DESIGNING, DEVELOPING, AND DEMOCRATIZING GUIDANCE FOR VISUAL ANALYTICS

A Dissertation
Presented to
The Academic Faculty

By

Arpit Ajay Narechania

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Interactive Computing
College of Computing

Georgia Institute of Technology

December 2024

DESIGNING, DEVELOPING, AND DEMOCRATIZING GUIDANCE FOR VISUAL ANALYTICS

Thesis committee:

Dr. Alex Endert, Advisor
School of Interactive Computing
Georgia Institute of Technology

Dr. Clio Andris
School of City and Regional Planning, School of Interactive Computing
Georgia Institute of Technology

Dr. John Stasko
School of Interactive Computing
Georgia Institute of Technology

Dr. Shamkant B. Navathe
School of Computer Science
Georgia Institute of Technology

Dr. Duen Horng (Polo) Chau
School of Computational Science and Engineering
Georgia Institute of Technology

Dr. Mennatallah El-Assady
Department of Computer Science
ETH Zürich

Date approved: November 19, 2024

*"Life is a constant dance between one's **desire** and **destiny**, embrace it."*

– A life lesson I inferred from Sri Swami Sivananda[1]'s **guidance** to Dr. APJ Abdul Kalam[2], as quoted from the latter's autobiography (*Wings of Fire: An Autobiography* [1]):

"**Desire**, when it stems from the heart and spirit, when it is pure and intense, possesses awesome electromagnetic energy. This energy is released into the ether each night, as the mind falls into the sleep state. Each morning it returns to the conscious state reinforced with the cosmic currents. That which has been imaged will surely and certainly be manifested. You can rely, young man, upon this ageless promise as surely as you can rely upon the eternally unbroken promise of sunrise... and of Spring."

"Accept your **destiny** and go ahead with your life. You are not destined to become an Air Force pilot. What you are destined to become is not revealed now but it is predetermined. Forget this failure, as it was essential to lead you to your destined path. Search, instead, for the true purpose of your existence. Become one with yourself, my son! Surrender yourself to the wish of God."

[1] Yoga Guru and Hindu Spiritual Teacher.
[2] "Missile Man" and former President of India.

For my grandparents, Anilkant and Arvinda

For my parents, Ajay and Rita

For my wife, Nupur

ACKNOWLEDGMENTS

This dissertation exists thanks to a lot of people who, knowingly or not, became co-authors (not mere acknowledgements) in what was truly a memorable entry into academia.

My first and deepest gratitude goes to my advisor, Alex Endert. Alex, thank you for 'hijacking' me into the doctoral program, when I was one year into my masters program. You truly changed my life! It was an absolute honor to be your student and teaching assistant. Thank you for being patient with me all these years. I still remember your feedback on the first draft of the SafetyLens paper, *"You don't have to write so dramatically."* At that time, I had not *read* many research papers, let alone write one. I was also a very shy kid, but I think I have gotten over that, so thank you. I would next like to thank John Stasko, for reading my epically long email application in the Fall of 2018 to be his research assistant, then meeting me, and later actually recommending me to Rahul Basole. Rahul eventually hired me, and I was fortunate to assist him (and also work with John and his student, Arjun) on my first research project(s) for a year, before joining team Alex as his Ph.D. student.

Dissertation Committee. Now, as this journey comes to an end, I thank my committee members for their **guidance** in shaping this work – Alex Endert, John Stasko, Deun Horng (Polo) Chau, Clio Andris, Shamkant Navathe, and Mennatallah El-Assady. Based out of different continents and with diverse backgrounds and expertise, this was truly a 'dream team'. John questioned why guidance must only be 'visual', which made me explore multimodal guidance in BiasBuzz. Polo questioned what happens if the guidance is wrong, and Clio questioned how guidance from an individual differs from the crowd – both of which made me run a study to understand the effect of guidance attribution during analysis. Sham, with whom I built DataPilot and DataCockpit, helped me tighten the research goals and the contributions! Menna, with whom I built ProvenanceWidgets, questioned how guidance is more desirable than just 'automating analysis' and Alex similarly questioned if there are downsides to too much guidance – both of which I explored (to some extent) in Lighthouse.

Collaborators. Thank you to all my collaborators for working on exciting projects and co-authoring impactful papers with me [2, 13, 22, 23, 24, 25, 26, 27, 28, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 14, 15, 16, 17, 18, 19, 20, 21]: Rahul Basole, Ahsan Qamar, Biswajyoti Pal, Michael Corral, Matthew Meinhart, Prashanth Dintyala, Joy Arulraj, Alex Endert, Arjun Srinivasan, John Stasko, Adam Fourney, Bongshin Lee, Gonzalo Ramos, Arthita Ghosh, Deven Bansod, Su Timurturkan, Emily Wall, Jamal Paden, Adam Coscia, Alireza Karduni, Ryan Wesslen, Rishab Mitra, Clio Andris, Fan Du, Atanu R Sinha, Ryan A. Rossi, Jane Hoffswell, Shunan Guo, Eunyee Koh, Nedim Lipka, Alexa Siu, Shamkant B. Navathe, Vasanthi Holtcamp, Saurabh Mahapatra, John Anderson, Prithvi Bhutani, Sonali Surange, Shivam Agarwal, Surya Chakraborty, Ryan Colletti, Masrur Tajwar, Abbass Srour, Jomar Noceda, Gurman Minhas, Javed Padinhakkara, Kaustubh Odak, Mennatallah El-Assady, Subham Sah, Wenwen Dou, Shiyao Li, Roshini Deva, Cindy Xiong Bearfield, Hongye An, Kai Xu. A special thanks to Ahsan Qamar from Ford Motor Company for a solid collaboration of six years. A special thanks to Shamkant Navathe for mentoring me throughout my Ph.D. including during job search; I am honored to be the last student on whose dissertation committee you served pre-retirement. A special thanks to Atanu Sinha, Jane Hoffswell, Fan Du, and Shunan Guo for mentoring me during two summer internships at Adobe and for continued collaborations since then. A special thanks to Clio Andris for mentoring my during my Ph.D. minor, and with whom I worked on three of my most favorite projects.

Georgia Tech Visualization Lab. I thank current as well as alumni faculty members and friends of the Georgia Tech Visualization Lab, including Alex Endert, John Stasko, Deun Horng (Polo) Chau, Clio Andris, Rahul Basole, Ben Shapiro, Jessica Roberts, Yalong Yang, Cindy Xiong Bearfield, and especially Jim Foley, who always had golden advice in terms of feedback on paper drafts and brainstorming sessions during lab meetings.

Next, a big thanks to my labmates: Emily Wall, Arjun Srinivasan, Bahador Saket, John Thompson, Minsuk Kahng, Fred Hohman, Hannah Kim, Subhajit Das, Po-Ming "Terrance" Law, Haekyu Park, Hayeong Song, Grace Guo, Shenyu Xu, Ángel Alexan-

der Cabrera, Will Epperson, Jay Wang, Rishab Mitra, Aishwarya Mudgal Sunil Kumar, Arpit Mathur, Jamal Paden, Yu Fu, Sichen Jin, Alexander Bendeck, Seongmin Lee, Tao Lu, Alex Yang, Donny Bertucci, Chenyang Zhang, Minsuk Chang, Kylie Lin, Songwen Hu, Yishu Ji, Adam Coscia; and Qian Zhu and Kam Wong who were visiting students from HKUST. A special thanks to Arjun and Emily for being perfect mentors during my first year, for making me experience the extremes of paper submission – either seconds before the deadline or days before. I like both, but I think I'm getting too old for the 'last-minute' strategy now, haha! A special thanks to Grace, who joined the program with me, with whom I explored conference cities, and on whom I could always count on for "Tin Drum" and the 'latest lab chatter'. I've ordered '(Tofu) Tikka Masala' from Tin Drum so often that the server, Stephanie, now has it ready for me by name when I walk in to pick it up.

Family. I thank my parents for their encouragement, all the way from my home in India. It is because of their sacrifices, as well as those of my grandparents, that I received the education I did and am where I am today. I am also grateful to my wife, Nupur, for her support and sacrifices throughout this journey, and in particular, for enduring all the "Hi everyone", "Hello everyone", "Hey all", as I re-recorded talk/demo videos for eternity, never fully satisfied with any take. I am the first person in my family to receive a Ph.D. degree and also become a professor. I hope there are more folks opting for this route :).

Project Teammates. I thank Prashanth Dintyala (CS 4460), Ishaani Mittal and Joshua Culver and Jason Paul (CS 7450), Jihwan Oh and Karla Wagner (CSE 6010), Richard Henneman and Henry Deng and Ishaani Mittal and Lindsay Kelly (Lumovia), Nirmal Venkatachalam and Mayur Mahajan (CSE 6730), Shenyu Xu (CS 7001), Kuhu Gupta and Saloni Dalal and Deepthi Nagesh (CS 8803), Abhay Kumar and Aditi Shetty and Akash Kumar Pillai and Neha Naik (CS 6750), Shenyu Xu and Shaotung Sun and Jinyang Han (CSE 6140), Arpit Mathur (GT VIS Website Redesign), Grace Guo and Shenyu Xu (CSE 6740), Anastasia Shauer and Yan Zhang and Suood A Alroomi and Allison Wilkens (CETL 8717), Breanna Shi and Abigail Diering (CETL 8713), for making me meet course requirements.

Course Instructors. I thank Shamkant Navathe, John Stasko, Rahul Basole, Joy Arulraj, Richard Fujimoto, Richard Vuduc, Lauren Lukkarila, Hyesoon Kim, Annie Anton, Charles Byrd, Kurt Belgum, Alex Endert, Gervais Wafo Tapobda, Yongsung Lee, Elizabeth Mynatt, Frederic Faulkner, Xiuwei Zhang, Kate Williams, Jacki Rohde, Rodrigo Borela, Li Chao, and Srinivas Aluru, for training me from academic writing to machine learning.

Teaching Assistants: I thank Katie Dai, Oliver Zheng, Jessie Tepper, Paige Thompson, Aparna Arul, Kaustubh Odak, Justin Blalock, Emily Layton, Chloe Devre–all for CS4460.

Center for Teaching and Learning Colleagues: I thank my fellow Graduate Teaching Fellows (GTF), Graduate Teaching Assistant (GTA), Communications Manager, and Faculty: Anastasia Schauer, Marina Haldopaulos, Mehmet Akif Ağlar, Chandler Thornhill, Declan Abernathy, Maugan Lloyd, Becky Rafter, Alisha Vira, Bethany Harris, Kate Williams, Sarah Kegley, Tammy McCoy, and David Lawrence, from whom I learnt a lot about academia. Special thanks to Kate, Sarah, and Tammy for their guidance and support!

IT, Admin, Finance, HR folks: I thank the fantastic Tim Trent, and other College of Computing Helpdesk staff for helping me maintain Ocular (the Georgia Tech Visualization lab's server). I also thank Chrissy Hendricks, Theresa Nash, Monica Ross, Carolyn Daley-Foster, Philicia Bellinger, Gilberto Moreno, Sharina Richardson, Karen Rodrigues de Melo, and Eric Santacruz for always trying their best to ensure I got hired, enrolled for the necessary course credits, and applied for and received conference travel reimbursements, all in a timely manner (which, for some reason, due to my 'destiny', I rarely did).

Financial Support: I am grateful for the financial support from the National Science Foundation (USA) under grant award numbers IIS-1813281 and IIS-1750474, Ford Motor Company (USA) as part of their broader university-alliance grant, Adobe Research (USA, India) through gift funding, GVU Center (Georgia Tech) and CRIDC (Conference organized by Student Government Association, Georgia Tech) for conference travel funding, and Georgia Tech via stipends. Any views and conclusions contained herein are my own, and do not necessarily represent the official positions, express or implied, of the funders.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

The ubiquity and utility of data across a wide variety of domains – from science and technology to commerce, education, healthcare, and beyond – have created an urgent need for automated systems that help users make sense of large volumes of complex information. While these systems can process vast amounts of data, human intuition and expertise remain critical for many tasks, necessitating effective collaboration between the two (humans and systems) for accurate and timely decision-making. However, challenges arise when human users of these systems must provide extensive input (e.g., to convey their analytic intent) or when automated actions by systems misinterpret user intent or are mistimed, which can increase users' perceptual and cognitive load and disrupt the analytic process.

Guidance – or any kind of help, advice, support, suggestion, assistance, or recommendation – offers a promising solution to bridge this 'knowledge gap' between human expertise and (humans' understanding of) system capabilities, while improving the quality and effectiveness of the analysis process and its outcomes. Guidance during analysis has also been shown to boost user confidence and refine user expertise, while making the process more engaging and enjoyable. Building on information visualization (InfoVis), visual analytics (VA) and human-computer interaction (HCI) literature, this dissertation deepens our understanding of how **guidance** can be communicated to/from users and how it can impact users' behavior during analysis. Specifically, this dissertation makes contributions across three thrusts, with broader implications for researchers, developers, and practitioners alike:

1. **Design.** *Design spaces* for provenance and guidance communication, derived from a series of design interventions for guiding users during various data analysis tasks.
2. **Develop.** *Guidance-enriched systems*, developed for and evaluated with end-users, revealing strengths and challenges, and informing the development of future systems.
3. **Democratize.** A *library of guidance-enriched user interface (UI) controls*, released as open-source software, helping developers prototype custom guidance systems.

CHAPTER 1
INTRODUCTION

The ubiquity and utility of data today has transformed processes across a wide variety of domains: technology companies analyze product usage patterns to inform future releases; medical researchers examine patients' biological markers in clinical trials to understand the effectiveness of a treatment; universities conduct outreach programs to meet diversity targets based on past student admission trends; and retailers make decisions about the distribution of products to stock based on past sales trends. Machines, through their superior computational power and working memory, can support many of these processes (e.g., predicting future product sales using machine learning models) but humans' dominant perceptual capabilities and adaptive analytic skills are equally desirable, especially when the stakes are high and/or where domain expertise is essential (e.g., decision-making during military operations). This necessitates the need for human-in-the-loop approaches to data analysis that build upon the strengths of both humans and machines [29].

Visual Analytics (VA) is one such human-in-the-loop approach that *"combines automated analysis techniques with interactive visualizations for an effective understanding, reasoning and decision-making on the basis of very large and complex data sets"* [30]. However, while interacting with VA systems, several phenomena may occur that can endanger the analysis outcome and the user experience. For instance, automated computations may require users to provide a lot of feedforward (e.g., manually configure input parameters [31]) as systems are still poor at guessing the users' needs. Furthermore, the processes and outputs of these computations may still need to be explained to the user to instill trust and enable subsequent refinement (of suboptimal outputs), the communication of which – through various modalities – may overload users' cognitive and perceptual memory.

Increasingly, VA systems have adopted concepts of mixed-initiative systems to *"enable*

users and intelligent agents to collaborate efficiently" [32] by sharing agency and control during analysis. A key idea is that these systems can infer a user's intent and take initiative on the user's behalf (and vice-versa). For example, following the 'human *is* the loop' viewpoint for VA [33], the system implicitly recognizes analysts' workflows from their interaction history, and naturally integrates analytics into their ongoing workflow [34]. However, interacting with these mixed-initiative systems can also deter the analysis process. For instance, automated actions by the system may be premature or flawed as the system's interpretation of the user's intent might be incomplete or incorrect, requiring users to provide extensive feedback (e.g., reject or downvote/unlike the output) and feedforward (e.g., configuring input parameters). Furthermore, the nature of the system's actions may also throw the user off their analysis (e.g., sudden popup notifications that block the UI); and a combination of these phenomena may require the user and the system to engage in an efficient dialog to minimize their "knowledge gap" and ensure proper analytic progress.

Guidance in VA is one such *"computer-assisted process that aims to actively resolve the knowledge gap between the user and the system during an interactive analysis session"* [35, 36]. Guidance also aims to help enhance analysis efficiency, validate insights, boost user confidence, refine user expertise, and increase awareness of and prevent biases – all while making the process more engaging and enjoyable [37]. Simply put, guidance is the act of helping somebody reach a goal by enhancing their skills and competencies, and enriching their journey toward that goal. We have all sought and received such guidance in life. For example, parents teach us to ride a bike, teachers hone our skills, and supervisors support career growth. Similarly, objects like compasses and signposts indicate direction, while software assistants like Microsoft's Clippy [38] and robots like Tesla's Optimus [39] provide task assistance. Even movie characters such as *J.A.R.V.I.S* in 'Iron Man' and television show aids like the *lifelines* in 'Who Wants to be a Millionaire?' offer help. Figure 1.1 and Figure 1.2 show additional examples in the real and reel worlds.

Scientific literature on guidance in the fields of visualization and human-computer inter-

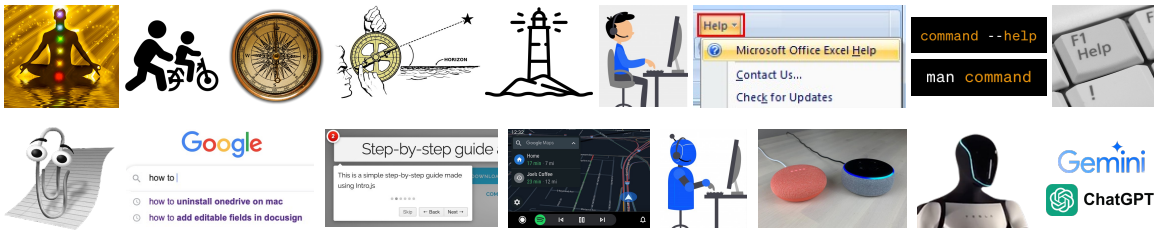


Figure 1.1: Brief timeline illustrating how humankind has either received or sought some kind of guidance from god (via prayers), family, map and compass, celestial phenomenon, lighthouse, human agent, software menu, command line-based documentation, 'F1' keyboard key, Clippy [38], Google Search, website onboarding tour, turn-by-turn navigation, chatbot, home assistant, humanoid robot (e.g., Sofia), and language model (e.g., ChatGPT).

Figure 1.2: Characters and/or objects from movies and television shows that provide(d) guidance, e.g., Master Shifu (Kung Fu Panda), Yoda (Star Wars), Mr. Miyagi (The Karate Kid), the broken East Dock signpost (Jurassic Park), Sonny (I, Robot), Tars (Interstellar), Mufasa (The Lion King), The Marauder's Map (Harry Potter and the Prisoner of Azkaban), Map (Dora the Explorer), Tia Dalma's compass (Pirates of the Caribbean: The Curse of the Black Pearl), J.A.R.V.I.S. (Iron Man), and the 'lifelines' (Who Wants to Be a Millionaire?).

action (HCI) has continuously evolved through definitions, theories, models, frameworks, techniques, tools, libraries, and empirical evaluations, all aimed at increasing our understanding of and optimizing how guidance supports analytical processes. As a matter of fact, guidance was previously referred to as "help," "tip," "advice," "support,", "assistance," or "recommendation," until Schulz et al. [40] grouped them under an umbrella term, "guidance." Schulz et al. [40] also characterized guidance for visualization comprising four aspects: *context* or the user's prior knowledge, *domain* or the basis of guidance, *target* or the goal of guidance, and *degree* or the amount of guidance. Ceneda et al. [35, 36] then proposed a conceptual model of guidance for VA, characterizing guidance based on the *knowledge gap* of users, the *input* and the *output* of a guidance generation process, and

the *degree* to which (or amount of) guidance is provided. Collins et al. [37] proposed a more practical model of guidance, incorporating just-in-time "facilitation" that addresses not only *where* and *what* type of guidance can be provided in the analysis process but also *how* it can be effectively implemented. Ceneda et al. [41] then proposed a guidance framework for designers comprising requirements, phases, and quality criteria to design effective guidance systems. Pérez Messina et al. [42] expanded an existing model of knowledge generation [43], focusing on the interaction between users and guidance systems, and developed a typology of tasks to better understand and evaluate current guidance systems. Next, Sperrle et al. [44, 45] applied concepts of mixed-initiative user interfaces [32] (UI) into VA, introducing the concept of a "co-adaptive" guidance process, wherein the user and the system teach and learn from one another during analysis [46]. Sperrle et al. [47] also provided a practical guidance framework, including an open-source library (Lotse) for developers to design custom guidance strategies for their own tools. Additionally, systematic literature reviews [48, 49, 50] have served as key checkpoints, examining existing literature and revealing opportunities for future research. Informed by prior work, this dissertation aims to extend literature on co-adaptive guidance processes [44] by building mixed-initiative UIs that efficiently minimize the knowledge gap between the user and the system [35] through an intuitive and enjoyable [37] guidance dialog with the system, while ensuring accurate analytic outcomes. In doing so, this dissertation aims to fill some gaps in existing work and also introduce fresh ideas for deeper exploration, as described next.

First, Ceneda et al. [35, 36] characterized guidance into three degrees: *orienting* (basic guidance through visual cues), *directing* (useful alternatives that the user may choose to follow), and *prescribing* (an automated process which proceeds towards a specified target along a 'best' path). However, from Ceneda et al. [48]'s review of guidance approaches in visual data analysis, and to the best of our knowledge, there is no system prototype or testbed that offer all three degrees of guidance while facilitating a dynamic transition between them during analysis. The ability to provide different degrees (or amounts) of guidance

at different points in time during analysis is essential because the knowledge gap between the user and the system is always evolving. A new user might require basic, continuous guidance early into their analysis than an experienced/expert user, who might require more on-demand, specialized guidance. So I asked: *How can we design mixed-initiative UIs that seamlessly adapt and transition between different guidance degrees during analysis?*

Second, assuming the system is able to transition between different guidance degrees, how can it automatically compute the appropriate degree during analysis? Is it based on the user's task, expertise, or preferences (e.g., some users may just like orienting guidance more, as it may be less distracting)? Or, is it more data- and model-driven, wherein the system continuously quantifies the user's knowledge gap, by processing their interactions with the system, to determine an appropriate degree of guidance (e.g., larger the gap, higher the degree of guidance to bring the user back on track, quickly)? Additionally, how can the user explicitly provide contextual information (feedforward) or respond to the system's guidance (feedback) for the system to implicitly infer the same? So I asked: *How can the system capture and adapt to user intent (through feedforward or feedback) and take initiative on the user's behalf to facilitate an effective co-adaptive guidance process?*

Third, Zhou et al. [50]'s state-of-the-art review, of how visualization systems surface content recommendations (i.e., provide guidance) to users during visual analysis, introduces a four-dimensional design space consisting of the *Directness* (context), *Forcefulness* (level of intrusiveness), *Stability* (timing), and *Granularity* (content) of the guidance recommendations. While this work characterizes prior systems based on the four dimensions, to the best of our knowledge, there is no design space for consistently visualizing and interacting with different degrees of the same guidance in the UI. For instance, how can common UI elements, including the visualizations, UI controls (e.g., range sliders), and even custom views (e.g., a "Guide Me" panel) present the same guidance via *orienting*, *directing*, and *prescribing* degrees [35]? Additionally, must guidance always be about a *future* action to perform (e.g., interact with a specific datapoint)? Can it be about showing

the *present* analysis state or the *problem* with it (e.g., an underemphasized or uninteracted datapoint), which then nudges *future* action(s)? So I asked: *How can we model a guidance state space and generalize its communication via different UI elements in VA systems?*

Lastly, my review of visualization and HCI literature revealed that most guidance systems are bespoke implementations for specific use-cases, with limited reusability. There are few open-source tools for building guidance systems, limiting broader access, experimentation, and adoption by researchers, developers, and practitioners alike. So I asked: *"How can we equip people with tools to build their own custom, guidance-enriched systems?"*

Evidently, these questions cannot be answered solely by building a single prototype system and conducting few experiments using it. Hence, as part of this dissertation, we designed and developed a series of mixed-initiative UIs enriched with co-adaptive guidance to investigate the role and utility of guidance in enhancing VA processes and outcomes. We also derived design spaces and built open-source tools, democratizing access by enabling developers to build custom guidance-enriched systems. Figure 1.3 serves as an overview of this dissertation, which is further detailed in subsequent (sub)sections.

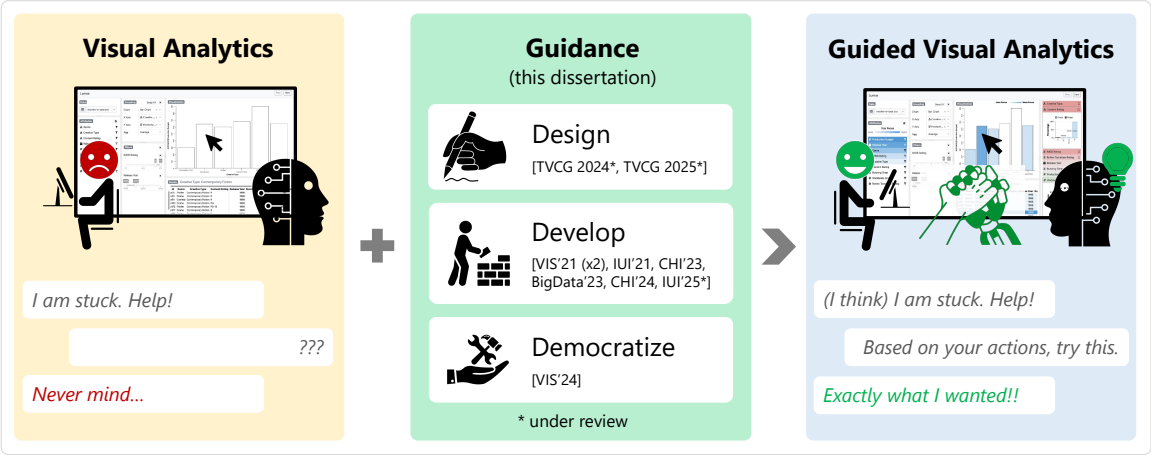


Figure 1.3: Dissertation overview: To study how guidance can enhance Visual Analytics (VA) processes and outcomes. When interacting with VA systems, there is a 'knowledge gap' between (1) users' abilities, analytic goals, and understanding of the system's capabilities and (2) the system's interpretation of the same. Guidance can help bridge this gap, aligning users and systems to collaboratively achieve the analytic goals, while also boosting users' confidence and ensuring an engaging and enjoyable overall experience for them.

1.1 Thesis Statement and Research Goals

The work as part of this dissertation is captured by the following thesis statement:

> ***"Facilitating co-adaptive guidance in mixed-initiative user interfaces, wherein the user and the system learn from and take initiatives on behalf of each other, enhances human-data interaction experiences as well as analytic processes and outcomes, while promoting the design of new tools that broaden access for researchers, developers, and practitioners alike."***

To validate the above statement, I break it down into four research goals (**RG1**–**RG4**), listed in Table 1.1 along with corresponding chapter numbers and associated publications:

RG1 *Investigate the role of guidance in enhancing analytic processes and outcomes in various data preparation and analysis workflows* (Chapters 3, 4).

A key question this dissertation aims to address is how to design mixed-initiative, co-adaptive guidance systems that adjusts the level of guidance based on the analysis needs and user preferences. I derived this goal to first study the different levels of guidance in isolation, in different analysis contexts, to understand their pros and cons.

To achieve this goal, we first built a visual data preparation system (**DataPilot**) that provisions guidance to enhance subset selection workflows. Essentially, this system models data quality insights (e.g., number of missing values) and data usage insights (e.g., how often was the data used and where) from large, unfamiliar tabular datasets to help users select effective subsets for use in downstream applications. These insights are presented as visual cues (to orient the user towards the good and bad aspects of data) as well as via interaction affordances such as filtering and sorting (to help users reduce and organize their search space). A user evaluation revealed that providing such visual and interactive affordances helps users select smaller, effective data subsets with greater success and confidence; however, to balance exploration

versus exploitation, caution must be exhibited about excessively relying on usage information. Extending DataPilot, we also built **DataCockpit**, an open-source Python toolkit with a visual monitoring tool that provides quality and usage insights for navigating and monitoring data lakes (a collection of multiple relational databases).

Next, we built a question-answering system, enhanced with an interactive, self-service debugging view (**DIY**), to help users interactively debug (i.e., inspect for, isolate, and fix errors in) the responses of a natural language (NL) to SQL model. This system provides end-users with a test-bed wherein they can interact with (1) the mappings between the question and the generated SQL query, (2) a small-but-relevant subset of the underlying database, and (3) a multimodal explanation of the generated query. End-users then employ a back-of-the-envelope calculation debugging strategy, e.g., manipulate the sample database to verify the system's strategy (serving as a proxy for executing the query on the production database) and fix errors by selecting the correct mappings. Unlike DataPilot users, who rely on system-generated guidance to select data subsets, *DIY users guide themselves* by posing *what-if* queries on the sample database to verify the system's response and establish trust. An exploratory user evaluation revealed the benefits of using DIY, including a variety of debugging strategies to assess the correctness of the system's responses.

RG2 *Design a mixed-initiative guidance system, wherein the user and the system learn from and take initiative on behalf of each other, co-adaptively steering the analytic process* (Chapters 5, 6).

The user evaluations of DataPilot and DIY revealed many pros and cons of communicating guidance. In particular, I noticed a need for (1) users to convey their intents and preferences (for the system to provide contextual guidance) and also for (2) systems to automatically infer the same and take initiatives by intervening depending on the analysis state and progress (e.g., by providing less/more guidance). To investigate

these aspects, I derived this research goal to design *mixed-initiative* systems wherein the user and the system continuously guide (and adapt to) each other during analysis.

To achieve this goal, we designed a new visualization technique ("interaction traces") to model and present the history of a user's *interactions* with a visualization system, to increase real-time awareness of biased analytic behaviors. Studying user interactions – or the ways in which users engage with and manipulate data visualizations [51] – is a dedicated area of visualization and HCI research called *analytic provenance* [52, 53]; it builds upon the database community's research area focused on *data provenance* (or data lineage), which tracks the origin, processing, and transformation history of data [54]. This dissertation employs users' analytic provenance as one of the foundations for provisioning guidance as it is a 'solid' way for the system to naturally and continuously learn about the users' intents and preferences.

Leveraging the technique of interaction traces, we first built a mixed-initiative visual data analysis system (**Lumos**), wherein we (1) colored already visited points in a visualization and (2) compared the distribution of user interactions to the underlying distribution of the data (to determine if the user over- or underemphasized). Additionally, we enabled users to compare their own focus against configurable baselines (i.e. target interaction behaviors), such as (1) the underlying data distribution, (2) an equal distribution (where each data item is expected to receive equal focus), or (3) a custom distribution (to satisfy a specific work requirement). By comparing users' focus against a target, the system then provides contextual guidance, facilitating a co-adaptive guidance dialog with the user. A user evaluation revealed that interaction traces increased people's awareness of analytic behaviors, often prompting self-reflection that sometimes changed subsequent interactions. A second user evaluation studied how interaction traces help mitigate human biases (e.g., gender bias) during decision-making, also suggesting they can help promote conscious reflection on decision-making strategies, but more such studies are needed for conclusive results.

Next, we enhanced Lumos with multimodal guidance affordances, wherein the system (**BiasBuzz**) processes the user's analytic behavior, and depending on the magnitude of its 'knowledge gap' with the user, provisions continuously evolving ('adaptive') guidance. Essentially, the user first specifies one or more data attributes to track (for subsequent bias mitigation); in response, the system provides visual guidance until a certain threshold of the knowledge gap, exceeding which, the system takes initiative and triggers an additional haptic stimulus (by vibrating a haptic mouse) to capture user's attention with the hope to instantly bring them 'back on track'. A user evaluation revealed that the dual guidance modalities (visual + haptic) can increase analytical awareness in some cases, but the haptic mouse vibrations can be distracting and disturbing, putting into context the design of such multimodal guidance systems.

RG3 *Establish a design space for guidance communication during analysis* (Chapters 7, 8).

Based on the designs and findings from subsequent user evaluations of Lumos and BiasBuzz, I identified commonalities in how the systems provided guidance and how users responded to it. I derived this goal to generalize these guidance designs, providing a foundation for building consistent and intuitive guidance-enhanced systems.

To achieve this goal, we established two design spaces: one for communicating provenance during analysis, and another for doing the same but for guidance. The provenance design space covers how users can visualize and interact with provenance via encodings and data transformations. This (provenance) design space sets the foundation for the guidance design space, which introduces additional concepts: (1) guidance *wildcards*, (2) a *state* space of guidance, and (3) different *levels* (or amount) of system intervention for the same guidance recommendation.

First, we modeled analytic provenance as an attribute that is available to users during analysis. We demonstrated this concept via two *provenance attributes* that track the *recency* and *frequency* of user interactions with data. We integrated these attributes

into a visual data analysis system (**ProvenanceLens**) wherein users can visualize their interaction recency and frequency by mapping them to one or more encoding channels (e.g., color, size) or applying data transformations (e.g., filter, sort).

Second, we introduced the concept of *wildcards*, defined as "entities about which guidance is provisioned," e.g., data quality and usage (used in DataPilot, DataCockpit) or interaction frequency and recency (used in Lumos, BiasBuzz, ProvenanceLens). These wildcards track four kinds of analysis *states*: previous state (past), current state (present), problem with the current state (problem), and the system-determined future state (future). These states represent different aspects associated with the same guidance recommendation, as derived from the basic idea of the 'knowledge gap'. For instance, when the system determines the user must next perform a specific interaction, we refer to it as the *future* state. This system recommendation is based on the 'knowledge gap', which we refer to as the *problem* state, wherein the system provides guidance about the problem (not the solution). Next, we refer the user's current state as the *present* state. Lastly, we also model a *past* state to guide users about their previous state. Together, these states cover the entire spectrum associated with a guidance recommendation. We integrated these guidance wildcards and states into a visual data analysis playground (**Lighthouse**) that additionally offers different *levels* (or amounts of) of guidance through adaptive UI elements: visualizations (via visual encodings), UI controls (via data transformations such as filtering and sorting), and custom panels (via text-based NL explanations).

