Safe Visual Data Exploration

Zheguang Zhao Emanuel Zgraggen Lorenzo De Stefani Carsten Binnig
Eli Upfal Tim Kraska
Department of Computer Science, Brown University
{firstname_lastname}@brown.edu

ABSTRACT

Exploring data via visualization has become a popular way to understand complex data. Features or patterns in visualization can be perceived as relevant insights by users, even though they may actually arise from random noise. Moreover, interactive data exploration and visualization recommendation tools can examine a large number of observations, and therefore result in further increasing chance of spurious insights. Thus without proper statistical control, the risk of false discovery renders visual data exploration unsafe and makes users susceptible to questionable inference. To address these problems, we present QUDE, a visual data exploration system that interacts with users to formulate hypotheses based on visualizations and provides interactive control of false discoveries.

1 Introduction

In the era of Big Data, interactive data exploration tools arise as an important mean to explore and derive insights from data through visualization. However, perceived interesting patterns in visual data representations, such as relationships or trends, may emerge from irrelevant random effects inherent to the data, such as random noises, large variances, insufficient samples, and biases. Without proper statistical control, users may mistake a distinct or dominant visual observation as statistically significant. On the other hand, systems that search and recommend visualizations automatically based on such interesting visual features further increase the chance of bogus insights. A recent study highlights these issues and shows that visualization and recommendation systems that do not consider the risk of false discovery, such as Vizdom [4], SeeDB [12] and Data Polygamy [3], become difficult to derive insights safely on real-world datasets [1].

False discovery due to random noise is pervasive in visual data exploration on real-world datasets. For example, when using Vizdom [4] to explore a recently conducted survey on personal habits and opinions [1], we observed that the preference on watching films on DVD produced visually different proportions of belief in aliens, as shown in Figure 1 (A and B). Just by visually examining these charts, users often falsely assumed that people who prefer to watch movies on DVD are more prone to believe in aliens even though this effect is not statistically significant.

SIGMOD'17, May 14 - 19, 2017, Chicago, IL, USA

ISBN 978-1-4503-4197-4/17/05…$15.00

DOI: http://dx.doi.org/10.1145/3035918.3058749

Such observation-based hypotheses may be accumulated quickly as the user continues to explore a dataset, and hence dramatically raises the risk of spurious findings. With the same survey dataset, after searching through a few different comparisons we stumbled upon a visualization that suggests hair color predicts whether one knows about Michael Stonebraker, and it would be statistically significant if considered as a lone hypothesis. This phenomenon is often referred to as data dredging or p-hacking [7], and formally known as the multiple comparison problem [10].

Several challenges exist to control false insights in such interactive data exploration. The first question is what could even be considered as a hypothesis in visual data exploration. In some cases users might explicitly specify certain hypotheses, but in other instances they do not formulate any hypothesis but still use visualization to infer about the data or to extract an insight. An ideal system should assist users in hypothesis formulation and corresponding hypothesis test selection.

Second, interactive data exploration mandates that hypotheses are formulated dynamically based on the process of human decision making. However the classical statistical procedures such as Bonferroni [2] and Sequential FDR [5] are not dynamic as they require collecting all the hypotheses a priori before finalizing any significance result. Moreover these traditional techniques assume complete pass of the dataset, and thus would make the system non-interactive on larger data. Thus an ideal system for interactive control of false discovery should follow progressive computation, which proves to be a more appealing paradigm [4, 8, 13].

Finally, parts of the data exploration process can be automated through algorithmic search and recommendation. Such automatic data exploration should also be subject to false discovery control.

In this demo, we present QUDE[1], a first "safe" visual data exploration tool that addresses these challenges. In QUDE, we implemented recent techniques in interactive false discovery control that are both dynamic and progressive [14], apply them to visual data exploration and visualization recommendation, and expose them through a pen- & touch-based user interface that simplifies and in some cases even automates the creation of hypothesis tests.

2 Controlling False Discoveries

The theory of controlling false discovery in interactive data exploration is introduced in [14]. Data exploration is modeled as a growing sequence of observations. Three different exploration settings are studied, namely, the *targeted exploration*, the *free-form exploration*, and the *uniform exploration* [14]. In targeted exploration, the user focuses on a predefined set of questions around narrow topic, such as "understanding what affects the salary distribution." The earlier discoveries are thus used to build up the

[1] QUDE stands for Quantifying the Uncertainty in Data Exploration.

