Exploring the "Double-Edged Sword" Effect of Auto-Insight Recommendation in Exploratory Data Analysis

Quan Li[a], Huanbin Lin[b], Chunfeng Tang[c], Xiguang Wei[d], Zhenhui Peng[e], Xiaojuan Ma[f] and Tianjian Chen[g]

[a]School of Information Science and Technology, ShanghaiTech University
[b]AI Group, WeBank
[c]AI Group, WeBank
[d]AI Department, Shenzhen Semacare Medical Technology Co.
[e]Department of Computer Science and Engineering, The Hong Kong University of Science and Technology
[f]Department of Computer Science and Engineering, The Hong Kong University of Science and Technology
[g]Department of Computer Science and Engineering, The Hong Kong University of Science and Technology

Abstract

Modern data analytics tools often provide visualizations as an accessible data window to users in exploratory data analysis (EDA). Still, many analysts feel lost in this process due to issues such as the high complexity of data. Auto-insight recommendations offer a promising alternative by suggesting possible interpretations of the data to users during EDA but might impose undesirable effects on users. In this study, we systematically explore the "double-edged sword" effect of auto-insight recommendations on EDA in terms of exploration assistance, message reliability, and interference. Particularly, we design and develop two versions of a Tableau-like visualization system termed *TurboVis*: one supports auto-insight recommendations while the other does not. We first demonstrate how typical visualization specification tools can be augmented by incorporating auto-insight recommendations and then conduct a within-subjects user study with 18 participants during which they experience both versions in EDA tasks. We find that although auto-insight recommendations encourage more visualization inspections, they also introduce biases to data exploration. The perceived level of message reliability and interference of auto-insight recommendations depend on data familiarity and task structures. Our work elicits design implications for embedding auto-insight recommendations into the EDA process.

Keywords

Visualization recommendations, exploratory data analysis, auto-insight

1. Introduction

Exploratory Data Analysis (EDA) refers to the critical process of performing initial investigations on data to discover patterns, spot anomalies, test hypotheses, and check assumptions with the help of summary statistics and graphical representations [1]. For example, financial analysts conduct EDA to identify the main trend of enterprise business metrics, discover data outliers to locate potential problems that need attention or even action, and form further hypotheses for more in-depth data explorations. The outcomes of EDA processes are data insights that are often integrated into a visual dashboard [2, 3]. In a sense, EDA is kind of a creative process [4], during which users leverage their knowledge and intuitions to inquire into the data and scaffold a graphical representation for insight interpretation and communication. In an ideal setting, analysts are supposed to immerse themselves into such a creative process without breaking the flow. However, in reality, analysts are often stuck with questions such as "*where should I start?*" and "*what else can I find?*" [5]. These issues could slow down the process and yield fewer meaningful EDA outcomes.

To support analysts to gain insights from the data, previous data analytics tools have proposed auto-insight recommendation services that suggest potential interesting visual patterns [6] or simulate exploration hunches [7]. In other words, as users visualize data, these tools could simultaneously conduct analysis, and recommend trends/patterns termed auto-insight recommendations [8]. For example, Mackinlay et al. proposed *Show Me* [9] which supports users to search for graphical presentations when analyzing data. However, current auto-insight recommendation techniques such as *Voyager 2* [10] and *Foresight* [11] primarily focus on the design of insight discovery algorithm or perceptually effective insight presentations without deep considerations on the representation of the intended goal of the recommendations, ease of understanding and contexts, and user prefer-

Joint Proceedings of the ACM IUI 2021 Workshops, April 13-17, 2021, College Station, USA
qliba@connect.ust.hk (Q. Li); bindylin@webank.com (H. Lin); jsontang@webank.com (C. Tang); marcus@semacare.com (X. Wei); zpengab@connect.ust.hk (Z. Peng); mxj@cse.ust.hk (X. Ma); tobychen@webank.com (T. Chen)
0000-0003-2249-0728 (Q. Li); 0000-0002-5700-3136 (Z. Peng); 0000-0002-9847-7784 (X. Ma)

ences. Consequently, the resulting auto-insight recommenders may introduce the following side effects to the EDA process. ① **Bias**. Prior studies in the field of recommender systems indicate that without a clear representation of the intended goal of the recommendations, elaborately designed recommendation algorithms have the potential to limit exploration breadth as users may unconsciously confine their explorations to the items recommended [12, 13, 14]. Systems such as *Foresight* and *Voyager 2* uncover visual insights independent of the EDA pipeline and represent visual insights in an implicit way [10, 11]. An anecdotal evidence shows that, if such auxiliary findings generated by these systems and the intended goal of these insights are not explicitly represented and mentioned, users still have little clue as to how the augmented information can be pieced into the final story and where it leads them in EDA [10]. Consequently, it brings bias to analysts in interpreting and exploiting the recommended visual insights for EDA. ② **Reliability.** Prior research in recommender systems suggested that recommendation service is context-specific and should improve its readability in order to make the results more reliable to users [15]. For one thing, previous studies illustrate that the perceived utility of recommendations is context-specific, i.e., with a limited knowledge of users, intelligent systems will be less competent in offering recommendation services [12, 16, 17]. For another, as we may not know the system users at all, "*some automatically recommending insights are still like a baffling mystery to laymen*" [18]. If the auto-insight recommendation service fails to present its suggestions in an easy-to-understand manner, the insights' reliability drops. ③ **Interruption.** Given that the structure of an EDA process could be anywhere from fully open exploration to target-oriented inspection, automatically recommending insights of the data may bring interruption to the EDA process when users prefer open exploration or examination on the non-suggested data aspects [14, 19, 20]. Auto-insight recommendation service could thus be a "double-edged sword" in EDA, and it would impair the analysis experience and results if it is designed inappropriately. Hence, to design an effective auto-insight recommendation service to support EDA, we need to first identify what are required and concerned in a smooth EDA pipeline, and then we need to systematically explore the potential "double-edged sword" effect of auto-insight recommendations on the EDA process and outcome.

To this end, we design *TurboVis*, a Tableau-like visualization system that supports analysts in EDA activities with features including auto-insight recommendation based on an extensible repository of statistic metrics, graphics matching, manual visualization specification, and dashboard editing and interaction. To evaluate how auto-insight recommendation could positively or negatively affect the EDA process and outcome, we create a simpler version of *TurboVis* that only has the later three features but no suggestions from the system. Our within-subjects (*TurboVis* with vs. without auto-insight recommendation) study with 18 industrial business analysts shows that the auto-insight design makes them inspect more visualizations but introduces bias to the direction of exploration. Auto-insight recommendations offer new perspectives when analysts are not familiar with the data or have a vague idea about how to proceed, and may distract analysts when they are facing a familiar dataset or usage scenario. Meanwhile, auto-insight recommendations would interrupt analysts more when they have specific target for exploration in mind than when they are completely open-minded. Based on these results, we further elicit design implications for embedding auto-insight recommendations into the EDA process.

2. Related Work

Literature that overlaps with this work can be classified into three categories: exploratory data analysis, visualization recommendations, and recommender system.

