VizRec 2.0: Towards a holistic workflow for visual data exploration

Doris Jung-Lin Lee, Aditya Parameswaran
{jlee782,adityagp}@illinois.edu
University of Illinois, Urbana-Champaign

Abstract

test abstract

1 Introduction

Visualization —¿ insights.

1. Visual analysis help discover insights, etc.
2. supporting cycle of visual data exploration.
3. In this paper, we introduce various challenges in the —- visual analysis, and discuss —- solutions —.
4. Challenge #1: Precise search
5. finding the right vis and relevant data is hard,
6. we discuss *zenvisage* as one use case and solution in this space (Section 2)
7. Challenge #2: Need for in-the-loop support. how do people query visually?
8. During our PD + others, hypothesis + in the loop considerations is important. We discuss relevant work and our findings in Section 3.
9. Need to formulate complex expressive queries, language is good, but need interface (e..g visual primitives) to support this. (Section 4)
10. top-down, bottom-up –¿ point to need for bottom-up recommendations (Section 5)
11. Challenge #3: Vague and intelligent search
12. Users might not always have something they want to start with, supporting vague querying (Section 4)
13. Challenge #4: Cold-start recommendation for data understanding. One aspect of this is to gain understanding of dataset, (e.g. representative and outliers in *zenvisage*, bottom up exploration)we discuss STORYBOARD as one example solution addressing this problem (Section 5).

Bulletin of the IEEE Computer Society Technical Committee on Data Engineering

2 Precise Visual Querying

Visual analysis often reveal important anomalies or trends in the data[5]. However, it is often challenging to find the appropriate piece of information to realize these insights.

2.1 Motivating Example

Astronomers from the The Dark Energy Survey (DES)[**?**] are interested in finding anomalous time series to discover astrophysical transients (objects whose brightness changes dramatically as a function of time), such as supernova explosions or quasars. When trying to find celestial objects corresponding to supernovae, which have a specific pattern of brightness over time, scientists need to individually inspect the visualizations of each object until they find ones that match the pattern. With more than 400 million objects in their catalog, each having their own set of time series brightness measurement, the process of manually exploring a large number of visualizations is not only error-prone, but also overwhelming for scientists who do not have extensive knowledge about their dataset.

The astronomy use case highlights a common challenge in exploratory data analysis (EDA). There is often a large space of possible visualizations that could be generated from a given dataset and manual search through this large collection is inefficient. Visualization authoring tools such as Tableau and Excel focusses on presenting one visualization at a time. There is no systematic way to create, compare, and query large collections of visualizations.

2.2 Effortless Data Exploration with *zenvisage*

The challenges presented earlier points to a need for a tool that enables users to create and search through large collections of visualizations. We developed *zenvisage* a visual query system that . *zenvisage* is built on top of a querying language called ZQL, which provides a mechanism for managing collections of visualizations[11]. Contrary prior work on visualization languages for specifying visual encodings of individual visualizations [12, 15], ZQL supports high-level queries over visualizations, such as composing, sorting, filtering a collection of visualization. Example functionalities of ZQL includes:

- Comparing across a collection of visualizations by iterating over one or more x, y, z attributes while fixing other attributes (e.g. Find me all the cities where the housing prices starts off high then becomes lower over time. Here we vary along CITY while keeping X=TIME,Y=AVG(PRICE) fixed.)
- Finding the top-k visualizations whose y values are most or least similar from a queried visualization.
- Finding a pair of X and Y axes where the visualizations for two specific products 'stapler' and 'chair' differ the most

Given a ZQL query, *zenvisage* parses it into a graph of visual component nodes(contains visualization information, such as X, Y columns) and task nodes (common and user-defined primitives for processing visual components, such as sort-filter). *zenvisage* then performs query optimization to merge together multiple nodes, as well as reducing the processing time required for individual visualization components. Using the optimized query plan, the executor compiles visual nodes into SQL queries for retreiving the visualization data and post-processes the result via the defined operations.

While ZQL provides powerful mechanism for expressively specifying queries on large collections visualizations, writing ZQL queries can be daunting for novice users. Therefore, we extracted a typical workflow of visualization querying (finding top-k most similar visualization from a collection with fixed X,Y while varying Z) to allow users to formulate ZQL queries through interactions (such as drag-and-drop, sketching), while allowing them to construct ZQL queries via a frontend table input. *zenvisage* maps the frontend interactions into ZQL queries submitted to the backend and renders the query results as a ranked list of visualizations in the Results panel as shown in Figure 2.2. *zenvisage* is a full-fledged visual querying system that supports a variety of querying interactions as described in Figure 2.2. In the following section, we will discuss the design process of how we developed this visual query system and the lessons that we have learned for designing future visual data exploration systems.

