GROOT: A System for Editing and Configuring Automated Data Insights

Sneha Gathani*
University of Maryland, College Park

Anamaria Crisan†
Tableau Research

Vidya Setlur‡
Tableau Research

Arjun Srinivasan§
Tableau Research

Figure 1: GROOT allows users to edit and reconfigure automated data insights by (1) selecting marks in charts to get recommendations of new insights based on the selection, (2) reconfiguring default insights by adjusting the template or insight generation thresholds, (3) adding new custom insights by specifying text templates for insights.

ABSTRACT

Visualization tools now commonly present automated insights highlighting salient data patterns, including correlations, distributions, outliers, and differences, among others. While these insights are valuable for data exploration and chart interpretation, users currently only have a binary choice of accepting or rejecting them, lacking the flexibility to refine the system logic or customize the insight generation process. To address this limitation, we present GROOT, a prototype system that allows users to proactively specify and refine automated data insights. The system allows users to directly manipulate chart elements to receive insight recommendations based on their selections. Additionally, GROOT provides users with a manual editing interface to customize, reconfigure, or add new insights to individual charts and propagate them to future explorations. We describe a usage scenario to illustrate how these features collectively support insight editing and configuration and discuss opportunities for future work, including incorporating Large Language Models (LLMs), improving semantic data and visualization search, and supporting insight management.

Index Terms: Automated data insights, insight reconfiguration, natural language templates.

1 INTRODUCTION

Automated data insights are now commonly found in both commercial and research-focused visualization systems [1, 3, 10, 11, 20, 22, 29, 32, 34, 35]. These insights are typically *textual statements highlighting key takeaways or data patterns* such as correlations, extreme values, or comparisons, among others. While these insights can aid data exploration and help interpret charts, the black-box nature of automated data insight systems limits their utility [4, 19].

Existing systems predominantly use a set of predefined heuristics and templates to extract key facts about a chart's underlying data and surface them as text alongside the chart. However, these heuristic approaches may not capture all types of interesting and relevant information from a chart [5]. For example, while existing heuristic approaches call out correlations between attributes, they tend to overlook insights that highlight the absence of a correlation between two attributes, which may be more noteworthy than its presence. Alternatively, default insights in current tools might call out extreme values in bar charts. However, if there is a significant difference between the two bars, it may be more interesting to call out this difference as an insight instead. Unfortunately, existing systems fail to provide users with flexibility and control over the insight generation process, limiting them to the binary choice of accepting or rejecting the default system-generated insights.

We explore the idea of allowing users to interactively specify or refine automated data insights in visualization tools. We present GROOT, a prototype system that allows users to proactively edit, customize, and reconfigure automated data insights. Through this system, users directly interact with charts to receive additional insight recommendations based on their selections. For instance, clicking on multiple bars in a bar chart triggers new insights that highlight the values for the selected bars or emphasize the differences between them (Figure 1-1). The system also allows customizing the underlying logic for generating predefined types of insights to highlight any relevant findings that were not captured by the system defaults (Figure 1-2). Furthermore, users can define altogether new insights using custom criteria and templates through an interactive editing interface (Figure 1-3). Besides refining and specifying new heuristics for generating insights, GROOT also supports basic data cleaning and transformation operations, including editing entries with specific values (e.g., removing/replacing nulls) and modifying attribute names. These data edits dynamically update the insights, enhancing their readability and alignment with users' analytical requirements.

*e-mail: sgathani@umd.edu. Sneha Gathani conducted this work while interning with Tableau Research.
†e-mail: acrisan@tableau.com
‡e-mail: vsetlur@salesforce.com
§e-mail: arjunsrinivasan@tableau.com

In this paper, we detail GROOT's design and implementation and demonstrate its utility through a usage scenario. Reflecting on our system design, we discuss opportunities for future work on improving agency in automated data insight systems and applications that can be enabled via such configurable systems.

2 Related Work

The definition of "insight" has been a long-standing topic of discussion within visualization and visual analytics research [5, 6, 16, 21]. These definitions range from insights being user utterances about a chart (e.g., a callout to a specific data point) or statistically derived data facts (e.g., differences, outliers) to more high-level definitions equating insights to hypotheses and knowledge links [5]. Numerous "automated data insight systems" have adopted these varying definitions of insights and presented approaches to automatically generate and help users explore data insights. A full review of these systems is beyond the scope of this paper and can be found in other survey manuscripts such as [5, 15, 16, 26, 37].