Based on the analytic needs as deemed automatically by the system or determined by the user's expertise and preferences, these UI elements provision different levels of guidance, making Lighthouse the first adaptive guidance system in visualization literature. We demonstrated the utility of Lighthouse system through usage scenarios covering all aspects of the introduced guidance design space.

RG4 *Create tools to help developers build custom guidance-enriched systems* (Chapter 9).

Building the above guidance-enhanced systems demanded significant development skills and time. To enable broader participation in this process, I derived this goal to develop tools to help other developers (easily) build custom guidance systems.

To achieve this goal, we built a JavaScript library of enhanced UI controls – such as sliders and dropdowns – that track and dynamically overlay analytic provenance, in situ and out-of-the-box (**ProvenanceWidgets**). By showing the user what they have done so far, these widgets can make the user reflect upon their present choices to influence subsequent ones. Additionally, if these widgets are preconfigured to show customized information (e.g., interaction behavior of peers), they can be used to nudge users in specific directions (e.g., interact with previously overlooked aspects). This library is open-source, customizable, and universally compatible, enabling developers to integrate provenance-tracking into existing systems and/or prototype new systems. Additionally, because provenance is often a basis for providing guidance, the library can be used to prototype custom guidance-enriched systems. We demonstrated the library's utility by replicating the core functionalities of three prior widget libraries and also conducting a developer-focused evaluation.

In summary, I first investigated the role of guidance in improving analysis workflows (**RG1**). Next, I built mixed-initiative guidance systems where the user and the system 'work together' (**RG2**), from which I derived design spaces (**RG3**). Lastly, to broaden access, I built an open-source library of UI controls for tracking and visualizing provenance (**RG4**).

1.2 Contributions

Through this dissertation, I make the following primary contributions to visualization and human-computer interaction (HCI) literature, with complementary contributions to database, ubiquitous computing, deep learning, and artificial intelligence literature:

Design, Implementation, and Evaluation of Techniques and Systems

- A data preparation system that presents data quality & usage metrics to guide users in selecting effective subsets from large, unfamiliar datasets (DataPilot [4], chapter 3).
- A mixed-initiative visual data analysis system, that presents real-time visual traces of a user's interactions ("interaction traces"), to increase awareness of biased analytic behaviors against configurable target analytic behaviors (Lumos [27], chapter 5).
- A mixed-initiative system that provides multimodal guidance (visual + haptic) to mitigate biased analytic behaviors during data analysis (BiasBuzz [7], chapter 6).

Design, Implementation, and Evaluation of System Test-beds / Playgrounds

- A question-answering system, integrated with an interactive, self-service debugging view, to help users debug natural language to SQL workflows (DIY [26], chapter 4).
- A visual data analysis system as a test-bed for demonstrating (and studying) the design spaces for provenance communication (ProvenanceLens [11], chapter 7).
- A visual data analysis system as a test-bed for demonstrating (and studying) the design spaces for guidance communication (Lighthouse [12], chapter 8).

Empirical Evaluations

- A series of in-lab and crowdsourced studies to understand how human biases (e.g., gender) impact the way people make decisions during analysis (Left, Right, and Gender [25], section 5.4). We found some evidence that "interaction traces" can increase awareness of unconscious biases, but additional confirmatory studies are needed.
- A crowdsourced study [10] to understand how the source of guidance–such as AI model or human expert–impacts people's perception and usage of guidance during analysis (chapter 10). We found that the source of guidance matters to users, but not in a manner that matches received wisdom; users utilize guidance differently, expressing varying levels of regret, despite receiving guidance of similar quality.

Design Spaces

- A design space for communicating analytic provenance by modeling it as an attribute, and mapping it to visual encodings and data transformations during analysis (ProvenanceLens [11], chapter 7).
- A design space for communicating guidance by modeling it as a *state*-space (past, present, problem, future) and presenting different *levels* (e.g., 1, 2, 3) via adaptive UI elements–visualizations, UI controls, external panels (Lighthouse [12], chapter 8).

Open-Source Libraries and Toolkits

- A Python toolkit that helps developers compute data quality and usage information from data lakes, along with a companion data visualization system to guide database administrators to navigate and monitor data lakes (DataCockpit [5], chapter 3).
- A JavaScript library of UI controls that helps developers prototype custom web applications with provenance-tracking (ProvenanceWidgets [9], chapter 9).

1.3 Associated Publications and Attributions

The content of this dissertation is, in part, based on manuscripts either previously published or under review at different venues (associated publications are listed in Table 1.1).

Even though I am the principal author of this dissertation, the associated publications are the result of collaborations with my advisor, Alex Endert, as well as mentors and colleagues at Georgia Tech, Emory University, Microsoft Research, Adobe Research, and ETH Zürich. I was the lead author of all but two publications, which I co-led with different colleagues: Emily Wall (former PhD student at Georgia Tech) and I co-led the "Left, Right, and Gender" project described in chapter 5. Jamal Paden (former Bachelor's student at Georgia Tech) and I co-led the "BiasBuzz" project described in chapter 6.

To acknowledge the collaborative efforts behind this dissertation, I will use "We" in applicable sections and "I" when referring to my own thoughts.

Table 1.1: Dissertation outline and publication summary. An additional relevant publication – x. **Narechania, A.**, Endert, A., Sinha, A. *"Guidance Source Matters: How Guidance from AI, Expert, or a Group of Analysts Impacts Visual Data Preparation and Analysis."* (under review) – is briefly described in chapter 10.

Research Goals	**Chapter**	**Publication(s)** [*Equal Contribution]
RG1 *Investigate the role of guidance in enhancing analytic processes and outcomes in various data preparation and analysis workflows.*	chapter 3	i. **Narechania, A.**, Du, F., Sinha, A. R., Rossi, R. A., Hoffswell, J., Guo, S., Koh, E., Navathe, S. B., Endert, A. *"DataPilot: Utilizing Quality and Usage Information for Subset Selection during Visual Data Preparation."* ACM CHI, 2023. ii. **Narechania, A.**, Chakraborty, S., Agarwal, S., Sinha, A. R., Rossi, R. A., Du, F., Hoffswell, J., Guo, S., Koh, E., Endert, A., Navathe, S. B. *"DataCockpit: A Toolkit for Data Lake Navigation and Monitoring Utilizing Quality and Usage Information."* IEEE BigData, 2023.
	chapter 4	iii. **Narechania, A.**, Fourney, A., Lee, B., Ramos, G. *"DIY: Helping People Assess the Correctness of Natural Language to SQL Systems."* ACM IUI, 2021.
RG2 *Design a mixed-initiative guidance system, wherein the user and the system learn from and take initiative on behalf of each other, co-adaptively steering the analytic process.*	chapter 5	iv. **Narechania, A.**, Coscia, A., Wall, E., Endert, A. *"Lumos: Increasing Awareness of Analytic Behavior during Visual Data Analysis."* IEEE VIS, 2021 (IEEE TVCG, 2022). v. Wall, E.*, **Narechania, A.***, Coscia, A., Paden, J., Endert, A. *"Left, Right, and Gender: Exploring Interaction Traces to Mitigate Human Biases."* IEEE VIS, 2021 (IEEE TVCG, 2022).
	chapter 6	vi. Paden, J. *, **Narechania, A.** *, and Endert, A. *"BiasBuzz: Combining Visual Guidance with Haptic Feedback to Increase Awareness of Analytic Behavior during Visual Data Analysis."* ACM CHI (LBW), 2024.
RG3 *Establish a design space for guidance communication during analysis.*	chapter 7	vii. **Narechania, A.**, Guo, S., Koh, E., Endert, A., Hoffswell, J. *"Utilizing Provenance as an Attribute during Visual Data Analysis Promotes Self-Reflection: A Design Probe with ProvenanceLens."* (under review).
	chapter 8	viii. **Narechania, A.**, Guo, S., Koh, E., Endert, A., Hoffswell, J. *"Lighthouse: A Design Space for Guidance Communication during Visual Data Analysis."* (under review).
RG4 *Create tools to help developers build custom guidance-enriched systems.*	chapter 9	ix. **Narechania, A.**, Odak, K., El-Assady, M., Endert, A. *"ProvenanceWidgets: A Library of UI Control Elements to Track and Dynamically Overlay Analytic Provenance."* IEEE VIS, 2024 (IEEE TVCG, 2025).

CHAPTER 2

RELATED WORK

2.1 Information Visualization

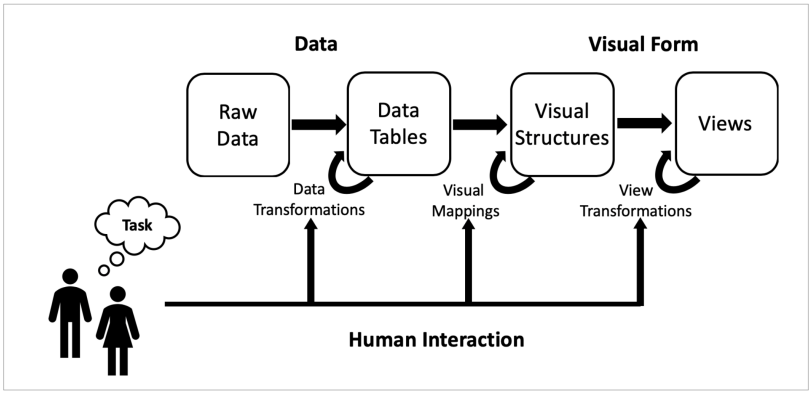


Figure 2.1: The InfoVis Reference Model by Card, Mackinlay, and Shneiderman [55].

Information Visualization (InfoVis) is a technique for simplifying complex data by creating visual representations of abstract concepts, processes, or datasets [56]. In doing so, users can reduce cognitive load, allowing improved reasoning during analytical tasks. Card, Mackinlay, and Shneiderman [55] introduced the first InfoVis reference model, which outlines a step-by-step process for transforming abstract data into an interactive visual form to solve a given task (Figure 2.1). In this model, "Raw Data" is first transformed into a computer-readable format such as "Data Tables". Next, visual marks and encodings, such as color and size, are mapped to this data to create "Visual Structures". Finally, one or more such visual representations are arranged and transformed into "Views" to make relevant information easily accessible. Interaction is integral to this entire process, enabling users to adjust each step of the model to influence the final outcome. Although InfoVis generally aids visual data exploration, it can struggle with the scale and complexity of modern datasets, necessitating more automated approaches, as described next.

2.2 Visual Analytics

To overcome issues associated with the scale and complexity of modern datasets, data mining [57] techniques such as clustering, dimensionality reduction, and anomaly detection can be applied. However, even these algorithmic solutions, while fast and accurate, often produce black-box outputs that are difficult to interpret, necessitating perceptual tools such as interactive visualizations that enhance human cognition and facilitate decision-making.

Recognizing the superior computational power and memory of machines alongside the perceptual and adaptive analytic skills of humans, researchers adopted human-in-the-loop approaches to leverage the strengths of both. Visual Analytics (VA) is one such human-in-the-loop approach that combines automated analysis techniques with interactive visualizations for an effective understanding, reasoning, and decision-making based on large, complex data sets [58, 30]. VA systems incorporate concepts from mixed-initiative systems to *"enable users and intelligent agents to collaborate efficiently"* [32] by taking initiatives on behalf of each other during analysis. Recently, VA systems have also embraced a 'human *is* the loop' perspective–centralizing the role of the user–by enabling the system to implicitly infer the user's workflow(s) and seamlessly integrating analytics [33].

VA researchers have proposed various process models [59, 58, 60, 29, 43, 61, 62] that describe how humans gather information, draw conclusions, formulate and evaluate hypotheses, extract evidence, and refine their understanding during analysis–also known as the "sensemaking" process. Although details vary, all VA models (implicitly) include the three key components of InfoVis models: (1) data, (2) human, and (3) visualization, along with a fourth component, (4) analytic model, which facilitates the sensemaking process.

Pirolli and Card [59] first studied the sensemaking process by performing a cognitive task analysis with intelligence analysts (Figure 2.2). They proposed that the sensemaking process could be roughly described by two loops: (1) a foraging loop to search for information and (2) a sensemaking loop to resolve an understanding of the information. Each of

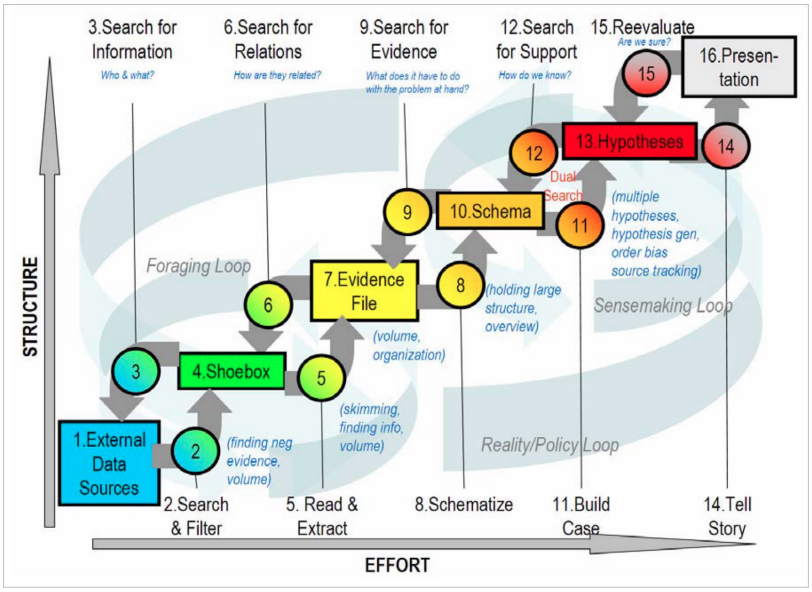


Figure 2.2: The Sensemaking Process Model by Pirolli and Card [59].

these higher-level processes is then decomposed into a series of cognitive actions, e.g., the foraging loop involves iteratively finding evidence from external data sources, compiling the evidence, and then skimming it to look for relevant information.

Alternatively, Klein et al. [60] studied the sensemaking process as an iterative framing and re-framing of information (Figure 2.3). They postulated that when examining data, analysts begin with some frame of reference and then continuously compare, refine, and create new frames throughout the analysis to refine their understanding of the data.

Next, Green et al. [29] presented a human cognition model that details the often-complex relationship between humans and computers during the VA processes of knowledge creation and hypothesis generation, describing how tasks and information should be distributed across the two to leverage their complementary strengths (Figure 2.4).

Next, Keim et al. [61] presented the VA Reference Model, describing the relationships between data, visualization, models, and knowledge (Figure 2.5). Sacha et al. [43] later extended this model to describe the process of knowledge generation in terms of the related roles of the human and the computer. This extended model consists of (1) an *Exploration Loop* for describing how findings are extracted from the data, thanks to exploration activ-

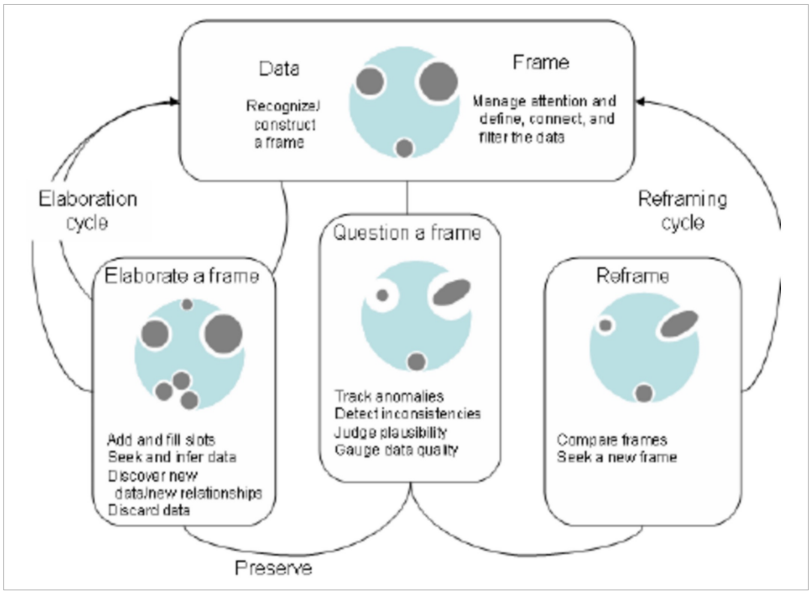


Figure 2.3: The Data-Frame Model of Sensemaking by Klein et al. [60].

ities; (2) a *Verification Loop* where the findings are grouped together to prove or disprove any working hypotheses and form insights; and (3) a *Knowledge Generation Loop* where the insights are condensed into new knowledge (Figure 2.6).

Recently, Booth et al. [62] surveyed nine process models from the VA and human-computer interaction (HCI) literature and presented a granular, descriptive model of human decision-making in VA. They examined the humans and computers presented in the models (entities), the divisions of labor between the entities (both physical and role-based), the behavior of the entities as constrained by their roles and agency, and the elements and processes which define the flow of data both within and between entities.

We drew inspiration from these models and applied them in various ways to design and develop a series of guidance-enriched VA systems as part of this dissertation.

2.3 Analytic Provenance

Data provenance (or data lineage) documents the history of a data item, including its source, the processes it has undergone, and any transformations applied [54]; as a type of metadata, it ensures data authenticity and enhances its reusability. In the context of information visu-

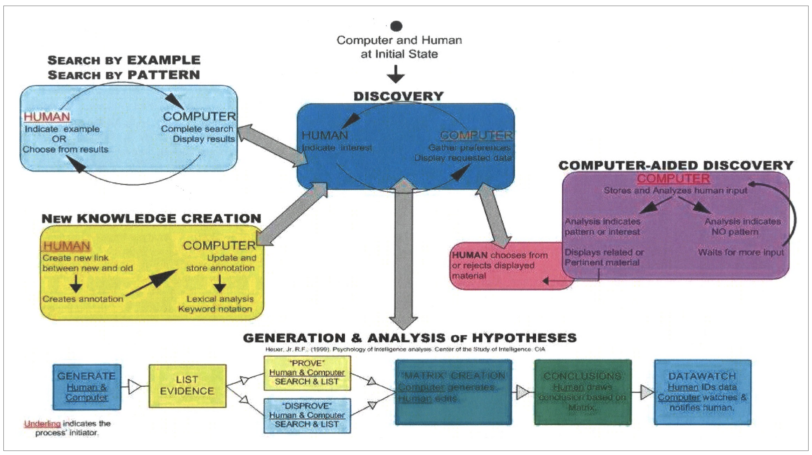


Figure 2.4: The Human Cognition Model by Green et al. [29].

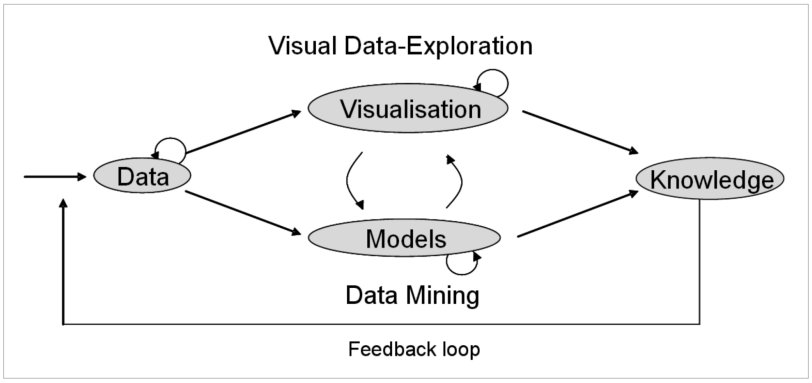


Figure 2.5: The VA (Visual Analytics) Reference Model by Keim et al [61].

alization and visual analytics (VA), this concept extends to capturing **interaction**–or how users engage with and manipulate data visualizations [51]. Interactions may occur through input devices like keyboards and mice [63, 64] or through modalities including speech [65], touch [65], eye gaze [66], hand gestures [67], and facial expressions [68], among others. This dissertation focuses on interactions via input devices (keyboards, mice)–such as typing, clicking, hovering, zooming, panning, and brushing–in a web browser-based user interface. Through such interactions, users can advance a visualization from one state to the next to effectively navigate and sensemake complex information [29] and facilitate human reasoning and decision-making [69]. Interaction data also contains rich information about users, such as their task performance and personality traits, that can help refine systems [70]. Because our memory has a finite capacity to track and remember everything [71,

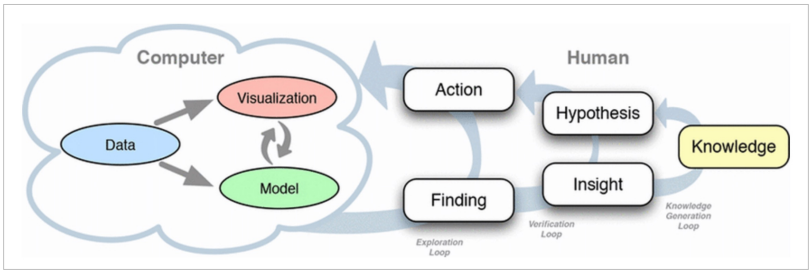


Figure 2.6: The Knowledge Generation Model by Sacha et al. [43].

72], interaction data can also be used as a record of the analysis process. An entire area of visualization and HCI research, known as analytic provenance, studies how interaction data with visualizations can be used to enhance analysis [52, 53], as described next.

Ragan et al. [53] present an organizational framework comprising five types of provenance information–data, visualization, interaction, insight, rationale–and six purposes for why they are desired in the field of visual analytics–recall, replication, action recovery, collaborative communication, presentation, meta-analysis (Figure 2.7). Xu et al. [49]'s survey notes that provenance has been used to support sensemaking [73, 74] and decision-making [75] with data, evaluate the usefulness of visualization systems [76, 77], design adaptive systems [35, 78], improve the performance of machine learning models [79], replay or replicate analysis sessions [80, 81], and automatically generate summary reports of an analysis session [82, 83]. Provenance has also been shown to enhance analysis through more unique data discoveries [84, 85], improved confidence [86] and inspiration [87] levels, increased contextual awareness of previously visited data [88] and recall [87].

Today, provenance tracking occurs in data analysis tools [52], code editors [89], computational notebooks [90, 91], workflow modeling systems [80, 92], collaborative environments [93, 94, 95], websites [96, 97], and video games [98, 99, 100]. Below we describe tools and techniques for capturing, modeling, and visualizing provenance.

Capturing Provenance. By default, interaction data is ephemeral, which means once it has triggered the appropriate system response, the information contained in the interaction

Types of Provenance Information	
Data	The history of changes and movement of data, which can include subsetting, data merging, formatting, transformations, or execution of a simulation to ingest or generate new data
Visualization	The history of graphical views and visualization states
Interaction	The history of user actions and commands with a system
Insight	The history of cognitive outcomes and information derived from the analysis process, including analytic findings and hypotheses
Rationale	The history of reasoning and intentions behind decisions, hypotheses, and interactions

Purposes for Provenance	
Recall	Maintaining or recovering memory and awareness of the current and previous states of the analysis
Replication	Reproducing the steps or workflow of a previous analysis
Action recovery	Maintaining the action history that allows undo/redo operations and branching actions during analysis
Collaborative communication	Communicating and sharing data, information, and ideas with others who are conducting the same analysis
Presentation	Communicating the insights or progression of the analysis with those who are not directly involved with the analysis themselves, such as general public, upper levels of management, or analysts focusing on other areas
Meta-analysis	Reviewing the analytic processes themselves in order to understand and improve aspects of the analysis (such as process efficiency, training efficiency, or analytic strategies)

Figure 2.7: Ragan et al.'s [53] organizational framework of provenance types and purposes.

is discarded. To avoid such data loss, several logging tools and frameworks have been developed to record and analyze interaction data [96, 97, 101, 102, 103, 104, 105, 106]. A prominent example is Trrack [105], which is an open-source library to capture provenance information in websites (e.g., clicks, hovers) to later visualize or replay it.

Modeling Provenance. Several metrics have been proposed that characterize user behaviors from provenance information. For instance, Feng et al. [107] quantify *exploration uniqueness* and *exploration pacing* as users interact with points in a scatterplot. Ottley et al. [108] use a hidden Markov model to capture user attention to predict clicks in a visualization. Gotz et al. [109] model and visualize the provenance of how a user's subset selections of the data differ from the dataset as a whole. Zhou et al. [110] introduce a for-

mal model of *focus* based on user interactions defined by (1) type of action and (2) focus of the action in the form of an additive model. Wall et al. [111] define metrics for quantifying *bias* by computing deviations of a user's interactions from a baseline of "unbiased" behavior based on (1) type of interaction and (2) object of interaction.

Visualizing Provenance. Heer et al. [112] have summarized an entire design space for visualizing interaction histories (or provenance). Our review revealed that provenance information is often displayed on or near the object of interaction, e.g., highlighting previously visited or interacted visualizations [84, 94, 95] or specific regions within them [88], regions and hyperlinks on a webpage [113, 106], options and ranges in UI controls [85], lines of code in a code editor [89], or a document's authorship and readership history [114, 115], among others. Other provenance visualizations are separately visualized in an external view or application, e.g., as a graph [80, 105, 116, 117, 118].

This dissertation extends visualization and HCI literature via multiple contributions to all three aspects of analytic provenance–capturing, modeling, and visualizing: a new provenance visualization technique [25] that was later expanded into a design space [11], multiple provenance-enabled visual data analysis systems [25, 27, 7, 11], and an open-source library for provenance-tracking [9]. Studying analytic provenance is essential as it forms the basis of provisioning *guidance* during analysis, as described next.

2.4 Guidance

Recall guidance is the act of helping somebody in various ways to reach a goal. Below, we describe existing visualization and HCI literature on guidance including its many roles and definitions, characterizations, models, frameworks, and tool examples.

Roles and Definitions of Guidance. Although Cambridge Dictionary defines guidance as *"help and advice about how to do something or about how to deal with problems connected with your work, education, or personal relationships"* [119], the role and definition

of guidance in visualization and HCI literature has evolved over the years. Smith and Mosier [120] first defined guidance as *"a pervasive and integral part of interface design that contributes significantly to effective system operation."* Dix et al. [121] stressed the need for guidance to bridge the gap between the user's knowledge and the tool's operational requirements. Thomas and Cook then introduced a complementary term to guidance, called "facilitation", to describe a VA system's role in supporting human data analysis more broadly [122]. Until 2013, guidance approaches were interchangeably referred to as any kind of "help," "tip," "advice," "support," "suggestion," "assistance," or "recommendation", before Schulz et al. [40] grouped them under an umbrella term called "guidance"– defining it as *"methods that have the goal of providing dynamic support to users, such as guiding data exploration or assisting users when choosing visual mappings for presenting analysis results."* Extending this work, Ceneda et al. [35] defined guidance as *"a computer-assisted process that aims to actively resolve a knowledge gap encountered by users during an interactive visual analytics session."* Recently, Collins et al. [37] determined that the role of guidance is to enhance analysis efficiency, validate insights, boost user confidence, refine user expertise, and increase awareness of and prevent biases.

Characterizations of Guidance. Engels [123] first characterized guidance into (1) a "what" dimension that defines the problem, which is decomposed into an "initial state" at the beginning of the analysis and a "goal state" that has to be reached; and (2) a "how" dimension that describes the functioning mechanisms to solve the problem, which are the discrepancies between the initial and goal states. Schulz et al. [40] characterized guidance for visualization comprising four aspects: *context* or the user's prior knowledge, *domain* or the basis of guidance, *target* or the goal of guidance, and *degree* or the amount of guidance.

Extending this work, Ceneda et al. [35, 36] characterized guidance into (1) (what we refer to as) the *knowledge gap* between the user and the system; (2) an *input* that consists of a list of resources the process could exploit to generate the necessary guidance, and an

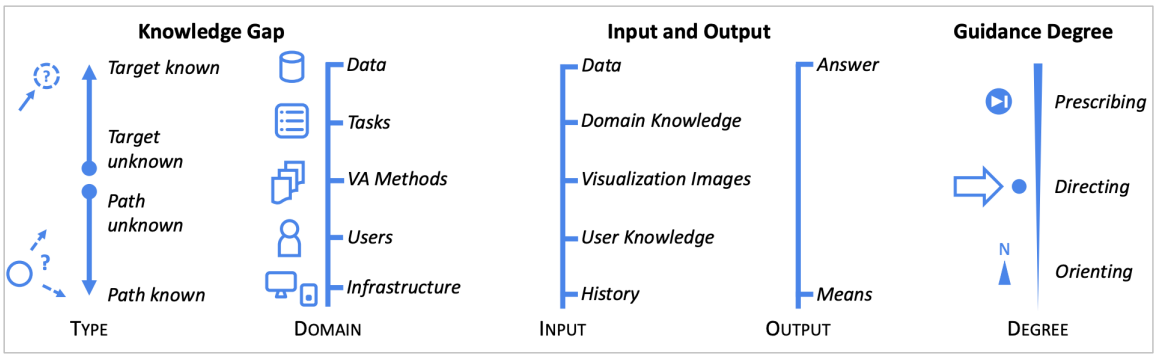


Figure 2.8: Ceneda et al. [35]'s characterization of guidance in Visual Analytics systems across "knowledge gap" (type and domain), "input and output", and "guidance degree".

output that is the computed answer to the user's knowledge gap and the visual mean(s) to communicate this answer; and (3) a degree, indicating the amount of assistance provided by the answer–orienting, directing, or prescribing (Figure 2.8). Orienting guidance is the lowest guidance degree that exploits the user's perceptual abilities and provides them with visual hints to make analytic progress. Orienting guidance is more focused on providing the means to an answer to the problem at hand instead of directly providing a ready-made answer. Prescribing and directing are higher degrees that are meant to provide a high level of assistance to the user, in the form of one (best) or more (a ranked list) suggestions, respectively. To determine which degree of guidance to provide and when, Ceneda et al. [124] conceptualized a guidance decision tree (Figure 2.9) to help designers.

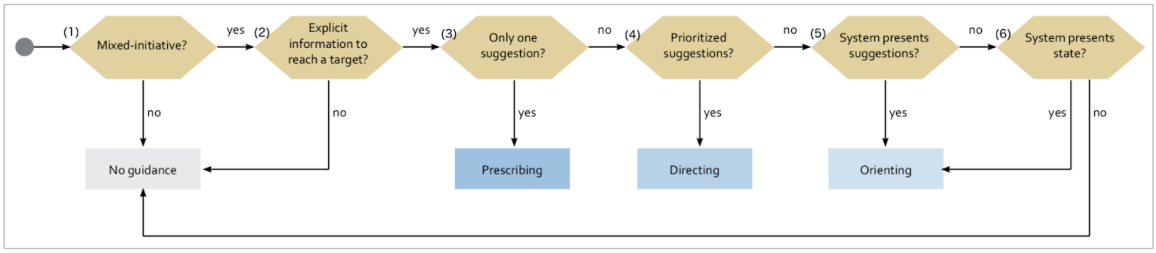


Figure 2.9: A decision tree to assess how much guidance to provide during analysis [124].

Collins et al. [37] later proposed a more practical characterization of guidance, incorporating just-in-time "facilitation" that addresses not only *where* and *what* type of guidance can be provided in the analysis process but also *how* it can be effectively implemented.

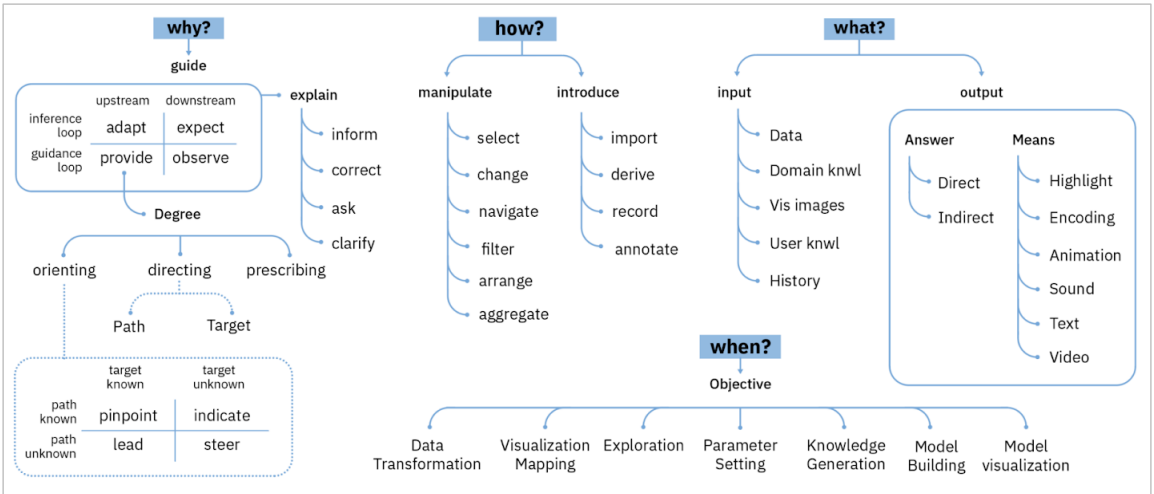


Figure 2.10: Typology of system guidance tasks by Pérez-Messina et al. [42]. It spans the three dimensions of the multi-level visualization task typology [125] plus a new dimension that captures the analytical objective of an analysis phase (when?). It allows describing: the system task intent (why?) by different detail levels (aim, first- and second-order degree), also with an accompanying explanation task (explain); the suggestion method (how?) in terms of data manipulations and means of communication; and the information inputs and type of output relative to the targeted user task (what?).

Recently, Pérez-Messina et al. [42] proposed a typology of system guidance tasks to describe and analyze guidance systems in VA and their interaction with users and their tasks (Figure 2.10). Compatible with and built upon Brehmer and Munzner's multi-level visualization task typology [125], this taxonomy spans the three main dimensions of user tasks: *Why* (describing the intent of the guidance), *How* (showing how an intent is translated into actions) and *What* (input/output). In addition, this taxonomy includes an extra dimension: the *When*, capturing the high-level analytical objective of an analysis phase.