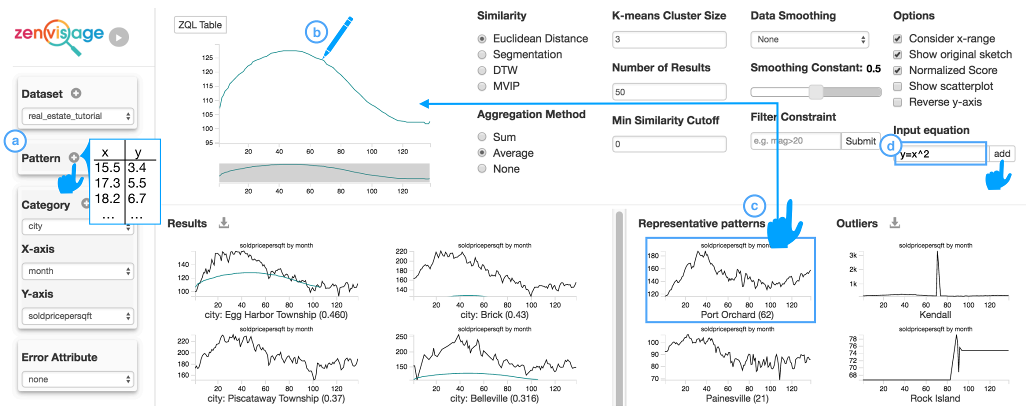

Figure 1: *zenvisage* offers a variety of querying modalities, including: a) uploading a sample pattern from an external dataset as a query, b) sketching a query pattern, c) dragging-and-dropping an existing pattern from the dataset, and d) inputting an equation as a query.

3 Hypothesis Formation and the cycle of visual analysis

3.1 Visual Querying in the *Sensemaking Framework*

In developing *zenvisage*, we collaborated with scientists from astronomy, genetics, and material science in a year-long participatory design process [4]. In particular, we study how various features impacts analyst's ability to rapidly generate new hypothesis and insights and perform visual querying and analysis. In addition, we were interested in how a visual query system (VQS) like *zenvisage* fits into the participant's analysis workflow. Our findings offered design guidelines for improving the usability and adoption of next-generation VQSs. More importantly, in this paper, we focus on applying our findings on visual querying to the context of supporting the full cycle of visual data exploration. We highlight two of the key findings related to this goal below:

Our participatory design findings points towards future —– in supporting a cycle of —. —-advocate —cycle of visual analysis.

- Essential ingredient in facilitating intelligent vague querying and exploration.
- This is a human process ([2, 7])
- Iterative Hypothesis Exploration/Refinement : argue that the following properties is important to sustain this "cycle of visual analysis"

Supporting Complex Vague Expressive Querying *zenvisage* is an example of a *precise visual querying system (PVQS)*, which accepts precise queries as an input, expressed through interactions or directly specified via ZQL. The expressiveness of PVQSs comes from the multiplicative effect of —— combinations —- into a custom workflow, combinations — workflow. For example, in our participatory design study, we found that —- (for example, filter first, then examine representative trends, then query with pattern, then adjust filter to update hypothesis, higher lower value).

The extensibility of these systems or querying language also comes with the cost of potentially overloading the users with too many potential options to chose from.

When users are querying with our VQS (which we now refer to as 'precise querying'), they often need to translate their ambiguous, high-level questions into an plan that consists of several interactions in series which addresses the desired query incrementally. However, the combination workflow is limited and sometimes, queries can not be expressed in this framework. Complex multistep queries. Give one example.

despite extensive work in database usability, there is an inevitable design trade-off between the query expressivity and interface usability[5, 3]. In Section 4, we survey related work in this area and advocate for —–.

Precise querying

Construction of multi-step queries - tie in many different types of interactions - goal is data understanding

Our *zenvisage* work ——- Bottom up and top down querying in VQS facilitates rapid insight discovery. workflow integration , complex queries, While our visual interface —, —-it was unable to capture all the —–. [Give some example queries that didn't work]. Balance between language/querying complexity versus expressiveness. More importantly pointed towards a need for vague querying. Give some examples of vague querying.