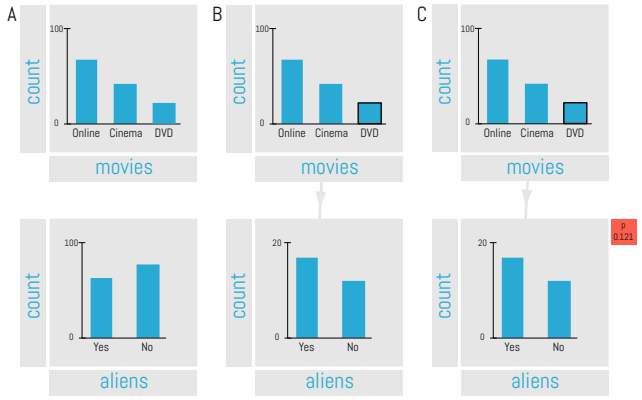

Figure 1: Example of a visualization network where users might be led to false discoveries without automatic hypothesis formulation. (A) two separate visualizations showing preferences for watching movies and how many people believe in alien existence; (B) the two visualizations combined where the bottom one shows proportions of belief in alien existence for only people who like to watch movies on DVD, displaying a noticeable difference compared to the overall population. (C) same visualizations as before but now with automatic hypothesis formulation turned on, highlighting that the observed effect is not statistically significant.

subsequent ones. Many data-driven scientific studies fall into this category. On the other hand, the user may not start with a focus, but may be interested in developing a focus as more data are explored. Such process is prevalent in interactive data exploration. Finally, when data exploration is less structured, such as in the case of recommendation engines, significant insights can be modeled as uniformly distributed in the process.

A procedure for false discovery control evaluates each observation by a corresponding hypothesis test, which outputs a p-value . A control procedure such as β-Farsighted [14] starts with a predefined amount of exploration *budget*, and invests a fraction of the budget as the significance level for each test. If the observation is deemed significant, or equivalently, the test is rejected, then a fraction of the investment is returned to the budget. The procedure halts when either there are no more tests or the budget reduces to zero. The guarantee for the entire data exploration is that the *marginalized false discovery rate* (mFDR), namely, the ratio between the expected number of false discoveries and that of all discoveries, is no greater than the fraction set as the initial exploration budget.

Contextual information is useful for the quality of false discovery control. If the number of observations the exploration is unknown or unbounded, the best strategy to use is β-Farsighted [14], which always conserves a proportion of the current exploration budget. Otherwise if the expected exploration length can be approximated, then ϵ-Hybrid [14] offers similar power but lower false discovery rate. These procedures are most efficient in terms of power and error rates for the aforementioned exploration settings [14].

3 Design

The system, QUDE, is based on Vizdom [4] and addresses the aforementioned challenges, namely,

- To formulate hypotheses via user interaction;
- To visualize the statistical significance and other contextual information for each observation;
- To control multiple hypotheses dynamically in exploration;
- To progressively compute the risk of false discovery.

Figure 2: User interface design showing QUDE's "risk-gauge" on the right which keeps track of all hypotheses and provides details for each of them.

3.1 User Interface

QUDE's user interface (UI) features an unbounded 2D canvas where visualizations can be laid out in a free form fashion (Figure 2). It is based on our visual data exploration tool Vizdom [4] but extended by three new features:

Explicit and implicit hypothesis formulation: For cases where users know which effect they want to statistically verify, we included support to explicitly create hypotheses through a gestural UI that poses minimal overhead to users. Additionally however, we found that in many cases users do not deliberately think about chains of visualizations as tests even though users gain insights from them. For these instances, that we call implicit hypotheses, we augmented our system to formulate tests and display test results automatically. The class of tests we formulate automatically includes comparisons of subsets against the global population of a dataset such as the one illustrated in Figure 1. The example shows a visualization chain where without an implicit hypothesis test (C), users might wrongly observe and perceive a significant effect (difference in bottom visualization between A and B).

Visualization recommendations: To speed up the potentially laborious process of manually exploring a dataset we added a visualization recommendation engine with false discovery control (Section 3.2). Similar to SeeDB [12] our *recommender* allows users to search for filter conditions that have a significant effect (positive or negative) on a given reference visualization. We expose this functionality though a gestural touch UI that can be accessed from any visualization.

