STORYBOARD: Navigating Through Data Slices with Hierarchical Summary of Visualizations

ABSTRACT

The task of navigating through a large, multidimensional dataset is a common challenge in exploratory analysis. Due to limitations on the number of visualizations that an analyst can examine at one time, the narrow scope of drill-downs can often lead to inductive fallacies. In this paper, we present STORYBOARD, an interactive visualization recommendation system provide safe guarantee during drill-down exploration by picking the proper visualization reference that leads to interesting and informative trends. Given a dataset and the x and y axes of interest, STORYBOARD intelligently explores the lattice of equivalent visualizations across data subsets, and recommends interesting and informative visualizations. The recommended visualizations are then displayed in an interactive dashboard, where the visualizations are organized into a hierarchical layout. Our evaluation study shows that visualization dashboards generated by STORYBOARD are interpretable and leads to higher performance in data analytic tasks compared to the competing baselines.

KEYWORDS

exploratory data analysis, visualization recommendation.

1 INTRODUCTION

To understand a multi-dimensional dataset, analysts often apply OLAP (Online Analytical Processing) operators to explore the space of attributes [7]. A common OLAP task includes generating visualizations to gain an overview of the data, then drilling down to interesting subsets to generate more visualizations. For example, a campaign manager may be interested in understanding the voting patterns across different demographics (say, race, gender, social class) using the 2016 US election exit polls[1]. A natural first step is to generate a bar chart for the entire population, where x-axis shows the election candidates and y-axis the percentage of votes for these candidates. He can then drill down to specific demographics of interest, say gender-based demographics by generating bar charts for female voters. In this exploration process each drill-down may lead to insights, which derive from the observed visualizations. As shown in Figure 1, an analyst can either arrive at the Black Females visualization by going through the purple or orange drill-down path. At random, an analyst that followed the purple path may be surprised at how drastically the Black Female voting behavior

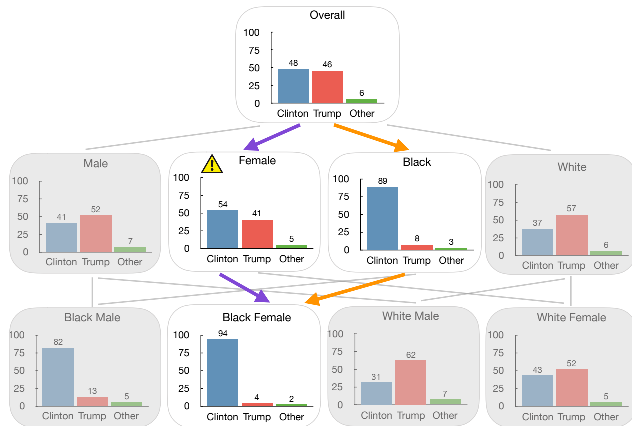

Figure 1: Example data subset lattice illustrating the misleading factor fallacy along the orange path as opposed to the informative purple path.

differs from the vote distribution for females. This behavior is no longer surprising if the analyst had went down the orange path, where the proper reference (vote distribution for Black) explains the behavior of the Black Female distribution. The misleading insight is a result of the order in which the drill-down has been performed. This example demonstrates a case of *drill-down fallacy*, which results from potentially confounding factors not explored along a drill-down path.

When an analyst explore a dataset by randomly selecting the attribute to drill down on, they may not come across the proper reference visualization that explains the behavior of the visualization of interest. Thus, they are at risk of falling prey to the drill-down fallacy. A naive solution to avoid this fallacy is to explore all potential pathways along the drill-down path. For example, generating and exploring visualizations for both race and gender based demographics, before exploring any of their combinations. Unfortunately, this approach does not scale with increasing number of factors in the drill-down path.