Conceptual Models of Guidance. Ceneda et al. [35, 36] conceptualized guidance in VA into a model that aims to show the fundamental mechanisms of guidance in relation to the visualization process it seeks to assist; they adapted and extended van Wijk's visualization model [126], as shown in Figure 2.11. Van Wijk's model was created to represent only the visualization process, similar to the model by Pirolli and Card [59]; it did not convey any special characteristics of VA, similar to models by Keim and Sacha [61, 43].

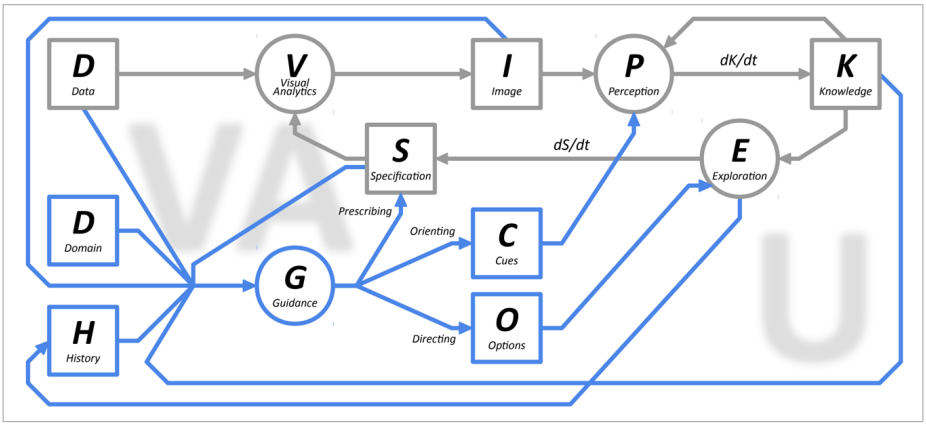


Figure 2.11: Ceneda et al.'s [35, 36] conceptual model of guidance in Visual Analytics (VA), adapted from van Wijk's [126] model (in gray) with newer guidance-related blocks (in blue); system aspects of guidance are on the left while user aspects (U) are on the right.

Ceneda et al. [35, 36] applied guidance to VA by including analytical processes into van Wijk's visualization model. Each square box represents different sources of input required for generating guidance or the output produced by active guidance processes; each circle represents the various analysis processes. Simply put, boxes are like artifacts and circles are like functions. Visual and analytical means (V) transform data [D] into images [I] based on some specifications [S]. The images are then perceived (P) to generate some knowledge [K]. Based on their accumulated knowledge, users can interactively explore (E) the data by adjusting the specifications (e.g., choose a different clustering algorithm or change the perspective on the data). As in the original model, gray boxes and circles abstract the entire VA process; to this, new components in blue are added to represent the specific aspects of guidance (boxes and circles) and the relations between each other (arrows).

Pérez-Messina et al. [42] (Figure 2.12) conceptualized guidance in VA by expanding Sacha et al.'s Knowledge Generation Model [43] (Figure 2.6). In particular, they added a new "Guide Side" (bottom portion of Figure 2.12), opposite to the User Side, that interacts with the system through a Guidance Loop, which is controlled by an Inference Loop. The information flowing from the bottom Guide Side to the top User Side can influence subsequent user actions and the overall analysis progress.

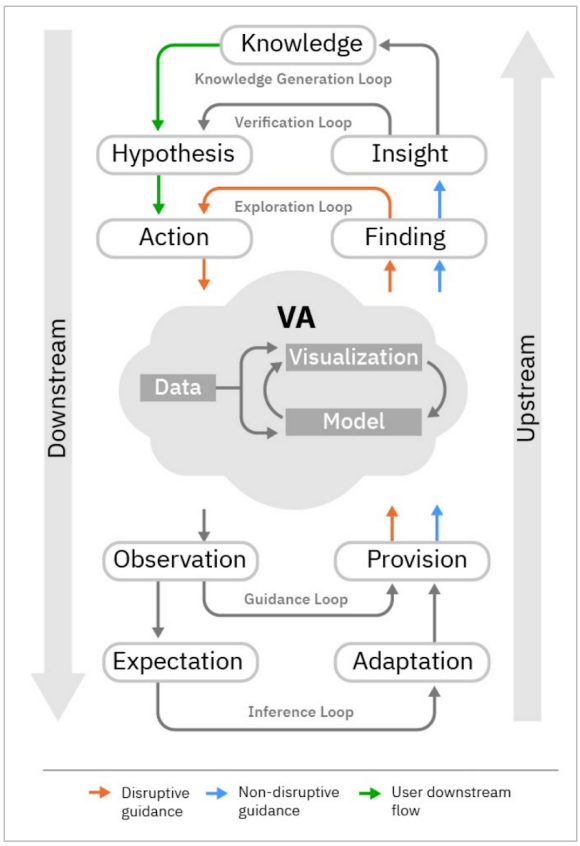


Figure 2.12: The Model of Knowledge Generation in Guided Visual Analytics (VA) showing how guidance contributes to the progress of the analysis [42]. The different arrows model the interactions between User (top) and Guide (bottom). Downstream (User-to-Guide) and Upstream (Guide-to-User) arrows signal the two directions of information flow. This model is an expansion of Sacha et al.'s [43] Knowledge Generation Model.

Conceptually, Horvitz [32] advocated for guidance-based systems to be mixed-initiative in nature, wherein both the users (human actors) and the system play an active role during analysis, by taking initiatives on behalf of each other. Sperrle et al. [44] then introduced the concept of co-adaptive guidance, building on the principles of initiation and adaptation. They argue that both the user and the system must adapt their data-, task- and system-/user-models over time. They propose reasoning about the guidance design space by introducing the concepts of learning and teaching that complement the existing dimension of implicit and explicit guidance, thus, deriving the four guidance dynamics user-teaching, system-teaching, user-learning, and system-learning (Figure 2.13).

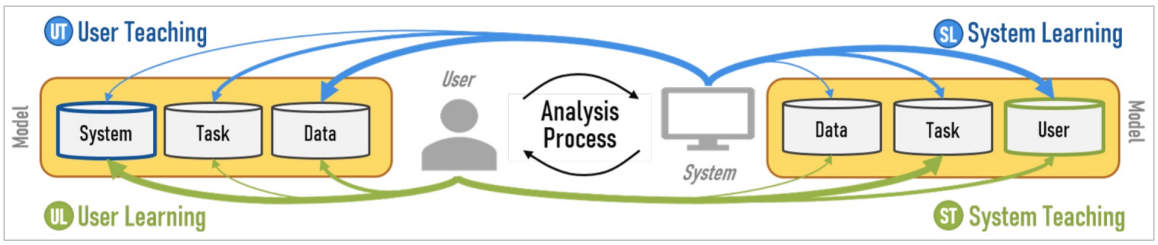


Figure 2.13: In a co-adaptive guidance process, both the system and the user initiate guidance with the goal of learning (adapting their own data, task and system/user models) or teaching (adapting the models of the other), to improve the shared analysis process [127].

Framework for Designing Effective Guidance Systems. Effective guidance refers to mechanisms that should help analysts complete a task while overcoming possible issues that could arise during the process. Ceneda et al. [41] list a set of qualities that influence the effectiveness of guidance in practical VA applications: (1) *Available*–users should be aware that guidance is available and accessible at any time; (2) *Trustworthy*–guidance must reduce uncertainty without adding confusion; (3) *Adaptive*–guidance systems should adjust to the current analysis state, dynamically changing knowledge gap; (4) *Controllable*–users need control to adjust, choose, or dismiss guidance as needed; and (5) *Non-Disruptive*–guidance should maintain analysis flow without interrupting the user's mental map.

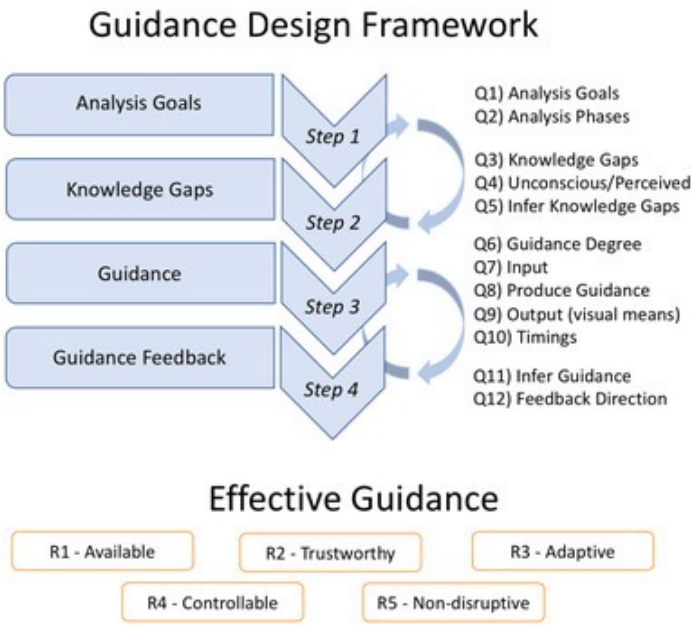


Figure 2.14: Design framework and evaluation criteria for effective guidance systems [41].

Based on these quality criteria, Ceneda et al. [41] defined a framework for designing guidance in practice. This framework comprised four nested steps aimed to identify and delineate a set of issues (i.e., knowledge gaps) that the user might encounter during the analysis (Figure 2.14), thereby pushing the designer to design appropriate countermeasures (e.g., guidance mechanisms) in a specific analysis environment; essentially, the quality criteria serve as guidelines for choosing from alternate designs.

Examples of Guidance Systems in VA. Multiple systematic literature reviews [48, 49, 50] have examined existing guidance-based tooling. For instance, orienting guidance has been provided using (1) visual properties such as *highlighting* (e.g., contrasting the color hue and intensity of important elements with those of the surroundings allows the users to quickly and pre-attentively identify them [84]), (2) *layout and form* (e.g., the 2D position, spatial grouping, and marks can attract our attention faster [128, 129]), (3) *motion* (e.g., flicker and animations are important pre-attentive visual features [130]), and (4) *suggestions* (e.g., analytical options for the user to proceed toward their goal [131].).

Next, directing guidance has been provided in data preparation to, e.g., suggest most suitable functions to transform the data [132], clean and polish the data [133], and support feature selection for data profiling [134]. In the visualization community, directing guidance has been provided to suggest different visualization alternatives, e.g., based on perceptual characteristics (e.g. [135, 136, 137]).

Finally, prescribing guidance has been offered by Horvitz et al. [38]'s system to software users by exploiting Bayesian user modeling to transform interaction into useful hints related to the user's intentions. Additionally, Chen and Scott et al. [82]'s system automatically calculates annotations of data snippets selected by the user; the user can directly modify the annotation, which again affects the generation of future annotations. Ip et al. [138]'s system guides the user through the visualization of large images by calculating and providing a step-by-step exploration of the most promising and interesting views.

Ceneda et al.'s [48]'s review revealed that orienting is the most common degree of guidance followed by directing and then prescribing, with the number of approaches providing multiple degrees very limited, and no approach providing all three. We hypothesize supporting multiple guidance degrees can enable effective and natural guidance solutions since they would allow for dynamically adapting the guidance degree as needed; this dissertation contributes such a co-adaptive, dynamic guidance system (described in chapter 8).

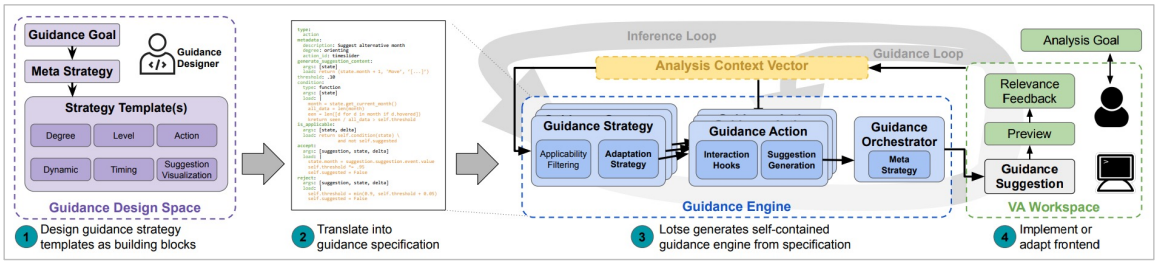


Figure 2.15: Lotse: A Practical Framework for Guidance in Visual Analytics [47].

To enable developers to build custom guidance-based systems, Sperrle et al. [47] developed Lotse (Figure 2.15), an open-source library that enables specifying guidance strategies in definition files and converting them to a running guidance system. Lotse tries to model the entire guidance process into YAML-based files, presenting the first step toward a declarative grammar of guidance. Lotse also monitors the analysis state to determine which guidance strategies to employ and which suggestions to provide, while facilitating developers to customize system behavior on acceptance or rejection of a suggestion. Through Lotse, developers are freed from implementing boilerplate code to orchestrate guidance; instead, they can focus on the design of effective strategies in the UI. Inspired by and to complement Lotse, this dissertation contributes an open-source frontend library of user interface controls for dynamically tracking and visualizing provenance as guidance [9].

2.5 Research Methodologies

When designing and evaluating visual analytics (VA) systems, appropriate considerations should be made about the context. For example, humans have traditionally been a central

figure in the design of VA systems. The phrase, 'human-in-the-loop' highlights the importance of user feedback to steer analytic processes [32]. Recently, this focus has since deepened to 'human-*is*-the-loop', reflecting a shift towards embedding analytics more seamlessly within users' workflows [33]. Below we review methodologies in visualization and HCI literature that focus on other aspects such as the analysis domain, data, and task.

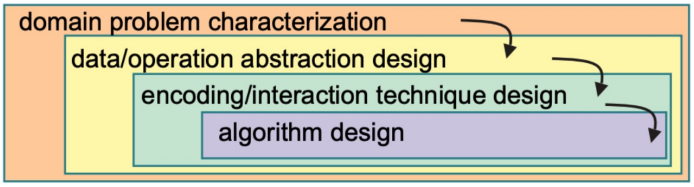


Figure 2.16: The nested model for visualization design and evaluation by Munzner [139].

First, Munzner introduced a practical framework for designing and evaluating data visualizations, structured into four nested stages: defining the domain problem, mapping problem characteristics to abstract data types, selecting visual encodings and interactions, and implementing algorithms for chosen representations [139].

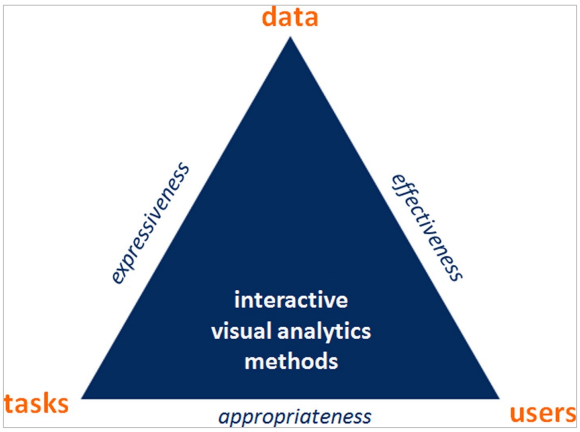


Figure 2.17: Design triangle depicting data, users, and tasks, that are major considerations during the design and implementation of visual analytics systems [140].

Next, to guide designers in choosing appropriate visualization and automated analysis techniques, Miksch and Aigner [140] proposed a design triangle framework (Figure 2.17) that addresses three main questions about the types of data users are working with (data), the users themselves (user), and the general tasks these users aim to accomplish (task).

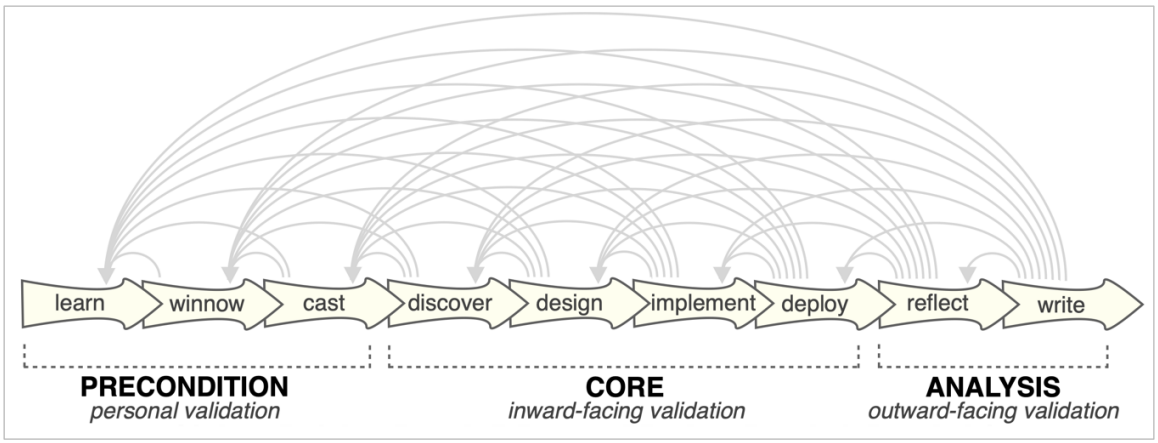


Figure 2.18: Nine iterative stages of the design study framework [141], grouped into three categories: a **precondition** phase, outlining steps to complete before beginning a design study; a **core** phase, detailing steps for carrying out the study; and an **analysis** phase, where researchers reflect on the accomplished work and write a paper at the study's conclusion.

Next, Sedlmair et al. [142] defined the design study methodology as *"a project in which visualization researchers analyze a specific real-world problem faced by domain experts, design a visualization system that supports solving this problem, validate the design, and reflect about lessons learned in order to refine visualization design guidelines."* This methodology is structured into three main phases: a **precondition** phase, outlining essential steps to complete before beginning a design study; a **core** phase, detailing the primary steps for carrying out the study; and an **analysis** phase, wherein researchers reflect on the work accomplished and in the end, write a paper (Figure 2.18).

Lam et al. [143] classified evaluation methodologies from 361 papers into seven scenarios, based on their focus on *process* or *visualization*. Process-oriented scenarios assess work practices, support analytical reasoning, evaluate communicative value, and facilitate collaborative analysis. Visualization-focused scenarios measure people's task performance and gather subjective feedback from them, and/or study an algorithm's characteristics.

For the works in this dissertation, we generally followed a user-centered design process involving a combination of interviews to understand the domain, design brainstorming sessions, system development, and evaluation for feedback and refinement.

CHAPTER 3

GUIDANCE FOR IMPROVING DATA PREPARATION WORKFLOWS

In this chapter, I describe two guidance systems that use data quality and usage information to improve data preparation workflows including navigation, discovery, selection, and monitoring. Essentially, these systems guide users by presenting *static*, *precomputed* insights about data quality and usage via visual cues and interactive affordances. This chapter is based on work published at ACM CHI 2023 [4] and IEEE BigData 2023 [5] and patented by Adobe Research [16, 17, 18].

3.1 Motivation and Background

Data are never truly raw [144] but still require processing through cleaning, integration, transformation, and selection before they can be utilized for their intended purposes [145]. Modern organizations often ingest all incoming data in their native form with the intent of performing analytics later [146]. The inherent information overload due to this "load-first" philosophy poses several challenges in data navigation and knowledge discovery [147, 148, 109]. No single user knows about all the datasets, let alone what each one contains; this unfamiliarity leads to adverse consequences. Consider a user task, *"analyze a large e-commerce dataset and build a dashboard visualizing recent geographic trends for predicting future sales."* To perform this task, users must first identify relevant data attributes pertaining to customers' locations (e.g., "ZipCode") and then select the desired data records by applying a temporal filter (e.g., monthly). This operation of reducing the size of the dataset is referred to as subset selection (or data reduction) [149, 150, 151]; it can be performed in two ways: feature set reduction (columns of a tabular dataset) or sample set reduction (rows of a tabular dataset). Feature set reduction is common when training ML models wherein users either drop irrelevant features [152] or reduce them through dimensionality reduc-

tion techniques [149]. Sample set reduction is common during market segmentation [153] wherein select groups of consumers are shortlisted to satisfy segment specific goals.

Unfortunately, subset selection can be challenging. New users unfamiliar with the data may adopt "trial and error" inspection strategies [154] resulting in the selection of irrelevant, inferior attributes while missing out on important attributes, undermining the outcome of the subsequent analysis. Even experienced users may rely upon their own past usage and not explore new attributes of a new dataset, also putting the analysis outcome into question. Furthermore, users may spend more time finding relevant data than performing the analytic task at hand [148]. So we asked, *"How to design user interfaces that provide guidance to users to analyze large, unfamiliar datasets and select relevant and effective subsets for downstream analytics tasks such as building dashboards and customer segmentation?"*

In response, we interviewed 14 data workers from a large technology company who select data subsets (extract a smaller set of attributes and records from a larger dataset) for making dashboards (data analysts), training machine learning models (data scientists), and running digital marketing campaigns (marketers). All data workers communicated the importance of the quality of data; some of them, who relied on others for preparing these data subsets as they lacked the necessary skill set, also reflected on the potential of surfacing other data characteristics such as their usage across users. This feedback from the data workers call for an interactive, self-service tool that facilitates data preparation with two kinds of auxiliary information: (1) quality and (2) usage.

Prior art defines data quality from multiple perspectives: consumer [155], business [156, 157, 158, 159, 160], and standards-based [161, 162]. A single definition covering the different contexts is difficult [155]. Contextual to this work, we define *quality* as *"the validity and appropriateness of data required to perform certain analytical tasks."* Quality is important because data are often messy, and organizations' "load-first" philosophy often results in "big data graveyards" [163] comprising large volumes of missing, erroneous, and irrelevant information. Ideally, these data deficiencies would trigger corrective measures or

even non-use; however, most organizations fail to maintain data quality standards [164] as "everyone wants to do the [ML] model work, not the data work" [165].

Regarding data *usage*, we define it as *"the historical utilization characteristics of data across multiple users,"* inspired by the "data utility" descriptor [166]. Users often collaborate at work [167, 165, 168, 169, 170], but much more around code than around data [171]. Understanding how data are created and shared inside an organization is underexplored [171]. We believe leveraging usage logs of current and past users, and meta-data can be one way to guide other users. In response, we modeled quality and usage information from the data, meta-data, and corresponding usage logs, and built DataPilot to visually present it to users to guide them during subset selection and analysis, as described next.

3.2 DataPilot

3.2.1 Design Goals

We derived six design goals based on our interviews to guide subsequent development.

DG1. Facilitate data preparation and visual data analysis, in situ. This goal supports subset selection and analysis within one tool to minimize tool-switching.

DG2. Model data quality and usage information as standardized scores. This enables non-technical users to interpret data quality via standardized scores (out of 100), derived through heuristics.

DG3. Provide visual guidance about data quality and usage. This entails offering guidance on data quality and usage, while balancing user agency and control.

DG4. Provide interaction and specification affordances for data discovery, subset selection, and visualization dashboard creation. This self-service goal includes UI controls to aid inspection of quality and usage data.

DG5. Enable control and context through configurability. This allows users to configure visibility for data quality and usage components, offering adjustable levels of control.

DG6. Design for scalability and performance. This goal aims to enhance user experience

by managing complex tasks through a scalable backend [172].

3.2.2 Modeling Data Quality

We modeled three dimensions of quality at an attribute-level: *completeness*, *correctness*, *objectivity* and two dimensions at a record-level: *completeness*, *correctness* **(DG2)**.

3.2.2.1 Attribute-level Quality Dimensions

Completeness is the percentage of *non-missing values* among an attribute's values, e.g., if 10 of 50 attribute values are *nulls* or *empty strings*, its completeness is 100*(50-10)/50 = 80%. Completeness can help users detect sparse attributes that can, for example, alter how well ML algorithms can make accurate predictions.

Correctness is the percentage of *correct values* among an attribute's values, e.g., if 5 out of 50 attribute values are incorrect, then its correctness is 100*(50-5)/50 = 90%. To calculate correctness, businesses can preconfigure SQL-like constraints in the DataPilot source code through relations (>,<,=), range (BETWEEN), pattern matching (LIKE), and membership (IN) operators; e.g., `WHERE email NOT LIKE '%_@__%.__%'` computes the number of records with incorrect email addresses. With correctness, users can assess the accuracy of individual attributes.

Objectivity is the extent that values conform to a target distribution, e.g., if the *Gender* attribute has 120 males and 45 females, then it is evidently skewed towards males and hence, from a gender equality standpoint, not objective. We utilize Wall et al.'s [111] Attribute Distribution (AD) metric for measuring the deviation between the observed and the expected objective distribution (baseline); AD scores range from [0,1] so we standardize them by multiplying by 100. With this dimension, users can detect anomalous phenomena, e.g., if the majority of applicants are of a specific gender, against expectations. Like *correctness*, businesses can preconfigure *objectivity* constraints in the DataPilot source code.

3.2.2.2 Record-level Quality Dimensions

Completeness is the percentage of *non-missing* values in each dataset record, e.g., if a record has 50 values (one for each attribute), 20 of which are *nulls* or *empty strings*, then its completeness is 100*(50-20)/50 = 60%. With this dimension, users can, e.g., discard sparse customer profiles (records) for marketing campaigns where success is determined by the profiles' richness.

Correctness is the percentage of correct values in each record, e.g., if a record has 50 attribute values, 15 of which are incorrect (based on set constraints), its correctness is 100*(50-15)/50 = 70%. With this dimension, marketers can discard customer profiles (records) with invalid email addresses and social media handles that are useless for running marketing campaigns.

Objectivity is inapplicable for record-level dimensions as each record comprises values from different, incomparable attributes.

Overall Scores: Aggregations and Customizations. We compute a configurable heuristics-based *overall* score for each attribute and record that defaults to the arithmetic *mean* of the corresponding dimensions. Based on work by Vaziri et al. [173], users can specify different weights for different dimensions (e.g., a user might prefer an overall dimension that comprises 75% completeness and 25% correctness, and ignores objectivity) as well as different attributes and records (e.g., a digital marketer may want to weigh the "Phone" attribute more than "Email Address" for correctness).

3.2.3 Modeling Data Usage

We modeled usage information **(DG2)** across three dimensions at an attribute-level: *in-subsets*, *in-filters*, and *in-visualizations* and one dimension at a record-level: *in-subsets*.

3.2.3.1 Attribute-level Usage Dimensions

In-subsets score of an attribute is the percentage of users who selected that attribute to be in their subset for later use, e.g., if 15 out of 20 users select a feature for training an ML model, then the *in-subsets* score is 100*15/20 = 75%. With this dimension, new users can, e.g., perform quick and efficient analysis by selecting highly used (important?) attributes based on subsets of prior users.

In-filters score is the percentage of users who applied a filter on that attribute, e.g., by choosing a multiselect dropdown option (*Gender="Female"*) or dragging range slider handles ($Age \in [40,50]$). With this dimension, digital marketers can, e.g., determine segmentation rules (filter criteria to pick certain customer profiles) for running marketing campaigns based on previous ones. Note that *in-filters* is not a subset of *in-subsets*; users can filter (or not) by an attribute and (not) select it in their subset and vice versa.

In-visualizations score is the percentage of users who assigned that attribute to one or more visual encodings (e.g., X axis) and utilized the resultant visualization in a dashboard. With this dimension, users can refer to popular (important?) attributes from past business reports to assist with the design of present ones.

3.2.3.2 Record-level Usage Dimensions

In-subsets score of a record is the percentage of users who selected that record to be in their subset (as a result of filters). With this dimension, users can, e.g., select a subset of popular (important?) records and re-run new marketing campaigns by targeting customer profiles (records) from previous successful campaigns. This dimension is in essence the same as record-level *in-filters* and *in-visualizations* usage dimensions because DataPilot treats a filtered dataset as the selected subset that is used in the visualization.

Overall Scores: Aggregations and Customization. Like overall quality, we computed a heuristics-based *overall* score for each attribute and record, but as the *maximum* of the constituent dimensions. Because attributes are seldom utilized simultaneously in subsets, filters, and visualizations, choosing *mean* would result in low scores that would be ineffective and demotivating for the user; hence, we chose *maximum*. Users can ignore one or more usage dimensions, e.g., *In-filters* usage, if it is irrelevant to their use-case.

3.2.4 User Interface

We integrate both quality and usage information into a visual data preparation and analysis tool, DataPilot. DataPilot facilitates preparing a subset from a large tabular dataset for building a visualization dashboard. Specifically, DataPilot computes a standardized score out of 100 for each of the quality and usage dimensions, e.g., *in-subsets* score for the "Profit" attribute is 94 out of 100. DataPilot also presents visual cues to guide users about the "good" and "bad" aspects of their data, e.g., highlighting missing and incorrect data values by coloring them in red. Lastly, DataPilot provides graphical user interface (GUI) controls as interaction affordances to assist users during subset selection, e.g., range sliders to filter out less popular data and sorting widgets to order and group data with similar characteristics together. To support these subset selection and analysis affordances in the same tool (**DG1**), we designed the DataPilot UI to have a three-step workflow with each step navigable from others via the top left corner (Figure 3.1 and Figure 3.2). We finalized this design based on pilot studies with four users.

3.2.4.1 Step 1: Review Raw Data

This step, also the landing page of DataPilot, enables users to analyze a dataset and select a relevant subset (Figure 3.1). It consists of the following views:

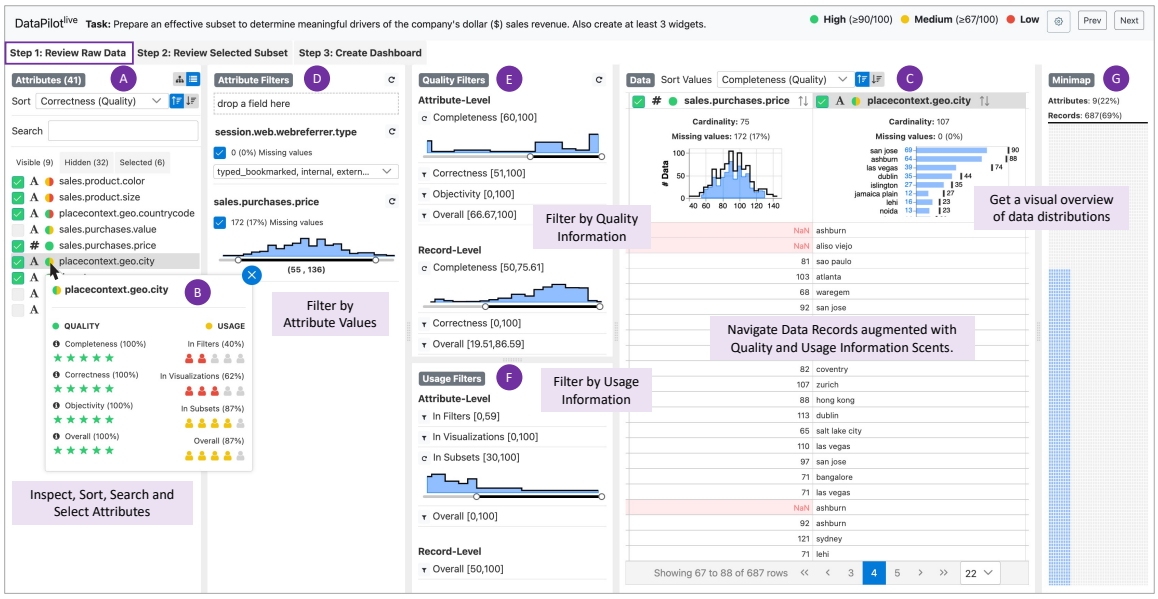


Figure 3.1: **The DataPilot user interface** showing Step 1 (Review Raw Data) of the three-step workflow. Users can inspect the list of dataset attributes (**A. Attribute View**), inspect quality and usage dimension scores for an attribute (**B. Attribute Detail View**), visualize attribute distributions and navigate dataset records (**C. Data View**), incrementally filter records by attribute values (**D. Attribute Filter View**), incrementally filter attributes and records by both quality (**E. Quality Filters View**) and usage dimensions (**F. Usage Filters View**) to reduce the search space, get a visual summary of this filtered dataset (**G. Minimap View**), and explicitly select attributes (**A. Attribute View**) and records (automatically selected based on filters) for the desired subset.

A Attribute View shows all attributes as a flattened list (☰) or as a nested list (🖧), the latter being helpful for hierarchical datasets. To efficiently display a large number of attributes, we utilize the virtual scrolling principle preventing unnecessary rendering of objects not visible in the viewport (**DG6**). A search field allows quick attribute lookup via keyword-based queries. Users can also sort by quality and usage dimensions at the attribute-level. Each list item shows the attribute's name (e.g., "sales.product.name"), its datatype (e.g., **A**: Categorical, **#**: Numerical), a bi-colored circular glyph (**DG3**), e.g., 🌓 (combination of green 🟢, yellow 🟡, red 🔴 colors), where the left-half shows the *overall quality* score and the right-half shows the *overall usage* score. Note that when the uploaded dataset has only either quality or usage information available, these bi-colored glyphs automatically transform into single-colored glyphs; users can also manually configure them

Figure 3.2: **DataPilot Step 2 (Review Selected Subset) and Step 3 (Create Dashboard)**. Users review their selected attributes (**H. Attribute View**) and records (**I. Data View**), assign attributes (**J. Attribute View**) to encodings (**K. Encodings View**), inspect the resulting visualization (**L. Visualization Canvas**) and save it to the dashboard (**M. Saved Visualizations**). Users can freely navigate between the three steps.

from the ⚙ settings in the top-right corner (**DG5**). The high (≥90), medium (≥67 but <90), low cutoffs (that determine the three categories) and the corresponding colors (to accommodate color-related accessibility concerns), can be configured from the legend in the top-right corner. Each checkbox allows users to select ☑ or deselect ☐ attributes in the subset (**DG4**). Hovering on an attribute's name shows its description in a tooltip. Clicking the bi-colored glyph opens the **Attribute Detail View**.