In our work, we adopt the definition of insights as "data facts" or statements describing the result of one or more statistical functions applied to the data (e.g., min/max, correlation). Given this scope, below we discuss examples from a subset of automated data insight systems that adopt the same definition and heuristically generate insights corresponding to low-level analytic tasks such as identifying extremes, correlations, outliers, and differences, among others [2, 9].

DataShot [35], for instance, extracts data facts from tabular data and presents them along with charts in an ordered layout as a fact sheet. Foresight [11] helps users rapidly explore large high-dimensional datasets by automatically generating and presenting different types of statistical insights as "guideposts". Voder [29] and DataSite [10] recommend data facts for manually specified charts while also presenting related facts across the dataset as "next steps" during visual data exploration. Sortilège [19] uses a Tarot card reading metaphor while presenting automated insights to encourage critique, reflection, and healthy skepticism. WhatsNext [7] paraphrases data facts from Voder [29] as data questions, allowing users to leverage these questions and an interactive relationship graph of insights to guide their analysis. InkSight [17] allows users to sketch on charts in computational notebooks and generate data facts relevant to the sketched area (e.g., selecting a region on a line chart generates statements about the underlying data trend). Besides these research systems, commercial tools like PowerBI's Quick Insights [12], Tableau's Explain Data [32], Salesforce's Einstein [24], Qlik Sense [22], Google Analytics [3], Alteryx [1] also present similar automated insights to highlight statistically salient data patterns.

GROOT builds upon this line of systems and adopts similar heuristics to present statistically salient information as insights during data exploration with two key differences. First, unlike existing tools that operate as a black box, GROOT offers greater transparency and control over the insight generation process by allowing users to inspect and edit the underlying insight generation heuristics and the insight text templates. Second, while existing tools only present a static set of insights, GROOT incorporates techniques from chart annotation and demonstration-based tools (e.g., [8, 9, 13, 23]) to present an interactive interface for specifying insights. In doing so, GROOT allows users to incorporate domain-specific knowledge as part of the generated insights or automatically call out specific patterns/data points of interest.

3 Design Requirements

To identify key challenges with existing tools, we conducted 30-minute structured interviews with six users of automated data insight systems. Participants were recruited on a first-come-first-serve basis via mailing lists and Slack channels at a data analytics software company. The participants included a UX designer, two executive decision makers, and three data analysts who worked with automated insight tools like Tableau's Explain Data [32], Tableau Pulse [33], and Microsoft Power BI [20]. We asked participants questions across four categories: (1) their roles in developing or utilizing automated data insight systems, (2) their existing processes for refining and evaluating the usefulness of insights, (3) challenges faced, and (4) suggested techniques and requirements for refining automated data insights. We derived four design goals from these interviews that guided our prototype development:

DR1. Help users interpret the default system-generated insights. Discussing challenges with current tools, the participants expressed concerns about the transparency of insight generation, as existing systems lack explanations or rationale behind suggested insights. To address this problem, we noted that the system should allow users to access the underlying heuristics to generate insights.

DR2. Support reconfiguring the default insight generation heuristics. Besides understanding how the system generates insights (**DR1**), users may also want to adjust the underlying generation functions or thresholds to determine when an insight is presented (e.g., changing the threshold value beyond which a data point is called out as an outlier). To this end, the system should present an interactive interface to access and edit the system's insight generation logic.

DR3. Allow specifying custom insights. Participants commented that while automated insight systems were good for bootstrapping data exploration and identifying salient information, the predefined insights would not always capture the nuances of a data domain or miss points of specific interest to a user. Considering this limitation, we noted that the system should allow users to interactively define new types of insights at a chart- or dataset-level.

DR4. Dynamically update insights based on data changes. Participants noted that the quality of automated insights, both in terms of what is called out or how the insights are phrased, is significantly impacted by the data quality (e.g., number of null values, poor attribute names). To support the tight coupling between data and automated insights, the system should allow making in-situ changes to the dataset and dynamically update the generated insights.