In this paper we develop a tool to help users explore a dataset while avoiding improper references that may lead to drill-down fallacy. Our tool automatically identifies the best possible drill-down paths that lead to *informative insights*, and summarizes the paths. The challenge of building such tool includes considerations for how each visualization influences user's perception on other visualizations and selecting a set of visualization that are collectively interesting amongst

[1] https://edition.cnn.com/election/2016/results/exit-polls

a large set of visualization. To address this challenge, we develop a notion of *informativeness*, defined as the capability of an reference visualizations to explain the visualization of interest. Informative visualizations helps users identify meaningful insights that arise from something *actually interesting* about the data (instead of confounding variables), thereby preventing users from the drill-down fallacy. Our user study result demonstrates that our tool that make use of this notion of informativeness can guide an analyst towards meaningful insights. The contribution of this paper include:

- Introducing the novel concept of *informativeness* that helps avoid drill-down fallacy in data exploration (Section 3),
- Designing a tool that automatically identifies the best possible drill-down paths based on informative insights, and summarizes those (Section 4),
- Demonstrating the efficacy of our system through a comprehensive user study evaluation (Section 5).

2 PROBLEM FORMULATION

In this section, we first describe how analysts explore the space of visualizations through drill-downs and introduce a common fallacy that arises when analysts have limited time and attention to examine all possible factors that contribute to the observed visualization. Then, we discuss how to resolve the problem of finding informative references along a drill-down path.

Research in visualization storytelling shows that people prefer hierarchically structured visualizations with increasing levels of aggregation [9, 10, 12]. In order to find meaningful insights, analysts often drill-down to explore data at different levels of granularity by adding one filter at a time. For each data subset that they encounter, they may want to visualize the data distributions through a bar chart. When analysts perform a drill-down by adding one additional filter, they naturally look towards the last visualization that they have seen (known as the 'parent') to establish what they expect to see in the current visualization (known as the 'child'). In this case, a parent is any visualization that can be obtained by removing one filter constraint from the child. For example in Figure 1, the visualizations Female and Black are the parents of the Black Female visualization.

As analysts perform drill-downs, they may be misguided by child visualizations that highly deviate from one of its parents, if one of the other potential factors that explain seemingly-anomalous behavior is overlooked (i.e. not along the chosen drill-down path). As exemplified by the exploration along the purple path in Figure 1, we refer to this phenomena as *drill-down fallacy*, since this type of fallacy arises from the inductive nature of the drill-down operation. While such fallacies can be prevented if the analyst exhaustively browses through all possible parents of any visualization that he observes in the dataset, the prohibitively large number of visualizations and limited memory and attention of analysts make this task impractical.

Due to these challenges, our goal is to develop a mechanism that would *provide safe guarantee by picking the proper informative parent* as a reference when analysts navigate through the space of data subsets. To model the informativeness of an observed parent in the context of an unseen visualization, we characterize the capability of the parent in predicting the unseen visualization. An observed parent is *informative* if its data distribution closely follows the data distribution of the unseen child visualization, since the visualization helps the analyst form an accurate mental picture of what to expect from the unseen visualization. Specifically, we formulate the informativeness of an observed parent V_i^j of an unseen visualization V_i as the similarity between their data distributions measured using a distance function $D(V_i, V_i^j)$. The most informative parents V_i^* of an unseen visualization V_i are the ones whose data distributions are most similar to the unseen.

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

We regard a visualization as informative if its distance falls within a user-defined threshold $\theta\%$ close to its most informative parent:

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

For example in Figure 1, while both visualization Black and Female visualizations are considered parents of the Black Female visualization, only the Black visualization are considered the 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 different distance metrics such as cosine similarity and earth mover's distance. Without loss of generality, we chose to use Euclidean distance metric for the remainder of our paper.

3 SYSTEM

While the concept of informativeness is useful for helping analysts determine which parent is the proper informative reference for a given visualization, in practice, users often do not have a preconceived knowledge of what visualizations would lead to useful insights. The ultimate goal of exploration is to discover insights, in particular finding visualizations that are *interesting* and lead to those insights. However, without knowing *what* subset of data contains an insightful distribution, manually exploring distributions from all possible data subsets can be tedious and inefficient. In order to accelerate the process of manual drill-down, our goal is to develop a system that automatically selects a small set of *interesting* visualizations to summarize the distributions within a dataset in an *safe and informative* manner.

We first highlight three of the challenges (*significance, safety, saliency*) that we face when building such a system and how they are each addressed in our system objective.

Safety: As discussed in Section 2, failure to select the proper reference for a given visualization can lead to the drill-down fallacy. To resolve this issue, we require that for every visualization except for the overall, at least one of its informative parents must be included within the set of selected visualizations. This enforces that every parent in the visualization lattice is guaranteed to be informative.