Hypotheses tracking: A "risk-gauge" on the right-hand side of the display (A) serves two purposes, namely, to give user a summary of the multiple hypothesis correction procedure (e.g., in this case α-investing is used with a false discovery rate of 5% and with current remaining budget of 70%), and to provide access to a scrollable list of all the hypotheses that have been explored. Each list entry can be expanded (in the example all are expanded) to display details about an observation and its statistical significance. The text labels describe the null and alternative hypotheses for each observation and the corresponding hypothesis test and p-value . Each color coded tile indicates whether the observation is statistically significant or insignificant, which corresponds to green or red respectively. The distributions of null and alternative hypotheses and the color coded effect size are also visualized (C). To help the user understand the effect of data collection, the sample size estimate for the current significance level is displayed for each hypothesis test assuming the effect size is fixed (B). For example, the five green squares in

(B) indicates approximately five times the current data size with the same effect size would make this observation significant. Finally, important insights can be marked by tapping the "star" icons (D).

3.2 Backend

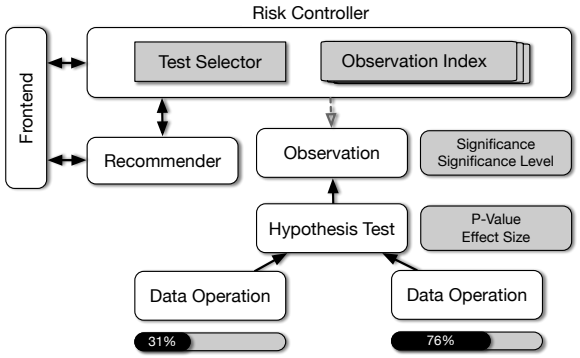

Figure 3: QUDE's backend. Uncertainties in data discovery from the user or the recommendation algorithms are automatically quantified and controlled. Hypothesis tests are chosen automatically for usability. Observations are tracked by the observation index to provide consistent false discovery control. The statistical computation is modeled as online aggregation for interactivity and scalability.

Several challenges exist in the design of QUDE backend. First, to free the user from the burden of choosing appropriate hypothesis tests, we design QUDE to automatically select the appropriate testing procedures based on iput observations. The user only needs to specify via pen-&-touch gestures what she observes, such as "the salary distribution looks more skewed towards the higher end among male than female." As in Figure 3, the *risk controller* at the backend receives and decodes the observation as a corresponding null and an alternative hypothesis. Observations can also be created by the backend recommendation algorithms. The risk controller implements the false discovery control procedures as introduced in [14], which bootstraps with an initial exploration budget and invests a fraction on each new observation. The *test selector* then chooses the appropriate test based on different criteria on the hypothesis and the data characteristics. For example, to compare two means being different, a two-sided t-test is chosen; χ^2-test is used to compare two histograms, but only when the frequencies are large enough; whereas for smaller frequencies, a permutation test is used.

The second challenge is to correctly manage the past observations to provide consistent false discovery control over the time and data. This not only has implication on performance but also on correctness for supporting multiple users and the QUDE internal recommendation engine, because repeated observations on the same data should *not* be tested as different steps within a false discovery procedure [14]. As in Figure 3, the *observation index* tracks the past observations to assist the risk controller on the global decisions. A loose analogy of the observation index is perhaps a version control system, such as Git [11], but in this case for observations. A side effect of such design is that it allows computing different observations in parallel, yet meanwhile enforces a linear ordering for the false discovery control procedures.

Finally, the risk control component should not incur significant overhead to impeded the interactivity of the system and the user productivity, and should scale to large datasets. However most statistical computing tools such as R[9] and MATLAB[6] only support batch-oriented functions, so the time complexity of any derived implementation of hypothesis testing grows at least linearly in the data size. On the other hand, computation that outputs approximate answers with online refinement proves to be most effective for user productivity [13]. Thus to scale to large datasets while maintaining interactivity, we take a different approach by designing the statistical procedures via online aggregation [8]. In particular, the approximate hypothesis testing result is computed based on an expanding subset of the data, and output a stream of updated results to the frontend within interactive response time. The result is incrementally updated as more data is scanned, as in Figure 3.

4 Demo Proposal

To demonstrate QUDE, we use various publicly available datasets and a dataset obtained via a survey on Amazon Mechanical Turk [1]. This survey collected answers to 69 questions from 104 participants. Questions cover a wide range of habits and opinions and are mostly unrelated conceptually, such as "Do you believe in aliens?", "What is your eye color?" and "How tall are you?".