Exploratory Data Analysis. EDA is a term coined by John W. Tukey for describing the act of "*looking at data to see what it seems to say*" [21]. In EDA, attempts are made to identify the major features of a dataset of interest and to generate ideas for further investigation. Particularly, Tukey drew an analogy between EDA and a series of detective work, during which analysts form a set of hypotheses by asking questions, and integrate their domain knowledge to obtain rich data insights [22, 21]. Data visualization is perhaps the most widely used tool in the EDA process. With the rise of interest in data science and the need to derive value from data, analysts increasingly leverage visualization tools to conduct exploratory data analysis, spot data anomalies, and correlations, and identify patterns and trends [23, 24]. The state of the art in data visualization involves a lot of manual generation of visualizations through tools such as *Excel*, *Tableau* [25] and *Qlik* [26] to facilitate the EDA process for non-expert analysts. However, it is still challenging for data visualization novices to rapidly construct visualizations during the EDA process. Grammel et al. [27] conducted an exploratory laboratory study in which data visualization novices explored fictitious sales data by communicating visualizations to a human mediator, who rapidly constructed the visualizations using commercial visualization software. Apart from identifying activities that are central to the iterative visualization construction process, they also found that the major barriers faced by the participants are translating questions into data attributes, designing visual mappings, and interpreting the visualizations. In this study, we explore the role of auto-insight recommendation in the EDA process.

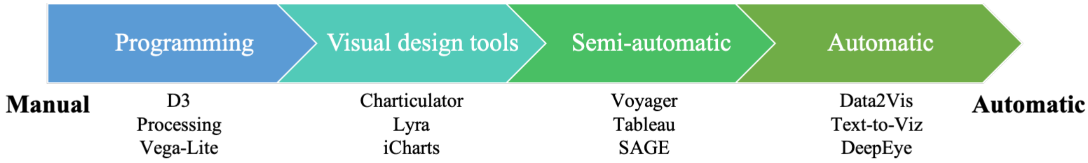

Figure 1: Visualizations are generated with a spectrum of tools automatically or manually.

Visualization Recommendations. As the demand for rapid analysis for visualization grows, there is an increasing requirement to design visualization tools allowing users to efficiently generate visualizations. Prior studies show that the relevant authoring tools are increasingly towards automatic [28], which can be classified into four categories. Initially, users have to manually write codes for visualizing data by using imperative languages and libraries such as *D3* [29], *Vega-Lite* [30], and *ECharts* [31], which are designed for users who are familiar with coding and visualizations. Later, researchers contributed visual building frameworks for easy visualizations including template editing [32, 33], shelf configuration [34], and visual building [35, 36]. These tools are designed for users who can write codes but not familiar with visualizations. Particularly, users need to "*pre-conceive blueprints, then interact with the system*" [28] to obtain more expressive, appropriate and aesthetic visualizations. Then, semi-automatic methods involved with few interactions were proposed for efficiently obtaining visualizations like *SAGE* [37] and *Tableau* [25]. Fully automatic methods are designed for no-human-involved tools for efficiently obtaining visualization recommendations such as *Text-to-Viz* [38], *Click2Annotate* [39], *Data2Vis* [40], and *DeepEye* [41]. These tools resolve the issues when users are not familiar with either visualizations or coding. On the other hand, to resolve the issues that analysts often have no idea what they are looking for especially at the initial stage of data exploration, researchers have developed various algorithms and systems to recommend insightful visualizations that can depict data trends and patterns [41, 42, 43]. In this study, we combine a semi-automatic method that involves user interactions along with algorithms that can recommend interesting insights on the basis of extensible metric repository. Therefore, analysts can benefit from a quick launch of data exploration from automated recommendations of potentially interesting data patterns.

The existing studies mainly focus on employing either machine learning algorithms or user-defined rules and visual embellishments into the creation of infographics to lower the barrier for data exploration by automatically generating visualizations. However, indicators from previous studies also point out that recommendations may potentially hamper users during data exploration [44, 45]. Inspired by the previous findings and to obtain a systematic understanding of how the auto-insight recommendation systems might pose a hindrance to the exploratory data analysis process, we design and develop two versions of a Tableau-like visualization tool and attempt to explore the "double-edged sword" effect of auto-insight recommendations on EDA in terms of exploration assistance, message reliability, and interference.

Recommender System. Recommender systems collect their target users' preferences for a set of items such as movies, songs, books, and travel destinations. They leverage different sources of information for providing users with predictions and recommendations of items [46]. With the ever-growing volume of online information, recommender systems have been an effective strategy to overcome such information overload [47], particularly useful when users do not have sufficient experience to make a choice from a large number of alternatives [48]. Existing research in the field of recommender systems mainly focus on the recommendation accuracy and the explainability of recommendation algorithms [46, 47, 49, 50, 51], which inevitably result in that people are increasingly relying on recommender systems that employ algorithmic content curation to organize, select and present information [12, 52, 53]. Despite its wide utility, researchers have indicated that its potential impact to improve problems related to over-choice should be concerned [14]. Therefore, inspired by the studies on the potential harm of such recommendations, we believe that the potential "double-edged sword" effect of auto-insight recommendations warrant a separate study.

3. Research Questions

The literature suggested that visualization recommendations encourage users to explore more visualizations [10], while prior studies in the field of recommender systems indicated that recommendations would introduce exploration bias, i.e., users might confine their exploration to the recommended items [13, 54]. In other words, recommendations can encourage exploration breath but may introduce a lack of breadth diversity. We posit that with the auto-insight recommendations, target users may explore more visualizations but will be biased to the visu-

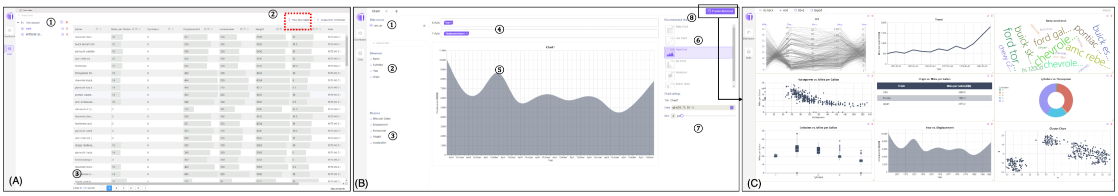

Figure 2: (A) Data processing module of *TurboVis*: ① Data menu; ② Entry to auto-insight recommendation and interactive visual analysis modules; ③ Data table that provides necessary processing functions. (B) Interactive visual analysis module of *TurboVis*: ① Data; ② Dimensions; ③ Metrics; ④ *x* and *y* axis; ⑤ Display area of visualization; ⑥ Visualization recommendations; ⑦ Chart configuration area. ⑧ (C) Dashboard editing and export module of *TurboVis*.

alizations that are only supported by the recommended items. Therefore, we have our first research question: **RQ①** ***How do analysts utilize auto-insight recommendations and manual specification of visualizations collectively as a latent impetus in their EDA process?***

Previous studies in the field of recommendations indicate that the perceived reliability of recommendations is context-specific [12, 15, 16, 17, 54]. For example, when users are exploring an unfamiliar dataset, recommendation services should improve its readability to make the results more reliable to users. Therefore, we posit a similar effect of auto-insight recommendations need to verify it in the second research question: **RQ②** ***How does the perceived reliability of auto-insight recommendations depend on the context of the EDA process such as dataset familiarity?***

Analysts may hold different purposes when conducting EDA. Previous studies indicate that the degree of applying auto-insight recommendation may vary in different EDA scenarios [55]. Given that the structure of an EDA process could be anywhere between a fully open exploration and a specific target-oriented inspection, we want to know **RQ③** ***Would auto-insight recommendations act differently due to different exploration purposes in EDA?***