Saliency: We want to select visualizations that are *visually-salient*, in other words, the visualization distribution is *interesting* if differs from the distribution of its parents. The use of distance-based metrics to quantify surprisingness or interestingness have been widely adopted in past work [5, 11, 16].

To model the interestingness of an visualization V_i in the context of its parent V_i^j, we characterize the deviation between their data distributions using a distance function $D(V_i, V_i^j)$. From the safety criteria, all parents in the dashboard are guaranteed to be informative, therefore the reference would not be misleading. visualization shown in the dashboard has an informative reference to compare against to create a connected story.

Significance: The danger of spurious patterns and correlations in visualizations that contain small subpopulation size is a well-known problem in exploratory analysis [4]. We take two preventive measures to avoid picking these misleading visualizations that are 'insignificant' in size. When constructing the visualization lattice, we allow users to select an 'iceberg condition' [2] (δ) to adjust the extent of pruning on visualizations whose sizes fall below a certain percentage of the overall population size. Second, we downweigh the interestingness edge utility $D(V_i, V_i^j)$ between a parent V_i^j and a child visualization V_i by the ratio of their sizes $U(V_i, V_i^j) = \frac{|V_i|}{|V_i^j|} \cdot D(V_i, V_i^j)$.

Given these objectives, we select k visualization to include in our dashboard that represent a connected set of visualizations that are collectively safe, salient and significant, based on maximizing the utility $U(V_i, V_i^j)$. The problem of finding a connected subgraph in the lattice that has the maximum combined edge utility is known as the maximum-weight connected subgraph problem [3] and is known to be NP-Complete, via a reduction from the CLIQUE PROBLEM [13]. Next, we discuss heuristic algorithms used for deriving a locally optimal solution for ensuring interactive runtime.

Algorithms

We discuss the algorithms used for generating the visualization lattice, and then present a high-level overview of our traversal algorithms to selecting the k-connected maximum-weighted subgraph.

Lattice Generation: Our system supports two variants of traversal algorithms based on the lattice generation procedure—offline variants that first generate the complete lattice and then work towards identifying the maximum utility solution, 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 large number of high-cardinality attributes, where we incrementally generate a partial lattice.

Lattice Traversal: Given the materialized lattice, the objective of the traversal algorithm 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 our discussion on the details of other algorithms that we have developed to the technical report.

As described in Algorithm 1, our algorithm obtains a list of candidate nodes known as the *frontier* nodes, which encompasses all neighbors of nodes in the existing subgraph solution. Any of the nodes in the frontier can be added to the current solution since their informative parent is present in the solution. The `getFrontier` function 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. At each step, our algorithm greedily picks the node with the maximum utility from the frontier to the current solution, and updates the frontier accordingly.

[2]The terminology is used in the discussion of iceberg cubes in OLAP literature [18].

Figure 2: Overview of the STORYBOARD interface for the Police Dataset [1]. Users can select x and y axes of interest, as well as a choice of an aggregation function. Default values are set for system related parameters such as the number of visualizations to show in the dashboard (k), iceberg condition for pruning (δ), and informative parent criterion (θ), which can be adjusted by the users via the sliders if needed.

Algorithm 1 Frontier Greedy Algorithm

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

User Interaction

Given the selected visualization to be included in the dashboard, we render the visualization in an interactive frontend interface, as shown in Figure 2. After the user selects the x and y axes of interest, aggregation function, and optional system parameter settings, an initial dashboard of k visualizations is displayed on the canvas. The system allows users to inspect the visualization dashboard through panning and zooming with navigation buttons, mouse clicks, and keyboard bindings.

After browsing through visualizations in the dashboard, users may be interested in getting more information about a particular node. STORYBOARD allows users to request additional summarizations based on a chosen visualization of interest as the new starting point for analysis. As shown in Figure 3 (left), the analyst starts with a 5-visualization dashboard on a police stop dataset. The dataset contains records of vehicle and pedestrian stops from law enforcement departments in Connecticut, dated from 2013 to 2015. The analyst learns that for the drivers who had contraband found in the vehicle, the arrest rate for drivers who are 60 and over is surprisingly higher than usual, whereas for Asian drivers the arrest rate is lower. In addition, he is also interested in learning more about the other factor that contribute to high arrest rate: duration=30+min. He clicks on the corresponding visualization and requests for 2 additional visualizations. Upon seeing the updated dashboard in Figure 3 (right), he learns that similar to the selected visualization, any visualization that involves the duration=30+min filter results 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 driver's race or age. 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 refinement* in other visual analytics system, which is essential for the users to iterate on and explore different hypotheses [8, 17].