Figure 4 shows an example storyboard of a user exploring the aforementioned survey dataset through QUDE. A user, Eve, starts out by looking at two attributes she is interested in: age and height (A top). She wants to see if there is a correlation between the two and thus creates a third visualization where she plots them against each other (A bottom). Just from visual inspection, age and height do not seem to be correlated. However, Eve wants to be sure and thus explicitly creates a hypothesis test. She uses a multitouch gesture (dragging the two visualization close to each other) and QUDE automatically picks and computes an appropriate test (a correlation test in this case) (B). This confirms Eve's intuition that the two attributes are not correlated. Eve continues to look at the answers to the question "Do you know what SQL is?". It looks like a bit more than half the people answered this question with yes (C left). Our user wants to find out what other attributes are good predictors if someone knows SQL. Her first hunch is to look at age. She creates a query that allows her to filter the SQL attribute by people who are over 50 years old. For this age group the y-axis ordering of the two bars switched: less people in this age group know of SQL (C right). Visually it seems that this is quite a big effect, however QUDE automatically executed a hypothesis test for this comparison that tells Eve this is in fact not statistically significant (C, red block on the right-hand side). Eve goes on to do a similar query. This time checking if people who know who Mike Stonebraker is have a higher chance of knowing SQL. The hypothesis test automatically computed by QUDE reinforces what Eve sees: this is a significant effect.

Eve realizes this manual exploration is becoming fairly laborious and decides to continue by using the automatic visualization recommender that QUDE provides. By tapping on the SQL attribute visualization she gets a handle to invoke a search for recommendation visualization. The handle shows that she has 70% of exploration budget left (E). The visualisation recommendation system works through touch gestures. Dragging away from the handle invokes it, whereby the angle and the length of the drag-path determine the amount of exploration budget that should be spent and if QUDE should look for similar or dissimilar visualization (+ and - signs on the handle) (F and G). Eve chooses to spend 20% of her budget and to look for dissimilar visualizations (G). QUDE progressively computes and ranks recommendations until the allotted budget is used up and presents thumbnails of results (H). Eve can press-and-hold on thumbnails to preview the result. Overlaid in purple she sees that people who know programming have a higher chance of also knowing SQL (I). She wants to see more detail about that relation and drags the thumbnail out which in turn creates a filter chain similar to the one she created manually before.