4. TurboVis

To understand how auto-insight recommendations could be leveraged to assist analysts' EDA processes and explore its potential "double-edged sword" effect, we design and develop *TurboVis*, an auto-insight recommendation-powered exploratory data analysis system. To enhance the generalizability of our findings to common data analytics tools, we design *TurboVis* by reference to existing commercial exploratory data analytics software and tools such as *Tableau*. *TurboVis* serves as instruments "for design and understand" and are not intended to suggest new interaction techniques [56]. Through a prolonged collaborative design process with data analysts, we iteratively refine the system through a series of informal usability testings. To be specific, *TurboVis* consists of four main modules, namely, the *data processing*, *auto-insight recommendation*, *interactive visual analysis*, and *dashboard editing and export* modules. Particularly, the *data processing* module (Figure 2(A)) handles common data formats, obtains and analyzes data fields, and provides necessary processing functions such as sorting, filtering, and editing. The *auto-insight recommendation* module serves as assistance to inspect potentially interesting data patterns including trending, correlation, distribution, clustering, and outlier detection based on an extensible metric repository (covered later). Particularly, data auto-insight recommendation is embedded in a typical EDA process, recommending interesting visualization and enables automatically modifying users' visualization specifications to achieve the desired visualizations. The *interactive visual analysis* module supports analysts to perform a drag-and-drop interaction on-demand to manually specify visualizations (Figure 2(B)) . The *dashboard editing and export* module (Figure 2(C)) allows analysts to interactively edit each data insight visualization by "linking + view" technique. The finalized dashboard can be exported on demand.

TurboVis without Auto-Insight Recommendation. *TurboVis* without auto-insight recommendation version only supports manual visualization specification on the basis of graph matching. To be specific, as shown in Figure 2(B), after loading the data (①), *TurboVis* automatically splits data fields into dimensions (②) and metrics (③). Analysts can perform a drag-and-drop interaction to drag any attribute(s) onto the x- or y-axis (④) and the display area (⑤) would simultaneously present the default chart ranking the first in the recommendation list (⑥). We do not explicitly provide other visual encoding channels such as size and color due to the observation that participants often get lost to determining where each selected attribute goes. Therefore, *TurboVis* automatically determines an appropriate visual encoding channel, e.g., color and size after analysts' specification of the *x*- and *y*-axis. Quantitative attributes can be aggregated in seven ways, i.e., *sum*, *mean*, *maximum*, *minimum*, *median*, *vari-*

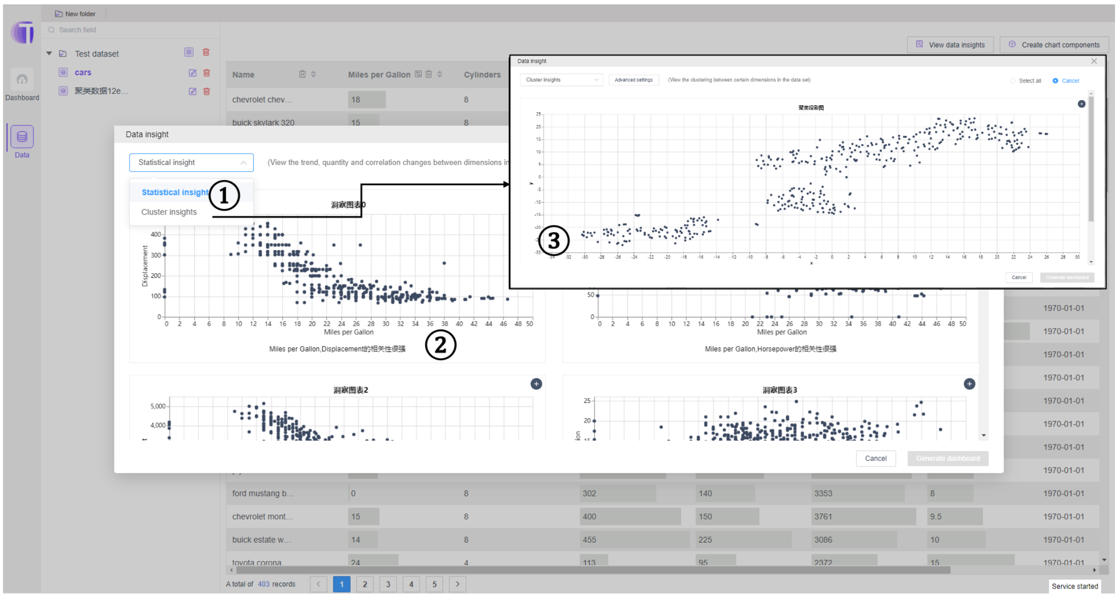

Figure 3: Auto-insight recommendation of *TurboVis*. ① Insight options; ② Auto-insights and the corresponding natural language descriptions; ③ Clusters and outlier detection.

ance, and *standard deviation*. For instance, analysts can drag a quantitative attribute to the *x*-axis shelf and another quantitative attribute to the *y*-axis shelf to generate a scatterplot; to create a bar chart, analysts can drag a nominal attribute to the *x*-axis shelf and a quantitative attribute by mean, and the chart can be replaced to a line chart if the *x*-axis is filled with a temporal attribute.

With respect to the recommendation list, in the version without auto-insight recommendation, the list only has the results from graph matching. Specifically, we classify basic visualizations based on input data format, which comes in the form of a decision tree [57] that leads to a set of potentially appropriate visualizations to represent the current data configuration. For example, considering quantitative attributes, if with only one numeric variable, graphs that are appropriate in this case are histogram and density plot. We adopt the graph matching on the basis of two underlying philosophies [57]. First, most data analysis can be summarized in about twenty different dataset formats. Second, both data and context can determine the appropriate chart. Therefore, our graph matching scheme consists of identifying and trying all feasible chart types to find out which one(s) suit(s) the data and idea best. In ⑦, analysts can rename the current visualization or adjust the color and size encoding if no additional attributes are encoded by color or size.

TurboVis with Auto-Insight Recommendation. We design the second version based on prior studies in proactive, e.g., *Voder* [58] and reactive, e.g., *DIVE* [59] insight-based recommendations. *TurboVis* with auto-insight recommendation serves as assistance to help achieve visualizations with interesting patterns. Asides from manual visualization specification, this version supports proactive unsolicited recommendations and reactive query-based recommendations. Particularly, proactive unsolicited recommendations list all potential auto-insights, and reactive query-based recommendations list all the recommended charts on the right side of the interface. This design considers the selected attributes a query and generates recommendations relevant to the selected attributes.

Regarding the proactive unsolicited recommendations, we put the entry to the auto-insight recommendation above the data table (Figure 2(A)②), maximizing the utility of auto-insight recommendation service. Particularly, as shown in Figure 3, different types of auto-insight recommendations are classified into pull-down list options (①), i.e., patterns measured by statistical metrics and clustering and outlier detection algorithms. Currently, auto-insight recommendation supports four insight types: *trend detection* shows line charts with obvious increasing or decreasing temporal pattern between temporal and quantitative dimensions; the *correlation* between two highly correlated attributes between two quantitative dimensions; *pairwise distribution comparison* concerns two groups where the distributions are significantly different, and *clustering and outlier detection* shows potential clusters and outliers. To facilitate quick browsing and inspection and easy-to-understand, each recommendation contains a concise natural language description which is generated by templates (②), such as "*Miles per Gallon and Weight have a strong correlation*". With respect to clustering and outlier detection, we select *t-SNE* as the dimensionality reduction technique because it shows superiority in generating 2D projection that "*can reveal meaningful insights about data, e.g., clusters and outliers*" [60]. In addition, advanced parameter settings are also provided such as quantitative attributes for projection and *t-SNE* parameters in terms of *perplexity*, *learning rate*, *maximum iterations*, and *distance metric* (③). By previewing all the auto-insight recommendations, analysts can select any of them by clicking on + to submit to the target dashboard. We employ exhaustion, match, generate for mining interesting visualizations. We show how to generate data auto-insights through the following four steps.

