordering and STORYBOARD follows a hierarchical ordering, which were more natural and interpretable for participants.

Algorithm	Parent	Sibling	Relative	Overall	Total
STORYBOARD	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. STORYBOARD 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, due to the limited number of visualizations (k), not all first-level visualizations were displayed in the dashboard. These incomplete comparisons can result in flawed reasoning, as observed in the Autism prediction task described earlier. In contrast, for STORYBOARD, almost all users compared against the overall and parents, while some also exploited sibling comparison information 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.)

The Danger of Small Subpopulations

The danger of spurious patterns and correlations in visualizations that contain small subpopulation size is a well-known problem in exploratory analysis [**?**]. When examining visualizations with many filters and extremely-skewed values in one or more bars (bars with 100% or 0%), 4 CLUSTER participants did not realize that charts with multiple filters may have a smaller subpopulation size. In contrast, 6 of the participants using STORYBOARD explicitly noted that while these extreme-valued visualizations may be interesting, they were less certain due to the unknown subpopulation size and should be investigated further. For example, P1.A1 noted that a visualization with warning=100% caught her eye, "*but I don't know what the N is, maybe it's one person, this makes me a little skeptical, that makes me want to go back to the raw data and look at what is the N and what drives something so drastic?*" We also found that this subpopulation-size fallacy was observed to be more severe for the Autism dataset, where participants had less intuition on the expected attribute behavior. Since BFS dashboards only displayed first-level visualizations, participants for BFS did not see visualizations with large numbers of filters that had small subpopulations during the study, so none of the BFS participants exhibited signs of this fallacy.

Interpretability of Hierarchical Layouts

In the post-study interviews, participants cited hierarchical layout as one of the key reasons why it was easier to follow contextual references in STORYBOARD. Users were able to easily interpret the meaning of the dashboards through STORYBOARD's hierarchical layout, even though they were never explicitly told what the edge connections between the visualizations meant. 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 STORYBOARD'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

	STORYBOARD	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 findings regarding the dataset or information regarding one or more attributes, that are 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.

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.

Since we did not inform participants about how the dashboards were generated, it was also interesting to note that some participants thought that the dashboards were handpicked by a human analyst and described what this person’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]). We encoded this phenomenon by looking at instances where a participant either explicitly referring to a person who picked out the dashboard or implicitly described their intentions through personal pronouns. As summarized in Table 4, 5 out of 12 participants referred to the STORYBOARD dashboards as if they were generated by a human, whereas there was only 1 participant for CLUSTER and none for BFS made such remarks. At the end of the study, many were surprised to learn that the STORYBOARD dashboard was actually picked out by an algorithm, indicating that STORYBOARD could automatically generate convincing dashboard stories similar to a dashboard that was authored with human intention. The interpretability of STORYBOARD 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 STORYBOARD

As described earlier, since the details of how the dashboards were obtained was not explained to the users during the study, some users expressed that they were initially confused by STORYBOARD 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 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.*” This indicates that STORYBOARD 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 STORYBOARD 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.

As discussed earlier, subpopulation size is important in establishing the significance of a trend observed in a visualization. While subpopulation size is taken into account implicitly in our objective, we should design interfaces that convey the notion of subpopulation size in our dashboard. Examples include Sankey-like flow diagrams indicating the percentage of the parent population broken down into individual subpopulations and subpopulation size explicitly specified via edge labels.

6 RELATED WORK

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

Guided Exploration of Multidimensional Data

Given a dataset, tools such as Tableau support automatic generation of visualizations based on perceptual graphical presentation rules [**? ?**]. A more recent body of work automatically selects visualizations based on statistical measures, such as scagnostics and deviation. Given a scatterplot, Anand et al. [**?**] applies randomized permutation tests to select partitioning variables that reveals interesting small multiples using scagnostics. Given a bar chart, Vartak et al. [**?**] 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 [**? ? ?**]. Wall et al. [**?**] 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. [**?**] 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. [**?**] presents a method to detect the presence of the multiple comparisons problem in visual analysis. In this paper, we concentrate on a novel type of fallacy 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 [**? ? ?**]. 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 [**?**]. 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).

Storytelling with Visualization Sequences

Visualizations are often arranged in a sequence to narrate a data-driven story. Existing work on visualization sequences and storytelling has studied the structures of narrative visualizations [**? ?**], effects of augmenting exploratory information visualizations with narration [**?**] and, more recently, ways to automate the creation of visualization sequences [**? ?**]. Most of these work have adopted a linear layout (motivated by slidedecks) to present the visualization sequences. Hullman et al. [**?**] found that most people prefer visualization sequences structured hierarchically based on shared data properties such as levels of aggregation. Kim et al. [**?**] modeled relationships between charts by empirically estimating transition (edge) cost between moving from one visualization (node) to another. They found that participants preferred "*starting from the entire data and introducing increasing levels of summarization*". Our work is the first to automatically organize visualizations in a hierarchical layout for summarizing data distributions across the space of data subsets.

7 CONCLUSION

Common analytics tasks, such as causal inference, feature selection, and outlier detection require studying data distributions at different levels of data granularity [**? ? ?**]. 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 STORYBOARD, 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 STORYBOARD 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 STORYBOARD 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.