Assistive tools for visualizing large lattices

Due to the amount of space occupied by the hierarchical layout when the number of visualizations gets large, we have developed tools to help users navigate through different parts of the dashboard interactively.

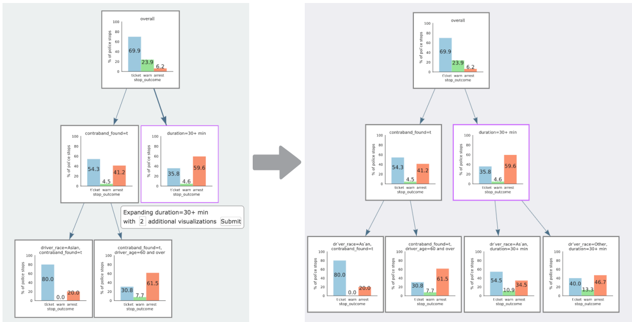

Figure 3: Left: Original k=5 dashboard with the duration=30+min visualization clicked. A pop-up is displayed to submit the request for additional summary visualizations to be generated. Right: Resulting dashboard after requesting for 2 more visualizations based on the visualization of interest.

Navigation Minimap: When the user zooms in on the dashboard, an overview mini-map is shown on the upper left-hand side of the canvas to help users identify which region of the dashboard they are currently exploring, as shown in Figure 5.
Collapsed visualizations: One observation that we found across several datasets was that many visualizations had identical distributions, which resulted in lots of wasted space. Apart from their attribute name, these visualizations are not very informative for the users, therefore, we offer an option to collapse these visualization, as demonstrated in Figure 4. A visualization can be collapsed if it has more than one redundant sibling and does not have any children, so that there are no hidden stories stemming from lower-level dependencies. As shown in Figure 5, collapsed nodes can be easily identified by an orange border and the details of which visualizations are in the collapsed node are displayed when the user hovers over the visualization.

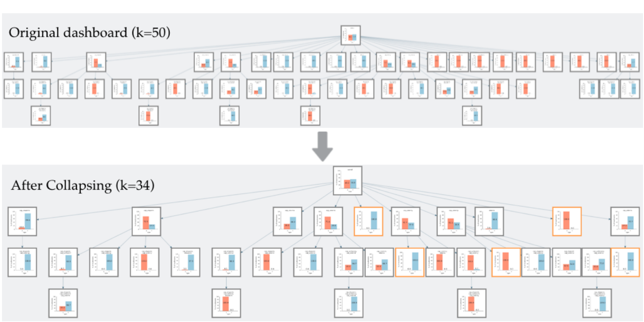

Figure 4: An example of the k=50 dashboard for the mushroom dataset [15], which contains type={posionous, edible} on the x-axis. The collapsed dashboard (bottom) removed 16 redundant visualizations from the original dashboard (top).

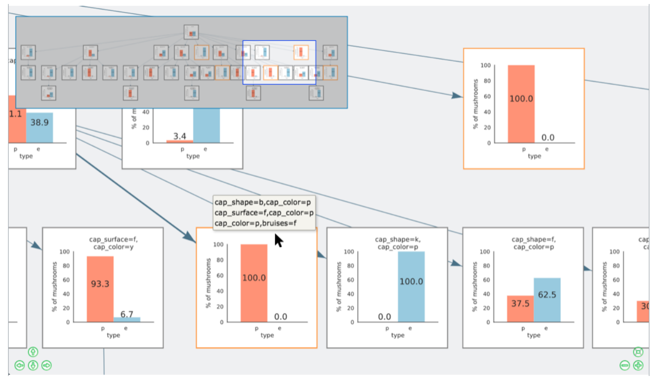

Figure 5: Zoomed-in version of Figure 4 showing the labels of a collapsed visualization when user hovers over the visualization. The navigation minimap is shown in the top-left to help users navigate through the large dashboard.