Step ① *Determining attribute types.* After loading a dataset, *TurboVis* first gathers metadata of data types by iterating on all data records. Particularly, we maintain several metadata to determine whether the value on a certain attribute is e.g., numeric, date, or coordinate. We also maintain the number of unique values and the maximum of replication corresponding to a certain data attribute. Data types can be thus determined for subsequent processing.

Step ② *Maintaining recommendation configuration.* Each data auto-insight corresponds to a recommendation configuration, which consists of six dimensions in terms of "*type_x*", "*type_y*", "*position_exchange*", "*measure*", "*supported_graphs*" and "*priority*". For example, trend corresponds to a bar recommendation configuration: {*type_x : temporal*, *type_y : quantitative*, *position_exchange : 0*, *measure : trend*, *supported_graphs : ['bar']*, *priority : 0* }, which means that when a temporal attribute meets a quantitative attribute, we can use bar to visualize the relationship with exchangeable axes. "*Priority*" indicates the recommendation priority when demonstrating all the data auto-insight patterns to audiences. Similarly, correlation corresponds to a scatterplot recommendation configuration with both "*type_x*" and "*type_y*" is a quantitative attribute and the "*supported_graphs*" can be *['scatter']*.

Step ③ *Preparing and matching all feasible combinations to recommendation configuration.* We initialize feasible combinations to mine potential patterns hidden in the combination of any two different attributes. To optimize the exhaustion process, we allow users to specify mask attributes thus the calculation will not consider those masked attributes. We then match each feasible combination against the targets of the auto-insight measures on the basis of "*type_x*" and "*type_y*".

Step ④ *Generating candidate auto-Insight recommendation.* Upon a match between a feasible combination and an auto-insight recommendation configuration, parameters in the corresponding candidate auto-insight configuration are automatically filled. For example, given a match of both "*type_x*" and "*type_y*" is a quantitative attribute, we generate a candidate auto-insight template with the following fields: '*mask*' : {'*type*' : supported_graphs[0], '*tooltip*' : True}, '*encoding*' : {'*x*' : {'*field*' : name_x, '*type*' : "quantitative"},'*y*' : {'*field*' : name_y, '*type*' : "quantitative"}}, '*priority*' : priority. Furthermore, according to different auto-insight measures, we compute different metrics for each candidate visualization specification. For example, x with a quantitative attribute and y with a quantitative attribute are evaluated by a *Spearman correlation coefficient* with associated *p-value* while x with a temporal attribute and y with a quantitative attribute are evaluated by the trend detection measure. Candidates with a metric value higher than a predefined threshold parameter are recommended to analysts. Based on the results, we fill the '*result*' in the template with {'*correlation*' : correlation, '*p-value*' : p-value}; '*message*' : "name_x and name_y has a strong correlation depending the value of p-value".

Regarding reactive query-based recommendations, *TurboVis* with auto-insight recommendations merges auto-insight recommendations into the recommendation list that appear in the right panel (Figure 2(B)⑥), which tailors the auto-insight recommendations into the EDA pipeline. In Figure 4, the left subfigure shows the graph matching based on a particular data attribute configuration and the right subfigure displays the auto-insights in the version with auto-insight recommendations. In other words, we only display graph matching in the version without auto-insight recommendations and display both in the version with auto-insight recommendations.

5. Experiment

To investigate the "double-edged sword" effect of auto-insight recommendation design on EDA, we conduct a within-subjects study with 18 data analysts in EDA tasks on two datasets.

Participants. We recruit 18 industrial data analysts (9 females, 9 males, age: 28 ± 3.03) from a local Internet bank, most of whom have 2 to 5 years of working experiences. We invite participants who need to conduct EDA almost every day according to their self report. Particularly, participants had used tools for EDA, including Tableau (10/18), Excel (18/18), Python (8/10), and R (12/18). They are representatives of our target users and could provide us more comprehensive insights. We compensate participants with a $20 gift card.

Datasets and Data Processing. We choose two datasets to evaluate the effects of the auto-insight recommendations. The first one is the happiness ranking dataset that our participants analyze less in their daily work, and it consists of 1093 records with attributes of *date*, *country*,

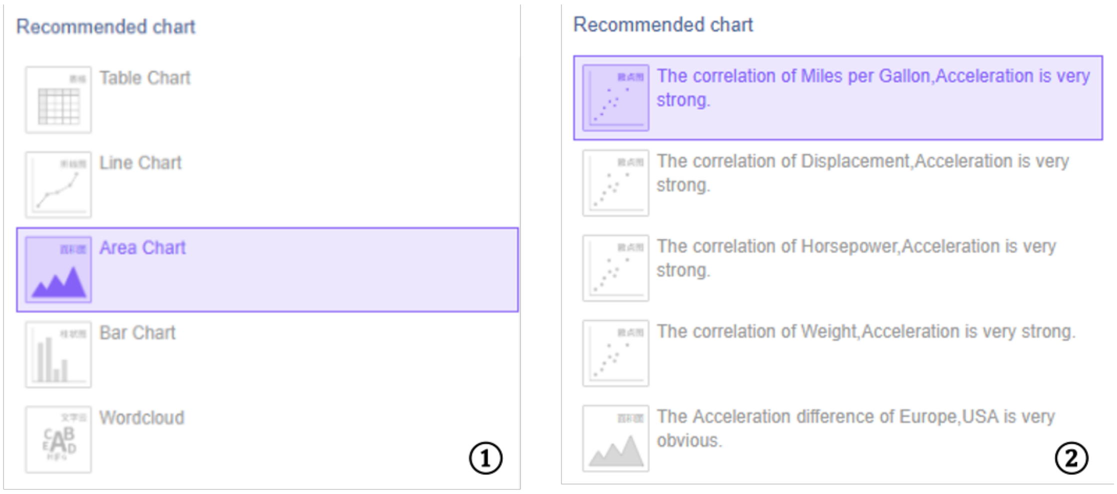

Figure 4: Graph matching and auto-insight recommendation results.

region, *happiness ranking*, *happiness*, *GDP per capita*, *GDP per family*, *healthy*, *freedom*, *trustness*, *generosity*, and *residence*[1]. The second dataset is the Chinese bank's annual report, which is closer to the type of data that our participants use everyday, and it comprises 1590 records with 18 financial attributes. As a demo to introduce our system to the participants, we also include a car dataset which consists of 403 records with attributes of *name*, *miles per gallon*, *cylinders*, *displacement*, *horsepower*, *weight*, *acceleration*, *year*, and *origin*.

To prepare datasets for the two versions, we split the happiness ranking dataset and bank dataset into two parts. During the study, the first dataset is used in the first data exploration session while the second one is used in the second session. This mitigates potential learning effects across the two sessions while ensuring that the data collected from the two sessions could be compared.