B Attribute Detail View is an overlay showing details of the attribute quality and usage, like LinkedIn's [174] profile completeness (**DG3**). Like the bi-colored glyph, the left

column shows data quality dimensions and the right shows usage dimensions along with the scores visualized on 5-point icon-array rating scales, e.g., "placecontext.geo.city" has a 100% completeness score (★★★★★) and an 87% overall usage score (). Hovering the info icon shows the dimension's definition (e.g., *"Completeness is the percentage (%) of non-missing values in the attribute"*) and any preconfigured rules (e.g., *"sales.purchases.price is considered correct if it is* ≥ 0*"*) to educate the user (**DG2**).

C Data View shows the entire dataset in an interactive table. The first row shows a summary view of attribute characteristics such as cardinality (number of unique values), missing values, and distribution plots (area charts for numerical #, bar charts for categorical A attributes that show the underlying data distribution in black and the filtered data distribution in blue) (**DG3**). Table cells that have missing or incorrect values (e.g., *"sales.purchases.price"*="NaN") are highlighted in red with details shown on hover (**DG3**). Standard operations such as search, pagination, and sorting are integrated within the table controls. Users can also sort by quality and usage dimensions at the record level (**DG4**). In Figure 3.1, the records are sorted by *completeness* (the "Sort Values" dropdown in the **Data View**) and the columns are sorted by *correctness* (the "Sort" dropdown in the **Attribute View**), both in the ascending order.

D Attribute Filter View enables users to filter the dataset by applying filters for each attribute by dragging them (from the **Attribute View** or the **Data View**) into this view's drop-zone (**DG4**). Multi-select dropdowns for categorical A and range-sliders for numerical # attributes along with visual scents (embedded visualizations that provide information scent cues for navigating information spaces [85]) for the distribution of attribute values in the original dataset (in black) and after applying filters (in blue) help the user determine appropriate filter criteria (**DG3**). Unlike selection of attributes, where one must explicitly check checkboxes to add to the subset, DataPilot automatically selects all remaining records after filtering into the subset.

Ⓔ **Quality Filters View** enables users to filter the dataset by quality dimensions at both an attribute and a record level (**DG4**). For example, applying the attribute-level completeness filter $\in$ [60, 100] removes all data attributes (columns) that have a completeness score outside the range. Similarly, a record-level completeness filter $\in$ [50, 75.61] filters out all records (rows) outside that range.

Ⓕ **Usage Filters View** enables users to filter the dataset based on usage dimensions (**DG4**), like the **Quality Filters View**. For example, applying the attribute-level *in-subsets* usage filter $\in$ [30, 100] removes all attributes that were selected by less than 30% of users.

Ⓖ **Minimap View** provides a novel, visual overview of the proportion of attributes and records originally in the dataset (gray), currently visible after applying filters (blue), and selected in the dataset subset (green) (**DG4**). We disabled the green (selected) state by default as our pilot users found it to be overwhelming. The width and height of the rectangular area encode the number of attributes and records, respectively. This view is discretized into small rectangles proportional to the dataset size.

3.2.4.2 Step 2: Review Selected Subset

This *review* step consists of the Ⓗ **Attribute View** and Ⓘ **Data View** with *just* the ☑ selected attributes and records (Figure 3.2). Viewing all selected attributes stacked together enables users to inspect the relative distributions of high, medium, and low quality and usage scores; this view also makes it easy to inspect the distribution of the red highlights (missing or incorrect values) in the selected table cells; both of these tasks would be difficult in **Step 1** in the presence of deselected attributes. This step makes users pause and reflect on their subset selection performance before moving onto building a dashboard (**DG1**).

3.2.4.3 *Step 3: Create Dashboard*

After reviewing the selected subset, this step helps users create and save univariate and bivariate visualizations, collectively forming a dashboard (Figure 3.2) (**DG1**). This step consists of the following views:

J **Attribute View** is the same as the **Attribute View** in **Step 2**.

K **Encodings View** allows users to create visualizations by specifying a chart type (bar chart, scatterplot, line chart), dragging attributes onto visual encodings (X, Y), and determining aggregations (sum, mean, max, min) wherever applicable (**DG4**).

L **Visualization Canvas** renders the visualization based on the specifications configured in the **Encodings View**. Users can save a visualization by giving it a title.

M **Saved Visualizations View** shows the list of all visualizations saved from the **Visualization Canvas**. This view also allows users to delete the saved visualizations (**DG4**).

3.2.5 Implementation

We developed the DataPilot frontend in Angular [175], which interfaces with a Python [176] server in real-time over the HTTP REST [177] and websocket [178] protocols. The datasets, user interaction logs (collected from the frontend), and auxiliary information were all stored in PostgreSQL, and queried later using SQL (**DG6**).

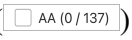
3.2.6 Example Scenarios

To illustrate how DataPilot can help users prepare relevant subsets from large, unfamiliar datasets, we developed two usage scenarios about two hypothetical users - Sunny (data engineer) and Kiran (data analyst); these scenarios were developed in collaboration with the domain experts to ensure domain relevance.

Case 1: Expert User, Improved Performance. Sunny, an experienced data engineer, often prepares data subsets for analysts who then prepare business reports. They open DataPilot, upload a recent batch of customer transactions data for an e-commerce app, and begin analysis. Given their domain expertise, they quickly lookup known attributes via the search field and select *five* attributes for their subset: "sales.product.name", "sales.purchase.price" (in USD), "placecontext.geo.countrycode" (e.g., 'IN' for India), "timestamp" (of purchase), and "environment.operatingsystem" (e.g., 'iOS').

They switch to **Step 2: Review Selected Subset** where they observe several cells in the data table with a red background. In particular, the "placecontext.geo.countrycode" column is highlighting cells with the value "*AA*" (

) and the "environment.operatingsystem" column is highlighting cells with blank (missing) values (). Realizing no country has "*AA*" as their code (as per DataPilot's *correctness* constraint and from their own knowledge) and that a majority (706 out of 1000) of values for operating system are missing, they go back to **Step 1: Review Raw Data** to make amends.

They drag the "placecontext.geo.countrycode" attribute from the **Attribute View** into the *Filter Panel* to remove all records with "*AA*" values (

) and separately alert the data collection team about this issue. To absolutely ensure that their data are correct across all attributes, they apply a record-level "Correctness" filter () to only keep 100% correct records. Finally, they deselect "environment.operatingsystem" from the subset and instead select another attribute "environment.browserdetails.useragent" that has similar information, e.g., *'Mozilla/5.0 (iPhone; CPU OS 12_0 like Mac OS X; en_US)'* and although it has not been used often before (right half is red), it is of high overall quality (left half is green). In this way, DataPilot helped Sunny become aware of issues with their data, guiding them to prepare a more complete and correct subset.

Case 2: New User, Effective Onboarding. Kiran recently joined a data analytics company and is tasked with becoming familiar with a client's data for designing future dash-

boards. They upload a client dataset of e-commerce transactions into DataPilot and start analyzing. The dataset is large and unfamiliar. They start inspecting the attribute names and descriptions from the **Attribute View** and the corresponding values and distribution plots in the **Data View** (

). Overwhelmed by the sheer size of the data and wanting to speed up their onboarding, they modify their strategy to only target *important* attributes.

They try to reduce the attribute search space by applying attribute-level filters in the **Quality Filters View** and **Usage Filters View** as proxies for *importance*. Specifically, they inspect the distributions over the respective range sliders and filter out attributes with an *overall* quality score < 75 () and an *overall* usage score < 25 (), reducing the number of attributes to a manageable 17. Finally, they sort these attributes by *overall* quality score in the descending order (

) and start inspecting their name, description, and 

and

scores in the **Attribute Detail View** (via the bi-colored circular glyphs). In this way, DataPilot helped Kiran get onboarded to a new, unfamiliar dataset quickly and effectively.

3.3 Evaluation: User Study Using DataPilot

We conducted a user study to investigate how the DataPilot user interface guides users (nudging them one way or another) to navigate a large and unfamiliar tabular dataset, prepare a relevant subset, and build a visualization dashboard.

Task: We designed a subset selection and visual analysis task for participants to:

> Explore a dataset of online customer behavior on an e-commerce website, prepare an effective subset[a] to determine meaningful drivers of $ (dollar) sales revenue for the company, and create a dashboard of at least three visualizations to convey their findings.
>
> [a] A data subset comprises attributes and records less than or equal to those in the original dataset.

Participants: We recruited 36 participants consisting of professionals and researchers from industry and academia: *students* (23), *business consultants* (2), *senior data ana-*

lysts (2), *assistant professor*, *associate product manager*, *data science manager*, *post-doctoral scholar*, *program manager*, *quality assurance engineer*, *scientist (clinical trials)*, *software developer*, and *UX designer*. Participants were pursuing or had received *bachelors* (3), *masters* (14), or *doctoral* (19) degrees in *computer science* (21), *human-centered computing* (4), *human-computer interaction* (2), *business administration* (3), *pharmaceutical sciences*, *economics*, *electronics engineering*, *systems engineering*, *data science*, or *information studies*. Demographically, they were in the *18-24* (13), *25-34* (19), *35-44* (3), or *preferred not to say* (1) age groups (in years) and of *female* (16), *male* (19), *other* (0), or *preferred not to say* (1) genders. They self-reported their experience performing any kind of data analysis using visual analysis tools (e.g., Excel, Tableau) or programming as either *everyday or part of the job* (10), *often* (13), *occasionally* (13), *rarely* (0), or *never* (0).

Dataset: To thoroughly evaluate all DataPilot capabilities and complete the task within the study duration, we used a random sample of 1000 records and 42 attributes from an open-source digital marketing dataset [179] and infused certain quality issues pertaining to *correctness* and *objectivity* (by setting appropriate constraints). We marked quality and usage (and overall) scores such that $\geq$90 is marked as *high* 🟢, $\geq$67 but $<$90 as *medium* 🟡, and the rest as *low* 🔴. We fixed these thresholds to realize a reasonable distribution of attributes and records across the three (high, medium, low) categories, so that participants are neither demoralized (all scores are low) nor overconfident (all scores are high).

System Configurations as User Study Conditions: To achieve **DG5**, we designed DataPilot to support four configurations: (1) neither quality nor usage, (2) only quality, (3) only usage, and (4) both quality and usage. Of these four configurations, we did not explicitly evaluate the (3) only usage configuration because our expert interviews highlighted addressing data quality concerns as most important and that usage information alone must never power "data-driven" analysis and decision-making, at least not without more important aspects such as quality. Hence, we utilized the other three DataPilot configurations as

standalone study conditions in a between-subjects evaluation, as described next.

[B] Baseline. With this configuration, we aim to understand user strategies *without* quality and usage information, also simulating what many current systems do (e.g., Tableau [180]). Specifically, the bi-colored glyphs next to the attribute name, filter and sort options, and visual scents (in the table) for usage and quality are all hidden.

[Q] Quality. With this configuration, we aim to understand how users utilize only quality information to perform the study task, also simulating what many current systems do (e.g., Profiler [133], Trifacta [181]). This condition would also enable us to compare against the following *D* configuration (that has both quality and usage information). Specifically, only single-colored circular glyphs next to the attribute name, sort and filter options, and visual scents (in the data table) that are relevant to quality are visible and enabled.

[D] DataPilot. This all encompassing configuration shows both data quality and usage information in the interface. Specifically, all features are enabled. Usage information for the *D* condition were computed by processing the interaction logs of the participants in the *B* and *Q* conditions (24 participants). We computed each attribute's *in-subsets* score as the percentage of participants who selected that attribute to be in their subsets, *in-filters* score as the percentage of participants who filtered by that attribute, *in-visualizations* score as the percentage of participants who assigned that attribute to a visual encoding, and an *overall* score as the maximum of the three aforementioned scores. Similarly, for each record, we computed the *in-subsets* score (also the *overall* score in this case) by computing the percentage of participants who selected that record (automatically as a result of applied filters) to be in their subsets. To disregard temporary, unplanned, and accidental selections during analysis, we compute this information only based on the final state of the interface at the end of the task (selected subset, applied filters, saved visualizations).

Study Session. We assigned participants to one of the three study conditions (*B*, *Q*, *D*). Each study session lasted between 60 and 90 minutes, with *D* taking longer than *Q* than *B* due to differences in the training and practice times. We compensated each participant with a $15 gift card for their time. We conducted the study remotely using Teams [182]; the experimenter provided participants access to the study interface by sharing their (experimenter's) computer screen and granting input control to the participant. After providing consent, participants saw a video tutorial on DataPilot's features (*B*:5, *Q*:7, *D*:10 minutes long). Participants then performed a practice task on a *dataset of houses* (adapted from [183]) to get acquainted with the UI before starting the actual task.

The actual task lasted a maximum duration of 30 minutes. Participants were not required to think aloud during the task to simulate a realistic work setting (although some participants felt comfortable doing so). During the task, participants' interactions with the system (e.g., the filters they applied, the data subsets they selected) were logged. The study ended with participants completing a questionnaire to rate the usefulness of DataPilot's features and a semi-structured debriefing interview for 10 minutes in which participants reflected on their overall experience, provided feedback, and answered other questions. At the end of the debriefing interview, the experimenter also demonstrated the *D* configuration to both *B* and *Q* participants to get their initial reactions and elicit feedback on how the new set of aids would have hypothetically helped them accomplish their task differently. Each debriefing interview was screen- and audio-recorded for subsequent qualitative analysis.

3.3.1 Hypotheses

We structure our study analysis according to the hypotheses below, predetermined before the study based on our expectations from the intended purpose of the tool, former perception studies, feedback from pilot studies, and our own instincts. $>$ implies *more or greater than*; $<$ implies *less or smaller than*.

H1 B (Baseline) > Q (Quality) > D (DataPilot) in terms of the number of attributes and records in the selected subsets.

H2 B > Q > D in terms of the proportion of attributes and records with *low* quality and usage in the selected subsets.

H3 B < Q < D in terms of the proportion of attributes and records with *high* quality and usage in the selected subsets.

H4 B < Q < D in terms of success and confidence after the task.

H5 B < Q < D in terms of amount of effort, temporal demand, mental demand, and frustration while doing the task.

H6 Participants will find quality information to have greater utility than usage information while doing the task.

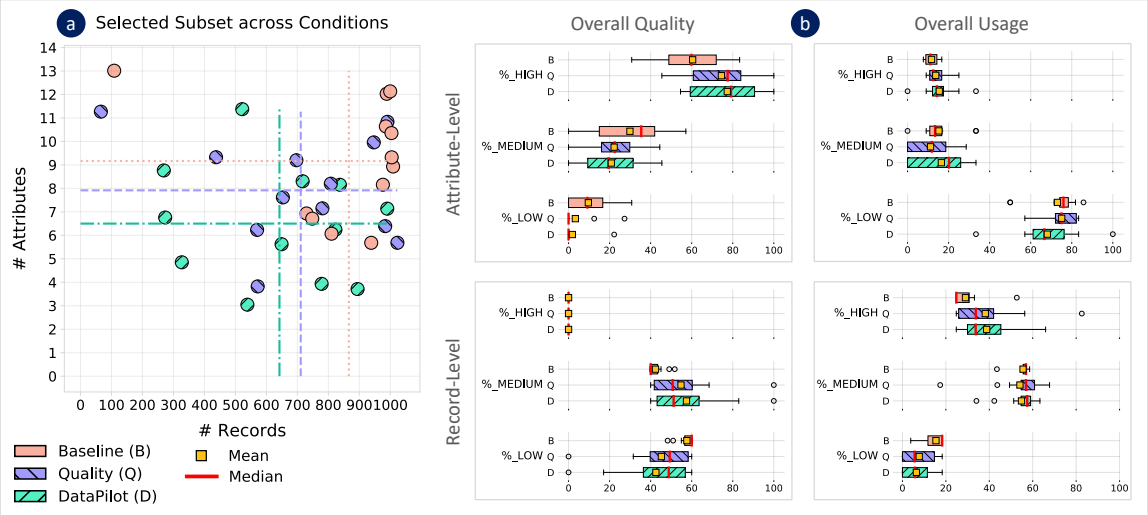


Figure 3.3: (a) Number of attributes and records in the participants' selected subsets and (b) attribute-level and record-level distributions of high, medium, low overall scores for both quality and usage across the three study conditions (Baseline, Quality, DataPilot).

3.3.2 Results

Below, we present findings from the user study and discuss them in the context of qualitative participant feedback. $P_C1,...,12$, $Q_{1,...,12}$, $D_{1,...,12}$ refer to the 36 participants in the Baseline (B), Quality (Q), and DataPilot (D) conditions, respectively. Participant quotes

spoken during the debriefing interview and responses written in the questionnaires were both coded and categorized using affinity diagramming [184], an inductive thematic analysis [185] technique. One experimenter came up with an initial set of categories that were then refined during iterations with three other experimenters until a consensus was reached; the final codebook consisted of 6 high-level categories with 43 detailed, low-level codes.

3.3.2.1 Feedback on DataPilot's Quality and Usage Information

DataPilot, the system. Overall, participants found DataPilot to be useful, reporting above average system usability (SUS [186]) scores across the three conditions as {B: 80.21, Q: 74.17, D: 71.67}. D_4 commented that *"Providing detailed auxiliary information such as the quality and usage of each data attribute is very important and missing in current tools like Tableau and PowerBI."* Q_8 also explained why quality and usage information are important noting, *"80-90% of true data analysis, data science, machine learning is [the data preparation] step. These [quality and usage] measurements that you're creating to allow users to start [working on their tasks] and make them explore some of the unintended consequences is very powerful. It has ample opportunity for future discovery to continuously make this a better product, so very very fascinating stuff."*

Quality information. Participants had overall positive feedback for the quality information. Q_{10} commented that *"There are invisible problems with your data and you don't necessarily find out until you start playing around with the visualizations. [Furthermore,] in aggregate visualizations, you either have limited or no ability to identify quality problems so I appreciate that DataPilot is just very explicit about these quality issues."* Q_7 noted that *"It is important for systems to provide such out-of-the-box insights so that users like me who don't write code don't completely ignore these aspects and can rely on the green attributes and just get started with analysis."* Q_8 saw *"a lot of value to enable users to more quickly filter [attributes and records] through the signal of these measurements of*

quality as opposed to learning [them] on their own." However, Q_8 also expressed caution about *"confounding factors, especially missing data, because many times data is not missing at random it is actually missing and telling a story,"* suggesting quality information can provide a good starting place, but additional analysis by users may still be required.

Usage information. There was mixed feedback regarding the usage information. Participants with positive feedback suggested using usage information to perform fast and efficient analysis (Q_6), to seek validation *"by performing little investigations"* (Q_1), *"to check if they have a similar opinion as others"* (D_4), *"to identify new things where other people are not looking"* (D_3), to seek guidance from predecessors (e.g., $Q_{2,11}$), to avoid repeating past mistakes (D_3), and to choose between conflicting choices (e.g., *"for some attributes it's not easy to decide...but usage can help choose"* - D_8). Participants with mixed and negative feedback said they would not care (D_3) or rely on what other people did as they do not know anything about the other users and would have to assume they did a great job with their analysis (Q_1, D_{10}). Participants also raised concerns around bias and following the crowd as *"one might miss out on an uncommon attribute that is also useful"* (P_C7).

3.3.2.2 Comparing Prepared Subsets

Table 3.1 and Figure 3.3 show the sizes of subsets (number of attributes out of 42 and records out of 1000) selected by the participants (Figure 3.3a) and the distribution of *high*, *medium*, *low* values of attribute- and record-level quality and usage scores (Figure 3.3b). Validating **H1**, *D* chose the fewest attributes and records followed by *Q* followed by *B*.

Furthermore, *D* chose a higher percentage of *high overall quality* attributes than *Q* than *B*. Because the dataset was sparse (a majority of values in each record were empty), no record had a *high overall quality* score, hence the corresponding μ_R, σ_R values for B, Q, D were all 0. *D* also chose a higher percentage of *high overall usage* attributes and records than *Q* than *B*. These results validate **H3**.

Table 3.1: Statistics associated with the prepared dataset subsets in terms of their **"Size"** and distribution of *high* ("% H"), *medium* ("% M"), *low* ("% L") values for attribute- (**"A"**) and record-level (**"R"**) quality and usage scores across the three study conditions (**B**, **Q**, **D**). The bolded and highlighted values in each row support our hypothesis, specifically H1, H2, H3, e.g., **6.5 (D)** has the smallest μ of number (**"Size"**) of attributes (**"A"**) selected in the subset, supporting H1. No record ("R") had a high ("% H") overall quality score because the chosen dataset was sparse. Medium ("% M") values were not part of our hypotheses; thus, the table cells corresponding to these values are neither highlighted nor formatted.

		Baseline (B)		**Quality (Q)**		**DataPilot (D)**	
		μ	σ	μ	σ	μ	σ
Size of Prepared (Selected Subsets)							
Size	A	9.17	2.44	7.92	2.19	**6.5**	2.32
	R	866.17	253.08	710.83	282.85	**642.17**	249.18
Distribution of Overall Quality Scores							
% H	A	60.45	16.63	74.47	18	**77.32**	17.73
	R	0	0	0	0	0	0
% M	A	29.90	20.43	22.22	13.94	20.83	16.36
	R	42.36	4.12	54.78	17.09	57.45	18.7
% L	A	9.65	10.32	3.31	8.36	**1.85**	6.42
	R	57.64	4.12	45.22	17.09	**42.55**	18.70
Distribution of Overall Usage Scores							
% H	A	11.67	33.20	13.72	4.54	**15.45**	8.27
	R	29.03	8.14	38.16	16.89	**38.78**	12.97
% M	A	15.29	9.53	11.18	9.81	16.46	12.95
	R	55.54	3.90	54.11	13.16	54.82	8.59
% L	A	73.04	11.37	75.09	7.87	**68.08**	16.45
	R	57.64	4.12	45.22	17.09	**42.55**	18.70

Similarly, *D* chose a lower percentage of *low overall quality* attributes and records than *Q* than *B*. Furthermore, *D* chose a lower percentage of *low overall usage* attributes and records than *Q* and *B*, validating **H2**. These findings suggest that quality and usage information nudged users to prepare smaller, more effective subsets.

3.3.2.3 Task Fidelity Scores

Figure 3.4 shows participant feedback on the fidelity of the task on a seven-point *Disagree* (1) to *Agree* (7) scale. *D* reported higher or comparable mental demand (M_D=5; M_Q=5; M_B=4.5; M=median), hard work (M_D=5; M_Q=4; M_B=4), and frustration (M_D=2.5; M_Q=2.5; M_B=2) than *Q* than *B*, finding some evidence in support of **H5**. We attribute this

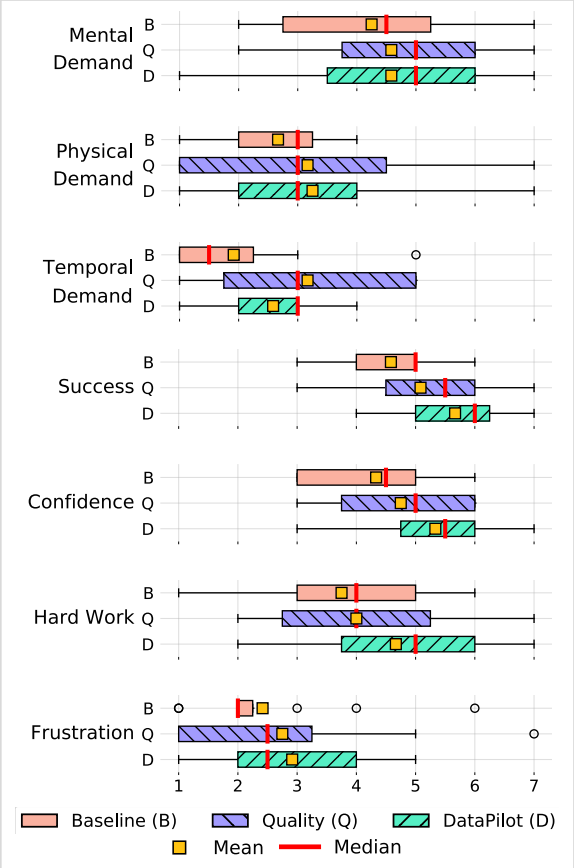


Figure 3.4: Task fidelity scores as reported by participants on a seven-point *Disagree* (1) to *Agree* (7) scale. *D* participants reported higher or comparable mental demand, hard work, and frustration but greater success and confidence at the end of the task than *Q* than *B*.

result to the increased complexity due to additional user interface elements in *D*, that may have affected users' cognitive load. However, *D* reported greater success (M_D=6; M_Q=5.5; M_B=5) and confidence (M_D=5.5; M_Q=5; M_B=4.5) in the end, validating **H4** and suggesting that the auxiliary information helped participants perform the task more effectively.

3.3.2.4 Importance of General, Quality, and Usage Information

We asked participants about the importance of different kinds of general, quality, and usage information in the interface on a *Not at all important* (1) to *Very important* (7) scale. Except attribute *datatypes*, other general information such as attribute *names*, *values*, *distributions*, *cardinalities*, and *descriptions* were mostly useful (Figure 3.5a).

Figure 3.5b, Figure 3.5c show that overall, both *Q* and *D* participants found data quality to be useful (M_D=5; M_Q=5; M=median). At the attribute-level, *completeness* (M_D=6; M_Q=6) was more important than *correctness* (M_D=5; M_Q=6) and *overall* (M_D=5; M_Q=5), while *objectivity* (M_D=3.5; M_Q=4.5) received mixed scores. Many participants felt completeness was the most important (Q_5, $D_{3,6,9,10}$) because *"[they were] not the one who set the rules for correctness and objectivity"* (D_6). Scores were mixed for the record-level dimensions: *overall* (M_D=4; M_Q=4), *correctness* (M_D=4; M_Q=3.5), and *completeness* (M_D=4; M_Q=4). Q_4 aimed for an authentic subset with mostly complete records, while Q_7 felt it counterproductive after applying attribute-level filters. *B* participants, when presented with quality information during the debriefing, stated that they either assumed there were no missing values (P_C2,10), forgot to look for them and vowed to be more alert next time (P_C9), or thought of but ignored them (P_C4,7).

Figure 3.5b, Figure 3.5d show that overall, *D* participants had mixed feedback about the usage information (M_D=5; M=median). $D_{2,7,8}$ found them useful, D_1 not so much, and $D_{3,4,5,10}$ raised concerns about bias and loss of originality, suggesting usage be provided with care in specific situations. At the attribute-level, *overall* (M_D=5) was more important than *in-subsets* (M_D=4), *in-visualizations* (M_D=3.5), and *in-filters* (M_D=3). Most participants also stated *overall* to be the most important dimension except D_6 who *"went for the highest [usage] in filters."* Participants found the record-level dimensions less useful (*in-subsets*: M_D=3). *Q* and *B* participants, when they were presented simulated usage information during the debriefing interview reflected that usage can *"give [them] more confidence in selecting attributes"* (Q_4), help verify their work (Q_1), and be guided by others' work (P_C8, Q_2). Overall, participants found quality to be more important than usage, as noted by D_4, *"Data quality is way more important in our daily life and only if there are several people working on the same dataset or tool, then data usage may be helpful"* and Q_{12}, *"If an attribute is of high quality but low usage, I would still pick that attribute."* Collectively, these results validate **H6**.

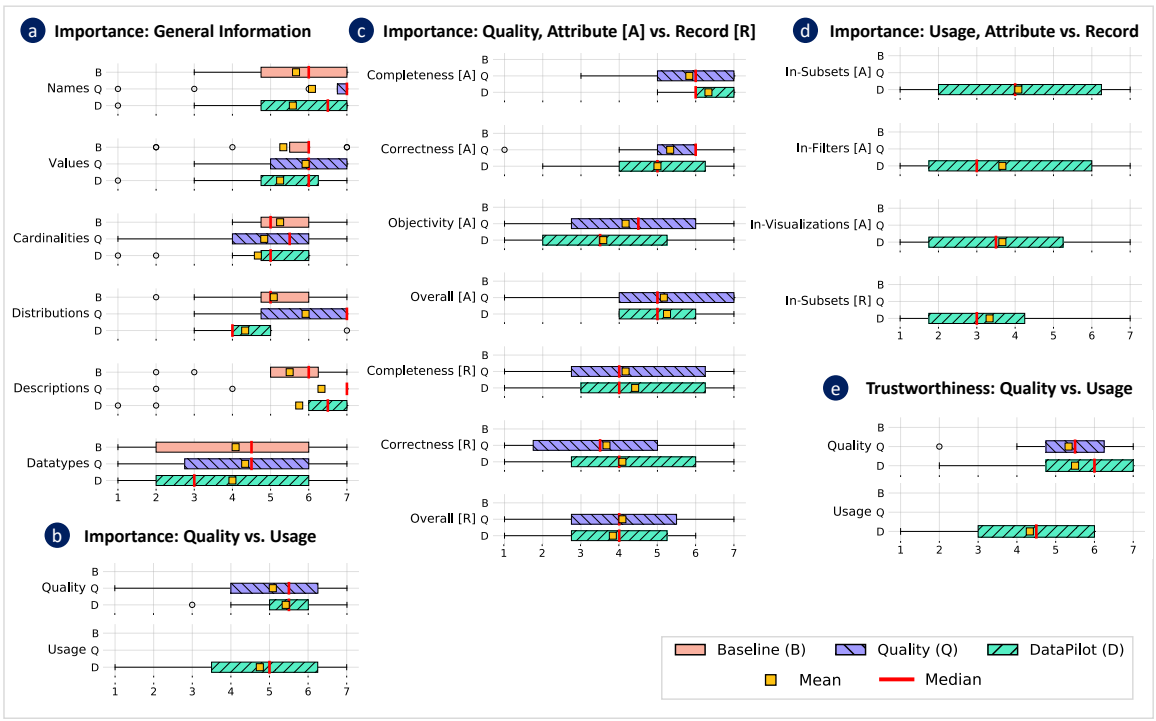


Figure 3.5: Importance and trustworthiness scores of general, quality and usage information for attributes and records across the three study conditions. There are no box plots for some study conditions, e.g., Baseline (B) in (b)-(e), as they were not applicable.

3.3.2.5 Participant strategies to select subsets.

Only quality. Ten Q and two D participants relied only on quality: Q_4 discarded incomplete records by applying a *completeness* filter, $D_{1,5}$ filtered out attributes based on *completeness*, and Q_3 looked for high 🟢 *overall* quality attributes via the colored glyphs.

Only usage. No D participant relied only on usage, vindicating our domain experts' judgment that quality is still the most critical information during data preparation and analysis.

Both quality and usage. Seven out of twelve D participants used both quality and usage. For example, D_9 applied quality filters and then focused on the bi-colored glyphs to avoid the low 🔴 usage attributes. D_8 sorted attributes by *overall* usage scores before applying quality filters, D_{11} inspected the *in-subsets* usage dimension after applying quality filters, and $D_{4,6}$ used quality to make initial selections and then usage to verify and validate.

Neither quality nor usage. All *B* (as they did not see any auxiliary information), two *Q* ($Q_{1,2}$), and three *D* participants ($D_{2,3,10}$) primarily relied on general attribute information (e.g., attribute names and descriptions) and correlation and trend analysis (e.g., by creating visualizations) to select their subsets.

Other non data-driven strategies. Participants also relied on their preconceptions (Q_3, D_4), common sense (D_1), intuition ($D_{2,3,5,7}$), and trial and error practices ($D_{3,6}$) as secondary strategies, highlighting the role of human-intelligence in data-driven analysis. Modeling auxiliary information such as quality, usage can minimize uncertainties and inconsistencies associated with such strategies.

3.4 Limitations and Future Work

DataPilot. DataPilot currently supports quality information for tabular datasets; future work may explore other structured (e.g., relational databases) and unstructured (e.g., text, documents) datasets. Additionally, there are other data-dependent (e.g., consistent representation, ease of manipulation, and timeliness [160]) and process-dependent (e.g., data collection [187]) aspects of quality, and similarly, other aspects of usage beyond a subset selection and dashboard building task (e.g., co-usage frequencies of multiple attributes in a visualization, frequency of visualization interactions such as zooming and panning [125]) that may be operationalized in the future. Next, DataPilot's dashboard view currently supports creation of disconnected visualizations; future work may explore the effects of interactive affordances such as brushing and linking. Lastly, the completeness, correctness, and objectivity quality constraints are currently hard-coded in the DataPilot source code in a SQL-like syntax. Future work can provide interactive affordances for the user to configure these constraints and also clean the data (e.g., handle missing values) directly via the UI.

User Study. During the user study, we made a fair assumption that our participants were unfamiliar with the dataset and hence exhibited similar expertise, supporting internal va-

lidity; however, this assumption may not hold true for real-world cases from an external validity standpoint [188]. Future work may incorporate weighting mechanisms to more accurately approximate usage based on recency of use (e.g., give more importance to recent data), user expertise (favor experts), or the criticality of the application that utilized the data. Next, because our participants were not domain experts, we did not have experts assess the selected subsets or final dashboards; future user studies with domain experts should further evaluate the quality of these results. Lastly, we focused on the particular task of exporting visualizations for a dashboard, which may have impacted how the attributes and records were chosen; future work should consider developing additional tools to study downstream analytics tasks other than subset selection such as ranking and clustering.

3.5 DataCockpit

Extending DataPilot's functionalities, we also built **DataCockpit**, a Python toolkit that utilizes quality and usage information to help users *navigate* and *monitor* data lake comprising multiple relational datasets and a logging framework.