4 The Groot System

Considering the design requirements, we developed GROOT as a prototype tool to explore the idea of allowing users to edit and configure automated insight systems.

4.1 Usage Scenario

Consider the following usage scenario as an illustration of GROOT's support for interactively editing and specifying automated insights (this scenario is also demonstrated as part of the supplementary

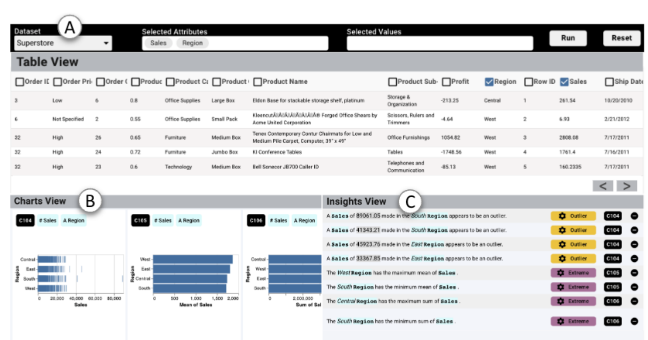

Figure 2: Three main views of GROOT: the data table view (A), the charts view (B), and the insights view (C).

Figure 3: A change in distribution observed for total `Sales` x `Region` bar chart before and after removing the rows having '*Not Specified*' `Order Priority`.

video). Imagine Phoebe, a business analyst at a fictional company, Superstore, exploring the company's sales data containing ∼8k records and 22 attributes, including **A** `Product Name`, **#** `Profit`, **#** `Sales`, **#** `Discount`, and **A** `Ship Mode`, among others.

Selecting attributes to filter charts and insights. Looking through the data table (Figure 2A), Phoebe selects the `Sales` and `Region` attributes to focus her exploration. In response, GROOT presents three charts (Figure 2B) along with a list of insights for each chart (Figure 2C). The charts include a strip plot showing sales variations across regions emphasizing *Outlier* insights, and two bar charts displaying the average and total `Sales` per `Region` with the insights highlighting *Extreme Values* in each chart.

Changing data to update insights. As Phoebe peruses the total `Sales` by `Region` bar chart, she notices that the *Central* region has the highest sales, while the *South* region has the least. However, reviewing the data table (Figure 2A), she notices that some entries in the `Order Priority` attribute are unspecified. To ensure that the insights are only derived from complete data, Phoebe excludes 1672 data rows (19.9% of total data) having an `Order Priority` of '*Not Specified*' by double clicking a corresponding cell to select all the rows having that value (Figure 3A) and pressing delete to remove all of them. This change in the data automatically updates the charts and insights according to the rest of the data (**DR4**). Phoebe confirms that the values and insights do not change significantly based on this data-cleaning step (Figure 3B, C) and continues with her exploration.

Using direct manipulation to get insight recommendations. Looking at the total `Sales` by `Region` bar chart, Phoebe is also intrigued by the steep difference in total sales between the *Central* and *South* regions. Specifically, based on her knowledge of the marketing team, Phoebe knows that the company has invested more heavily in the southern market than the central region. However, because the default list of insights does not call out this difference, Phoebe clicks on the bars for *South* and *Central* (Figure 1-1A). In response, GROOT recommends two new insights, one listing the sales of each region (i.e., a *Retrieve Value* insight) and one about the *Difference* between the two regions (Figure 1-1B). Phoebe clicks the **+** *Add* button on the suggested difference insight and selects *Propagate*. This results in the system adding a new insight to the current chart but also updates the system's insight generation engine to call out differences between the *South* and *Central* regions by default across all bar charts (e.g., Figure 4A) (**DR3**).

Using an interactive editing interface to customize insights. Next, Phoebe is interested in validating whether increasing `Sales` leads to increased `Profit`. This is confirmed when she views a scatterplot comparing the two and observes GROOT's default insight of a strong correlation between them (Figure 4B). Aware of the fact that the operations team has recently raised concerns regarding excessive backstock, Phoebe is curious if there exists a similar correlation between `Sales` and `Discount`.

She explores the corresponding scatterplot but finds no default correlation insights generated by the system (Figure 1-2A). To explicitly call out the lack of a correlation, she clicks the **+** *Add* button and chooses the *Correlation* option to open GROOT's insight editing interface (Figure 1-2B).