Procedure. After obtaining the participants' consent, we conduct the experiment in four sessions, each with a subset of a dataset and one version of *TurboVis*. In other words, every participant gets to explore both datasets in different tasks using both versions of our tool. We counterbalance the order of *TurboVis*'s version in each dataset to minimize the learning effects. We arrange the experimental procedure in the following steps. First, we give a tutorial on how to use both versions of *TurboVis* (each for 10 minutes; the system is running on ThinkPad X270 notebook with a 12.5-inch display) to explore data with the car dataset and then allow the participants to freely explore the tool for another 10 minutes. During this process, we encourage the participants to raise any question about the usability, functions, and features of *TurboVis* to ensure that they have no problems conducting the subsequent tasks on their own. In the main study, for participants who start with the one *TurboVis* version, we ask them to conduct a 15-minute data exploration of the happiness ranking sub dataset (session 1). Then, they proceed to another 15-minute data exploration of another happiness ranking sub dataset using another version of *TurboVis* (session 2). Participants are also asked to think aloud their ideas when performing all the tasks. Their exploration processes are automatically recorded as system logs for the subsequent quantitative analysis. Then, we repeat the above process by using another dataset, i.e., the bank annual report dataset for session 3 and 4. After finishing all the tasks, participants are required to complete a questionnaire with 7-point Likert scale questions, followed by a semi-structured interview with each participant to make sense of their ratings and collect their opinions about auto-insight recommendations. The whole experiment lasts around 90 minutes for each participant.

Measures. In the above-mentioned experiment, we collect 72 log files (18 participants × 4 sessions). The log data details every action and the associated entities conducted by the participant. The actions include but are not limited to click, select, delete, and drag, and objects are like the name of auto-insight recommendation and attributes. We then derive our dependent measures from these data in relation to the previously mentioned three research questions.

M① *Number of inspected visualizations.* We count the number of visualization if a specific visualization is loaded, selected, edited, or added in a dashboard. This measure is a conservative estimate of the inspected visualizations for the exploration with recommendations.

[1]https://www.kaggle.com/mathurinache/world-happiness-report

M② ***Proportion of supported visualizations.*** To understand whether the potential exploration bias, i.e., confining to the scope of auto-insight recommendations, we compare the proportion of visualizations in terms of *correlation, trend, pairwise distribution comparison*, and *clusters and outliers* among those visualizations logged as manual or auto-insight recommendations between with and without auto-insight recommendation versions.

M③ ***Number of manually specified visualizations.*** We count the number of manually created visualizations for each session. This measure indicates the adoption or potential reliance on the auto-insight recommendations for EDA, since participants might utilize the auto-insights and thus create fewer visualizations on their own.

M④ ***Time duration between opening and closing auto-insight recommendations.*** To investigate the effect of how auto-insights may advance their EDA process, we record and calculate the time intervals between participant opening and closing (actions recorded in the logs) the auto-insight recommendation panel when given the version of our tool with this function, to explore different datasets.

M⑤ ***Number of modifications to the recommended visualizations.*** When participants are exploring the data using the auto-insight recommendation version, they can select any visualization from both the channels of graph matching and the tool's recommendations. When participants drag "asset size" to one axis and the display area would immediately present the auto-insight recommendation relevant to this attribute and fill the other axis with information e.g., asset size that has a high correlation relationship. However, one issue we frequently observe is that if this "automatic completion" is inconsistent with participants' intent, they would modify the recommendation result. Therefore, to investigate to what extend participants would directly accept the auto-insights when they are exploring dataset with different familiarity, we calculate the number of modification actions immediately occur after a recommendation result is populated in the display area.

To complement the quantitative data and provide in-depth understanding of users' perceptions towards the auto-insight recommendations, we also collect the participants' responses to an end-of-study questionnaires, in which we ask them about their preference of tool versions when conducing EDA with a clear exploration task, and whether the auto-insights offer new knowledge. Particularly, we have: **M⑥** ***usefulness of auto-insights in an open exploration***, **M⑦** ***usefulness of auto-insights with a target-oriented inspection*** (1 - Extremely unuseful, 7 - Extremely useful), **M⑧** ***version preference when exploring in an open exploration***, and **M⑨** ***version preference when exploring with a target-oriented inspection*** (1 - Prefer non-auto-insight version a lot, 7 - Prefer auto-insight version a lot).

6. Results and Analysis

We report the quantitative analysis of participants' operation logs and quantitative ratings and feedback on the three research questions, as shown in Figure 5. Particularly, we analyze the first five measures (**M①** - **⑤**) using Wilcoxon signed-rank tests (very appropriate for a repeated measure design where the same subjects are evaluated under two different conditions) with a significance level of 0.05 and we report the median values for the subjective measures collected from the questionnaires on each item (**M⑥** - **⑨**).

RQ① Utilizing auto-insights and manual visualization specification collectively. As shown in Figure 5(a), we find a significant difference in the **M①** *number of inspected visualizations* at a significance level of 0.05 (both $Z = -3.726, p = 0.00019$) by using happiness ranking and bank datasets, respectively. On average, participants inspect more visualizations when using *TurboVis* with auto-insight recommendations ($M = 45.33$ with happiness ranking and $M = 44.11$ with bank dataset) than in the without auto-insight recommendation condition ($M = 12.61$ with happiness ranking and $M = 15.44$ with bank dataset), indicating a wider coverage of visualizations with auto-insight recommendations. **M②** *Proportion of visualizations that are supported by recommendation* is higher with auto-insight recommendations ($M = .465$ with happiness ranking and $M = .36$ with bank dataset) than without this feature (Figure 5(b)). The difference is significant for both the happiness ranking ($Z = -3.725, p = .000196$) and the bank dataset ($Z = -3.483, p = .000499$), indicating that the auto-insight recommendations bias participants towards certain types of visualizations during the EDA process.

Generally, participants specify more visualization in the absence of auto-insight recommendations ($M = 6.61$ with happiness ranking and $M = 9.94$ with bank dataset) than in the presence of this service ($M = 4.5$ with happiness ranking and $M = 8.67$ with bank dataset), as shown in Figure 5(c) (**M③**). However, we observe different results of the manual specification of visualization on our two datasets. Specifically, the difference is significant when participants explore the happiness ranking dataset ($Z = -2.166, p = .03$), but not significant ($Z = -1.742, p = .081$) when using a bank dataset. Considering that our participants frequently analyze bank data and rarely inspect happiness data in their daily work, this implies that our participants have a tendency to resort to auto-insight recommendations for inspecting visual patterns rather than constructing visualizations manually when this service is available for exploring an unfamiliar dataset, but not as much when facing a familiar dataset. "*I think I can do more inspection without auto-insight recommendations*" (P12, male, age: 31). "*With my intuition and knowledge, I just want to see the*