Like DataPilot, DataCockpit computes quality and usage characteristics for each column (e.g., number of times the column was queried for subsequent use in downstream applications) and row (e.g., number of non-missing values, valid values) and assigns scores out of 100, that are then aggregated to a dataset-level. DataCockpit provides a customizable and extensible Python API to compute, persist, and query usage metrics such as who used which dataset, how, when, and why; and quality metrics namely *completeness*, *correctness*, *objectivity* [4, 160]. Using DataCockpit, we developed a visual monitoring tool that presents usage and quality information with interactive affordances for data lake navigation and monitoring. Figure 3.6 shows the user interface, and it consists of two main tabs.

Data Lake View. This view is the landing page and provides an overview of a single data lake (e.g., "Asia/Pacific") configurable via the dropdown. It includes an interactive table

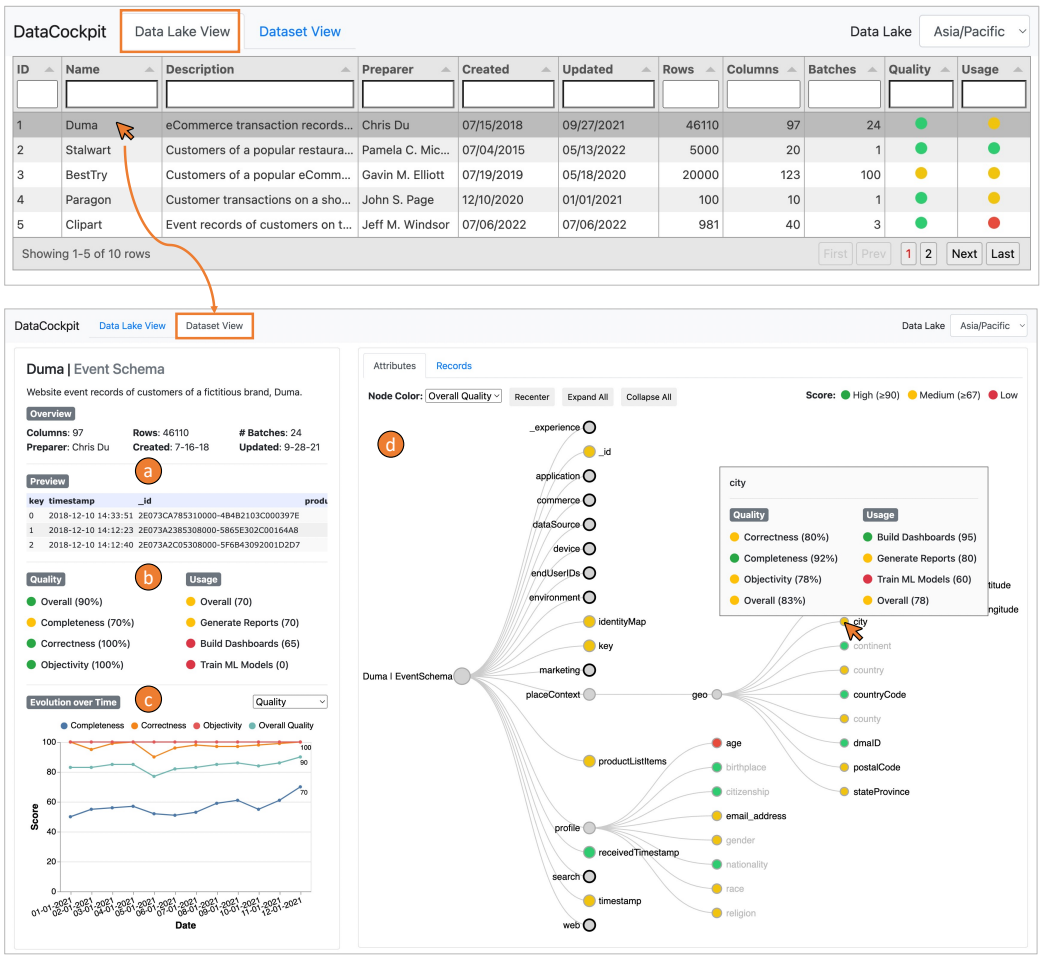


Figure 3.6: DataCockpit's Visual Monitoring Tool: the **Data Lake View** lists all datasets in the data lake; the **Dataset View** provides additional information (e.g., a preview) about a specific dataset (a), overall quality and usage scores (b), temporal evolution of these scores (c), and a visualization showing attribute and record-level quality and usage scores (d).

with information on constituent datasets, e.g., {"Id", "Name"}. The last two columns correspond to *overall* "Quality" and "Usage" scores, heuristically classified as *high* ● (≥90), *medium* ● (≥67 but <90), and *low* ●, using the same cutoffs as DataPilot [4][1]. Users can utilize this information along with search, filter, and pagination interaction affordances to (1) **navigate**: strategically explore the datasets in the data lake based on their quality and/or usage (not by their name and/or create/update timestamps), (2) **discover**: find high quality, high usage, relevant datasets for a downstream application, (3) **monitor** the 'health' of the data lake, and (4) **housekeep**: find low quality, low usage irrelevant datasets for

[1]Shown quality and usage information is simulated for this demonstration.

archival. Clicking a table row presents additional details about the corresponding dataset in the *Dataset View*.

Dataset View.

- (a) **Overview and Preview Views** present factual information for the selected dataset, like the *Data Lake View*, along with five sample records (as a preview).

- (b) **Quality and Usage Views** visualize the computed quality and usage scores as colored glyphs: *high* 🟢 ($\geq$90), *medium* 🟡 ($\geq$67 but $<$90), and *low* 🔴. For example, Figure 3.6 shows that the "Duma" dataset has a *correctness* of 100% 🟢 and has been used more to generate reports (70) 🟡 and less to build dashboards (65) 🔴.

- (c) **Evolution over Time View** visualizes the evolution of overall usage and quality scores for this dataset over time. These scores are (re)computed whenever new records (a new batch of data) are appended to the dataset (even when new datasets are ingested), and/or on a scheduled (e.g., weekly) basis, helping users monitor the health of datasets.

- (d) **Attribute and Record Explorer View** presents quality and usage information for each attribute in an interactive list or tree visualization (Figure 3.6), and record as a tabular visualization. The tree visualization is useful for hierarchical data schemas (e.g., "placeContext.geo.city", "placeContext.geo.point.latitude") and lets users pan, zoom, expand, and collapse attribute nodes to promote overview first and details on demand visual exploration [189]. Nodes can be colored based on the quality or usage scores of the corresponding attributes. Node label colors correspond to whether an attribute in the mapped schema is in the dataset (black) or not (gray). Hovering an attribute node (e.g., "city" in Figure 3.6) shows corresponding quality and usage scores in a tooltip. For record-level information, a datatable provides similar capabilities.

3.6 Summary

In this chapter, I described two guidance systems that use data quality and usage information to improve data preparation workflows: (1) DataPilot for selecting an effective subset from a large, unfamiliar tabular dataset and (2) DataCockpit for data navigation, discovery, and monitoring within data lakes. DataPilot is a visual data preparation and analysis tool that models two kinds of auxiliary information, *quality* and *usage*, to assist users in analyzing a large and unfamiliar tabular dataset, selecting a relevant subset, and building a visualization dashboard. DataPilot is an outcome of a design study with 14 data workers over a period of two months who communicated the importance of data quality and also suggested surfacing data usage characteristics to guide users during data preparation. A user study with 36 participants suggested that quality and usage information together help users select smaller, effective data subsets with greater success and confidence; however, to balance exploration versus exploitation, our participants sounded caution about users relying excessively on usage information. DataCockpit is an open-source Python library that similarly models *quality* and *usage* information for data lakes (relational databases with logging enabled). Through DataCockpit we enable developers to build custom data navigation, discovery, and monitoring tools because we believe that through quality *and* usage information, organizations can build collective intelligence, increasing transparency and accuracy to foster closer collaboration and cooperation among teams. DataCockpit and the tool are released as open-source software at **https://github.com/datacockpit-org**. For details, I refer the reader to associated publications [4, 5] and patents [16, 17, 18].

CHAPTER 4

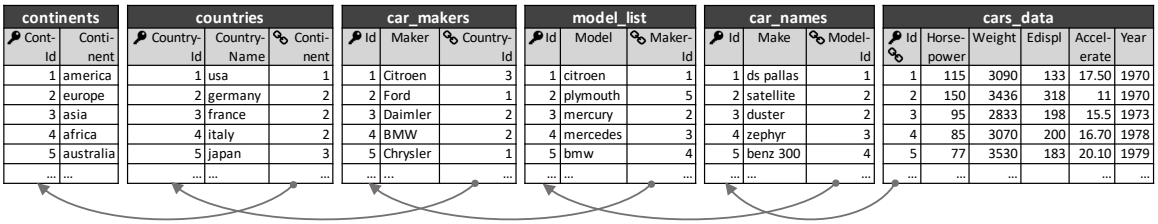
GUIDANCE FOR DEBUGGING QUESTION-ANSWERING WORKFLOWS

In the previous chapter, I described a system (DataPilot) that presented *static*, *precomputed* insights about data quality and usage to enhance subset selection workflows; essentially, this system offered visual cues and interactive affordances as guidance, for users to select relevant subsets from large, unfamiliar datasets. In this chapter, I describe a question-answering chatbot system, integrated with an interactive, self-service debugging view, that helps users interactively debug (i.e., inspect for, isolate, and fix errors in) natural language to SQL (NL2SQL) scenarios, partly achieving **RG1**: *Investigate the role of guidance in enhancing analytic processes and outcomes in various data preparation and analysis workflows*. Unlike DataPilot, this system offers a test-bed for users to interactively guide themselves by exploring *what-if* scenarios to dynamically debug NL2SQL responses. This chapter is based on work published at ACM IUI 2021 [26] and patented by Microsoft [15].

4.1 Motivation and Background

Current advances in machine learning make it possible for many systems to let their users express and fulfill their goals through natural language (NL) in what are known as natural language interfaces (NLIs). A particular family of these systems, NLIs for querying databases, have been studied by researchers in natural language processing [190, 191, 192], databases [193, 194, 195, 196, 197, 198, 199, 200], and human-computer interaction [201, 202, 203, 204, 205, 206, 207, 208]. Systems employing these NLIs receive a natural language (NL) question as input, translate it into a formal database query and execute the query on the underlying database to compute an answer. Existing systems present these responses using a combination of the computed answer, the generated query, any associated meta-data (e.g., mappings between the question and the generated query), easy-to-understand expla-

nations of the aforementioned artifacts (using, for example, NL and visualizations), and UI control augmentations (e.g., drop-downs) that facilitate fixing errors and disambiguation.



continents

Cont-Id	Conti-nent
1	america
2	europe
3	asia
4	africa
5	australia
...	...

countries

Country-Id	Country-Name	Conti-nent
1	usa	1
2	germany	2
3	france	2
4	italy	2
5	japan	3
...	...	...

car_makers

Id	Maker	Country-Id
1	Citroen	3
2	Ford	1
3	Daimler	2
4	BMW	2
5	Chrysler	1
...	...	...

model_list

Id	Model	Maker-Id
1	citroen	1
2	plymouth	5
3	mercury	2
4	mercedes	3
5	bmw	4
...	...	...

car_names

Id	Make	Model-Id
1	ds pallas	1
2	satellite	2
3	duster	2
4	zephyr	3
5	benz 300	4
...	...	...

cars_data

Id	Horse-power	Weight	Edispl	Accel-erate	Year
1	115	3090	133	17.50	1970
2	150	3436	318	11	1970
3	95	2833	198	15.5	1973
4	85	3070	200	16.70	1978
5	77	3530	183	20.10	1979
...	...	...	...	...	...

(a) The **cars** database (production database); → depicts the Foreign Key - Primary Key relationships; [...] imply more rows.

```
SELECT car_makers.Maker FROM car_makers
JOIN countries
    ON countries.CountryId=car_makers.
        CountryId
WHERE countries.CountryName='usa';
```

(b) Generated SQL query for the "Which car makers are American?" question

car_makers

Maker
Ford
Chrysler

(c) Answer on production database

Figure 4.1: An example natural language (NL) to SQL (NL2SQL) scenario.

These systems present challenges for users who may be familiar with the domain but are not fluent in the database query language. In particular, assessing the correctness of an answer that is output from an NLI can be challenging. For example, in a system that answers questions about a cars database (Figure 4.1a), a user asks a question *"Which car makers are American?"* The system first translates it into a SQL query (Figure 4.1b), and then runs it on the database to compute the answer—*Ford, Chrysler* (Figure 4.1c). An expert on cars might suspect the answer to be correct based on their knowledge, but it might not be so. In this case, the question contains *"American,"* which is syntactically similar to *"america"* of the `continents.Continent` column and semantically similar to *"usa"* of the `countries.CountryName` column. Just by looking at the computed answer, it is hard for users to tell if this NL ambiguity was successfully resolved. Showing the generated SQL query can clarify these issues, but this only helps those who understand the query language.

Prior work has studied ways to explain such question-answering workflows using NL [209, 210, 211, 212, 213] and visualizations [214, 215, 216, 217], including ways to communi-

cate and resolve ambiguities using multimodal interactive widgets [202, 204, 206, 207, 198]. For example, in terms of NL, NaLIR [198] reveals these issues by mapping entities from the input query to the entities in the database schema and presenting them to the user using NL and dropdowns. Su et al.'s [218] system converts a Seq2Sql model API output into NL, augmented with GUI widgets that support error-fixing and disambiguation using fine-grained user interaction. DataTone [207] leverages mixed-initiative interaction through dropdown menus called "Ambiguity Widgets" to resolve ambiguities in the input query. In terms of visualizations, QUEST [216] connects matching entities from a query's input to a database structure. QueryVis [214] automatically generates diagrams of SQL queries that capture their logical intent. Berant et al.'s [215] cell-based provenance model explains the execution of a SQL query using provenance-based highlights on tabular visualizations (e.g., highlighting relevant cells that match a WHERE condition).

We followed a different approach, using the *data* itself to explain the query and the execution process. Additionally, we provision an interactive, self-service debugging view for users to *guide themselves* through the system's execution process, as described next.

4.2 DIY: Debug-It-Yourself

We developed Debug-It-Yourself (DIY; Figure 4.2 and Figure 4.4), an interactive, self-service debugging tool that enables users without specialized knowledge of a query language (e.g., SQL) to assess the responses of a state-of-the-art NL2SQL system for correctness. Specifically, DIY lets users inspect for, isolate, and if possible, fix errors in the system's output; essentially, guide themselves through the system's execution process. DIY presents users with a sandbox where they interact with (i) a *small-but-relevant* subset of the underlying production database, which we refer to as the sample testing database; (ii) mappings between the entities in the question and the generated query; and, (iii) multimodal explanations of the execution of the generated query on the sample testing database. DIY's intended users include domain experts with limited databases experience and information

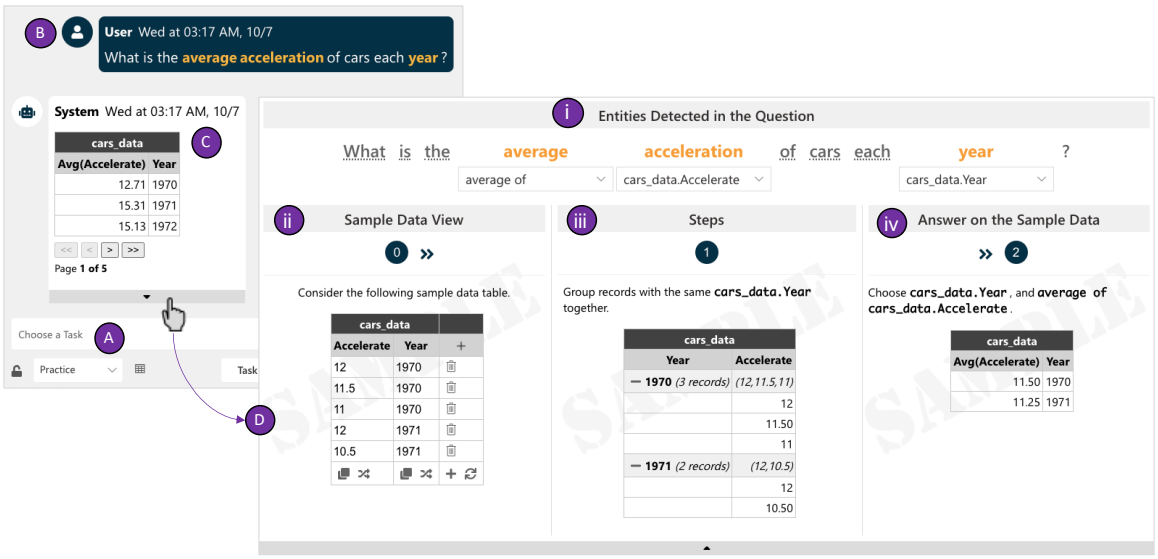
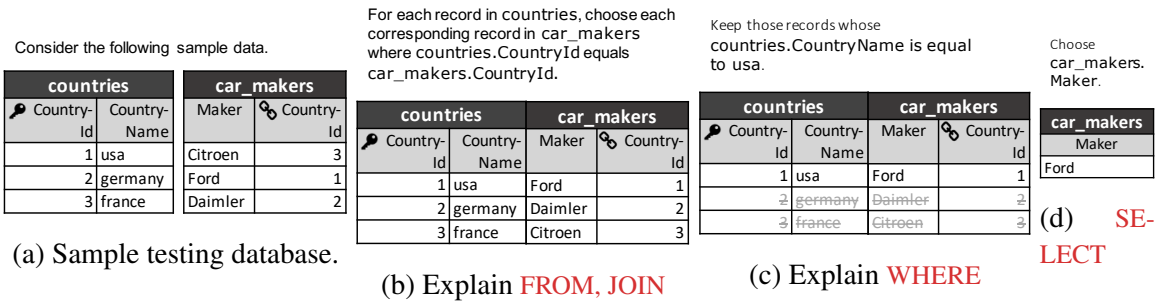


Figure 4.2: The DIY (Debug-It-Yourself) technique implemented in a QA (Question-Answering) shell. **(A)** Query input, **(B)** Annotated Question View shows the question with important tokens highlighted, **(C)** Answer on Production Database View shows the query result on the production database (DB), and **(D)** Debug View. **(i)** Detect Entities View shows the mappings between the question and the query, **(ii)** Sample Data View shows a *small-but-relevant* subset (sample testing DB) of the production DB, **(iii)** Explainer View provides step-by-step explanations of the query, and **(iv)** Answer on Sample Data View shows the query result on the sample testing DB.

workers who are not familiar with writing complex database queries.

For example, Figure 4.3 shows DIY applied to our earlier scenario (Figure 4.1). DIY first identifies relevant tables and columns from the query and samples a few relevant records from the underlying production database (Figure 4.3a). The query is then broken into three subqueries that are sequentially executed on the sample testing database. Each subquery explains one or more SQL clauses: FROM, JOIN (Figure 4.3b), WHERE (Figure 4.3c), and SELECT (Figure 4.3d), respectively using NL and tabular visualizations. Figure 4.3d is also the final answer when the query is executed on the sample testing database.

The sample data and sandbox environment allow users to employ different back-of-the-envelope calculation debugging strategies to assess the correctness of the query. For example, users can experiment by modifying the sample testing database (e.g., edit a cell's value) that updates the subsequent steps including the answer. Experimenting with the



(a) Sample testing database.

(b) Explain FROM, JOIN

(c) Explain WHERE

(d) SELECT

Figure 4.3: A query explained using the DIY technique.

sample testing database can build trust in the system's interpretation of the original question and its output on the production database. If a problem is detected, DIY also presents users with the means to fix errors and resolve ambiguities by allowing them to adjust the mappings between the question and the generated query. The adjusted query is automatically applied to the sample testing database and can eventually be applied to the production database. In this way, DIY guides the user to debug NL2SQL scenarios.

4.2.1 Design Goals

DIY's design was driven by three goals, based on prior work on NLIs for visualization [207, 202] and databases [198, 215], and our hypotheses for enhancing user experience.

DG1. Explain system responses. DIY aims to clarify system responses for users without SQL knowledge, using natural language and visual explanations of queries.

DG2. Support error isolation. DIY aims to enable users to identify errors arising from incorrect or incomplete NL references, by showing mappings between the entities in the question and the generated query to the user.

DG3. Facilitate error correction. Upon discovery, DIY aims to let users directly correct errors and ambiguities through user interface (UI) controls, promoting human-machine collaboration without paraphrasing on the human's part.

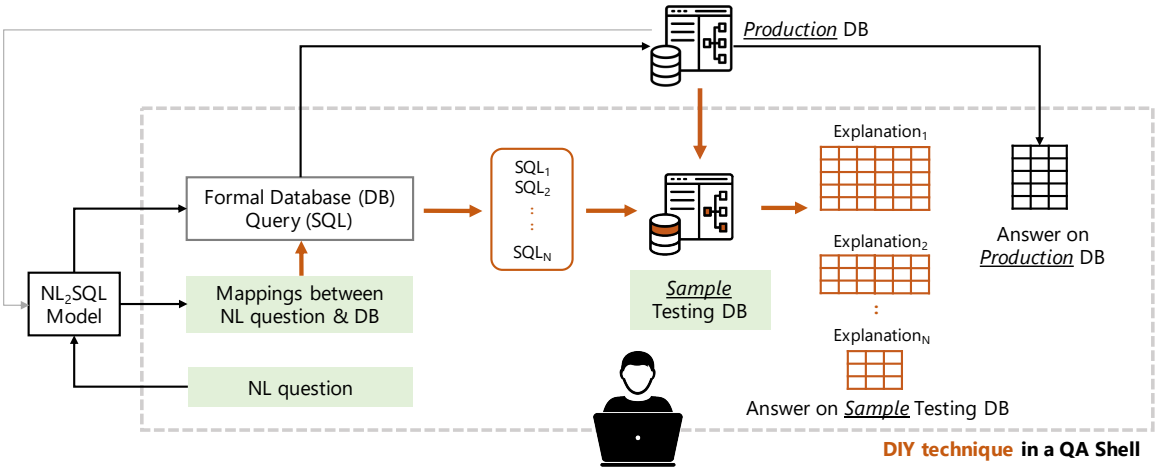


Figure 4.4: Overview of the DIY technique in a QA shell.

4.2.2 User Interface

With the above design goals in mind, we designed DIY and embedded it into a QA shell (Figure 4.2), with the following key components:

4.2.2.1 Generating the Sample Data

DIY's key element and contribution is the use of a sample testing database to provide a sandbox for simplified inspection, testing and debugging. To generate the sample testing database, we clone the production database schema and show only those tables and columns that are part of the generated query. We then apply one of the following two strategies to populate each sample table with *five* records[1].

Smart Constraints. To generate a *small-but-relevant* sample database, the system first lists all entities and expressions specified in the query. Based on these, it identifies *smart constraints* that the data sampling algorithm must satisfy. For example, consider the SQL query: `SELECT Id FROM cars_data WHERE Horsepower>200`. In this query, the filter expression `WHERE Horsepower>200` leads to a constraint requiring that *at least one* of the sample

[1]We chose *five* based on feedback from pilot studies and UI design considerations with respect to visual clutter; currently this limit and sampling criteria are pre-configured.

rows has *Horsepower>200* so that the final result set is non-empty. We also add a second constraint requiring that at least one row has *Horsepower≤200*. This ensures that both sides of the boundary condition are represented so that, when the relevant subqueries are executed on the sample testing database, the subanswers create before-after scenarios that help to visualize the effects of specific operations. Table 4.1 catalogs the *sample constraints* that we considered and implemented for various SQL constructs. We implemented those constraints that had primitive entities, for example, a simple WHERE clause, `WHERE Horsepower>200` comprises {"Horsepower", '>', 200}. On the other hand, both a subquery (e.g., `WHERE Horsepower > (SELECT AVG(Horsepower))`) and a HAVING construct (e.g., `HAVING AVG(Price)>2000`) require an additional computation step using a SQL engine. We did not implement these types of constraints.

Table 4.1: **Smart Constraints:** A catalog of constraints to generate a sample testing database that can effectively explain the execution of the SQL query. IEU* = INTERSECT, EXCEPT, UNION SQL keywords. **Status** = Status of Implementation.

SQL Entity	Constraint Operation	Status
SELECT	Choose all columns mentioned in the SELECT clause.	✓
FROM	Choose all tables mentioned in the FROM clause.	✓
JOIN	Choose records from each *to-be-joined* table such that the *joined* state has at least one record.	✓
GROUP BY	Choose records such that the *grouped-by* columns have duplicate values.	✓
HAVING	Choose records such that the *grouped-by* state satisfies the HAVING expression(s).	×
WHERE	Choose records such that at least one satisfies the WHERE condition, and at least one fails.	✓
DISTINCT	Choose records such that the *grouped-by* column has duplicate values.	✓
LIMIT	Choose records such that the result set has enough records to apply the LIMIT operation.	✓
IEU*	Choose records such that the execution of the subqueries have intersecting subanswers.	×
Subquery	Choose records such that the execution of this subquery produces a non-empty final result set in the query.	×
Functions	Aggregation functions (COUNT, SUM)	×
Operators	Wildcards (*, %), LIKE	×

Human-in-the-loop. For any generated query, it is not always possible to satisfy all *smart constraints*. This can be due to: (i) Practicality: Records that satisfy all constraints may not be common, and the database may not be structured to support efficient sampling of certain

constraint combinations. In such cases, collecting five records may require a linear scan of the entire production database, and this may be too computationally costly to be practical in an interactive setting; (ii) Feasibility: satisfying certain constraints may be impossible. For example, the positive constraint for the question *"How many **car makers have their headquarters on Mars**?"* will be *car_makers.Headquarter="Mars"* which cannot be satisfied. In such scenarios, the system generates *partially-relevant* sample data. Users can then optionally modify the tables in the sample testing database to add records or to modify existing records to make them *more relevant*.

4.2.2.2 Generating Multimodal Explanations

To break the SQL query into subqueries, we consider the order of execution of different SQL clauses. Each step generates a virtual table that is used as the input to the following step. If a certain clause is not specified in a query, the corresponding step is skipped. DIY considers only the forms of SQL queries output from the underlying NL2SQL model (Listing 4.1). This is a subset of valid SQL queries, excluding clauses like TOP and WITH.

Logical Order of Execution of a SQL query. As shown in Listing 4.1, the FROM clause and the subsequent JOINs are executed first to determine the working set of data. Next, the WHERE constraints are applied to the individual rows, discarding the rows that do not satisfy the constraints. The remaining rows are then grouped based on common values as specified in the GROUP BY clause.

If the query has a HAVING clause, it is then applied to the grouped rows – the groups that do not satisfy the constraints are discarded. Next, the expressions in the SELECT clause are computed. This may include columns, or aggregation of functions, or subqueries. If a DISTINCT keyword is present, duplicate records are discarded. Likewise, if an ORDER BY clause is present, the rows are sorted accordingly. Finally, the rows that fall outside the range specified by LIMIT and OFFSET clauses are discarded, leaving the final result set.

(6) SELECT (7)
DISTINCT *select_list*
(1) FROM *left_table*
(2) *join_type* JOIN
right_table
ON *join_condition*
(3) WHERE
where_condition
(4) GROUP
BY *group_by_list*
(5) HAVING
having_condition
(8) ORDER
BY *order_by_list*
(9) LIMIT *count* (10)
OFFSET *count*
(12) *left_SQL* IEU*
right_SQL
(11) *right_SQL*;

4.1: General form of a SQL query, with step numbers assigned according to the order in which each clause is logically processed. *left_SQL* & *right_SQL* represent SQL queries with Steps 1-10. IEU* stands for INTERSECT, EXCEPT, UNION.

(I) *left_SQL*
(a) SELECT * FROM JOIN;
(b) SELECT * FROM JOIN WHERE;
(c) SELECT * FROM JOIN WHERE GROUP BY;
(d) SELECT * FROM JOIN WHERE GROUP BY HAVING;
(e) SELECT *select_list* FROM JOIN WHERE GROUP BY HAVING;
(f) SELECT DISTINCT *select_list* FROM JOIN WHERE GROUP BY HAVING;
(g) SELECT DISTINCT *select_list* FROM JOIN WHERE GROUP BY HAVING ORDER BY;
(h) SELECT DISTINCT *select_list* FROM JOIN WHERE GROUP BY HAVING ORDER BY LIMIT OFFSET;
(II) *right_SQL*
(III) *left_SQL* IEU* *right_SQL*;

4.2: Sequence of subqueries generated by DIY at each step of the Explainer View for a general SQL query represented in Listing 4.1. The underlined text shows the difference with the previous subquery.

Natural Language (NL) Explanations. Prior work has explored methods to translate SQL queries to natural language (SQL2NL). In the context of an NL2SQL system, this can be used to allow the "DMBS to talk back in the same language" as the users, allowing users to verify if their question was interpreted correctly [213]. Several SQL2NL strategies have been explored: Kokkalis et al. [211] and Elgohary et al. [209] employ a template-based approach while Su et al. [210] employ a grammar based approach.

We follow a heuristics-based approach to generate NL explanations for each step in the Explainer View. One notable aspect of these explanations is that they explain the *difference* between the current and the previous subquery. For example, consider two consecutive sub-

queries: (i) SELECT * FROM cars_data and (ii) SELECT * FROM cars_data WHERE Horsepower>200 . The generated NL explanation for subquery (i) is *"Choose all columns from the cars_data table."* and for subquery (ii) is, *"Keep those records whose Horsepower is more than 200."* We hypothesize that this approach can help users to not only understand each step but also enable them to detect and isolate specific errors. Table 4.2 shows the complete list of templates that are currently being used to explain each subquery.

Table 4.2: **Natural Language (NL) Explanation templates** for different SQL clauses. Each template scales to multiple instances (e.g., two WHERE clauses) using punctuations (e.g., ',') and conjunctions (e.g., 'and').

SQL keyword	Natural Language Template
FROM	Choose columns from the {table} table.
FROM + JOIN	For each record in $\{table_1\}$, choose each corresponding record in $\{table_2\}$ where $\{column_1\}$ {operator} $\{column_2\}$.
WHERE	Keep those records whose {column} {operator} {value}.
GROUP BY	Group records with the same {column} together.
HAVING	Keep those groups where {aggregation} of {records/column} {operator} {value}.
SELECT	Choose the {column}.
DISTINCT	Keep unique records.
ORDER BY	Sort the records by {column} in the {orderType} order
LIMIT	Choose the first {N} record(s).
INTERSECT	Choose all records that are common to the answers of Step {M} and Step {N}.
EXCEPT	Choose all records from the answer of Step {M} that are not in the answer of Step {N}.
UNION	Combine all records from the answer of Step {M} and the answer of Step {N}.

Tabular Visualizations. An interactive datatable complements each NL explanation displaying the result *after* the corresponding subquery is executed on the sample testing database. Table headers communicate the table names and column names of data values. Each table is treated as the input to the following step. For example, observe *steps 2, 3, and 4* in Figure 4.6. **Step (2)**, explains the WHERE clause, and the rows that do not satisfy the corresponding constraints are ~~faded out and struck-through~~. Similarly, **Step (3)**, explains the GROUP BY clause, and the grouped records are accordingly visualized.

4.2.3 Implementation

Semantic parsing of NL to SQL has recently surged in popularity thanks to the creation of dataset benchmarks such as WikiSQL [196], Spider [219], SParC [220], and CoSQL [221]. These dataset and associated benchmarks have led to the development of many deep learning models that address semantic parsing [222, 223, 224, 225, 226, 227]. Among these, RAT-SQL [225] uses relation-aware self-attestation and encodes the names of columns and tables, as well as the values of data, into a common dense representation. RAT-SQL achieved state-of-the-art performance on the Spider dataset in early 2020; hence we used it as DIY's underlying NL2SQL engine. We implemented DIY as a ReactJS [228] web application, making API requests to a deployed instance of RAT-SQL over HTTP REST. We utilized SQL.js [229] to manage the sample database directly in the user's browser.

4.2.4 Example Scenarios

In the following two scenarios, we illustrate how DIY can help users assess the generated queries and answers for correctness and detect & fix errors.

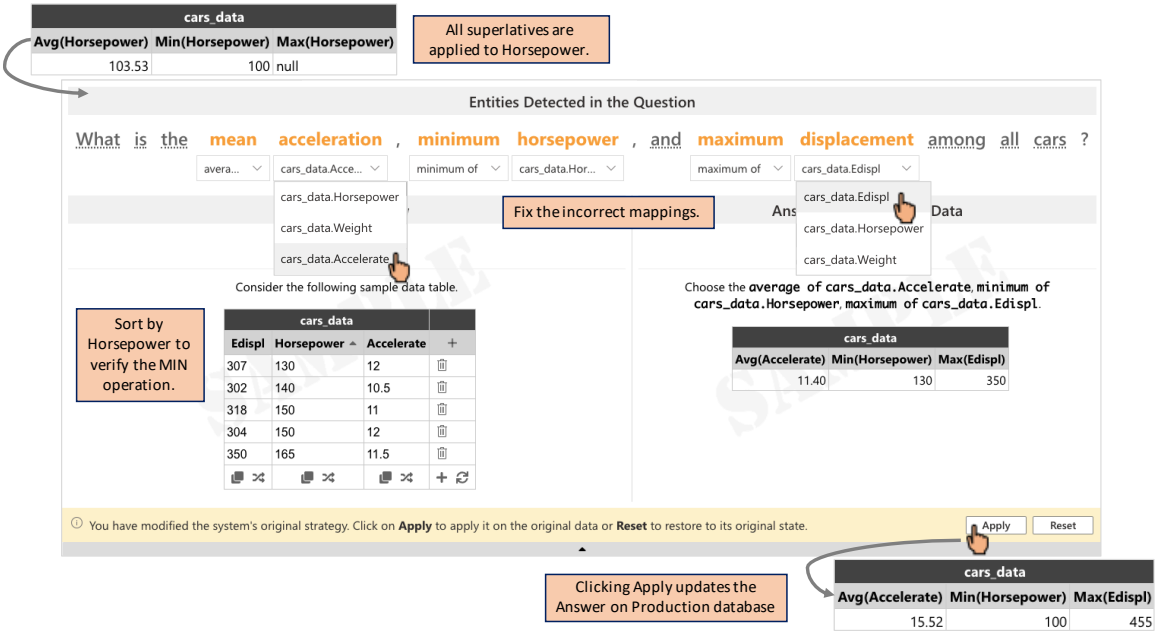


Figure 4.5: Scenario 1: DIY being used to correct a misclassified NL2SQL scenario.