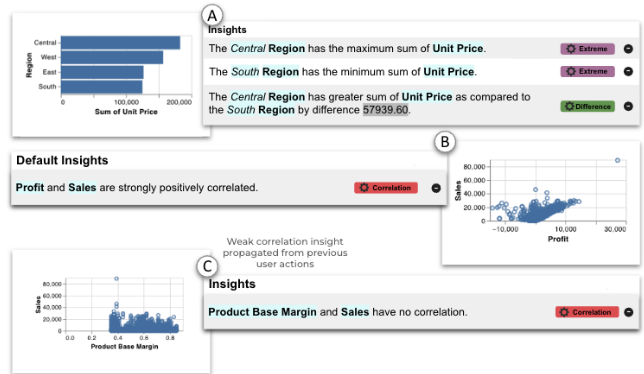

Figure 4: *Difference* insight between *Central* and *South* regions from Figure 1-1 propagated to the `Unit Price` × `Region` bar chart (A). A *Correlation* insight is generated by default for a scatterplot of `Sales` × `Profit` (B). New *Correlation* insight propagated from Figure 1-2 to a scatterplot showing `Sales` × `Product Base Margin` (C).

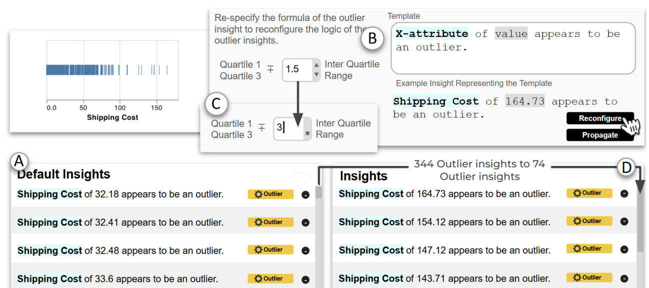

Figure 5: Reconfiguring the *Outlier* insights for `Shipping Cost` (A) by examining its formula (B) and editing it (C) results in a reduced number of items highlighted as outliers (D).

The interface helps Phoebe understand that the underlying generation heuristic for the *Correlation* insights is based on Pearson's correlation coefficient for the selected attributes falling within a specified range (**DR1**). She also learns that the correlation coefficient between `Sales` and `Discount` is −0.02, which is below the predefined threshold (0.7) and hence is not listed as a default insight. To add this insight, Phoebe sets a lower threshold range (−0.3 to 0.3) via the slider and edits the insight template to indicate the fields are not correlated. As she adjusts the range and template, the system presents an example insight that will be generated, helping her preview how the insight will appear. Phoebe selects *Propagate* (Figure 1-2C) to ensure other scatterplots also automatically present insights about the lack of correlations. With this change, when Phoebe subsequently inspects the scatterplot of `Product Base Margin` and `Sales`, GROOT uses the new insight generation rule to highlight the lack of correlation between these fields (Figure 4C).

Reconfiguration of default insights. Next, to explore cost optimization strategies, Phoebe turns her attention to examining the distribution of `Shipping Costs` (Figure 5A). She notices numerous *Outlier* insights, including many low-cost items. Given her knowledge of the inventory, she questions whether the items are truly outliers and decides to adjust the system's outlier-generating heuristic accordingly. To do this, she clicks on the ⚙ button to ac-

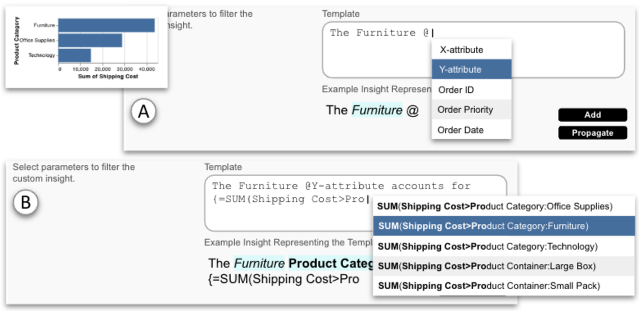

Figure 6: Specifying template for adding new *Custom* insight. “@” allows referencing attributes in the dataset (A), “=” is used to reference mathematical calculations (B), and “{}” allows defining formulae within the interactive editing interface.

cess the insight editing interface (Figure 5B). The system presents the underlying interquartile range-based formula driving the *Outlier* insights (**DR1**). To exclude lower shipping costs from being considered outliers, she modifies the default threshold (Figure 5C) (**DR2**) and clicks the *Add* button to apply it just to the strip plot of `Shipping Cost`. This updates the generated insights and significantly reduces the number of *Outlier* insights (Figure 5D), ensuring the system only calls out items with very high or low values.