4 USER STUDY EVALUATION

Methods

We evaluate the utility of our tool by performing a user study focusing on addressing the research questions:

- RQ1: How effective is our tool at discovering visualizations of interest?
- RQ2: How effective is our tool at helping user evaluate the importance of attributes within a given dataset?
- RQ3: How effective is our tool at guiding users towards safe and informative visualization references?

We recruited 18 participants with prior experience working with data. Participants include undergraduate and graduate students, researchers, and data scientists, with 1 to 14 years of data analysis experience (average = 5.61). There were 8 female participants and 10 male participants. No participants reported prior experience in working with the two datasets used in the study.

In this between-group study, participants are randomly assigned two of the three dashboards with k=10 visualizations generated by following conditions.
STORYBOARD: The dashboards for this condition is generated by the frontier greedy algorithm (described in Section 3) and displayed in a hierarchical layout (as seen in Figure 2). In order to establish a fair comparison with the two other conditions, we deactivated the interactive node expansion and dashboard navigation functionalities described in Section 3, especially since the k=10 dashboard was small enough to function without the navigation tools.
BFS: Starting from the overall visualization, k visualizations is selected in level-wise order: sequentially adding visualizations at the first level with 1-filter combination one at a time, proceeding with the 2-, 3-, etc. filter combinations, until k visualizations have been added to the dashboard. This

baseline is designed to simulate the dashboard generated by a meticulous analyst who exhaustively inspects all possible visualization combinations. The chosen visualizations are displayed in a 5x2 table layout in the traversed order.

CLUSTER: K-Means clustering is performed on the dataset with k clusters, corresponding to k, the number of visualizations to be shown in the dashboard. For each representative cluster, we select the visualization that has the least number of filter conditions for interpretability[3] and display them in a 5x2 table layout.

We randomize the ordering for each task combination to prevent confounding learning effects. The study begins with a 5 minute tutorial using dashboards generated from the Titanic dataset [2]. To prevent participant's bias, participants were not provided an explanation of how the dashboard is generated and why the visualizations were arranged in a particular way. Then, participants proceeded onto the Police dataset [14] , which contains a total of 312948 records of vehicle and pedestrian stops from law enforcement departments in Connecticut, dated from 2013 to 2015. We generate a dashboard of visualizations with bar charts with x-axis as the stop outcome (whether the police stop resulted in a ticket, warning, or arrest/summons) and y-axis as the percentage of police stops that led to this outcome. The attributes in the dataset include driver gender, age, race, and the stop time of day, whether a search was conducted, and whether contraband was found.

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 are 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 interpret 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 well participants are at retrieving interesting visualizations (RQ1).

Attribute Ranking: Participants were given a worksheet with all the attributes listed and asked to rank the attribute in order of importance in contributing to a particular x-axis value (e.g. stop outcome = arrest, autism = yes). Attribute ranking tasks are common in feature selection and other data science tasks. The goal of this task is to measure how well participants understand the relative importance of each attribute in contributing towards an outcome (RQ2).

Shallow Prediction: Participants were given a separate worksheet and asked to draw an estimate for a visualization that is not present in the dashboard. The visualization to be estimated is considered "shallow" if it is a visualization with 2 filter combinations, with one parent present in the given dashboard. After making the prediction, participants are shown the actual data distribution and asked to rate on a Likert scale of 10 how surprising the result was.

Deep Prediction: Similar to the shallow prediction, except that the visualization to be estimated is "deep" in the sense that it has 3 filter combinations, with only one parent in the given dashboard. Since we can not directly test compare between misleading and informative drill-down paths for RQ3, we use both prediction tasks as a proxy to how informative the selected visualizations in the dashboard are, by measuring how accurate the participants are at predicting an unseen visualization.

The second dataset in the study is the Autism dataset[6], 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. Participants are not given the descriptions of the questions nor the answers corresponding to the labels. We generate dashboard visualizations based on whether the participant is diagnosed with autism or not. We repeat the same study procedure described above for the Autism dataset. At the end of the study, we asked two open-ended questions regarding the stories and insights that they have learned and what they like or dislike about each dashboard.