Scenario 1: Fixing the Mapping. Chris, an automotive enthusiast without much knowledge on SQL, loads a database on *cars* (Figure 4.1a) and asks *"What is the mean acceleration, minimum horsepower, and maximum displacement among all cars?"* (Figure 4.5). Upon reviewing the system's response, Chris notices that even though the system correctly identified the three superlatives (mean, minimum, maximum) and attribute keywords (acceleration, horsepower, displacement), it applied all superlative operators only to the *Horsepower* attribute. Convinced that the result is not correct, Chris expands the Debug View to repair the output. From the Detect Entities View, Chris notices the incorrect mappings, and selects the correct attributes from the respective drop-downs (cars_data . Accelerate and cars_data . Edispl). Based on these new mappings, the system automatically updates the sample data, and produces a new answer for inspection.

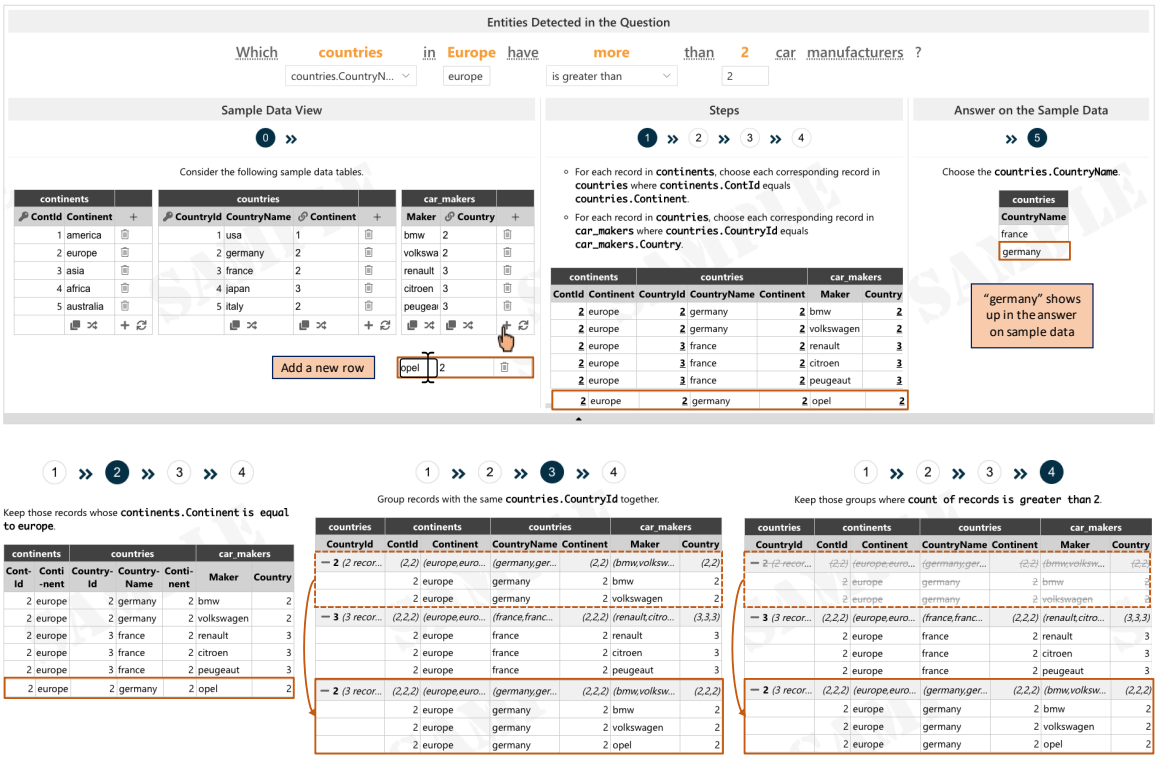


Figure 4.6: Scenario 2: DIY being used to debug a complex NL2SQL scenario.

Next, Chris wants to verify if the superlatives were interpreted correctly. Chris sorts the records by Horsepower by clicking on the Horsepower column in the Sample Data View and

verifies that the first (i.e., smallest) record matches the computed value in the Answer on the Sample Data View. After similarly checking the *Max* operation, Chris is convinced that the system now correctly performs the query. At this point, Chris notices a system alert indicating that the mapping changes have only been applied to the sample testing database, and that they must Apply or Reject them. Chris applies the changes and the Debug View closes. The answer on the Production Data updates accordingly.

Scenario 2: Checking the System Strategy. Being curious about the European automobile industry, Chris now asks the system, *"Which countries in Europe have more than 2 car manufacturers?"* (Figure 4.6). While checking the answer—*Germany, France*—they wonder based on prior knowledge why *Italy* was not included. To investigate this, Chris expands the Debug View and inspects the Detect Entities View. After confirming that the existing mappings are correct, Chris checks how the query is being executed on the sample testing database by reading through the four steps shown in the Explainer View: Step 1 joins the three tables; Step 2 removes non-European countries; Step 3 groups the rows by `countryId`; and Step 4 first counts the number of rows per `countryId` and then removes those groups that have less than or equal to 2 rows (indicated by ~~gray color and strike-through~~).

To verify the system's strategy, Chris decides to test it by manipulating the sample data. In the `car_makers` table of the sample testing database, they add a new row for the German car manufacturer "opel." On inspecting the updates to the subsequent steps, Chris confirms that "germany" now has three records, is no longer removed by step 4, and thus appears in the final answer (step 5). Satisfied that Italy was likely excluded for having fewer than 3 records, Chris closes the Debug View ⏶ without making or applying further changes.

4.3 Evaluation: Exploratory User Study Using DIY as a Design Probe

After developing the DIY prototype and receiving an approval from our ethics board, we conducted an exploratory user study and design probe with 12 participants. With this study,

we aim to understand how users utilize DIY to assess the generated results for correctness, and detect and fix errors in NL2SQL scenarios. In the following sections we describe the participants, detail the high-level procedure, and present the specific study tasks. We then present and discuss our findings.

4.3.1 Participants and Procedure

We recruited 12 participants (4 female, 7 male, 1 preferred not to say). They worked for a large technology company in different roles including UX Designers, Design Researchers, Site Reliability Engineers, Data Scientists, Cloud Solution Architects, Program Managers, and Research Interns. We compensated each participant with a $25 Amazon Gift card.

Due to the COVID-19 pandemic, we leveraged numerous Internet collaboration tools to conduct the study remotely. Participants were asked to complete a brief online demographics questionnaire, and to connect with the experimenter using the Microsoft Teams teleconferencing software. Participants were then quickly briefed about the study, and were presented with a 5 minute tutorial video that demonstrated the features of DIY. Following the video, the experimenter provided participants access to the study environment by sharing the study computer's screen and granting input control. Participants were then asked to complete 8 tasks of varying difficulty using DIY, and to think out loud while interacting with the system. Participants were free to ask questions at any time, and the experimenter occasionally asked questions to probe participants' strategies. The study ended with a debriefing in which participants completed a system usability score (SUS) [230] questionnaire, discussed their overall experience with the system, and provided suggestions for improvements. The entire session took 90 minutes to complete. All sessions were screen-recorded, and transcripts were later generated using automated software.

The eight tasks were organized into sections according to complexity: 3 easy, 2 medium, and 3 hard tasks (Table 4.3). Inspired by Spider [219], we determined the complexity based on the count and types of SQL clauses (e.g., GROUP BY, INTERSECT, MIN()) and the count

and types of errors in the generated SQL query (e.g., wrong operator, missing column). For example, the query corresponding to **Task #6** in Table 4.3 is *hard* because it has two JOINs and one each of: WHERE, GROUP BY, HAVING, and SELECT.

Within each complexity level, half of the tasks resulted in correct outputs from DIY, which participants might still choose to verify, while the other half contained NL2SQL translation errors requiring correction. We curated these tasks by posing varied questions to the RAT-SQL model and inspecting the responses, ensuring an equal distribution of correct and incorrect outputs across a range of SQL complexities and error types.

Table 4.3: **Tasks used in the evaluation of DIY.** Each task includes: (1) the natural language question input, (2) if it has errors (Yes or No), (3) type of error (e.g., Wrong operator), and (4) the heuristically determined overall task complexity (Easy, Medium, or Hard).

No.	Question	Error	Error Type	Complexity
1	What is the mean acceleration, minimum horsepower, and maximum displacement among all cars?	Yes	Wrong columns	Easy
2	What is the average acceleration of cars each year?	No	-	Easy
3	Which products are manufactured in Austin?	Yes	Wrong column	Easy
4	Which products by Sony are priced above 100?	No	-	Medium
5	Which car models are produced since 1980?	Yes	Wrong operator	Medium
6	Which countries in Europe have more than 2 car manufacturers?	No	-	Hard
7	Which continent has the most car makers? Also list the count.	Yes	Missing column	Hard
8	Which car models are lighter than 3500 or built by BMW?	No	-	Hard

4.3.2 Results and Discussion

Our observations revealed the benefits of using sample data to help users assess the correctness of the system's responses, and both a range of debugging rationales and strategies across participants. We present these observations below, and discuss observations that could indicate where DIY may benefit from additional refinement.

General Reactions. Overall, participants liked DIY's approach of using sample data to explain the system's strategy. P10 commented, *"It is important to have this transparency and to show people how the system is working and to let them control it. This is a great*

example of that." P8 commented, *"I think it's really cool and I think there are a lot of customers who would benefit from something like this."* P14 liked the multimodal explanations, commenting *"I really liked your idea of the linear stuff...kind of a visual explanation of the query path."* Also, participants rated their experience with an average SUS of 65.42; while this is encouraging, further refinement is possible.

Debugging Rationales. Our system's initial response highlighted important tokens in the question and the computed answer on the production database (Figure 4.2B,C). Participants inspected these first and then, based on their assessment, optionally chose to expand the Debug View. We observed different rationales for electing to debug, including: (i) the option was available (*"Just because I can!"* – P8), (ii) they detected an error (everyone), (iii) they just wanted to double-check the answer or strategy (almost everyone, *"I want to verify the Average."* – P1), (iv) they did not have enough domain expertise to trust the answer on production database (*"I am not good with cars"* - P1), (v) they had some domain expertise that led them to suspect the answer (*"I can think of a couple more rows so I'm just gonna verify"* - P8). At times, participants did not utilize the Debug View because: (i) they had begun to trust the system (*"I will probably not verify, I trust the system by now."* – P1, *"Assuming the math is correct, this seems fine."* – P7), (ii) they were satisfied with the orange highlights in the Annotated Question View (*"it looks to me that it highlighted the right keywords"* – P5, P14). Many participants also expressed a need for additional context to interpret the answers, even if that context was not explicitly requested in the original question. For example, for the question "Which products by Sony cost more than $100?" the system's response returned only `Products.Name` values. P1, P7, and P17 wished to see more columns, including the manufacturer and price, to facilitate inspection. Likewise, P7 suggested adding rows that do not match the criteria, and striking them out using the same visual convention employed by the DIY Explainer View.

Strategies. With the Debug View, participants employed three broad strategies to verify the query. In the first category were three participants (P1, P16, P17) who expressed concerns about modifying the sample data, and predominantly utilized inspection (e.g., of term mappings and explanations) to assess the correctness of the query. P1 commented, *"My judgement is based on the result I see, if I manipulate the data, I don't trust the result anymore, and I don't trust the system anymore."*

In the second category were participants who modified the sample data to explore counterfactual *what-if scenarios*. Specifically, participants modified sample data to (i) generate positive (or negative) scenarios (*"Now that I have been able to generate an affirmative case, I am more happy with this"* – P8) or (ii) to test specific boundary conditions (*"I want to make sure I have tested the right boundary conditions"* – P15). We observed participants manipulate the sample testing database in several ways. One participant chose to *delete irrelevant rows* from the sample data (*"I might as a matter of figuring this out remove everybody I don't care about"* - P8). One participant chose to *add a new test row* (*"as I did not want to manipulate existing data"* - P15). One participant sorted the sample data tables to verify the *MAX* and *MIN* superlatives in the question. Most participants edited specific cells in the sample testing database, e.g., *"ford"* to *"bmw"* to verify a WHERE clause, or change the `CountryId` from *1* to *2* to create a successful JOIN.

Finally, in the third category participants manipulated the mappings in the Detected Entity View to, for example, test boundary conditions. One participant modified the operator mappings "is greater than" to "is less than" to test a reverse scenario (P8). Another participant changed the attribute *Price* to *Revenue* to verify the query response updated accordingly. This strategy is interesting because modifying the mappings changes how the system interprets the user's original question. Accordingly, these affordances were intended for *fixing errors* or *resolving ambiguities*. Some participants were aware of this and planned to revert to the original mapping after testing. Other participants were reminded by the notification at the bottom of the Debug View.

4.4 Limitations and Future Work

System Limitations. While the system supports modifying existing mappings from question tokens to database columns or operators, it is more limited in what new mappings can be added. For example, unmapped tokens may only be mapped to columns previously implicated by the system's original interpretation of the question. Likewise, the sample data generation and the explanation generation modules currently do not support all SQL constructs. For example, neither module generates *smart constraints* or *multimodal explanations* for window functions (e.g., OVER) or wildcard operators (e.g., LIKE, %), since the NL2SQL backend does not currently support these constructs, though future versions may.

Minimizing confusion between production and sample data. The DIY technique currently presents two distinct answers for any given query: one for the production database, and a second for the sample testing database. At multiple points during the study, participants (P1, P8, P11, P14) exhibited confusion as to why the two answers did not match. P11 commented, *"OK, so it says monitor here (in the Answer on Sample Data View), which is what I was expecting. Why does it say CD drive, DVD drive (in the Answer on Production Data View)?"* They failed to recognize that the sample testing database is a very small subset of the production database. We will further refine the user interface to clearly distinguish between the two kinds of databases and minimize this confusion.

Generating a smarter sample testing database. The sample data generation module identifies entities from the query and defines constraints that, if satisfied, would generate a relevant sample. As discussed earlier, practicality and feasibility related restrictions further constrain sampling. Participants pointed out this limitation when they encountered a sample testing database that they felt they need to modify further to enable relevant debugging. P1 commented, *"For me, to build trust with the system, I would want the system to be smart enough and return sample data relevant to the question."* They went on to suggest the

human to be more involved in the generation of the sample data, *"I wonder if I could tell the system to return sample data post 1980 so then I can verify if the answer is indeed correct."* Thus, we will continue to refine our sample data generation algorithm.

Improving the multimodal explanations. Some participants found it challenging to follow the explanations for certain SQL constructs. For example, P5 did not understand multi-table JOIN conditions. P7 worried that the use of (too many) *IDs* in the sample testing database and the JOIN condition resulted in added complexity. Some participants failed to interpret compound SQL clauses (e.g., UNION) as it was presented in a linear manner just like other SQL clauses. We will thus explore alternate representations for these SQL clauses (e.g., representing subqueries in a tree-like representation, and using animations to visualize multi-table join operations).

Handling ambiguities between conversational and formal language. For one of the task questions–*"Which car models are produced since 1980?"*–the NL2SQL system mapped the token "since" to the "greater than" operator. In colloquial conversation, "since" often implies a "greater than or equal to" operation, and thus this mapping needed to be fixed. It was interesting that six participants (P1, P7, P11, P12, P15, P17) pointed out this ambiguity, commenting that it is sometimes up to the user's interpretation. In the future, we will put guardrails to caution the user when making decisions about such ambiguities.

Leveraging manipulation to facilitate understanding. Recall some participants modified the mappings between the question and the database entities to either test a boundary condition (e.g., *is greater than* $\rightarrow$ *is less than*) or to observe a change (e.g., *Price* $\rightarrow$ *Revenue*). This was interesting as the participants deliberately modified what were already correct mappings. We envision this to be an opportunity to support data exploration. For example, consider a scenario wherein a user first asks for cars with Acceleration$>$100 and upon inspecting the answer, is interested in cars with Acceleration$>$200 instead. Answer-

ing this question in a QA system generally involves paraphrasing the original question.

Teach SQL. We believe NL can be a powerful tool for teaching SQL. Existing tools (e.g., *SQL Fiddle* [231], *Tryit Editor* [232]) already provide users with a sandbox for executing SQL queries on datasets. Integrating DIY could provide an NL interface to help novices formulate SQL queries along with step-by-step multimodal explanations.

4.5 Summary

In this chapter, I described a question-answering chatbot system, enhanced with an interactive, self-service debugging view (**Debug-It-Yourself** (**DIY**)), for users to interactively debug (i.e., inspect for, isolate, and fix errors in) natural language to SQL (NL2SQL) workflows; essentially, guide themselves through the system's execution process. DIY provides users with a sandbox where they can interact with (1) the mappings between the question and the generated query, (2) a *small-but-relevant* subset of the underlying database, and (3) multimodal explanations of the generated query. Through an exploratory user study with 12 participants, we investigated how DIY helped users assess the correctness of a state-of-the-art NL2SQL system's answers and isolate and fix errors. Our observations revealed how DIY helped participants assess the correctness of the system while providing insights about different debugging strategies, and associated challenges. For details, I refer the reader to the associated publication [26] and patent [15].

CHAPTER 5

DESIGNING A MIXED-INITIATIVE, CO-ADAPTIVE GUIDANCE SYSTEM

In this chapter, I describe a mixed-initiative system that facilitates a co-adaptive guidance dialog between the user and the system, Lumos [27].

Lumos helps increase awareness of exploration biases by visualizing traces of a user's interactions. Additionally, Lumos also allows users to customize the target/baseline interaction behavior and receive contextual guidance, partly achieving **RG2**: *Design a mixed-initiative guidance system, wherein the user and the system learn from and take initiative on behalf of each other, co-adaptively steering the analytic process*.

This chapter is based on work published in IEEE TVCG [27].

5.1 Motivation and Background

Visualizations take advantage of people's perception to facilitate intuitive understanding of data. Interactive features of visualizations become critical when considering complex data, allowing people to progressively refine visual representations of data, e.g., by adjusting encodings to represent different attributes of the data or employing filters to reduce the scope of the data at hand. While it can aid in comprehension of large and complex data, certain patterns of interactivity can signal insular data analysis practices. Users may be unknowingly stuck inside an "echo chamber", where their own unconscious biases may lead them to attend to certain parts of the data while neglecting others.

Unconscious biases can take many forms, some of which are relatively innocuous (e.g., preference for a particular chocolate flavor) while others can lead to costly incorrect decisions or engender harmful societal stereotypes (e.g., dark-skinned people are denied parole [233]). Apart from implicit biases and stereotypes, there are other cognitive and perceptual biases that also influence people's analytic behaviors. Cognitive biases describe

systematic errors that can result from the use of "fast and frugal" heuristics [234] to make decisions. Several biases have been shown to affect decision-making tasks involving visualizations (e.g., [235, 236, 237, 238]). Yet, common visual data analysis tools such as Tableau and Microsoft Excel that help users see and understand their data do not report analytic behaviors that may correspond to such biases. So we asked: "*how much can understanding data analysis and decision-making behaviors reduce the potentially negative influences of potential cognitive, perceptual, or societal biases, if users were simply more aware of these often unconscious factors?*"

In response, we built a visual data analysis system (Lumos) and study how showing a user prior interaction history (introducing the concept of **"interaction traces"**) might be used to mitigate potential biases that may be driving one's data analysis and decision-making, as described next.

5.2 Lumos

Lumos is a visual data analysis system that visualizes interaction history with data (i.e., **interaction traces** [25]) to increase **awareness** of potential interaction biases that influence data analysis and decision-making processes. Using *in situ* and *ex situ* visualization techniques, Lumos provides real-time feedback about a user's **analytic behavior** for self-awareness and self-reflection to potentially change future course. For example, Lumos remembers and highlights datapoints that have been previously examined in the same visualization (in situ) and overlays the interacted datapoints on the underlying data distribution in a separate visualization (ex situ) for comparison. Furthermore, Lumos allows users to configure a custom target distribution to reflect decision-making goals, e.g., a university admissions committee in a computer science department may define an analysis target of 60% female to promote increased gender diversity in the department, even if only 40% of applicants are female. In doing so, Lumos facilitates a co-adaptive guidance dialog [44].

5.2.1 Design Goals

Our development of Lumos was driven by **five** key design goals. We compiled these goals based on a combination of similar prior visual analysis tools [84, 239, 109], formative feedback from pilot studies, and our own hypotheses with respect to usability.

DG1. Capture and present analytic behavior with attributes. Overemphasis (or underemphasis) on specific attributes during data exploration may lead to unconscious biases (e.g., not interacting with a Gender attribute may practically result in a bias towards men if the dataset has more men than women). This goal translates to capturing user interactions with *attributes*, modeling analytic behavior, and showing interaction traces to increase awareness to influence changes in subsequent interactions.

DG2. Capture and present analytic behavior with datapoints. Overemphasis (or underemphasis) can also occur on specific values of data (e.g., interacting mostly with a few top candidates for university admissions may come at the expense of neglecting other candidates). This translates to the same goal as **DG1** but at the datapoint-level.

DG3. Facilitate configuring different target distributions. Determining overemphasis (or underemphasis) on specific attributes or data requires comparing a user's analytic behavior with a known target distribution (e.g., the underlying data) as a baseline. However, different domains, tasks, attributes, or social norms may call for different target distributions. This goal translates to allowing users to configure different target distributions to suit their requirements.

DG4. Facilitate comparison between analytic behavior and a baseline distribution. This goal translates to visualizing the user's analytic behavior and the configured target distribution and quantifying the difference between the two distributions.

DG5. Facilitate visual data exploration while showing awareness. This goal ensures that system usability is not sacrificed by the added awareness visualizations.

5.2.2 Quantifying Analytic Behavior

We quantify *analytic behavior* using (1) the attribute distribution (AD) metric [111] and (2) the relative frequency of interactions with data and attributes. The AD metric characterizes how a user's interactive behavior deviates from expected behavior, and ranges from 0 (no bias) to 1 (high bias). By default, the system chooses a **proportional** baseline of expected behavior, wherein interactions with any given datapoint are equally likely, reflecting the true underlying distributions of attributes in the dataset. For example, if a user primarily interacts with PG-13 movies in a dataset that predominantly contains G-rated movies, the AD metric for the Content Rating attribute will be high (more emphasis). If the user instead spent more time interacting with G-rated movies, proportional to the distributions in the dataset, the AD metric value for Content Rating would be low (less emphasis).

Additionally, Lumos enables users to define their target interactive analytic behavior (or alternative baselines) in multiple ways: (1) by **proportional** interactions across the various attribute distributions of the dataset, (2) by **equal** interactions across the categories of the dataset, and (3) by defining a **custom** target distribution of interactions across the data (**DG3**). For example, consider a dataset of job applicants, where 50% of applicants identify as male, 40% of applicants identify as female, and 10% of applicants identify as non-binary. A **proportional** baseline would define the target distribution of interactive behavior such that 50% of interactions should be with male applicants, 40% with female applicants, and 10% with non-binary applicants, while an **equal** baseline would set the target distribution of interactions with 33.3% male applicants, 33.3% female applicants, and 33.3% non-binary applicants. If, for instance, diversity is a target in filling this particular role, then a **custom** baseline might be set, where the target interaction distribution is 40% female, 40% non-binary, and 20% male applicants. Figure 5.3 summarizes these settings in the context of a dataset about *movies* for the Content Rating attribute, and shows (in blue) how the user's actual analytic behaviors compare to the target. Users can configure **proportional**, **equal**, or **custom** target distributions per attribute in Lumos. In the **custom**

mode for categorical attributes, users are presented with an interactive bar chart where they can drag individual bars (each representing a category) to their desired relative weights. For quantitative attributes, users can sketch a target distribution by clicking (to add new quantiles) and dragging points in the presented interactive histogram. Section 5.2.4.3 describes an example usage scenario demonstrating **equal** and **custom** target distributions.

5.2.3 User Interface

Figure 5.1: The Lumos UI includes traditional visual data analysis functions – **A Data Panel**, **B Attributes Panel**, **C Encoding Panel**, **D Filter Panel** – and shows analytic behavior as in situ and ex situ interaction traces in the **B Attributes Panel**, **E Visualization Canvas**, **F Details View**, and **G Distribution Panel** as the relation between the user's analytic behavior and a target distribution (e.g., the underlying data), and a **H** Settings Panel to configure different targets (e.g., proportional (default), equal, and custom).

The Lumos user interface consists of the following views:

- **A Data Panel** shows the currently loaded dataset.
- **B Attribute Panel** shows dataset attributes and their datatypes: {Nominal (N; **A**), Quantitative (Q; **#**), Temporal (T; 📅)} and buttons to apply a filter (⏷).
- **C Encoding Panel** shows UI controls (dropdowns) to create visualizations by specifying different encodings: {Chart Type, X Axis, Y Axis, Aggregation}. Lumos

currently supports *four* visualization types: {*scatterplot, strip plot, bar chart, line chart*} and *five* aggregation types: {*count, sum, minimum, maximum, average*} depending on the attribute data type combinations.

- (D) **Filter Panel** shows UI controls (range sliders for {Q, T} and multiselect dropdowns for {N} attributes) to filter data. Filters can be added by clicking on ▼ in the Attribute Panel (B) **(DG5)**.
- (E) **Visualization Canvas** renders the visualization based on the Encoding (C) and Filter (D) Panel specifications.
- (F) **Details View** shows additional information when the visualization elements (e.g., point, bar, strip) in (E) are interacted with. Hovering on a single datapoint (e.g., a strip in a strip plot) shows a list of all attribute values for the given datapoint (Figure 5.1F). Hovering on an aggregation of datapoints (e.g., a bar of a bar chart) shows a table of all datapoints that belong to that aggregation (the bar) with attributes as columns and values as rows (Figure 5.2).
- (G) **Distribution Panel** shows a list of attribute cards similar to the Attribute Panel where clicking on a card toggles open/close a visualization that overlays user's interaction traces on datapoints (blue area) on the target distribution (black curve) **(DG3)**.
- (H) **Settings Panel** shows UI controls (radio buttons) for each attribute to switch between target distributions. Currently, Lumos supports *three* types of target distributions: *Proportional* (default), *Equal*, and *Custom* **(DG4)**. For the *Custom* type, users are presented with an interactive bar chart (N) or an interactive area chart (Q, T) to drag and sketch custom target distributions.

5.2.3.1 In situ Interaction Traces

Visualization Canvas. Lumos tracks user interactions with visual representations of datapoints (e.g., bars, lines, points, strips) and colors them on a white→blue scale based on the relative frequency of interactions, e.g., dark blue color represents more interactions (Fig-

ure 5.2) and white represents no interactions. Lumos captures mouseover interactions as a proxy for modeling analytic behavior from interactions with datapoints. Lumos employs a heuristic to ignore mouseovers that are active less than a 350 milliseconds threshold, regarded as random, accidental, or unintentional.

An interaction with a unit visualization (e.g., hovering on a point in a scatterplot of *Running Time* and *Worldwide Gross*) is handled differently than an interaction with an aggregate visualization (e.g., hovering on a bar showing average *Running Time* of *Action* movies). In the former scenario, Lumos treats it as one *complete* interaction with the corresponding datapoint incrementing its interaction counter by 1. In the latter scenario, Lumos treats the interaction as a set of *partial* interactions with all constituent datapoints (e.g., all *Action* movies), incrementing their corresponding interaction counters by 1/N where N=number of constituent datapoints.

Details View. When an aggregate visualization element (e.g, bar) is hovered on, the Details View below the chart shows a table with each datapoint. Lumos captures a mouseover on a table row, treats it as an interaction with the corresponding datapoint, and leaves an interaction trace by updating the table row's background color (Figure 5.2).

Attribute Panel. Like datapoints, Lumos also tracks user interactions with attributes. Each attribute card in the Attribute Panel is colored on a white→blue scale based on the corresponding number of interactions (white=no interaction; darkest blue=most interactions). Lumos captures attribute assignments to encodings (e.g., X, Y) and filters (e.g., Gender=Male) as a proxy for modeling analytic behavior from interactions with attributes. These interactions are totaled and normalized relative to the most interacted attribute to determine the resultant shade of blue, e.g., in Figure 5.1B, *Genre* has been interacted with most (dark blue) and *Worldwide Gross* has not been interacted with at all.

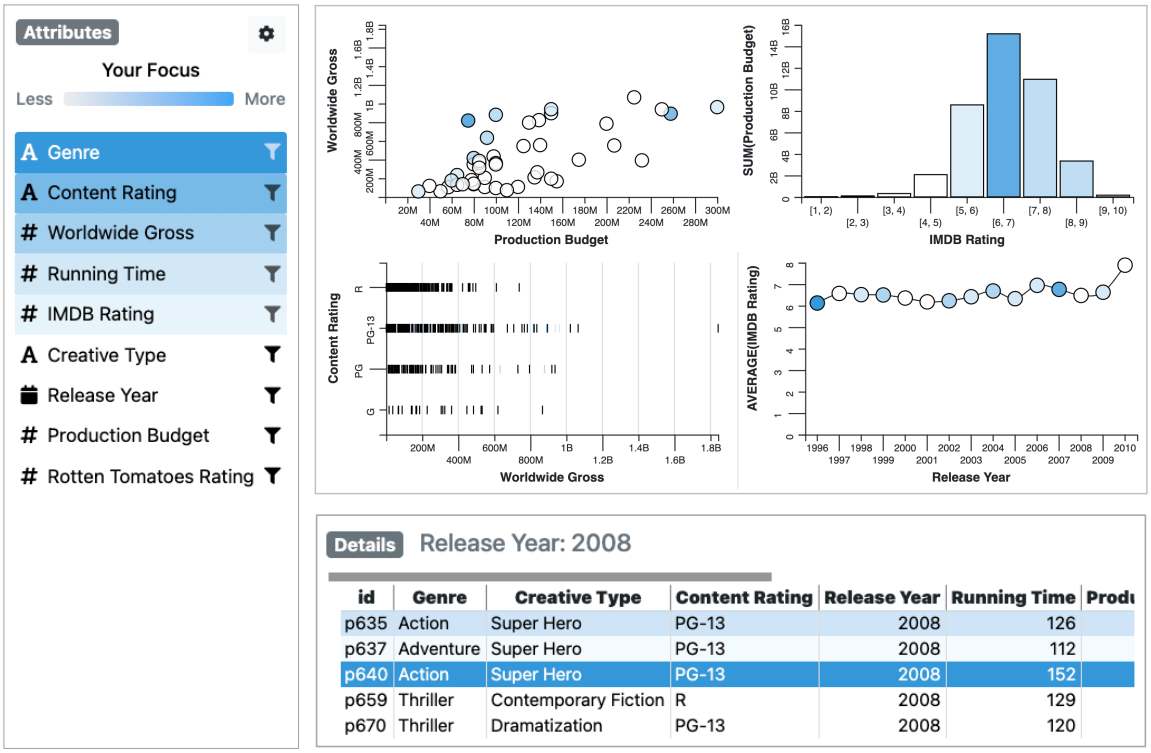


Figure 5.2: In situ Awareness of Interaction Traces

5.2.3.2 Ex situ Interaction Traces

Lumos allows users to compare their analytic behavior with a target distribution. Attribute cards in the Distribution Panel are colored on a red→gray→green scale (Figure 5.1G) based on the difference between their respective analytic behavior and underlying distributions (as quantified by Wall et al.'s [111] AD metric). In this evaluation, we set the target distribution to the underlying data. A red background indicates that the user's analytic behavior is different from the target distribution (green background indicates similarity). For example, inspecting the visualization for *Production Budget* shows the analytic behavior peaking around 150M (blue area) when most movies have a budget under 50M (black curve); the magnitude of the deviation is high resulting in a red background. Similarly, the computed analytic behavior (blue bars) on *Content Rating* is more closely matching the underlying data (black strips) resulting in a green background.