Top-down and Bottom-up Querying Modalities We employed Pirolli and Card's [7] information foraging framework for domain-experts to contextualize our study results. Pirolli and Card's notional model distinguishes between information processing tasks that are *top-down* (from theory to data) and *bottom-up* (from data to theory). In the context of visual querying, users employ top-down approaches by starting with a hypothesis on what patterns to look for and express it through sketching or inputting an equation (Figure 2.2b,d). On the other hand, bottom-up approaches originate from the data (or equivalently, the visualization). For example, the user may drag and drop a visualization of interest in the dataset as the input query or upload a visualization from an external dataset (Figure 2.2a,c).

Our interactions with the scientists showed that *bottom-up querying via drag-and-drop was more intuitive and more commonly used than top-down querying methods when the users have no desired patterns in mind*, which is commonly the case for exploratory data analysis. One of the main reason why participants did not find sketching useful was that they often do not start their analysis with a pattern in mind. Later, their intuition about what to query is derived from other visualizations that they see in the VQS, in which case it made more sense to query using those visualizations as examples directly. Similarly, while functional fitting is a common operation in scientific data analysis, querying by equation is also unpopular, since it is challenging to formulate functional forms in an prescriptive, ad-hoc manner without seeing what the common patterns in the dataset are.

While the usage of each querying feature may vary from one participant to the next, a key design principle that came from this finding was the need for visual query systems to provide visualization recommendations that can help analysts jumpstart their exploration. We found that many users made use of the representative trends and outliers visualizations provided by *zenvisage* as contextual information to better understand their data (e.g. after a filter is applied) or to query based on these recommended visualizations (e.g. find visualizations that are similar to the one in the largest representative clusters).

Recommendation facillitate smoother flow of analysis, ensures that user is never stuck or out of ideas. it does this by going towards better data understanding, accurate understanding of the context of analysis and scope of data. should not only close the loop between the two modalities of querying and exploration, but also contribute towards —- data understanding. In Section 5, we advocate the importance of building recommenders that contributes towards data understanding but also —–.

3.2 Challenges Ahead

The goal here is to help novice submit precise queries without SQL background, easy to use interface. Our study found that VQS does more than just this, but still not enough.

- Precise Search Fail to understand intricacies of user need/intent, need more expressivity/flexibility for querying.

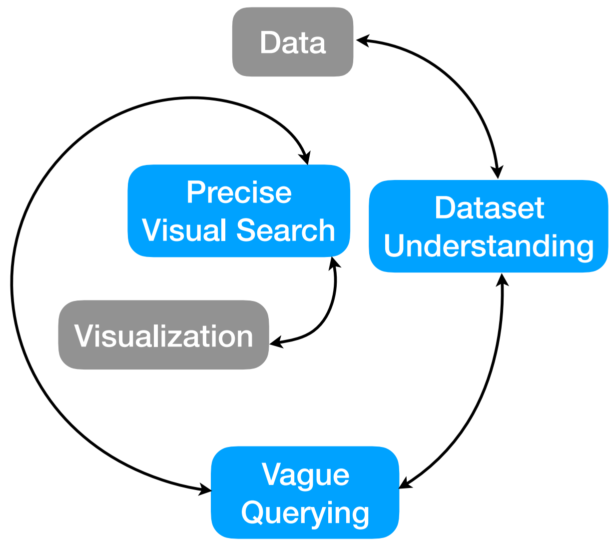

Figure 2: Cycle of visual data exploration.

- No perfect training workload, real-world data + task is noisy and complex.
- towards more holistic model for insight discovery

4 Vague Intelligent Search

Accounting for user interaction, mental models. More global objective taking into account user with the goal of dataset understanding rather than task completion.

4.1 Challenges

- Inferring user intent in querying and context is important (both in terms of user input and what is recommended)
- tools can not assume user has querying intention. exploration without intention, user don't know what they are searching for –¿ Recommendation.
- The important thing here is identifying what should be done by the system v.s. requested from user. Inappropriate choice of these will result in lack of expressibility and user feeling lack of control of analysis, limiting exploration.
- Need for a unified framework of inference to take all of these into account (e.g. natural language, etc)

4.2 Related Works

Most systems design exhibits a trade-off between how expressive can the query be and how usable the interface is. For example, while querying language (such as SQL) are highly expressive, formulating SQL queries that maps user's high-level intentions to specific query statements is challenging. Therefore, query builders have been developed to address this issue. Form-based query builders often consist of highly-usable interfaces that ask users for a specific set of information mapped onto a pre-defined query. However, form-based query builders are often based on templated queries with limited expressiveness in their linguistic and conceptual coverage, which makes it difficult for expert users to express complex queries. The extensibility of these systems or querying language also comes with the high engineering cost, as well as potentially overloading the users with too many potential options to chose from.