Quantitative Results

Retrieval (RQ1): Using the click-stream data logged from the user study, we record whether each user is interested, not interested, or have not selected a visualization in the dashboard. Since we do not have a objective ground truth on which visualization is interesting or not interesting, we devise a voting-based measure that measures how interesting is a visualization amongst all participants. Here i indexes the visualization and j indexes the user. As shown in Equation 3, we assign a vote δ_{ij} of 1 if a user is interested in a visualization, 0 if they leave it unselected, and -1 if they are not interested in a visualization.

$$\delta_{ij} = \begin{cases} 1 & \text{interested} \\ 0 & \text{unselected} \\ -1 & \text{not interested} \end{cases} \tag{3}$$

We obtain a consensus score for each visualization to measure how frequently the visualization is regarded as interesting by summing over all user's vote on that visualization.

$$\text{consensus}(V_i) = \sum_{j \in user} \delta_{ij} \tag{4}$$

Given a consensus measure of how interesting a visualization is, we can define a rating score which measures how

[3]Due to this requirement, the overall visualization is guaranteed to be picked as one of the displayed visualizations.

good a particular user's rating is, by taking the product of the consensus interestingness score and the rating value, as shown in Equation 5. Intuitively, a rating should be rewarded more if it has retrieved interesting visualization agreed by many other users, likewise, ratings that does not retrieve such visualizations should be penalized more heavily.

$$\text{rating score}(V_{ij}) = \text{consensus}(V_i) \cdot \delta_{ij} \quad (5)$$

Table 1 summarizes results of rating scores averaged over the tasks that the user performed.

Dataset	STORYBOARD	Cluster	BFS
Police	0.89	0.87	1.65
Autism	3.04	3.00	1.90

Table 1: Average consensus-agreement score for different algorithm and datasets.

Due to the highly subjective nature of the retrieval task, the interestingness selection for the Police dataset was biased by participant's priors and intuition about the attributes. For example, while all participants who have seen the visualization "duration=30+min" verbally noted that stop duration is a crucial factor that leads to arrest, only 4 users marked it as interesting. 5 participants marked the visualization as not interesting and 4 left it unselected, because the visualization was not very surprising as it agreed with their intuition that "*if the police stop is taking a long time, something has probably gone wrong*".

Since the attributes in the Autism dataset are simply question numbers, participants could not associate any priors to their interestingness selection. In this prior-agnostic case, participants who used STORYBOARD found more visualizations of interest that corresponded to the consensus, indicating that there are more interesting visualizations picked out by STORYBOARD than compared to the two baseline-generated dashboards.

Attribute Ranking (RQ2): To determine attribute importance ranking for a dataset, we computed the Cramer's V statistics between attributes to be ranked and the attributes of interest. Cramer's V test makes use of the chi-square statistics to determine the strength of association between attributes. Using the ranks determined by Cramer's V as ground truth, we compute the normalized discounted cumulative gain (NDCG@k) of each participant's ranking average over all tasks[4], as detailed in Table 2. We see that STORYBOARD performs better than clustering in both cases. Since clustering seeks for a set of visualization that exhibits diversity in the shape of the data distribution, it results in visualizations with many filter combination, which is hard to interpret without appropriate context to compare against. BFS performs better than STORYBOARD in the Police dataset, but not in the Autism dataset. BFS may have performed better than STORYBOARD in the Police dataset for a combination of two reasons: 1) since BFS exhaustively displays all attributes sequentially, for the Police dataset it had happened to select several of the important attributes (related to contraband and search) to display as the first 10 visualizations and 2) as discussed earlier, some participants had priors on the data attribute which influenced their ranking. However, with a budget of k=10, only visualizations regarding diagnostic questions 1-5 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 important attributes (questions 6 and 9) given the limited budget. In general, our results indicate that using STORYBOARD, users gain a better understanding of variable influence and correlation.

Dataset	STORYBOARD	CLUSTER	BFS
Police	0.63	0.45	0.84
Autism	0.50	0.30	0.24

Table 2: NDCG@10 scores for the attribute ranking task.

Prediction (RQ2): In order to measure how accurate participants' decisions are, we computed the Euclidean distance between their predicted distributions and ground truth data distributions. As shown in Figure 6 (top), all the shallow predictions made by using information from the STORYBOARD is closer to the actual distribution compared to the baselines. This aligns with our findings in the formative study and indicates that users are able to more accurately reason about how unseen data would behave with STORYBOARD. Figure 6 (bottom) also shows that participants who used STORYBOARD reported that they were less surprised when the unseen visualization is revealed, which again indicates that participants had a more accurate mental model of prediction.