Adding Custom Insights. Having looked at the distribution of `Shipping Cost`, Phoebe shifts her focus to understanding which `Product Category` incurs the highest `Shipping Cost` using a bar chart (Figure 1-3A). To call out the percentage of cost coming from the top product category, Phoebe adds a new *Custom* insight (**DR3**) by selecting the **+** *Add* button (Figure 1-3A) and defines a parameterized template for the new insight (Figure 1-3B, Figure 6). The system uses Pheobe's template and formula to compute the percentage and presents a preview of the insight. Phoebe adds this new insight to the bar chart and continues with her analysis.

Besides allowing Pheobe to configure insights based on her domain knowledge, GROOT also implicitly records her changes so that other users see the updated set of insights instead of the system defaults.

4.2 Implementation

We implement GROOT as a web application developed using JavaScript, HTML, CSS, and the React framework. Visualizations are specified and rendered using Vega-Lite [25]. User interactions with visualizations and interactive editing of insights for customization are recorded within the web application. This information is communicated with the back-end to refine insights and update the insight generation engine, ensuring that future sessions yield the edited insights.

GROOT currently supports six chart types (including histograms, dot plots, bar charts, strip plots, scatterplots, and stacked bar charts) and generates five types of pre-defined insights: outliers, extreme values, distribution, differences, and correlations. We build upon mappings and heuristics from prior auto-insight systems (e.g., [10, 11, 29, 35]) to generate these insights by default. Specifically, *Correlation* insights are generated for scatterplots if $|r| > 0.75$ where r is Pearson's correlation coefficient. *Extreme* and *Difference* insights are presented with bar charts to highlight the categories with highest/lowest values or to call out categories that have a difference of $\geq 2.5x$ in their value. *Distribution* insights are generated for strip plots and histograms to highlight the range of values that fall between $Q1$ and $Q3$, which are the first (25th percentile) and the third quartile (75th percentile), suggesting where most points lie. Lastly, *Outlier* insights are generated if data points fall outside the threshold of $\pm 1.5 \times IQR$, where IQR is the interquartile range (the difference between the first and third quartiles). Note that the default heuristics and templates for all types of insights can be adjusted via a reconfiguration interface similar to the one shown in Figure 5.

5 Discussion and Future Work

The development of the current prototype was motivated by prior tools and was informed by interviews with six users of automated insight systems. We plan to expand the prototype to support more charts, insight types, and parameters in the insight editing interface. Additionally, we will conduct a formal summative evaluation to assess the utility and impact of a system like GROOT during data exploration and analysis. Besides these next steps, we see the following broader avenues for future research:

Incorporating LLMs. We currently employ heuristics and templates for prototyping as it is a common approach in prior work [10, 11, 29, 35] that offers control over insight generation and configuration. However, rapidly evolving LLMs are increasingly capable of generating compelling data narratives and insights [17, 18, 30, 31, 36]. To this end, one short-term extension is to see how we can integrate LLMs into GROOT to suggest improved insight phrasings (instead of using predefined or user-defined text templates). Another extension may be to use LLMs to generate insights and explain the underlying logic to users, or for complex tasks like ranking or filtering the insights. However, this raises open questions about presenting and adjusting the LLMs' rationale for generating data insights, which is a rich area for future work.

Supporting other downstream applications. Beyond adjusting the generated insights, the metadata and insights generated from editing actions in GROOT can also be leveraged in other applications. For example, user-defined insights can be used as metadata during visualization and insight search (e.g., [14, 27, 28]) to return more relevant and domain-aware results. Similar to other programming-by-demonstration systems (e.g, [13]), GROOT stores user changes as steps to enable propagating insights defined for one chart as a general set of rules across a dataset (e.g., Figure 1-1 and Figure 1-2). Building upon this, one area for future work is to explore how these changes made to one dataset can be applied to another, supporting insight configuration in scenarios where datasets update progressively or have a similar structure but different values.