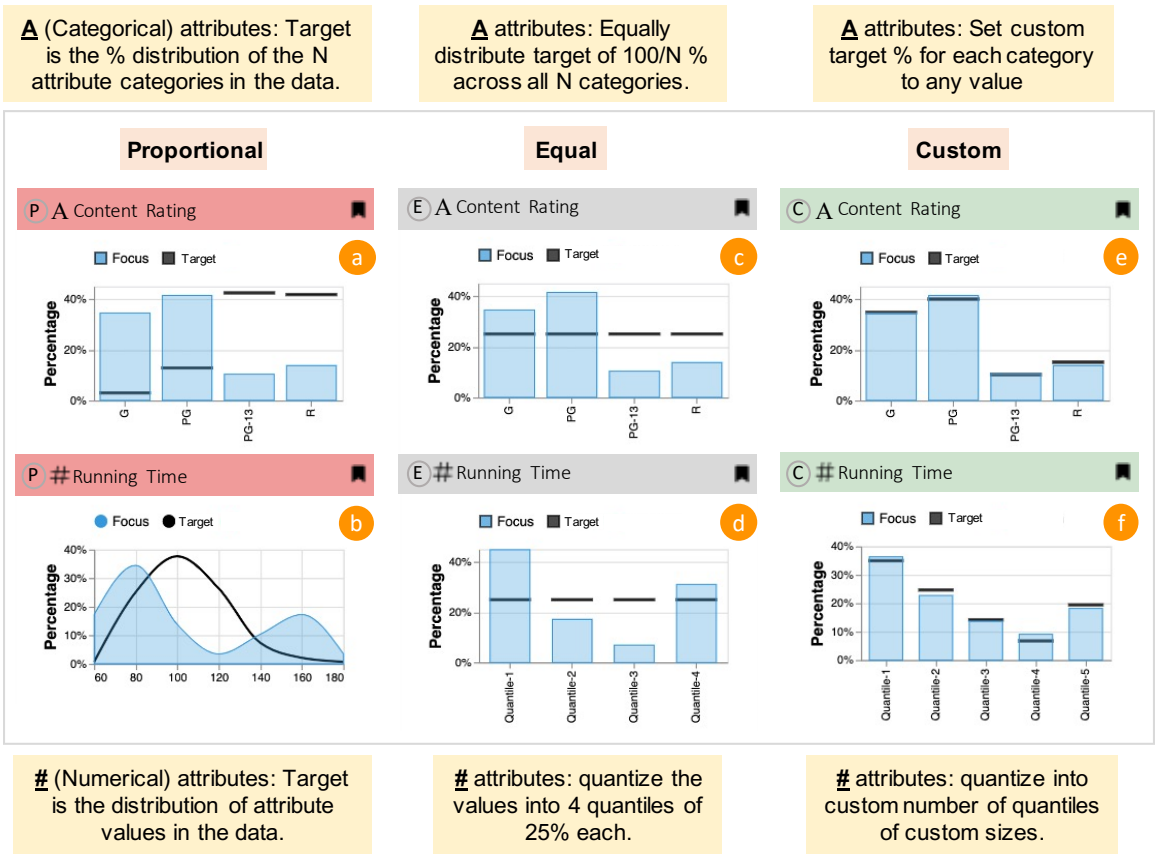


Figure 5.3: Ex situ interaction traces for three modes of target distributions (*Proportional*, *Equal*, *Custom*). These targets in the charts are presented as black curves/strips along with user behavior (blue area). Lumos also computes the difference between target and observed behavior and encodes it as the background color of the corresponding attribute card (red, gray, green colors where redder=more different; greener=more similar).

5.2.4 Example Scenarios

5.2.4.1 Scenario 1: Increasing awareness of analytic behavior

Assume Austin is looking for a new home and is exploring a housing dataset in Lumos (Figure 5.4). After acquainting themselves with the attributes, they apply three filters that match their criteria: $\{$*Home Type*=*Single Family*; *Price* $\leq$ *\$300K*; *Satisfaction* $\geq 7\}$ (Figure 5.4a). Then, they create different visualizations by specifying encodings (Chart Type, X, Y, and Aggregation) in the Encoding panel (Figure 5.4b).

While interacting with these different visualizations, they observe visualization elements (e.g., bars, points) changing colors to different shades of blue. For example, in the

scatterplot configuration with *Lot Area* (Y axis) and *Year* (X axis) (Figure 5.4c), Austin observes their focus has been on smaller ($Lot\ Area \leq 60K$) and more recently constructed ($2009 \leq Year \leq 2010$) homes. Similarly, in the barchart configuration with *Foundation Type* (X axis) and *Average(Price)* (Y axis) (Figure 5.4d), they observe they have not focused on two types of *Foundation Types*: $\{$*Brick & Tile*, *Poured Concrete*$\}$ (white).

During their analyses, they also observe different shades of blue in the Attributes Panel (Figure 5.4e) inferring they have not focused on all attributes equally, e.g., they focused more on *Price* and *Satisfaction* (darker blues), not so much on *Year* and *Foundation Type* (light blues) and not-at-all on *Fireplaces* and *Heating Type* (white).

After acknowledging that they did focus on the blue attributes, they start inspecting the five white attributes. They state they do not care about $\{$*Lot Config* and *Fence Type*$\}$ but regret not focusing on the other three attributes (*Heating Type*, *Fireplaces*, *Central Air*) associated with climate control as the city faces severe winters. Accordingly, they apply new filters and encodings and continue their analyses.

In this way, Lumos helped Austin in house-hunting by making them more aware of their analytic behavior with data and attributes.

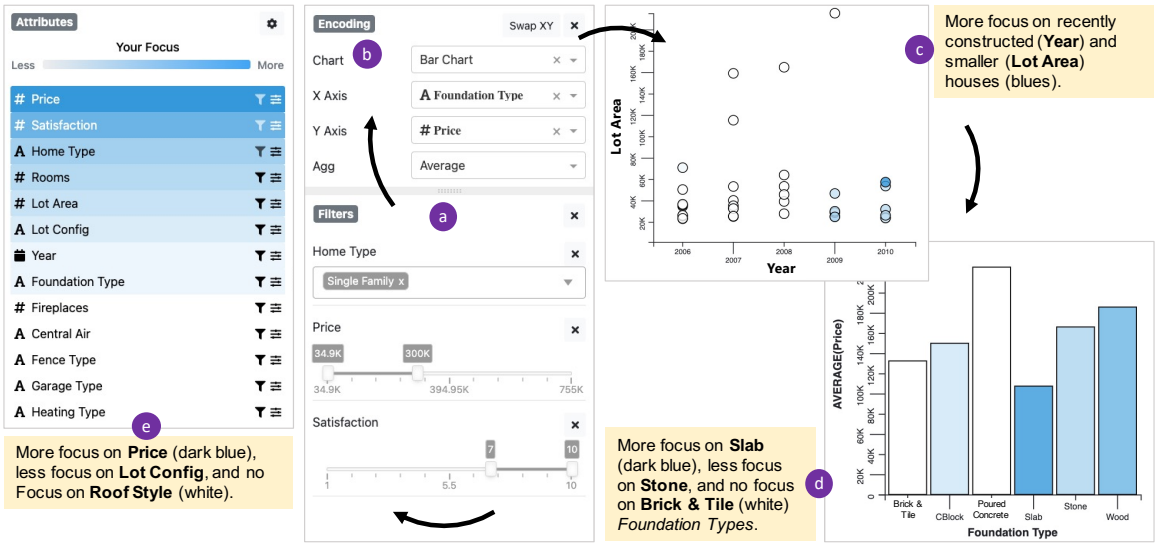


Figure 5.4: Lumos Example Scenario 1: Increasing Awareness of own Data Analysis

5.2.4.2 *Scenario 2: Mitigating biased analytic behavior*

Kiran, a loan officer, is using Lumos to analyze loan applications to determine credit-worthiness. After exploring the dataset for a while, they observe a red *Home Ownership Type* attribute card and a green *Age* attribute card (Figure 5.5a) in the Distribution Panel. They express happiness at not exhibiting any age bias but are concerned that their interactions with *Home Ownership Type* significantly deviate from the underlying data (target) distribution. They click on the card to toggle it open and begin inspecting the visualization. They observe they have unknowingly overemphasized on *Own* and *Rent* and underemphasized on *Mortgage Home Ownership Type*. Willing to correct their behavior, they apply a (reverse) filter: {Home Ownership Type=*Mortgage*} (Figure 5.5b) and analyze a few previously unconsidered (white) points (Figure 5.5c). They finally see a greener *Home Ownership Type* card (Figure 5.5d) and are more content.

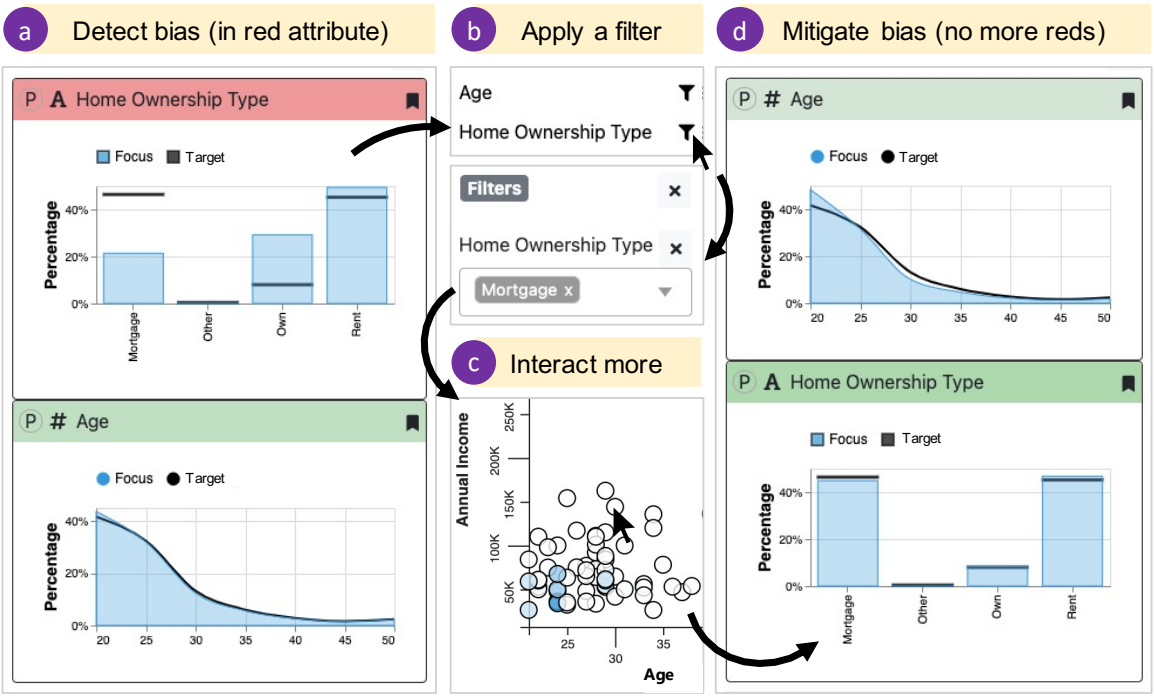


Figure 5.5: Lumos Example Scenario 2: Mitigating Biased Analytic Behavior

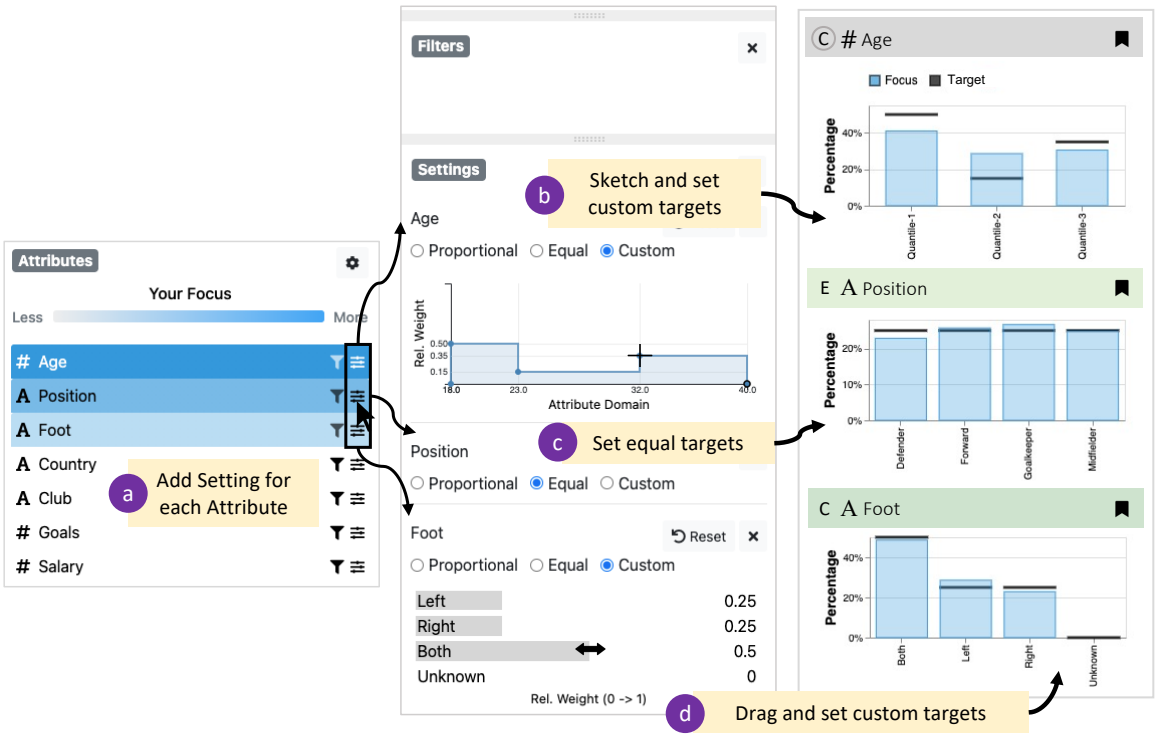


Figure 5.6: Lumos Example Scenario 3: Configuring Custom Baselines

5.2.4.3 *Scenario 3: Configuring custom baselines*

Viktor, a sports journalist, is using Lumos to analyze a dataset of European soccer players to write a news article (Figure 5.6) and they have a specific criterion to focus their analysis. They want to **equally** focus on player positions: {*Goalkeeper, Defender, Midfielder, Forward*} (and *not* **in proportion** to the underlying data distribution which may result in favoritism due to different proportions of player positions). In the Attribute Panel, they click on ⚌ next to *Position* (Figure 5.6a) and check the *Equal* radio button. The corresponding visualization in the Distribution Panel immediately updates with the black curves now all set at 100/4=25% (Figure 5.6c).

Next, they want to write a special section on the rising stars (younger players) and the old guard (older players). They check the custom radio button for *Age* and sketch a target distribution: {50% for ages under 23, 15% between 23 and 32, 35% above 32} by clicking (to add new quantiles) and dragging points in the visualization canvas (Figure 5.6b).

Finally, they want to focus on ambipedal players (comfortable shooting with either foot). They again check the Custom radio button and are presented with an interactive bar chart (*Foot* is an N attribute) and drag the bars with their mouse until their target distributions are reached: {50% Both, 25% Left, 25% Right, 0% Unknown} (Figure 5.6d).

Subsequent interactions will trigger the recomputation of the metrics but based on these updated target (baseline) distributions. Hence, Lumos helped Viktor specify different target distributions to compare their analytic behavior based on the task.

5.3 Evaluation 1: User Study to Understand How Interaction Traces in Lumos Increase Awareness of Analytic Behaviors

5.3.1 Participants and Procedure

We conducted a between-subjects qualitative study with the aim of understanding how Lumos helps users increase awareness of their analytic behaviors.

Participants. We recruited 24 participants who were either students, researchers, or industry professionals with a background in computing (e.g., computer science, human-computer interaction, human-centered computing) and a self-reported visualization literacy ≥ 3 (on a 5-point Likert scale). Participants were randomly divided into either a **Control** or **Awareness** condition, which determined the system version they used for the study. Participants in the Control condition *did not* see the Distribution Panel (along with the *ex situ* interaction traces) nor did they see the *in situ* interaction traces in the Visualization Canvas, Details View, and Attribute Panel. We set the target distribution to *Proportional* and hid the Settings Panel for both conditions.

Task. We tasked participants to:

> Analyze a tabular dataset of movies to recommend the characteristics of movies that a movie production company (e.g., Netflix [240]) should make next.

The dataset consisted of 709 movies (rows) and 9 attributes (columns): *Production Budget* (#), *Worldwide Gross* (#), *Running Time* (#), *IMDB Rating* (#), *Rotten Tomatoes Rating* (#), *Release Year* (), *Content Rating* (A), *Genre* (A), and *Creative Type* (A). Participants were encouraged to think aloud and their interactions were recorded.

Hypotheses.

H1 Interaction traces will increase awareness of analytic behavior.

H2 There will be differences in interactive behaviors of Awareness v. Control participants (as measured by bias metrics [111], differences in use of filters, and number of charts created).

H3 Participants will find the ex situ awareness features to have greater utility than in situ awareness features.

H4 Participants in the Awareness condition will react to interaction traces in ways to reduce potential biased analytic behaviors.

5.3.2 Results

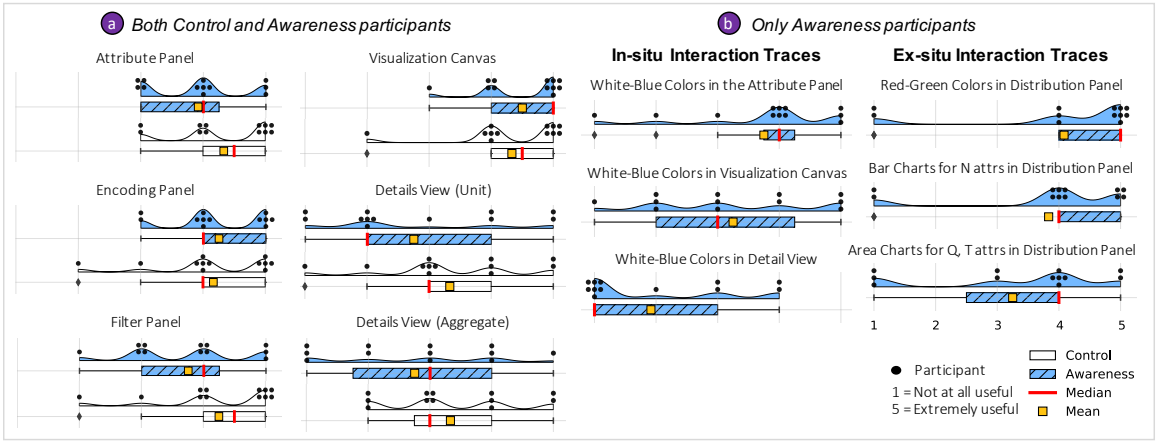


Figure 5.7: Summary of usefulness scores of all Lumos features as reported by participants in the post-study questionnaire, as RainCloudPlots [241].

Below, we present our study findings and discuss them in context of participant feedback. P_A01-P_A12 and P_C13-P_C24 refer to the 24 participants in Awareness and Control

conditions, respectively. Participant quotes and moments of awareness were coded and categorized using affinity diagramming. The lead author proposed an initial set of categories that were then iteratively refined with co-authors until a consensus was reached.

5.3.2.1 General Feedback.

Overall participant feedback was positive with P_A01 commenting, *"[they] haven't seen many things like [Lumos] before...really good technique."* P_A09 mentioned that *"[Lumos] can remove the internal bias of things users think are of the most interest."* Participants found *"the ability to keep track of [their] provenance, interaction history [as] interesting"* (P_A08) and *"communicating it back [how they are doing] as something [they] would use in [their] tools"* (P_A07). P_A09 found the Distribution panel *"a great idea to show users what their focus was"* and P_A05 found it *"very helpful as [they] don't need to create visualizations in the Vis panel for each attribute to see [their] distributions."* P_A05 suggested *"integrating this tool into existing tools such as Tableau [as] they don't have a feature that tells [them] what attributes haven't been used yet"* (P_A12). P_A07 suggested *"there are lots of use-cases for this technique in journalism and social media, e.g., you have only looked at Trump's negative tweets, but what about Biden's?"* Two participants found the Distribution Panel less useful as they *"didn't know exactly what to do about the [red-green] cards"* (P_A5) or felt it *"out of focus"* on the right side of the screen (P_A10).

5.3.2.2 Usefulness Scores: Lumos user interface.

Figure 5.7a summarizes Lumos's usefulness scores (*1=Not useful at all; 5=Extremely useful*) as reported in the post-study questionnaire:

Attribute Panel. Participants generally found the attribute list useful ($\geq$3 out of 5, median$_A$=4, median$_C$=4.5) along with *"their data types"* (P_C21) *"unlike, e.g., Excel where they aren't always on-screen"* (P_C17).

Encoding Panel. 23 participants found the Encoding Panel to be useful (median$_A$=4, median$_C$=4, *"it is standard in a good way"* - P_C21) except P_C23 who found it *"only slightly useful."* Four participants noted that the system messages to fix incorrect encodings were intuitive and helpful (*"it is sometimes hard to know what's wrong in Tableau"* - P_A05) but two found them confusing and suggested the app *"prevent [them] from choosing incorrect encodings"* (P_C18) by *"filtering out the chart types that are not allowed"* (P_A11). Five participants also suggested additional features (*"add color as a third encoding"* - P_C24) and enhancements (*"support drag-drop attributes to the Encoding Panel"* - $P_C\{17,18,19\}$, *"support text entry for the dropdowns"* - P_A12).

Filter Panel. Participants utilized filters (median$_A$=4, median$_C$=4.5) *"to remove outliers and to confirm hypotheses about the data"* (P_C22), to see the different values for a categorical attribute (P_C19, P_C23), and to mitigate any unconscious biased analytic behavior (e.g., *"Comedy and Drama are high percentage in the dataset, and I haven't interacted with them at all, so it might be worth my time to look at them."* - P_A09). P_A10 did not utilize filters as they were *"being more exploratory with [their] analysis and if [they] wanted to look at finer details, [they] would have used more filters."* Three participants also requested enhancements to *"specify precise inputs [for Q,T values]"* (P_C22), *"allow hover on a value in the categorical filter and highlight in the Visualization"* (P_A09), and *"support select- and deselect- all for categorical values"* (P_A11).

Visualization Canvas. 23 participants found the Visualization Canvas useful (median$_A$=5, median$_C$=4.5), utilizing it to *"observe patterns and outliers"* (P_A10), and *"see the different categories and values in the attributes"* (P_C14). P_C18 *"did not find it useful because a third attribute encoding, e.g., color was not supported"*.

Details View (Unit). This view received mixed usefulness scores from our participants (median$_A$=2, median$_C$=3). P_C17 *"liked the Details portion and being able to hover over*

points for more details." P_C19 and P_A11 *"didn't find the Details view for single data points super useful"* because they wanted the name of the film to bring prior knowledge to the analysis and spark different hypotheses[1].

Details View (Agg). This view also received mixed usefulness scores from our participants (median$_A$=3, median$_C$=3). P_C18 found it to be *"really useful because it is not apparent from the bar shape and size that some bars only have one point in them versus some bars having six or seven points."* P_A07 said *"[they] don't really hover on things in e.g., a scatterplot but information shown on hovering a bar chart [the details view agg] was awesome because you showed individual data"*. Also, *"it shows all information in one space."* (P_C20, P_C24) and *"could be useful for multivariate hypotheses"* (P_A10). Two participants found it *"hard to draw conclusions from lists of words and data"* (P_C17, P_C19) and preferred to see the information visually utilizing the Details View *"only as a reference"* (P_C19). P_C18 utilized the Details View *"because the visualization wasn't as helpful"*.

5.3.2.3 Usefulness Scores: Lumos technique of presenting interaction traces

Participants in the Awareness condition also saw *in situ* and *ex situ* (Distribution Panel) interaction traces in the user interface. Figure 5.7b summarizes the usefulness scores (*1=Not useful at all; 5=Extremely useful*) as reported in the post-study questionnaire:

Difference between analytic behavior and target distribution. Ten participants found the difference between their analytic behavior and the underlying data (red-green coloring in Distribution Panel) very useful (median$_A$=5) *"giving a sense if I'm looking at the dataset in an unbiased way" (P_A06)*.

Ex situ interaction traces in Distribution Panel. The overall feedback on the *ex situ* interaction traces in the Distribution Panel was positive; participants found the bar charts for

[1]Note: the movie title (*Title*) was deliberately not shown in the Detail View to prevent personal experiences with data to influence the analysis.

N attributes (median$_A$=4) more useful than the area charts for Q,T attributes (median$_A$=4) with P_A06 nicely summarizing, *"I can track and channel my focus based on discrete bar charts by applying a filter...but it is difficult to discretize and track a continuous [Q,T] attribute"*. For eight users, these real-time traces helped increase awareness (*"Geez, I haven't looked at Drama movies at all"*-P_A07), influencing them to interact differently (e.g., by creating a bar chart with *Genre* to inspect movies of other potentially underemphasized genres) while two participants either ignored them (*"I never looked at the individual distributions of attributes"*-P_A12) or preferred to look at them after analysis *"as the bars will be moving, and that's distracting"* (P_A09).

In situ interaction traces in Visualization Canvas. There was mixed response to the *in situ* interaction traces (white-blue colors) in the Visualization Canvas (median$_A$=3). For P_A06, they *"helped in tracking visited points"* nudging them to interact with uninteracted points, while for P_A05 they were confusing and distracting, nudging them to re-interact with them. P_A08 did not want the colors to stay persistent but *"be able to clear existing interactions and start a new session with a new set of model movies for comparison"*.

In situ interaction traces in Details View. Only four participants found the *in situ* interaction traces (white-blue colors) in the Details View to be useful (median$_A$=1).

In situ interaction traces in Attribute Panel. Participants generally found the *in situ* interaction traces (white-blue colors) in the Attribute Panel to be useful (median$_A$=4). They helped increase awareness of already-interacted attributes (*"I see that I have spent a lot of time on Release Year so I'll now see something else"*–P_A05) but also required some time to get acquainted with (*"the coloring in the Attributes panel...I did't use it initially, and later on it hit me that I had this feature. Once I noticed it, it was very useful"*–P_A*12*).

Summary. Comparing the distributions of scores for the aforementioned features (Figure 5.7b), participants found the *ex situ* interaction traces more useful than the *in situ* interaction traces, supporting **H3**, consistent with experimental results from [25]. We believe this is because *in situ* interaction traces are always visible to a user whereas *ex situ* interaction traces can be used more on-demand without side-tracking the analysis task at hand. Furthermore, *in situ* traces block an otherwise common attribute encoding channel, *color*, that can be undesirable for and cause inconvenience to some users.

5.3.2.4 Awareness Moments

In situ traces in the Attributes Panel. There were instances when Control participants expressed a need for tracking the already-interacted attributes. For example, we observed P_C13 use hand gestures to recollect and count the attributes that they had already visited and P_C14 exclaimed, *"I hope I have interacted with all (attributes)"*. Awareness participants, on the other hand, saw the interaction traces and had several instances of *awareness* during their respective analyses. Two participants *acknowledged* their choices (*"I don't think Release Year should matter too much, hence I am not interacting with it."*-P_A04, *"I don't think Running Time is important to me"*-P_A07) while two participants also suggested correcting future course via *interaction* (*"I see that I have spent a lot of time on Release Year so I'll now see something else"*-P_A05, *"[on seeing a white attribute bar] now I'm going to interact with Running Time"*-P_A22), also supporting **H4**. Two participants reflected upon their choices while answering questions pertaining to self-reported focus on individual attributes in the post-study questionnaire (*"actually I forgot, had I remembered, it might have been interesting to not click on the same thing over and over."*-P_A03, *"I didn't use the blue attributes panel but now that I see these questions, I would've seen them more"*-P_A08). These and the desire for awareness moments by Control participants validate **H1**.

In situ traces in the Visualization Canvas and Details View. Eight participants found the *in situ* interaction traces to be useful; two participants took some time to get acquainted with them (*"very useful but I learnt about them slightly afterwards"*-P_A01, *"I was initially confused but then over use I got used to them and found them useful in tracking visited points"*-P_A06). P_A03 found the colors to be useful but questioned the technique because *"if it is based on [me] hovering on a point again and again, it might not be 100% correct."* P_A05 was *"getting drawn to the visited points (instead of the white un-visited points)."* There was minimal commenting on the *in situ* traces in the Details View but it led to some awareness for P_A04 who hovered on an uninteracted (white) bar in a bar chart and observed *"there aren't many blue rows which means I haven't been focusing on it."*

Ex situ traces in the Distribution Panel. There were multiple instances of awareness among participants (supporting **H4**). P_A04 verified the interactions traces by comparing it with ground truth (*"distribution of my focus on Running Time (blue) is representative of the applied filter"*). P_A05 reflected upon seeing three red attributes in the Distribution Panel and hypothesized that they were *"just thinking aloud and exploring and will (now) follow a more targeted approach"*. P_A05 reflected upon seeing a red *Content Rating* attribute (*"Seems I didn't interact with R-rated movies enough so this view nudges me towards those"*) and applied a filter to show only R-rated movies. P_A07 reflected upon the white *Drama* category in the Distribution Panel and justified that they *"didn't care about Dramatization movies [...] who cares?"* At one point, P_A07 observed many red cards and exclaimed, *"this is so biased but whatever."* P_A11 tried to correlate the effects of their interactions with different attributes (*"I noticed that Adventure is representative of most values in Rotten Tomatoes Rating and IMDB Rating [...] This is because I mostly interacted with just Adventure movies and that caused those attributes [in the Distribution Panel] to be colored green"*). P_A10 did not use the Distribution Panel as they *"didn't know how to use it in the context of what [they were] doing"*.

5.3.3 Discussion

5.3.3.1 Using Color to visualize interaction traces

Fun, Focus, Distraction, or Inconvenience? Our participants had mixed opinions on seeing the changing colors in the visualizations (e.g., scatterplot points). Most participants found the changing colors fun and helpful as they helped them be more aware of their (bad) behavior, triggering a shift in their emphasis towards correction / mitigation. However, many participants also found them to cause inconvenience or be a distraction motivating the need to study alternate encoding approaches. One participant was drawn to the already visited (colored) points instead of the unvisited points. Another participant confused the blue colors with an attribute encoding; this is a disadvantage of encoding interaction traces to the color channel. Another participant was cautious of their interactions so as not to skew their interaction trace (interactions). Another participant questioned using the mouseover (hover) interaction as a proxy for focus and altogether ignored the resultant coloring, suggesting using proximity and not an exact hover as the metric to determine focus. Another participant noted Lumos to be color-blind safe from an accessibility standpoint.

While in this evaluation of Lumos, we studied the usage of color (shades of blue, red, and green) to encode interaction traces, there are other visual variables that can be modified to encode and convey the same information, e.g., *stroke color*, *stroke width*, *size*, *shape*, *orientation*, *opacity*, etc. It must be noted that some visual variables might work better for certain visualizations than others (e.g., modifying color works better for a scatterplot than a strip plot) and that some variables may not even be applicable for certain visualizations (e.g., it is difficult to modify the shape of a line chart). Exploring this design space will help derive guidelines for effective in situ and ex situ visualizations.

5.3.3.2 *The role of target distributions*

Target distributions in Lumos serve as a benchmark against which analytic behavior can be compared. For some tasks, meaningful target distributions may exist (e.g., forming committees with specific representation from certain groups). However, it may be harder to articulate a target distribution for other decision-making tasks, in which case standard baselines for comparison are more meaningful. Lumos allows users to modify these or use the data distribution as a default.

5.3.3.3 *False positives in modeling analytic behavior*

The Lumos technique of presenting interaction traces can be subject to false positives. For example, users might intentionally not interact with an attribute because it is either not important or they are not interested in it. Labeling these as underemphasized may not be correct, as it was a conscious decision by users to ignore them. Furthermore, a categorical attribute, when encoded along one of the scatterplot axes, can lead to the formation of visual clusters that offload a cognitive task to a perceptual one, rendering that attribute's filter somewhat redundant. On the other hand, users can also unintentionally neglect aspects of data, e.g., the Attribute Panel may not be able to fit all attributes of a dataset, causing the attributes that are outside the viewport to be potentially neglected during analysis. Lumos helps the user tackle both: by showing unintentionally uninteracted attributes and by allowing users to intentionally set custom target distributions for attributes. As a potential feature, Lumos can explicitly present an attribute-level flag that allows the user to tag the attribute as important during analysis or not.

5.3.3.4 *Toward additional mitigation strategies*

Based on Lumos results, interaction traces help increase the user's awareness of their analysis practices, sometimes influencing them to interact and mitigate unconscious biased analytic behaviors. We believe this is a *passive* mitigation strategy since the user has to inspect

the difference between their analytic behavior and the target distribution and devise an appropriate strategy, e.g., by applying a filter. Some participants suggested we implement a more *active* mitigation feature with *"a button to automatically apply a reverse filter [instead of them having to manually apply it]"*, *"especially for continuous attributes"*. For example, one participant saw their interactions with different Genres (Concert, Documentary, and Western) and reflected *"[they] should now interact with Drama since that is maximum and these are almost nil"*. They applied a filter to correct their unintended underemphasis but after a few interactions found themselves overemphasizing *towards* Drama movies and reversed the filter. This act of balancing focus across all attributes can lead to frustration, sidelining the analysis task at hand. This was our motivation to build mixed-initiative systems that more actively assist the user in mitigating biased analytic behaviors, e.g., explicitly recommending addition (or removal) of a set of filters that negate the overemphasis (or underemphasis) or automatically taking action and interacting on behalf of the user.

5.4 Evaluation 2: Left, Right, and Gender – User Study to Understand How Interaction Traces Can Mitigate Human Biases

In addition to evaluating Lumos – the system, we conducted a crowd-sourced experiment to understand how interaction traces help mitigate human biases during decision-making. Our users performed two tasks in the domains of (1) politics and (2) movies. In the political scenario, we curated a dataset of fictitious politicians in the U.S. state of Georgia and asked participants to select a committee of 10 responsible for reviewing public opinion about the recently passed Georgia House Bill 481 (Georgia HB481), banning abortion in the state after 6 weeks. In this scenario, several types of bias may have impacted analysis, including gender bias (i.e., bias favoring one gender over another), political party bias (i.e., voting along political party lines, regardless of potential ideological alignment from candidates in another party), age bias (i.e., preferential treatment of candidates based on age), and so on. Participants in the experiment also completed a parallel task in the domain of movies: to

select 10 representative movies from a dataset of similar size and composition. Note that, regardless of domain, our goal was not to address overt biases (e.g., in the form of discrimination); rather, we believe visualization *can* have an impact on increasing user awareness of potential unconscious biases that may impact decision-making in critical ways.