STORYBOARD did not perform as well compared to the baselines for the Autism deep prediction task. One possible reason for this is due to the fact that the shallow and deep prediction tasks for the Autism dataset were correlated. Therefore, after learning about the insights that answering 1 on question 9 results in a very high probability for an autism diagnosis, some participants made use of that information when tackling the subsequent deep prediction task. By discussing with the baseline participants on how they have obtained the prediction estimates, they described how surprised they were by the finding in the shallow prediction and therefore adjusted the autism diagnosed values to be higher to compensate for their mistake in the subsequent deep prediction task.

We also compute the variance of participants' predictions across the same task. In this case, low variance implies that

[4]Since participants are asked to examine all attributes, the k for NDCG@k corresponds to total number of attributes in that dataset.

any user 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 their predictions, so instead participants relied on different priors or guessing to form the prediction. These trends can be observed in Figure 7, where the prediction variance amongst participants who used STORYBOARD is much lower than the variance from the baselines.

REFERENCES

[1] [n. d.]. Connecticut Racial Profiling Prohibition Project Data Portal. http://ctrp3.ctdata.org/. Accessed: 2018-07-16.

[2] [n. d.]. Titanic: Machine Learning from Disaster. Kaggle. https://www.kaggle.com/c/Titanic

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

[4] Carsten Binnig and Lorenzo De Stefani. 2017. Towards Sustainable Insights or why polygamy is bad for you. *Cidr* (2017).

[5] 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. https://doi.org/10.1109/TVCG.2016.2598618

[6] Fadi Fayez Thabtah. 2017. Autism Screening Adult Data Set. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml

[7] 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. https://doi.org/10.1023/A:1009726021843

[8] 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). https://doi.org/10.1109/TVCG.2017.2744684

[9] Jessica Hullman, Steven Drucker, Nathalie Henry Riche, Bongshin Lee, Danyel Fisher, and Eytan Adar. 2013. A deeper understanding of sequence in narrative visualization. *IEEE Transactions on Visualization and Computer Graphics* 19, 12 (2013), 2406–2415. https://doi.org/10.1109/TVCG.2013.119

[10] Jessica Hullman, Robert Kosara, and Heidi Lam. 2017. Finding a Clear Path : Structuring Strategies for Visualization Sequences. 36, 3 (2017).

[11] Laurent Itti and Pierre Baldi. 2009. Bayesian surprise attracts human attention. *Vision Res* 49, 10 (19 May 2009), 1295–1306. https://doi.org/10.1016/j.visres.2008.09.007 18834898[pmid].

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

[13] 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. https://doi.org/10.1145/1871437.1871555

[14] 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. https://openpolicing.stanford.edu/data/

[15] Jeff Schlimmer. 1987. Mushroom Data Set. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml

[16] Manasi Vartak, Sajjadur Rahman, Samuel Madden, Aditya Parameswaran, and Neoklis Polyzotis. 2015. SEEDB : Efficient Data-Driven Visualization Recommendations to Support Visual Analytics. (2015).

[17] 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. https://doi.org/10.1109/TVCG.2015.2467191

[18] Dong Xin, Jiawei Han, Xiaolei Li, Zheng Shao, and Benjamin W. Wah. 2007. Computing iceberg cubes by top-down and bottom-up integration: The starcubing approach. *IEEE Transactions on Knowledge and Data Engineering* 19, 1 (2007), 111–126. https://doi.org/10.1109/TKDE.2007.250589