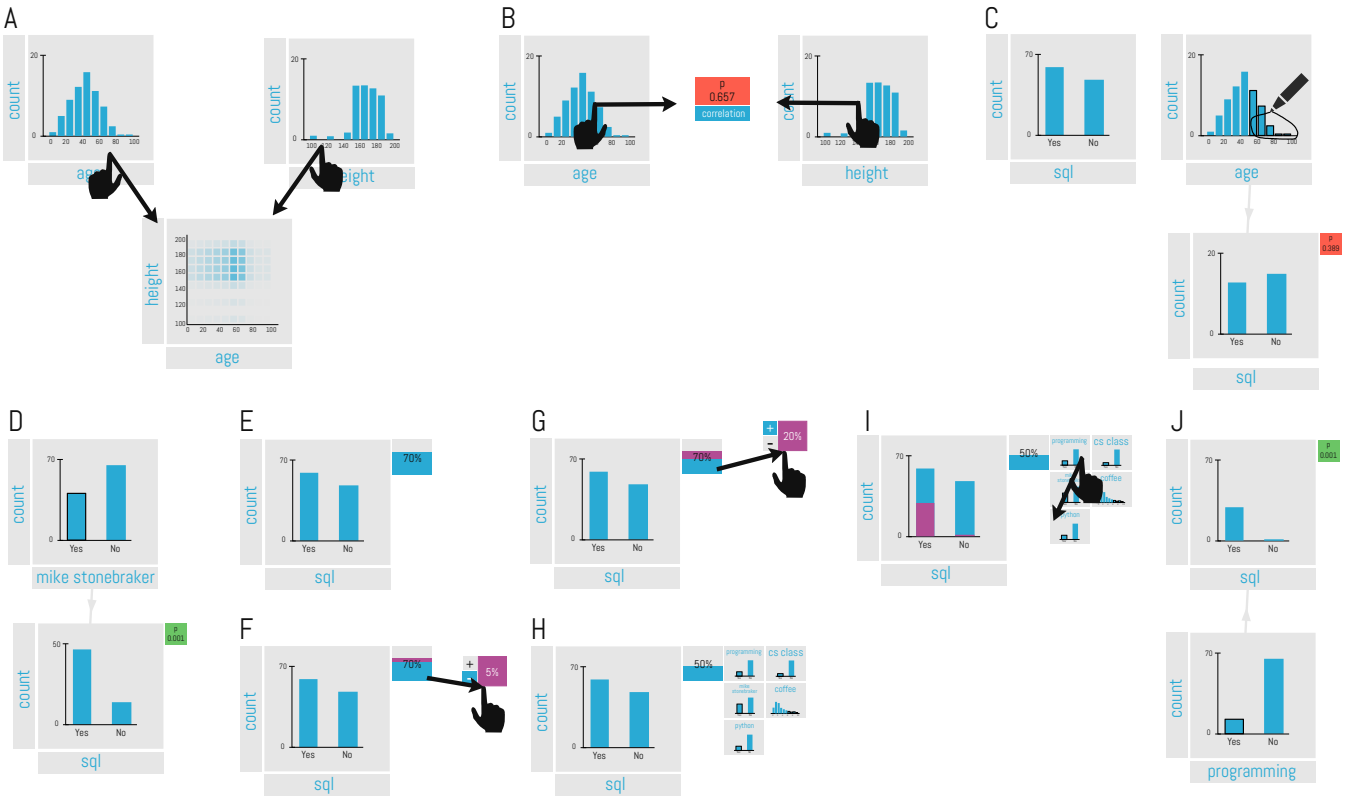

Figure 4: Storyboard showing a user exploring a dataset through QUDE.

All hypothesis test results are automatically tracked and subject to multiple hypothesis correction by QUDE and displayed in a scrollable list through which Eve can, at any time, inspect and obtain additional information about any hypothesis (Figure 2).

5 Conclusion

Visual data exploration and recommendation systems allow users to examine large number of observations either manually or automatically. If not treated as hypotheses and correctly controlled, visually significant results might be mistaken as statistically significant. Recent advance in multiple comparison problem and control procedures adds the desirable property of interactivity to false discovery control for data exploration [14]. QUDE implements these interactive procedures and exposes them through a pen-&-touch UI that allows for explicit and implicit formulation of hypotheses and user-driven steering of visualization recommendation.

6 Acknowledgments

This research is funded in part by the Intel Science and Technology Center for Big Data, DARPA Award 16-43-D3M-FP-040, NSF CAREER Award IIS-1453171, NSF Award IIS-1514491, NSF Award IIS-1562657, Air Force YIP AWARD FA9550-15-1-0144, and gifts from Google, VMware, Mellanox, and Oracle.

7 References

[1] C. Binnig, L. D. Stefani, T. Kraska, E. Upfal, E. Zgraggen, and Z. Zhao. Toward sustainable insights, or why polygamy is bad for you. In *CIDR 2017, 8th Biennial Conference on Innovative Data Systems Research, Chaminade, CA, USA, January 8-11, 2017, Online Proceedings*, 2017.

[2] C. E. Bonferroni. *Teoria statistica delle classi e calcolo delle probabilita*. Libreria internazionale Seeber, 1936.

[3] F. Chirigati et al. Data polygamy: The many-many relationships among urban spatio-temporal data sets. In *SIGMOD*, 2016.

[4] A. Crotty et al. Vizdom: Interactive analytics through pen and touch. *PVLDB*, 8(12), 2015.

[5] M. G. G'Sell, S. Wager, A. Chouldechova, and R. Tibshirani. Sequential selection procedures and false discovery rate control. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(2):423–444, 2016.

[6] M. U. Guide. The mathworks inc. *Natick, MA*, 4:382, 1998.

[7] M. L. Head, L. Holman, R. Lanfear, A. T. Kahn, and M. D. Jennions. The extent and consequences of p-hacking in science. *PLoS Biol*, 13(3):e1002106, 2015.

[8] J. M. Hellerstein et al. Online Aggregation. In *SIGMOD*, pages 171–182, 1997.

[9] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2016.

[10] J. P. Shaffer. Multiple hypothesis testing. *Annual review of psychology*, 46, 1995.

[11] L. Torvalds and J. Hamano. Git: Fast version control system. *URL http://git-scm. com*, 2010.

[12] M. Vartak et al. SEEDB: efficient data-driven visualization recommendations to support visual analytics. *PVLDB*, 8(13), 2015.

[13] E. Zgraggen, A. Galakatos, A. Crotty, J.-D. Fekete, and T. Kraska. How progressive visualizations affect exploratory analysis. *IEEE Transactions on Visualization and Computer Graphics*, 2016.

[14] Z. Zhao, L. De Stefani, E. Zgraggen, C. Binnig, E. Upfal, and T. Kraska. Controlling false discoveries during interactive data exploration. *arXiv preprint arXiv:1612.01040*, 2016.