Given that there is no one size fit all interface for query specification for users of different expertise levels and workload, future visual data explorations systems needs to take into account a wide spectrum of queries of different input types and varying degrees of specificity that could be potentially generated from different interfaces. There is a need for vague or ambiguous specification.

As illustrated in Fig.**??**, these can range from cold-start (no supervision) to input examples, input relations to complete specification.

natural language

4.3 Towards 3'I's of rapid hypothesis generation support

Given our observations from the participatory design study, we distill several desiderata for the next generation VQSs. Towards 3'I's Interactive, Iterative, Informative (Give examples from the ZV-TVCG paper)
Integrated: should always be aware of the context of data and user
Interactive flow: (how natural is it to move between analysis steps, facilitate fluid analysis and not get "stuck") : interactivity, feedback (latter is quite unexplored), and recommendation, expressivity (how easy is it to express what to do via interactions) and diversity of actions that could be performed.
Iterative: query refinement, dialogue (not a one-shot query) Joining the flow: Section 4 focuses on the first two items .
Informative: not just task-based interestingness but more explanation-based (causality, introduce distribution awareness notion in viz-sum), focussed on data understanding, which we will discuss in Section 5

5 Towards Dataset Understanding

One of the key goals of visual data exploration is to promote a better understanding of the dataset that enables users to make actionable decisions. While our focus in the previous sections have focussed on intention-driven queries, where users have some knowleldge of what types of questions he may be interested in. This section discusses general query-free recommendations and continual provenance that helps users become more aware of the dataset with respect where they are in their analysis workflow.

Situations where there is an absence of explicit signals from the user can happen in two scenarios: 1) user is at the beginning of their analysis (commonly known as the 'cold-start' problem) and 2) user doesn't know what to query for, which is the situation derived from our *zenvisage* finding in Section 3. In this section, we will describe STORYBOARD, a system that provides data summaries and guides users through infromative subsets of data, as an example of ——. Then, we will discuss three different types of data understanding and awareness during visual data exploration and highlight the challenges ahead and opportunities for these notions of data understanding.

5.1 STORYBOARD: Navigating Through Data Slices with Hierarchical Summary of Visualizations

Common analytics tasks, such as causal inference, feature selection, and outlier detection requires studying the distributions or patterns at different levels of data granularity [1, 16, 2]. However, without knowing *what* subset of data contains an insightful distribution, manually exploring distributions from all possible data subsets can be tedious and inefficient [8, 9]. In order to explore different data subsets, a user would first have to construct a large number of visualizations corresponding to all possible data subsets, and then, navigating through this large space of visualizations to draw meaningful insights. The lack of a systematic way to perform these excercises makes the

To this end, we present STORYBOARD, an interactive visualization summarization system that automatically selects a set of visualizations to summarize the distributions within a dataset in an informative manner. Figure **??** illustrates an example dashboard generated by STORYBOARD from the Police Stop Dataset [6], which contains records of police stops that resulted in a warning, ticket, or an arrest. 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. We requested STORYBOARD to generate a dashboard of 9 bar chart visualizations 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. First, at the top of our dashboard, STORYBOARD highlights three key data subsets that results in a high arrest rate, which looks very different trend than the overall (where the majority of stops results in tickets). Following along the leftmost branch, we learn that even though in general when a search is conducted, the arrest rate is almost as high as ticketing rate, when we look at the asian population, whether a search is conducted had less influence on the arrest rate and the trend resembles more like the overall distribution.