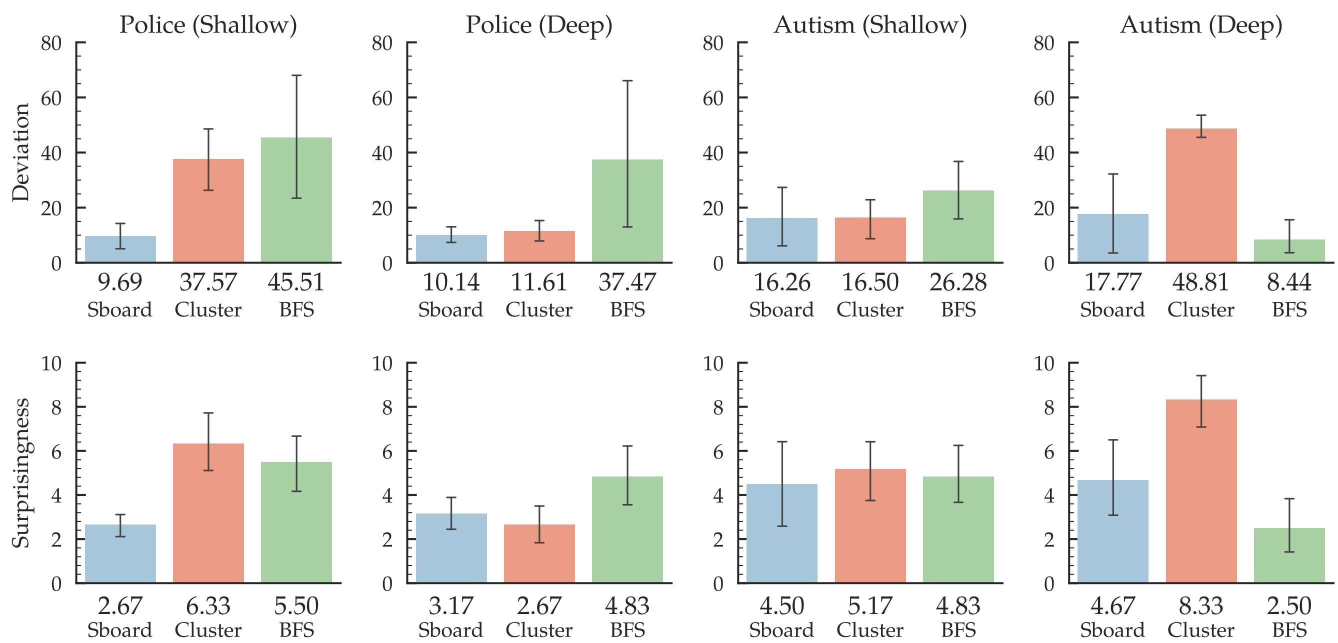

Figure 6: Top: Euclidean distance between predicted and ground truth. Bottom: Surprisingness rating reported by users after seeing the actual visualizations on a Likert scale of 10.

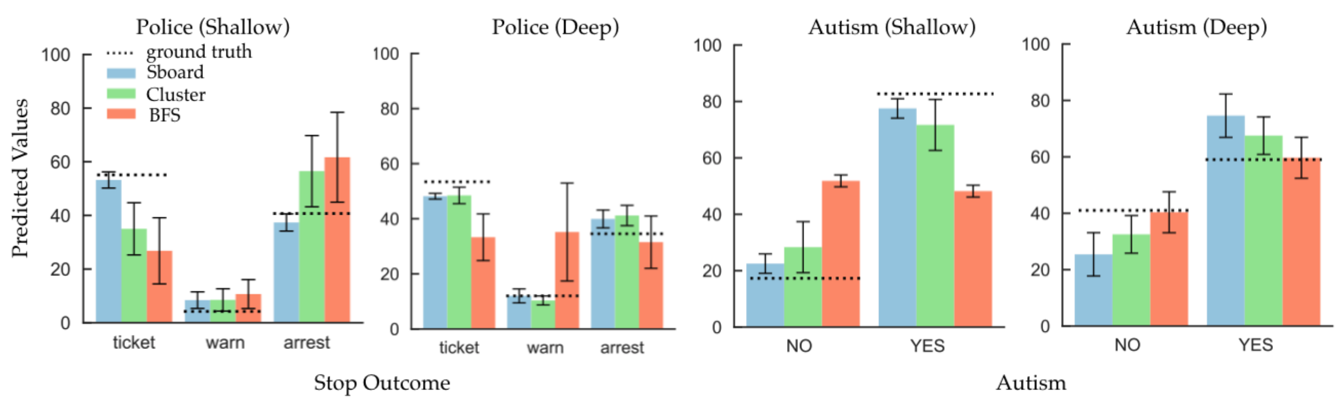

Figure 7: Mean and variance of predicted values. Predictions based on STORYBOARD exhibits lower variance (as indicated by the error bars) and great proximity to the ground truth values (dotted).