Expanding into automated insight management systems. GROOT currently assists in refining and customizing individual automated data insights. However, as the number of insights generated by default increases and more complex ones are introduced, mitigating information overload becomes crucial [15]. Extending GROOT to evolve into an insight management system can address this challenge. Such insight management systems would not only facilitate customization and reconfiguration of insights but also enable users to intelligently rank, browse, and track desired insights. Moreover, effective management tools can systematically track metadata related to user editing and customization actions, including desired data modifications, insight heuristic reconfigurations, and newly wanted customized insights. Analyzing this metadata could reveal common user refinements that are domain-specific, potentially benefiting other datasets within the same domain. This approach not only scales the refinement of automated insight systems across multiple datasets but also saves valuable time and effort in managing insights from scratch. Furthermore, it ensures that insights are informed by users' domain knowledge, thereby empowering users to discover more effective insights.

In conclusion, through GROOT's design, implementation, and usage scenario, we highlight how visualization tools can help users interactively edit and specify automated data insights. We hope this work can guide future research on building more transparent and configurable automated insight systems.

REFERENCES

[1] Alteryx. Alteryx Auto Insights. 1, 2

[2] R. Amar, J. Eagan, and J. Stasko. Low-level Components of Analytic Activity in Information Visualization. In *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.*, pp. 111–117. IEEE, 2005. 2

[3] G. Analytics. Google Analytics 4 Analytics Insights. 1, 2

[4] L. Battle and A. Ottley. What Do We Mean When We Say "Insight"? A Formal Synthesis of Existing Theory. *IEEE Transactions on Visualization and Computer Graphics*, 2023. 1

[5] L. Battle and A. Ottley. What Exactly is an Insight? A Literature Review. *2023 IEEE Visualization and Visual Analytics (VIS)*, pp. 91–95, 2023. 1, 2

[6] R. Chang, C. Ziemkiewicz, T. M. Green, and W. Ribarsky. Defining Insight for Visual Analytics. *IEEE Computer Graphics and Applications*, 29(2):14–17, 2009. 2

[7] C. Chen, J. Hoffswell, S. Guo, R. Rossi, Y.-Y. Chan, F. Du, E. Koh, and Z. Liu. WHATSNEXT: Guidance-enriched Exploratory Data Analysis with Interactive, Low-Code Notebooks. In *2023 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pp. 209–214. IEEE, 2023. 2

[8] Y. Chen, S. Barlowe, and J. Yang. Click2annotate: Automated Insight Externalization with Rich Semantics. In *2010 IEEE Symposium on Visual Analytics Science and Technology*, pp. 155–162. IEEE, 2010. 2

[9] Y. Chen, J. Yang, and W. Ribarsky. Toward Effective Insight Management in Visual Analytics Systems. In *2009 IEEE Pacific Visualization Symposium*, pp. 49–56. IEEE, 2009. 2

[10] Z. Cui, S. K. Badam, M. A. Yalçin, and N. Elmqvist. Datasite: Proactive Visual Data Exploration with Computation of Insight-based Recommendations. *Information Visualization*, 18(2):251–267, 2019. 1, 2, 4

[11] c. Demiralp, P. J. Haas, S. Parthasarathy, and T. Pedapati. Foresight: Recommending Visual Insights. *Proc. VLDB Endow.*, 10(12):1937–1940, aug 2017. doi: 10.14778/3137765.3137813 1, 2, 4

[12] R. Ding, S. Han, Y. Xu, H. Zhang, and D. Zhang. Quickinsights: Quick and Automatic Discovery of Insights from Multi-dimensional Data. In *Proceedings of the 2019 International Conference on Management of Data*, pp. 317–332, 2019. 2

[13] S. Kandel, A. Paepcke, J. Hellerstein, and J. Heer. Wrangler: Interactive Visual Specification of Data Transformation Scripts. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 3363–3372, 2011. 2, 4