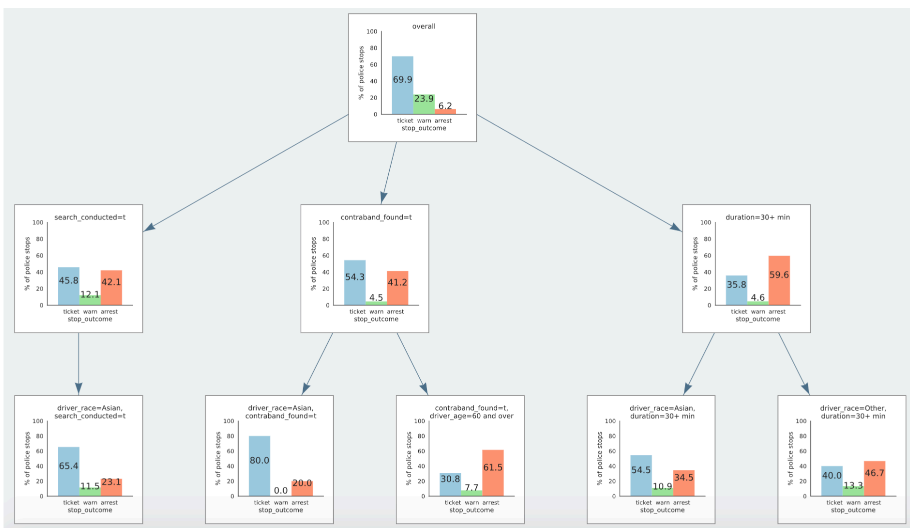

Figure 3: Example dashboard generated by STORYBOARD summarizing the key insights in the Police dataset.

While such summary dashboards are useful for making sense of relationships between data subsets, finding effective visualizations to summarize a dataset is not as trivial as picking individual visualizations that maximizes some statistical measure, such as deviation [13], coverage [10], or significance testing [1], which can often result in misleading summarizations. The key insights of our work is

For example, our insight regarding

The above example demonstrates a scenario where the selection of an improper reference (female) for comparing the visualization (black female) against results in misleading insights. In STORYBOARD, we formulate an objective where a visualization is *actually* interesting when it deviates from and can not be explained by *even* its most informative reference.

Our user study evaluations show that STORYBOARD guides users to make better predictions regarding unseen visualizations, ranking attribute importance, and retreival of interesting visualizations compared to the baselines. The effectiveness of STORYBOARD largely comes from how it helps analysts become more *distributionally aware* of the dataset. We define *distribution awareness* as the aspect of data understanding in which analysts make sense of the key distributions across different data subsets and their relationship in the context of the dataset. Even though it may be infeasible to examine all possible data subsets, with distribution awareness, the analyst will still be able to draw meaningful insights and establish correlations about related visualizations by generalizing their understanding to make predictions regarding the unseen visualizations.

How —– is underexplored future research building systems that —

5.2 Challenges Ahead

Distribution awareness highlights one example of data understanding that —–. In this section, we will discuss several other types of data understanding that is essential for effective visual data exploration. Recommendation providing better understanding for overall dataset and understanding. While the notion of distribution awareness is useful when we are looking at user understanding at a static point in time in the analysis (e.g. during cold start), we introduce two complementary notions of data understanding (contextual and situational awareness) when considering dynamic visual exploration in the context of an analytic workflow.
Contextual awareness serves to —- in a dynamic exploration situation, keep track of what filter is in play, what dataset/ schema am I looking at, which operations have been applied to the data that I'm looking at? Related to provenance for the current time. Within a dataset, structure and provenance is essential to help users navigate and provide users with sense of coverage and completion. This is an important but underexplored area. (viz-sum, Sarvghad et al 2017) Mechanisms that provides distribution awareness can effectively couple with contextual awareness in a dynamic exploration situation. For example, the representative and outlier patterns in *zenvisage* provides summaries of the current context of the data. When a dataset is filtered, the representative trends are updated accordingly. By understanding both the context (i.e. I'm only looking at data filtered with ....), the distribution awareness learn things on the context, provides an overview of typical trends for the data to be queried. better understanding of what's in the dataset that I'm looking at.
Situational awarenss: related to provenance, as a function of time. - provenance of schema and attribute understanding (coverage, etc) . Similar to situational awareness (cite Tory)

Note that while the discussion above have been focussed on how to design systems that can help facillatate these aspects of user's awareness in dataset understanding, these ideas can be generalized to principles in deisinging the types of intelligent querying systems discussed in Section 4. An intelligent visual exploration system needs to be distributionally, contextually and situationally aware, by make use of information about the data (distribution awareness), the analytic context, and situation jointly in making timely recommendations. For example, contextual awareness can inform the system that the user's current —- x,y, main visualization, while a distributionally aware system may recommend a highly-skewed data subset as interesting, a sitational aware system may realize a variable have been explored extensively in the past and recommends it accordingly. In other words, these intelligent visual query system not only needs to facillatate these aspects of data understanding, but also need to make use of this information to make inference and recommendations in an interpretable manner that can guide analysts towards meaningful stories and insights for further investigation.

inference and descisions intepretable.