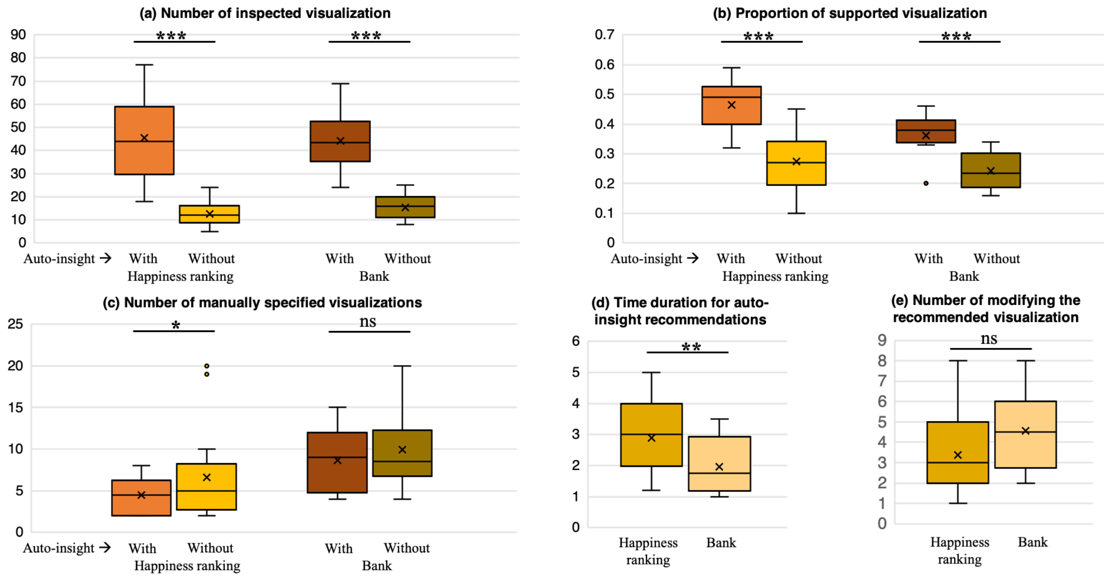

Figure 5: Results of log data in happiness ranking task and bank task, either with or without auto-insight recommendations. ns: $p \geq .05$, $*$: $p < .05$, $**$: $p < .01$, $***$: $p < .001$.

data this way" (P10, female, age: 29). "*I feel like I can reason more on my own than the recommendations*" (P6, male, age: 26).

RQ② Auto-insight reliability in EDA. To evaluate the acceptance of participants towards the auto-insights conveyed by the recommendations, we measure the **M④** *time duration between opening and closing the auto-insight recommendation* panel given the version of *TurboVis* with this feature. As shown in Figure 5(d), we find a significant difference in this measure ($Z = -3.301, p = .001$). An average duration of 2.88 minutes is spent on browsing the results of auto-insight recommendations of the happiness ranking dataset, compared with an average duration of 1.97 minutes on the bank dataset's recommended auto-insights. "*I am not familiar with happiness ranking dataset so I have a lower expectation with auto-insight recommendations*" (P2, male, age: 28). "*I would try to find why these auto-insights are recommended when I am exploring the happiness ranking dataset*" (P14, female, age: 25). "*I probably would not have figured out the outliers by intuition and I am happy that it has been recommended*" (P6, male, age: 26). With respect to a relatively more familiar dataset (e.g., bank dataset in our case), the auto-insight recommendation serves as assistance for quick verification, "*I can quickly identify the interesting auto-insights since I am familiar with them*" (P8, female, age: 30).

RQ③ Auto-insights in open exploration and target oriented inspection. We investigate the acceptance of auto-insights in different EDA tasks (i.e., an open exploration or with a target-oriented inspection) in terms of perceived usefulness and version preference. We ask participants in the questionnaire about the perceived usefulness and preference of auto-insights when they conduct an open exploration or a target-oriented inspection in our EDA tasks. During the experiment, we often find that participants modify the recommended results by deleting the attribute that has been automatically populated on one axis. Therefore, we obtain the **M⑤** *number of modifications to the recommended visualizations* and compare the counts between the two dataset. As shown in Figure 5(e), we find that although the mean value of the number of modification differs, i.e., $M = 3.39$ with the happiness ranking dataset and $M = 4.56$ with the bank dataset, the difference is not significant ($Z = -1.579, p = .114$). The questionnaire item of **M⑥⑦** *usefulness of auto-insight recommendations in an open exploration or with a target-oriented inspection* also shows that participants appreciate the usefulness of auto-insight recommendations regardless of the tasks (open: $M = 6.11, SD = .96$ and target: $M = 6.06, SD = .87$), suggesting that the acceptance of auto-insights in different EDA tasks does not change significantly.

However, in participants' response to the question **M⑧⑨** *version preference in an open exploration or with a target-oriented inspection*, the median rating was 5.89 with an SD of 0.83 for open exploration on a scale from preferring without auto-insight recommendation much more (1) to preferring without auto-insight recommendations much more (7), suggesting that participants prefer having the service much more when they only have a vague idea about what they are looking for. "*When you introduce a new dataset that I haven't see before, I don't know where to start*" (P2, male, age: 28). "*It is hard for me to figure out where to go first and auto-insights help me with the first step*" (P14, female, age: 25). However, when they have specific questions to investigate, they prefer *TurboVis* without auto-insight recommendations ($M = 3.8$, $SD = 1.1$). "*I have a very clear target in my mind so I directly turn to the manually specifying visualizations interface to see what I can get*", "*since I am quite familiar with the data and I know how to select attributes that have relationships*" (P8, female, age: 30). "*When I was exploring on my own, I feel like I am creating what I want*" (P12, male, age: 31).

7. Discussion

In this section, we first discuss the identified "double-edged sword" effect of auto-insight recommendations on the EDA process. Then, we elicit the design implications regarding the observed findings. In the end, we reflect on the limitations of this study.

Exploration Bias. We further our awareness of the side effects of auto-insight recommendations on EDA by first highlighting potential exploration bias and excessive reliance. Analysts tend to adjust their degree of reliance on auto-insight recommendations or manual visualization specifications on the basis of data familiarity and task structure. For one thing, the domain experts with a high degree of familiarity with data and analytic tasks are more likely to explore more visualizations on their own. For another, they believe that their domain knowledge and intuitions can help them achieve a smooth EDA process when facing familiar data and scenarios. However, if they encounter a new dataset, they might heavily rely on the auto-insight recommendations by immersing themselves in browsing the recommendation results. When auto-insight recommendations were present, participants demonstrated less desire to explore data on their own, e.g., the number of manually-specified visualizations drops significantly. Auto-insight recommendation service can produce a large number of recommendations to implicitly impel users to explore more visualizations, thus, leading to biased data exploration. A potential design implication is that auto-insight recommendations should be designed differently based on how people can tolerant auto-insights. In scenarios that welcome diverse exploration results, recommendations should be hidden or at least receive less concerns. Also, tooltips can be provided to explicitly inform analysts that how many auto-insight recommendations or the ratio of auto-insights to the overall visualizations have been added to the dashboard.

Message Reliability. When analysts are quite familiar with the dataset and exploration scenarios, they have a higher expectation of the auto-insight recommendations. They would try to draw conclusions by observing the auto-insight recommendation results, e.g., determining whether these insights make sense or not, i.e., they may question the message reliability. Otherwise, when they have a vague idea about what they are looking for, they have a lower expectation on the auto-insight recommendations; they appreciate the interestingness of the recommended patterns, instead of identifying whether these insights are right or wrong. A design implication is that it is necessary to note next to the auto-insight recommendations what methods are adopted to generate these recommendations and what the system has done in order to make analysts clear about the underlying recommendation mechanisms.