[14] H. Kim, K. D. Le, G. Lim, D. H. Kim, Y. J. Hong, and J. Kim. DataDive: Supporting Readers' Contextualization of Statistical Statements with Data Exploration. In *Proceedings of the 29th International Conference on Intelligent User Interfaces*, pp. 623–639, 2024. 4

[15] P.-M. Law, A. Endert, and J. Stasko. Characterizing Automated Data Insights. In *2020 IEEE Visualization Conference (VIS)*, pp. 171–175. IEEE, 2020. 2, 4

[16] P.-M. Law, A. Endert, and J. Stasko. What are Data Insights to Professional Visualization Users? In *2020 IEEE Visualization Conference (VIS)*, pp. 181–185. IEEE, 2020. 2

[17] Y. Lin, H. Li, L. Yang, A. Wu, and H. Qu. Inksight: Leveraging Sketch Interaction for Documenting Chart Findings in Computational Notebooks. *IEEE Transactions on Visualization and Computer Graphics*, 2023. 2, 4

[18] P. Ma, R. Ding, S. Wang, S. Han, and D. Zhang. InsightPilot: An LLM-Empowered Automated Data Exploration System. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 346–352, 2023. 4

[19] A. McNutt, A. Crisan, and M. Correll. Divining Insights: Visual Analytics through Cartomancy. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, pp. 1–16, 2020. 1, 2

[20] Microsoft. Power BI Quick Insights. 1, 2

[21] C. North. Toward Measuring Visualization Insight. *IEEE Computer Graphics and Applications*, 26(3):6–9, 2006. 2

[22] Qlik. Qlik Sense Automated Insights. 1, 2

[23] B. Saket and A. Endert. Demonstrational Interaction for Data Visualization. *IEEE Computer Graphics and Applications*, 39(3):67–72, 2019. 2

[24] Salesforce. Einstein Insights. 2

[25] A. Satyanarayan, D. Moritz, K. Wongsuphasawat, and J. Heer. Vega-lite: A Grammar of Interactive Graphics. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):341–350, 2016. 4

[26] H.-J. Schulz, T. Nocke, M. Heitzler, and H. Schumann. A Systematic View on Data Descriptors for the Visual Analysis of Tabular Data. *Information Visualization*, 16(3):232–256, 2017. 2

[27] J. Sechler, L. Harrison, and E. M. Peck. Sightline: Building on the Web's Visualization Ecosystem. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, pp. 2049–2055, 2017. 4

[28] V. Setlur, A. Kanyuka, and A. Srinivasan. Olio: A Semantic Search Interface for Data Repositories. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pp. 1–16, 2023. 4

[29] A. Srinivasan, S. M. Drucker, A. Endert, and J. Stasko. Augmenting Visualizations with Interactive Data Facts to Facilitate Interpretation and Communication. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):672–681, 2018. 1, 2, 4

[30] N. Sultanum and A. Srinivasan. Datatales: Investigating the Use of Large Language Models for Authoring Data-driven Articles. In *2023 IEEE Visualization and Visual Analytics (VIS)*, pp. 231–235. IEEE, 2023. 4

[31] M. Sun, L. Cai, W. Cui, Y. Wu, Y. Shi, and N. Cao. Erato: Cooperative Data Story Editing via Fact Interpolation. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):983–993, 2022. 4

[32] Tableau. Tableau Explain Data. 1, 2

[33] Tableau. Tableau Pulse. 2

[34] ThoughtSpot. SpotIQ. 1

[35] Y. Wang, Z. Sun, H. Zhang, W. Cui, K. Xu, X. Ma, and D. Zhang. Datashot: Automatic Generation of Fact Sheets from Tabular Data. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):895–905, 2019. 1, 2, 4

[36] L. Weng, X. Wang, J. Lu, Y. Feng, Y. Liu, and W. Chen. InsightLens: Discovering and Exploring Insights from Conversational Contexts in Large-Language-Model-Powered Data Analysis. *arXiv preprint arXiv:2404.01644*, 2024. 4

[37] A. Wu, Y. Wang, X. Shu, D. Moritz, W. Cui, H. Zhang, D. Zhang, and H. Qu. Ai4vis: Survey on Artificial Intelligence Approaches for Data Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 28(12):5049–5070, 2021. 2