, rather than the system's awareness of the user's context, situation ,etc. Ideally, an intelligent system should

related works have focussed on making specification easier, but not really trying to understnad user intent or what might the user want to see.

6 Concluding Remarks

Data is agnostic to the user, intention —, by building tools—, Section 2 to 4 have focussed on extracting what user want from data. bridging together what user want from data, what data has to offer, supporting interactive discourse between the two. Either using one-size-fits-all statistics, templates, heuristics as a solution or problem only applicable to a subset of analytic tasks[13, 14].

References

[1] Anushka Anand and Justin Talbot. Automatic Selection of Partitioning Variables for Small Multiple Displays. 2626(c), 2015.

[2] Jeffrey Heer and Ben Shneiderman. Interactive Dynamics for Visual Analysis. *ACM Queue*, 10(2):30, 2012.

[3] H. V. Jagadish, Adriane Chapman, Aaron Elkiss, Magesh Jayapandian, Yunyao Li, Arnab Nandi, and Cong Yu. Making database systems usable. In *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data*, SIGMOD '07, pages 13–24, New York, NY, USA, 2007. ACM.

[4] Doris Jung-Lin Lee, John Lee, Tarique Siddiqui, Jaewoo Kim, Karrie Karahalios, and Aditya G. Parameswaran. Accelerating scientific data exploration via visual query systems. *CoRR*, abs/1710.00763, 2017.

[5] Kristi Morton, Magdalena Balazinska, Dan Grossman, and Jock Mackinlay. Support the Data Enthusiast: Challenges for Next-Generation Data-Analysis Systems. *Proceedings of the VLDB Endowment, Volume 7, pp. 453–456, 2014*, 7:453–456, 2014.

[6] E. Pierson, C. Simoiu, J. Overgoor, S. Corbett-Davies, V. Ramachandran, C. Phillips, and S. Goel. A large-scale analysis of racial disparities in police stops across the united states, 2017.

[7] Peter Pirolli and Stuart Card. The Sensemaking Process and Leverage Points for Analyst Technology as Identified Through Cognitive Task Analysis. A Notional Model of Analyst Sensemaking.

[8] S Sarawagi, R Agrawal, N Megiddo, Generalitat Valenciana Ajuntament Valencia Univ Politecn Valencia, and E T H Zurich Oracle Sybase Softlab Iberia Edbt Fdn. Discovery-driven exploration of OLAP data cubes. *6th International Conference on Extending Database Technology (EDBT 98)*, pages 168–182, 1998.

[9] S. Sarawagi. User-adaptive exploration of multidimensional data. *Proc of the 26th Intl Conference on Very Large*, pages 307–316, 2000.

[10] Ali Sarvghad, Melanie Tory, and Narges Mahyar. Visualizing Dimension Coverage to Support Exploratory Analysis. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):21–30, 2017.

[11] Tarique Siddiqui, Albert Kim, John Lee, Karrie Karahalios, and Aditya Parameswaran. Effortless Data Exploration with zenvisage: An Expressive and Interactive Visual Analytics System.

[12] C. Stolte, D. Tang, and P. Hanrahan. Polaris: a system for query, analysis, and visualization of\nmultidimensional relational databases. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):1–14, 2002.

[13] Manasi Vartak, Samuel Madden, and Aditya N Parmeswaran. SEEDB : Supporting Visual Analytics with Data-Driven Recommendations. 2015.

[14] Manasi Vartak, Silu Huang, Tarique Siddiqui, Samuel Madden, and Aditya Parameswaran. Towards Visualization Recommendation Systems. *ACM SIGMOD Record*, 45(4):34–39, 2017.

[15] Leland Wilkinson. *The Grammar of Graphics (Statistics and Computing)*. Springer-Verlag, Berlin, Heidelberg, 2005.

[16] Eugene Wu and Samuel Madden. Scorpion: Explaining Away Outliers in Aggregate Queries. *Proceedings of the VLDB Endowment*, 6(8):553–564, 2013.