Exploration Interruption. When we inferring user intention from interaction log data, we observed that auto-insight recommendations sometimes interrupt analysts. When they were exploring a familiar dataset and trying to construct a desired visualization for inspection, they commented that they prefer the without auto-insight recommendation version in the drag-and-drop process. For example, when analysts drag a data attribute that has been involved in an auto-insight recommendation to one axis, TurboVis automatically refreshes the recommendation view and lists all the auto-insight recommendations related to the specific data attribute and even populates the other fields, "*I was intending to put 'acceleration' and 'cylinders' together to see what kind of visualization results could appear, but the recommended auto-insights grabbed my attention.*" A plausible hypothesis is that the low cognitive cost of gleaning insights from the recommendations makes them too tempting to consume, thereby inducing undesirable interruption effects. To mitigate the interruption effects of auto-insight recommendations, one alternative is to hide the auto-insight recommendation services in a toolbar by following *Show Me* or split the recommended results from the existing panel and display them in a separate panel.

Limitations. First, although we conducted a prolonged collaborative design with a limited number of industrial domain experts, *TurboVis* is still limited in its current form with respect to the raised requirements collected from their feedback. Second, we derived design alternatives by surveying prior systems and conducting iterative design with our collaboration experts. Admittedly, we only tapped into a limited design space of recommendations, i.e., by providing a button to see the auto-insight recommendation on-demand and linking auto-insight recommendations to manual user interaction. With a different design of auto-insight recommendation service, users may perceive differently. Third, we design the auto-insight recommendation service only based on a limited number of an extensible repository of statistic metrics, which quantify interesting visualization patterns in basic charts. Meanwhile, only experienced data analysts working on a particular set of problems were included in the user study. The results therefore might generalize only to this kind of users. Furthermore, auto-insight recommendations fail to recommend any patterns if involving multiple attributes. Our collaboration experts also commented that there should be more types of recommendations. Future work will systematically conduct more investigation into more real-world business scenarios to identify more preferable auto-insight types.

8. Conclusion

In this study, we explore the potential "double-edged sword" effects of auto-insight recommendations on the EDA process. We demonstrate how auto-insight recommendations could be incorporated into a self-developed Tableau-like visualization tool termed *TurboVis*. By comparing two versions of *TurboVis*, we find that auto-insight recommendations not only encourage more visualization inspections but also introduce biases to data exploration. Meanwhile, the perceived level of message reliability and interruption of auto-insight recommendation service depend on data familiarity and task structures. Our work offers initial implications for embedding auto-insight recommendations into the EDA process.

Acknowledgments

We are grateful for the valuable feedback and comments provided by the anonymous reviewers.

References

[1] R. Goyal, A. Tamizharasan, A. Singhal, Exploratory data analysis (1999).
[2] M. Diamond, A. Mattia, Data visualization: An exploratory study into the software tools used by businesses., Journal of Instructional Pedagogies 18 (2017).
[3] J. W. Gustafson, C. H. Jones, L. Pape-Haugaard, Designing a dashboard to visualize patient information, in: Proceedings from The 16th Scandinavian Conference on Health Informatics 2018, Aalborg, Denmark August 28–29, 2018, 151, Linköping University Electronic Press, 2018, pp. 23–28.
[4] R. K. Sawyer, Explaining creativity: The science of human innovation, Oxford university press, 2011.
[5] J. W. Tukey, We need both exploratory and confirmatory, The American Statistician 34 (1980) 23–25.
[6] M. Vartak, S. Rahman, S. Madden, A. Parameswaran, N. Polyzotis, Seedb: Efficient data-driven visualization recommendations to support visual analytics, in: Proceedings of the VLDB Endowment International Conference on Very Large Data Bases, volume 8, NIH Public Access, 2015, p. 2182.
[7] K. Wongsuphasawat, D. Moritz, A. Anand, J. Mackinlay, B. Howe, J. Heer, Voyager: Exploratory analysis via faceted browsing of visualization recommendations, IEEE transactions on visualization and computer graphics 22 (2015) 649–658.
[8] Z. Cui, S. K. Badam, M. A. Yalçin, N. Elmqvist, Datasite: Proactive visual data exploration with computation of insight-based recommendations, Information Visualization 18 (2019) 251–267.
[9] J. Mackinlay, P. Hanrahan, C. Stolte, Show me: Automatic presentation for visual analysis, IEEE Transactions on Visualization & Computer Graphics 13 (2007) 1137–1144.
[10] K. Wongsuphasawat, Z. Qu, D. Moritz, R. Chang, J. Heer, Voyager 2: Augmenting visual analysis with partial view specifications, in: the 2017 CHI Conference, 2017.
[11] Ç. Demiralp, P. J. Haas, S. Parthasarathy, T. Pedapati, Foresight: Recommending visual insights, arXiv preprint arXiv:1707.03877 (2017).
[12] Q. Li, Z. Peng, H. Zeng, Q. Chen, L. Yi, Z. Wu, X. Ma, T. Chen, Friend network as gatekeeper: A study of wechat users' consumption of friend-curated contents, in: The eighth International Workshop of Chinese CHI, 2020, pp. 21–31.
[13] A. Jameson, Adaptive interfaces and agents, in: The human-computer interaction handbook, CRC Press, 2007, pp. 459–484.
[14] A. D. Jameson, Understanding and dealing with usability side effects of intelligent processing, AI Magazine 30 (2009) 23–23.
[15] F. Du, C. Plaisant, N. Spring, K. Crowley, B. Shneiderman, Eventaction: A visual analytics approach to explainable recommendation for event sequences, ACM Transactions on Interactive Intelligent Systems (TiiS) 9 (2019) 1–31.
[16] D. Kelly, J. Teevan, Implicit feedback for inferring user preference: a bibliography, in: Acm Sigir Forum, volume 37, ACM New York, NY, USA, 2003, pp. 18–28.
[17] J. Xiao, J. Stasko, R. Catrambone, et al., An empirical study of the effect of agent competence on user performance and perception, in: AAMAS, volume 4, Citeseer, 2004, pp. 178–185.
[18] G. Dove, K. Halskov, J. Forlizzi, J. Zimmerman, Ux design innovation: Challenges for working with machine learning as a design material, in: Chi Conference on Human Factors in Computing Systems, 2017.

[19] D. Booth, The human-computer interaction handbook: fundamentals, evolving technologies and emerging applications, L Erlbaum Associates Inc 15 (2012) 85–87.
[20] M. Vartak, S. Huang, T. Siddiqui, S. Madden, A. Parameswaran, Towards visualization recommendation systems, Acm Sigmod Record 45 (2017) 34–39.
[21] J. Tukey, Exploratory Data Analysis, Addison-Wesley Pub. Co.„ ????
[22] J. T. Behrens, Principles and procedures of exploratory data analysis., Psychological Methods 2 (1997) 131.
[23] P. Hanrahan, Analytic database technologies for a new kind of user: the data enthusiast, in: Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data, 2012, pp. 577–578.
[24] K. Morton, M. Balazinska, D. Grossman, J. Mackinlay, Support the data enthusiast: Challenges for next-generation data-analysis systems, Proceedings of the VLDB Endowment 7 (2014) 453–456.
[25] M. D'Agostino, D. M. Gabbay, R. Hähnle, J. Posegga, Handbook of tableau methods, Springer Science & Business Media, 2013.
[26] O. Troyansky, T. Gibson, C. Leichtweis, QlikView Your Business: An Expert Guide to Business Discovery with QlikView and Qlik Sense, John Wiley & Sons, 2015.
[27] L. Grammel, M. Tory, M.-A. Storey, How information visualization novices construct visualizations, IEEE transactions on visualization and computer graphics 16 (2010) 943–952.
[28] S. Zhu, G. Sun, Q. Jiang, M. Zha, R. Liang, A survey on automatic infographics and visualization recommendations, Visual Informatics (2020).
[29] M. Bostock, V. Ogievetsky, J. Heer, D^3 data-driven documents, IEEE transactions on visualization and computer graphics 17 (2011) 2301–2309.
[30] A. Satyanarayan, D. Moritz, K. Wongsuphasawat, J. Heer, Vega-lite: A grammar of interactive graphics, IEEE transactions on visualization and computer graphics 23 (2016) 341–350.
[31] D. Li, H. Mei, Y. Shen, S. Su, W. Zhang, J. Wang, M. Zu, W. Chen, Echarts: A declarative framework for rapid construction of web-based visualization, Visual Informatics 2 (2018) 136–146.
[32] R. López-Cortijo, J. G. Guzmán, A. A. Seco, icharts: charts for software process improvement value management, in: European Conference on Software Process Improvement, Springer, 2007, pp. 124–135.
[33] M. Mauri, T. Elli, G. Caviglia, G. Uboldi, M. Azzi, Rawgraphs: a visualisation platform to create open outputs, in: Proceedings of the 12th biannual conference on Italian SIGCHI chapter, 2017, pp. 1–5.
[34] N. J. Pioch, J. O. Everett, Polestar: collaborative knowledge management and sensemaking tools for intelligence analysts, in: Proceedings of the 15th ACM international conference on Information and knowledge management, 2006, pp. 513–521.
[35] Z. Liu, J. Thompson, A. Wilson, M. Dontcheva, J. Delorey, S. Grigg, B. Kerr, J. Stasko, Data illustrator: Augmenting vector design tools with lazy data binding for expressive visualization authoring, in: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, 2018, pp. 1–13.
[36] D. Ren, T. Höllerer, X. Yuan, ivisdesigner: Expressive interactive design of information visualizations, IEEE transactions on visualization and computer graphics 20 (2014) 2092–2101.
[37] S. F. Roth, J. Kolojejchick, J. Mattis, M. C. Chuah, Sagetools: An intelligent environment for sketching, browsing, and customizing data-graphics, in: Conference companion on Human factors in computing systems, 1995, pp. 409–410.
[38] W. Cui, X. Zhang, Y. Wang, H. Huang, B. Chen, L. Fang, H. Zhang, J.-G. Lou, D. Zhang, Text-to-viz: Automatic generation of infographics from proportion-related natural language statements, IEEE transactions on visualization and computer graphics 26 (2019) 906–916.
[39] Y. Chen, S. Barlowe, J. Yang, Click2annotate: Automated insight externalization with rich semantics, in: 2010 IEEE Symposium on Visual Analytics Science and Technology, IEEE, 2010, pp. 155–162.
[40] V. Dibia, Ç. Demiralp, Data2vis: Automatic generation of data visualizations using sequence-to-sequence recurrent neural networks, IEEE computer graphics and applications 39 (2019) 33–46.
[41] Y. Luo, X. Qin, N. Tang, G. Li, Deepeye: Towards automatic data visualization, in: 2018 IEEE 34th International Conference on Data Engineering (ICDE), IEEE, 2018, pp. 101–112.
[42] R. Ding, S. Han, Y. Xu, H. Zhang, D. Zhang, Quickinsights: Quick and automatic discovery of insights from multi-dimensional data, in: Proceedings of the 2019 International Conference on Management of Data, 2019, pp. 317–332.
[43] B. Tang, S. Han, M. L. Yiu, R. Ding, D. Zhang, Extracting top-k insights from multi-dimensional data, in: Proceedings of the 2017 ACM International Conference on Management of Data, 2017, pp. 1509–1524.
[44] J. Heer, Agency plus automation: Designing artificial intelligence into interactive systems, Proceedings of the National Academy of Sciences 116 (2019) 1844–1850.
[45] K. Wongsuphasawat, Z. Qu, D. Moritz, R. Chang, J. Heer, Voyager 2: Augmenting visual analysis with partial view specifications, in: the 2017 CHI Conference, 2017.

[46] J. Bobadilla, F. Ortega, A. Hernando, A. Gutiérrez, Recommender systems survey, Knowledge-Based Systems 46 (2013) 109–132.

[47] S. Zhang, L. Yao, A. Sun, Y. Tay, Deep learning based recommender system: A survey and new perspectives, ACM Computing Surveys (CSUR) 52 (2019) 1–38.

[48] D. Parra, S. Sahebi, Recommender systems: Sources of knowledge and evaluation metrics, in: Advanced techniques in web intelligence-2, Springer, 2013, pp. 149–175.

[49] J. L. Herlocker, J. A. Konstan, L. G. Terveen, J. T. Riedl, Evaluating collaborative filtering recommender systems, ACM Transactions on Information Systems (TOIS) 22 (2004) 5–53.

[50] P. Pu, L. Chen, R. Hu, A user-centric evaluation framework for recommender systems, in: Proceedings of the fifth ACM conference on Recommender systems, 2011, pp. 157–164.

[51] R. Sinha, K. Swearingen, The role of transparency in recommender systems, in: CHI'02 extended abstracts on Human factors in computing systems, 2002, pp. 830–831.

[52] M. Eslami, A. Rickman, K. Vaccaro, A. Aleyasen, A. Vuong, K. Karahalios, K. Hamilton, C. Sandvig, " i always assumed that i wasn't really that close to [her]" reasoning about invisible algorithms in news feeds, in: Proceedings of the 33rd annual ACM conference on human factors in computing systems, 2015, pp. 153–162.

[53] E. Rader, R. Gray, Understanding user beliefs about algorithmic curation in the facebook news feed, in: Proceedings of the 33rd annual ACM conference on human factors in computing systems, 2015, pp. 173–182.

[54] A. Jameson, Understanding and dealing with usability side effects of intelligent processing, Ai Magazine 30 (2009) 23–40.

[55] P. D. Adamczyk, B. P. Bailey, If not now, when? the effects of interruption at different moments within task execution, in: Proceedings of the SIGCHI conference on Human factors in computing systems, 2004, pp. 271–278.

[56] J. Wallace, J. Mccarthy, P. C. Wright, P. Olivier, Making design probes work, in: Sigchi Conference on Human Factors in Computing Systems, 2013.

[57] Y. Holtz, C. Healy, From data to viz, 2018, URL https://www. data-to-viz. com (2019).

[58] A. Srinivasan, S. M. Drucker, A. Endert, J. Stasko, Augmenting visualizations with interactive data facts to facilitate interpretation and communication, IEEE transactions on visualization and computer graphics 25 (2018) 672–681.

[59] K. Hu, D. Orghian, C. Hidalgo, Dive: A mixed-initiative system supporting integrated data exploration workflows, in: Proceedings of the Workshop on Human-In-the-Loop Data Analytics, 2018, pp. 1–7.

[60] Q. Li, K. S. Njotoprawiro, H. Haleem, Q. Chen, C. Yi, X. Ma, Embeddingvis: A visual analytics approach to comparative network embedding inspection, in: 2018 IEEE Conference on Visual Analytics Science and Technology (VAST), IEEE, 2018, pp. 48–59.