Thus, in this task, we anticipated that participants' decisions would be driven by idiosyncrasies of their individual preferences. For the given tasks, we utilized a *lite* version of Lumos wherein we only allowed users to work with a scatterplot (no bar chart, strip plot, and line chart), and dropped the ability to configure the target behavior (we fixed it to proportional). Additionally, to support decision-making, we enabled users to click datapoints to *select* them and add to their lists. Furthermore, we assessed four interface variations: CTRL, SUM, RT, and RT+SUM. The CTRL interface served as the control system, which we compared to variations that provided either *real-time* (RT) or *summative* (SUM) views of the user's interaction traces (or both, RT+SUM).

Our experiments yielded mixed results, offering support that interaction traces, particularly in a summative format, can lead to behavioral changes or increased awareness, but not substantial changes to final decisions. Interestingly, we find that increased awareness of unconscious biases may lead to amplification of individuals' conscious, intentional biases. For details, I refer the reader to the associated publication [25].

5.5 Limitations and Future Work

Lumos currently supports only a small set of visualization types; however, we chose them to test across different aggregation types. Also, analytic behavior is modeled only from interactions, which may not be a complete proxy for attention; in the future, one may consider user gaze or other sources to more accurately approximate it. Lastly, Lumos models analytic behavior by equally weighting all interactions; given users' declining attention span over time, we may incorporate interaction recency in the future.

Finally, interactions with aggregate visualizations (e.g., hovering on a bar in a bar chart)

are currently considered as N equally weighted interactions of magnitude 1/N where N = number of data points belonging to that element. This has variable impact on the metrics due to different statistical tests used to compute the analytic behavior model (AD [111]) depending on the attribute type (e.g., χ^2 test for categorical attributes, Kolmogorov-Smirnov test for numerical distributions). Future work can explore alternative computations for analytic behavior models that may reflect a user's attention and intentions more precisely.

5.6 Summary

In this chapter, I described a mixed-initiative system, Lumos, that facilitates a co-adaptive guidance dialog between the user and the system. Lumos helps increase awareness of exploration biases during analysis, but additionally allowing users to customize the target interaction behavior and receive contextual guidance. A user evaluation of Lumos revealed that interaction traces can enhance user awareness of analysis behaviors in real-time, fostering self-reflection and acknowledgment of users' intentions. A second user evaluation studied how interaction traces help mitigate human biases (e.g., age bias, gender bias) during decision-making, (also) suggesting they can help promote conscious reflection on decision-making strategies, but further studies were needed for conclusive results. These results can have far-reaching implications, such as helping to mitigate biased decision-making, supporting diversity goals (e.g., in hiring), and promoting transparency in analysis processes. Lumos is available as open-source software at **https://lumos-vis.github.io**. For details, I refer the reader to the associated publications [27, 25].

CHAPTER 6

DESIGNING A MIXED-INITIATIVE, MULTIMODAL GUIDANCE SYSTEM

In this chapter, I describe a mixed-initiative system, BiasBuzz [7], that provides multimodal guidance (combination of visual and haptic) to increase awareness of biased analytic behaviors during visual data analysis, partly achieving **RG2**: *Design a mixed-initiative guidance system, wherein the user and the system learn from and take initiative on behalf of each other, co-adaptively steering the analytic process*. This chapter is based on work published at ACM CHI (Late Breaking Work) [7].

6.1 Motivation and Background

In chapter 5 (Lumos), I described how visually presenting interaction traces during analysis (e.g., coloring visited data points darker than others) increased user awareness of their analytic behaviors, yet sometimes led to confusion or went unnoticed. Essentially, we believe presenting visual cues alone may be a passive form of guidance that also adds to users' cognitive load already engaged in *visual* data analysis. We hypothesized stronger cues are needed in the user interface to promptly capture user attention to 'bring them back on track' without significantly increasing cognitive load or hindering user experience. So we asked: *"How can we use alternate modalities as a stimulus to the existing visual feedback, to strengthen and reinforce the overall guidance during analysis?"*

In response, we reviewed alternative modalities such as natural language [26] and ambient display media (light, airflow, sound) [242] and selected haptic[243] as our additional cue. Haptics refers to the science and technology involving the sense of touch, particularly focusing on the creation and study of tactile sensations and feedback [244, 243]. Haptic devices have been used in many applications such as remote systems for visually impaired people [245], anxiety and depression treatment [246], assistive communication

technologies for children with autism [247], wrist-mounted devices for alerting users of warnings in a cybersecurity context [248], affecting the state of mind of users watching the news [249], and gaming [250, 251]. Akamatsu et al. [250] showed that with a haptic mouse, users move faster and click targets within a wider area than users with a typical mouse. Kyung et al. [252] studied how a unique mouse with "force feedback" was more effective than a normal mouse at helping users recognize shapes in a task. Terry et al. [253] found that haptic mice reduce the response time spent on visual tasks on a computer. Han et al. [254] used an off-the-shelf haptic mouse for a guidance study using visualization and participants who used the haptic features performed better than users who did not.

For this work, we explored how a combination of visual guidance and haptic feedback can help users be *even more* aware of their analytic behavior during a visual data analysis task. In particular, we built BiasBuzz, as described next.

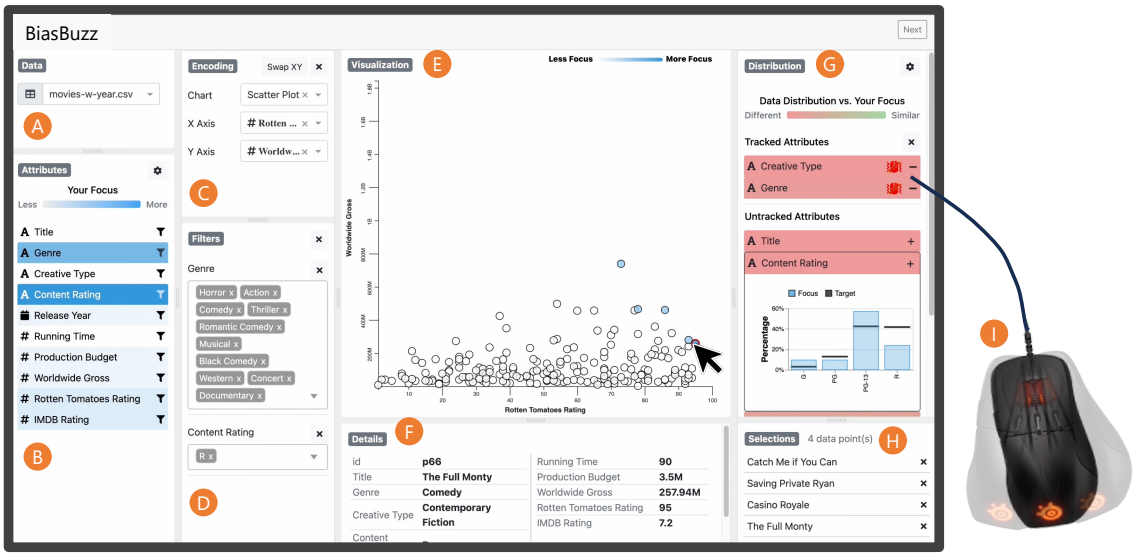


Figure 6.1: An existing visual data analysis tool, Lumos [27] (A)-(H), enhanced by wiring it to a gaming mouse [255] (I) to increase awareness of exploration biases. This new enhanced system (BiasBuzz) provisions visual guidance by highlighting a user's prior interactions (blue) and deviations from expected behavior (red, green) along with haptic feedback from a gaming mouse when there is significant deviation.

6.2 BiasBuzz

BiasBuzz is an extension of an existing visual data analysis system, Lumos [27], by interfacing it with a mouse capable of generating haptic feedback. This enhanced system tracks user interactions with data, measures exploration biases, and communicates them to the user in the form of mouse vibrations (haptic feedback) and simultaneous display of contextual information in the user interface (visual guidance). This combination of visual and haptic elements seeks to create a more engaging experience for users during data analysis.

6.2.1 Haptic Feedback: Design Choices and Considerations

To design a visual data analysis system integrating visual and haptic feedback, we identified key considerations and design choices, illustrated through a scenario. Consider a visual data analysis tool (e.g., Lumos [27]) where users upload a tabular dataset and perform analysis by creating different visualizations and applying relevant filters. To help the user not exhibit exploration bias, i.e., emphasize certain attributes and records more than others, the system tracks the user's interactions and visually presents any bias back to the user, in real-time. This tool achieves this by (1) highlighting already visited data attributes and records and (2) presenting the deviation of user's interaction pattern from the underlying distribution, computed as the AD metric by Wall et al. [111].

Recall from chapter 5 (Lumos), that the AD metric characterizes how a user's interactive behavior deviates from expected behavior and ranges from 0 (no deviation, no bias) to 1 (high deviation, high bias). By default, the system chooses a **proportional** baseline of expected behavior, wherein interactions with any given datapoint are equally likely and also reflecting the true underlying distributions of data attributes. For instance, if a user interacts primarily with 'Drama' movies among a dataset of movies that contains predominantly 'Action' movies, the AD metric for the *Genre* attribute will be high (more emphasis). If the user instead spent more time interacting with 'Drama' movies, proportional to the distribu-

tions in the dataset, the AD metric value for *Drama* would be low (less emphasis).

Now, we intend for the feedback in the tool to "guide" the user to exhibit less exploration biases in their interactions than they did prior to the feedback. This means discouraging "biased" exploration methods and reinforcing "unbiased" exploration methods without highlighting specific data points in the interface. Consider this tool is connected to a haptic-enabled gaming mouse, such as the SteelSeries 710 [255], that appropriately vibrates from time to time to capture the user's attention. There are several considerations in designing the timing, duration, intensity, and pattern of the resultant vibrations, as described next.

Vibration Timing: When to vibrate? We considered triggering a mouse vibration every time exploration bias is detected for an attribute. The AD metric [111], used to quantify the deviation of user's interaction pattern of an attribute from its underlying distribution ranges from [0, 1], where 0 implies less deviation and 1 implies more deviation. Based on our own testing and pilot studies, we set a threshold of AD=0.7 above which an attribute is considered to be interacted with bias. This can still result in multiple vibrations, depending on how many attributes have AD values above the threshold. Thus, we decided to allow the user to select the attributes they wish to track and only consider this subset for vibration.

Vibration Duration and Cooldown Period: When should the vibration end? Gaming mice have vibration motors built into them to provide haptic feedback during gameplay. These motors often generate heat when they are in use for extended periods or at high intensity. To prevent these motors from overheating, as a protective mechanism, these mice *cooldown* for a short time period before vibrating again. One of the implications of this behavior in our visual data analysis scenario is that if the user interacts twice in quick succession, and both times bias is detected, the mouse would still only vibrate once. Only after the cooldown period, if the detected bias is still active, will the mouse vibrate again. Between this timeframe, the vibrations can be considered 'lost', necessitating an alternative modality (e.g., visual) to communicate the same information.

Vibration Intensity: How strongly to vibrate? Gaming mice often enable customization of the vibration intensity (or strength) and pattern during gameplay. In our visual data analysis scenario, we can map intensity to the amount of exploration bias (e.g., less bias is `'z'` whereas more bias is `'Z'`), where `'z'` and `'Z'` represent one vibration pulse. During our own testing and pilot studies, we noticed less variance between different vibration intensities, making it difficult for users to differentiate between them. Thus, we decided to set the default vibration intensity at the highest level (`'Z'`).

Vibration Pattern: How to vibrate? Gaming mice often enable customization of the vibration pattern. To design our visual data analysis scenario, we reviewed the design space of haptics [256, 257, 258] and considered mapping different vibration patterns to different attributes (that are exhibiting bias). For example, given a dataset about *movies*, `'Z'` represents one vibration pulse. A biased *Genre* attribute would make the mouse vibrate as `'ZZ..ZZ..ZZ'`, wherein the mouse vibrates for a short duration two times every time bias is detected in the *Genre* attribute. Also in this example, a biased *Budget* attribute may vibrate as `'ZZZ.ZZZ.ZZZ'`, wherein the mouse vibrates three times every time bias is detected in the *Budget* attribute.

In summary, our own testing and pilot studies, we found that keeping track of different vibration patterns can become confusing for the user. Thus, we set the default vibration pattern to a single, long pulse set to the highest strength and show additional contextual information, e.g., the attribute(s) name and its AD value, visually in the user interface (UI). Note that a gaming mouse generally does not have its own display to show this information, hence we have chosen to utilize the tool's UI.

6.2.2 User Interface

We enhanced an existing, open-source visual data analysis tool, **Lumos** [27] (Figure 6.1). Lumos enables users to load a tabular dataset **(A)**, inspect its attributes and corresponding

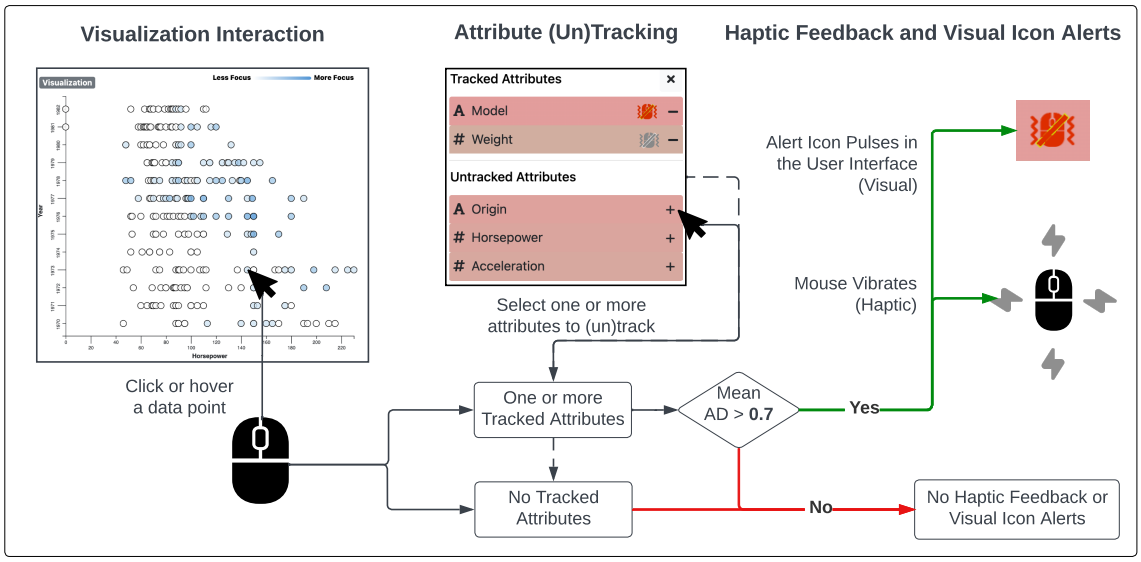


Figure 6.2: The interaction sequence diagram to trigger haptic feedback and visual icon alerts in BiasBuzz. When a user interacts with a datapoint, and tracks one or more attributes (for bias mitigation), and if the mean AD metric value for these tracked attributes is greater than a predetermined threshold of 0.7, the mouse vibrates and the corresponding visual alert icons pulse in the UI. In all other scenarios, there is no haptic feedback or visual guidance.

data distributions **(B)**, apply filter criteria **(D)**, and assign attributes to visual encodings **(C)** to eventually create visualizations **(E)** and inspect raw data records **(F)**. Lumos tracks users' interactions with data attributes and records and presents them back to the user in the form of visual highlights (e.g., by coloring visited data points in shades of blue **(E)**). Lumos also determines if the user has over- or underemphasized certain attributes and records and by how much by computing the AD (Attribute Distribution) metric [111] **(G)**. The AD metric values lie between 0 and 1; higher the AD metric, higher the deviation between the user's interaction with a certain category/quantile of data attribute and its underlying data distribution, implying higher exploration bias. Lastly, the Selections panel **(H)** shows the list of selected data records (movies).

We used a **SteelSeries 710 gaming mouse** that can be made to vibrate programmatically [255] **(I)**. We chose this specific model because of its diverse vibration-related capabilities (e.g., timing, duration, intensity, pattern), ease of setup via an extensive API and documentation [255], and prior usage in a research study related to visualization [254]. We

made the following enhancements to the Lumos UI to orchestrate the interactions with the haptic mouse, which is illustrated in Figure 6.2.

(Un)Tracking Attributes. Not all attributes from a dataset might be important or relevant to the user's task (e.g., the *Age* attribute is irrelevant if the user's task is to ensure *Gender* diversity). Thus, we added the ability to "track" specific attributes, and only communicated AD bias for these "tracked" attributes. Lumos already supports the ability to bookmark one or more attributes, and we repurposed this to instead track one or more attributes **(G)**. When a user tracks one attribute, that attribute's AD metric value is compared to a preconfigured *high-bias threshold*=0.7, on a scale from 0 to 1. If the value is greater than the threshold, exploration bias is detected and reported. When a user tracks multiple attributes, the *mean* of the AD metric values of the tracked attributes is computed and compared with the threshold (0.7). If the value is greater than the threshold, exploration bias is reported.

Haptic Mouse Vibrations and Visual Icon Alerts. To report exploration bias(es) for the tracked attribute(s), we provided two modes: haptic mouse vibrations and visual icon alerts. Whenever exploration bias is detected, the mouse vibrates once for a split second. Note that our haptic mouse does not come with any kind of display; it just vibrates and lights up. Hence, it can only convey *when* there is bias but not *why or who is responsible for it*. Transmitting this information via Morse (or equivalent) code is out of scope for this study. Thus, to put the vibration into context, it is very important for the Lumos visual interface to show the corresponding attribute(s) and the AD metric values. To achieve this, we added visual alert icons next to each tracked attribute in the Distribution panel **(G)**.

Whenever the mouse vibrates, corresponding visual alert icons start flashing in a pulse animation (i.e., continuously increase and decrease in size), connecting the vibration to the corresponding attribute. When a user tracks multiple attributes and the mouse vibrates (i.e., when the mean AD metric value is greater than the threshold), the mouse also vibrates but the visual alert icon starts flashing only for those attributes whose individual AD met-

ric value is greater than the threshold (i.e., who are, in a way, responsible for the overall exploration bias). This was a design choice to help the user formulate concrete *next step* interactions with specific attributes (e.g., the ones with the highest bias).

(Un)Muting Attributes. We anticipated users wanting to stop experiencing the mouse vibrations either temporarily or permanently because of a personal preference or feeling of distraction. Thus, we enabled users to toggle the tracking of an attribute's AD metric and associated vibrations by clicking its corresponding visual alert icon.

6.3 Evaluation

We conducted a formative study to understand how visual and haptic feedback can together help increase user awareness of analytic behaviors during visual data analysis.

6.3.1 Participants and Procedure

Participants. We recruited nine participants enrolled in a bachelors degree program in a computing or related field at a public university in the United States. We screened these participants based on their self-reported visualization literacy ($\geq$3/5). Demographically, our participants comprised seven men and two women in the age range of 21 to 32 years.

Dataset. 709 movies with 9 attributes: *Production Budget* (#), *Worldwide Gross* (#), *Running Time* (#), *IMDB Rating* (#), *Rotten Tomatoes Rating* (#), *Release Year* (📅), *Content Rating* (A), *Genre* (A), and *Creative Type* (A).

Task. We tasked participants to:

> Create a list of 10 movies that you would like to watch. These movies should reflect the underlying dataset as it relates to Release Year, Genre, and Content rating. Feel free to use the tracking feature to help you achieve your goal.

Study Session. We conducted the study in-person in a controlled lab environment. After providing consent, participants saw a video tutorial that demonstrated the features of the visual data analysis tool and the gaming mouse (5 minutes). Participants then performed a practice task on a dataset of cars to get acquainted with the study interfaces (5 minutes) before starting the actual task on the dataset of movies (20 minutes). After the task, participants provided feedback via a post-study questionnaire and a short debriefing interview (5 minutes). Each study session lasted about 60 minutes for which we compensated each participant with a $15 gift card. We encouraged participants to think aloud during this task recorded the screen and audio for subsequent qualitative analysis.

6.3.2 Results

6.3.2.1 Qualitative Feedback

For qualitative analysis, We transcribed participant audio recordings, divided the resultant transcripts into smaller sections, and two coders applied open coding [185].

The overall feedback was mixed. In the post-study questionnaire, participants scored their perceived utility of key aspects of the study on a Likert scale from 1 ("not useful at all") to 5 ("very useful"). All aspects including visual icon alerts ($\mu = 2.56$), haptic mouse vibrations ($\mu = 3.33$), the ability to track attributes ($\mu = 3.56$), and the ability to mute attributes ($\mu = 2.67$) received mixed scores. Findings from the qualitative analysis also resonated with the aforementioned sentiment, as described next.

P1-P9 refer to the nine participants. P1, P5, P9 were positive about both the haptic mouse and the visual icon alerts. P5 said, *"the mouse vibrations and visual alerts are [both] very good at drawing your attention towards data points you've been missing out on."* P1 said, *"I think [the vibrations] reminded me of my goal, so they changed my attention to focus on the tracked attributes."* P1 also noted they *"didn't notice the visual attribute alerts as much compared to the haptic feedback, but [they] like that it shows red when [they] haven't looked at data proportionate to that attribute."*

On the contrary, P2, P3, P6 disliked both. P2 said, *"I barely spent any time [with the Distribution Panel] near the beginning of the task and I already feel punished [due to high AD values]."* They projected *"[they] might get immune to it eventually and discard it as a nuisance rather than something that's giving helpful information."* P2 said, *"I'd prefer a post-facto email with suggestions rather than instant haptic punishments."* P6 did not understand the mouse vibrations or visual icon alerts very well. According to them, *"There was [high] latency between the event and the vibration so [they] had a hard time linking the vibration to its meaning."* Because *"[they] did not figure out how the mouse vibration worked [they] did not understand the [visual] icon [alerts] either."* P3 were more hopeful, suggesting *"the [haptic and visual alerts for attributes] would be more useful if they were more relevant to the way I was looking through the dataset [instead of comparing with the underlying data distribution as the baseline],"* suggesting alternate baselines to be employed [27]. These sentiments indicate a strong rejection of the mouse's vibrations.

P8 did not like the mouse vibrations but liked the visual icon alerts. They said, *"[the visual icon alerts] affected my data exploration because it made me want to avoid that datapoint that it vibrated on."* P4 and P7 liked the mouse vibrations but not the visual icon alerts. On the mouse vibrations, P4 said, *"There was one time [the vibrations] went off, and I was like 'ok we need to look at action thriller' and another time I was like 'hey you need to get a drama'."* P7 said, *"When I felt the vibration, I switched my focus to the [Distribution] panel and get some additional information."* On the visual icon alerts, P4 said, *"I feel like because the window was really small I had to scroll to find exactly what attribute was setting it off."* This issue can potentially be mitigated with interface enhancements. P7 said, *"I think the problem I had is that I am not familiar with the [visual icon alert] meaning. I thought the red icon indicates I am making some errors or mistakes, so I am thinking to 'fix' it."* All of these observations demonstrate the wide range of reactions to our enhancements.

6.3.2.2 Quantitative Analysis

Tracked Attributes. Participants tracked attributes for a total of 40 times (μ=4.44, η=4, σ=1.07, max=7, min=3). *Genre* was tracked the most (12 times) and *IMDB Rating*, *Worldwide Gross*, and *Running Time* were tracked the least (once). Participants scored (μ=3.55) the ability to track attributes relatively higher than other features like visual icon alerts, haptic mouse vibrations, and the ability to mute attributes. While P2 said, *"[the tracking] is a necessary piece for the whole design,"* P5 said, *"[the tracking] helped me notice the parts where my focus was deviating from expectation."*

Mouse Vibrations. The mouse vibrated a total of 142 times across all nine participants (μ=15.77, σ=8.72, max=36, min=3). Of these, *Genre* was above the exploration bias threshold ($= 0.7$) the most and vibrated 114 times. *Worldwide Gross*, *Rotten Tomatoes Rating*, *Running Time*, and *IMDB Rating* were all tracked by participants at one point or another, but none of these attributes were above the bias threshold to trigger vibrations. Many participants had interesting things to say about the mouse vibrations. P1 said, *"The only useful part about the haptic feedback is that it reminded me I hadn't reached my goal of selecting and viewing a proportionate amount of different movies with respect to the specific attributes I was tracking."* P4 *"could tell towards the end that the vibration is indicating that you need to work on something."*

Mouse Muteness. Participants muted the vibrations for a total of 56 times (μ=15.77, σ=9.40, max=32, min=0). *Genre* was muted the most (41 times). P8 said *"There are some attributes that I was not considering, so it was great to be able to mute these specific attributes."* P1 muted an attribute only once, and they did this because *"the haptic feedback wasn't too distracting, so I didn't see the need to mute the alert for specific attributes.*

Exploration Bias Mitigation: Did the AD metric values decrease? We plotted the total number of vibrations for each of the three attributes (*Genre*, *Content Rating*, *Release Year*)

against their corresponding final AD values. We observed no correlation suggesting that the vibrations did not reduce the AD values. P3, even though they experienced the most number of vibrations (n=36), said they did not feel they needed the vibrations to do well in the task. They said, *"[Vibration] definitely has a place for some tasks, but I didn't need it all that much for this one."* Similarly, P9 experienced the least number of vibrations (n=3) and found them less useful, noting, *"[Vibrations] didn't affect my data exploration process because I was focused on the task of creating a list of 10 movies more than anything."*

Temporal Analysis: Did the vibrations nudge users to respond by interacting differently? Even though there was no overall decrease in AD values, there were instances when participants actually changed their subsequent interaction behavior after the mouse vibrated, either temporarily or permanently. For instance, P4 experienced several vibrations towards the end of their session and the AD value of *Content Rating* dropped in tandem with those vibrations. They said, *"[Vibrations] helped me kinda narrow down genres toward the distribution."* Although the AD value of *Genre* did not drop significantly, their comment suggested that users can be made more aware and reflect on their choices during the task. Notably, P2 experienced several vibrations due to *Content Rating*, but ended-up muting it, because of which its AD value remained high throughout the task.

6.3.3 Discussion

Haptic vibrations can take some time to get used to. Unlike games, data analysis systems generally utilize a single 'visual' modality. Thus, it will take time for other modalities to gain acceptance. For instance, P4 said they *"didn't notice [the vibrations] at first, but after some repetition, got used to looking at the distributions after feeling the vibrations."* P9 *"wondered if the mouse vibrating was a technical issue."* P8 *"found [the mouse vibrations] were very clear but [were] just not sure why the mouse was vibrating."* P7 even said, *"If you asked me to do it again (the task with the mouse), I could get more used to it."*

Haptic vibrations can be a positive stimulant to aid analysis. Many participants were positively stimulated in one way or another directly after the mouse vibrated, lending credibility to the practicality of offering haptic modality as a more "aggressive", "active" form of guidance in visual data analysis. For example, the vibrations acted as a reminder of the analysis goal (P1), realization of missed out data points (P5) and attributes (P7), all of which nudged them to change subsequent interaction strategy.

Haptic vibrations can also be a source of distraction during analysis. While the mouse achieved the desired effect of increased analytic awareness for some participants, there were multiple instances where it negatively affected the participant's analytic goals, which is undesirable. P7 said, *"It encouraged me to explore different movie attributes instead of the ones I am interested in."* P8 said, *"It affected my data exploration because it made me want to avoid that datapoint that it vibrated on. Am I supposed to avoid this datapoint?"*

6.4 Limitations and Future Work

The capabilities of the SteelSeries 710 mouse limited this study. As a common off-the-shelf mouse, its vibration intensity was not very high and due to its cooldown requirement, it could not vibrate for long time periods. As a result, even though this mouse supported different vibration patterns, we could not exploit this to different aspects of the user interface (e.g., unique pattern per attribute or a certain level of bias). Studying these via a custom-built mouse that is capable of stronger vibrations, more vibrations in rapid succession, and different types of vibrations, is future work. Furthermore, exploring additional modalities such as natural language [26], ambient display media (light, airflow, sound) [242], or squeeze-haptics [259] to communicate appropriate guidance is also future work.

6.5 Summary

In this chapter, I described a mixed-initiative system that provides multimodal guidance to increase user awareness of biased analytic behaviors during visual data analysis. In particular, we wired a gaming mouse to an existing visual data analysis tool (Lumos, described in chapter 5); we made this mouse vibrate and reinforce the tool's existing ability to detect and visually communicate exploration biases exhibited by the user. A formative study with nine users revealed that the dual guidance modality of visual and haptic feedback can sometimes increase user awareness of biased analytic behaviors, but the mouse vibrations can also be distracting and disturbing, putting into context the design of future multimodal guidance-enriched user interfaces for visual data analysis. For details, I refer the reader to the associated publication [7].

CHAPTER 7
DESIGN SPACE FOR COMMUNICATING ANALYTIC PROVENANCE

In this chapter, I describe a design space for communicating analytic provenance, by utilizing provenance as an attribute during analysis, mapping it to visual encodings and data transformations. Because provenance information is often the basis for provisioning guidance, this design space sets the foundation for the guidance design space (described in chapter 8), partly achieving **RG3**: *Establish a design space for guidance communication during analysis*. This chapter is based on work currently under review [11] and also sets the foundation for the design space for guidance communication, described in chapter 8.

7.1 Motivation and Background

Analytic provenance records the history of analytical actions, showing how data was obtained, transformed, and analyzed. For data visualization, analytic provenance also tracks how users interact with visualizations as a representation of their reasoning process [52]. A frequent use of provenance is to help users recall steps taken during analysis [53]. While effective for forensic purposes, other tools have explored how to show provenance to users during analysis. For example, existing systems [27, 84, 85] leave visual traces of the user's interactions to encourage them to pause and reflect on their behavior, potentially influencing subsequent analysis. However, analytic questions such as *"How many data points has the user interacted with so far?"* or *"Which were the first attributes that the user interacted with?"* are often only answerable post-analysis (after analyzing the provenance logs). In the moment, it is impractical for the user to manually count the interaction traces one-by-one or outright remember their interaction history in detail, which makes such iterative reflection during analysis difficult [71, 72]. Thus, our main research question asks: *"How can we make analytic provenance available to the user during visual data analysis?"* In re-

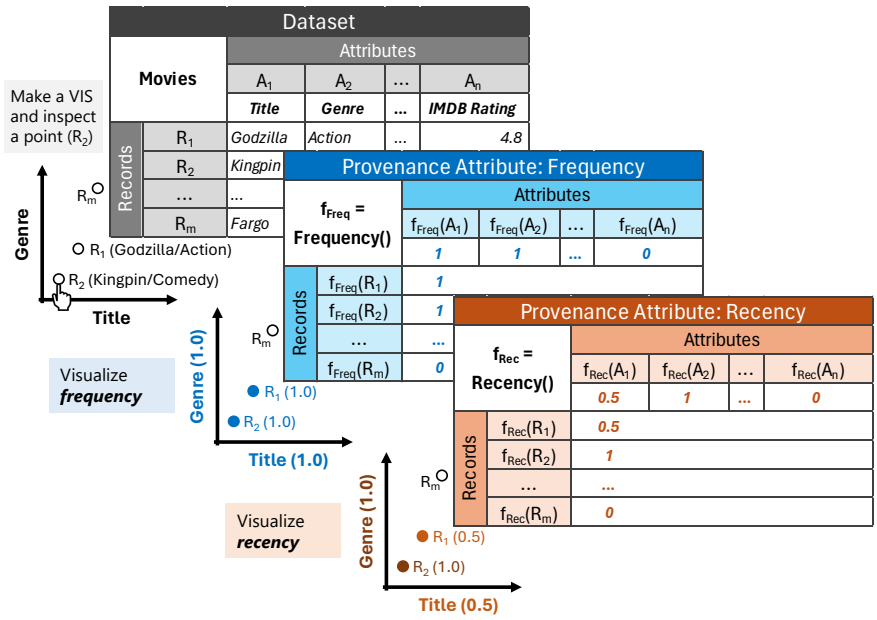


Figure 7.1: Illustration of two provenance attributes, frequency and recency, modeled for each dataset attribute (A_1–A_n) and record (R_1–R_m), on a 0 (low) to 1 (high) range. Consider a user creates a scatterplot visualization of **Title** (A_1) × **Genre** (A_2) and then clicks two datapoints one after another $R_1 \rightarrow R_2$, indicating interactions with two attributes and two records. Regarding data attributes, **Title** (A_1) and **Genre** (A_2) both receive a *frequency* score of 1.0 (each interacted once, hence maximum score), while other attributes score 0.0; for *recency*, **Genre** (A_2) (most recently interacted) scores 1.0 and **Title** (A_2) scores 0.5, while other attributes score 0.0. Likewise, regarding data records, R_1 and R_2 both score 1.0 on *frequency*; for *recency*, R_2 (most recently interacted) scores 1.0 and R_2 scores 0.5, while other records score 0.0. These scores are derived by evenly spacing the interactions between 0 and 1, based on their count and order of occurrence in the interaction history.

sponse, we investigate two key components: (1) how to model provenance during analysis and (2) how to present provenance back to the user.

7.2 Design Space: Utilizing Provenance as an Attribute

In this section, we describe how we (a) track user interactions with data attributes and records to (b) model provenance attributes and later (c) visualize them during analysis.

7.2.1 Tracking Provenance: Which User Interactions to Log?

In a typical visual data analysis system, users analyze their dataset in various ways: they can inspect an attribute's summary statistics (e.g., via distribution plots), examine individual data records (e.g., from a data table), apply data transformations (e.g., filter and sort), and create visualizations (e.g., by mapping attributes to visual encodings). To achieve our overarching goal of tracking, modeling, and visualizing provenance during analysis, we track a subset of relevant user interactions and map them to an individual data attribute or record. In this work, we track interactions with a data *attribute* when a user inspects its summary profile, maps it to a visual encoding, or uses it to filter and sort data records; we track interactions with a data *record* when a user hovers on a visualization mark (e.g., a point in a scatterplot) or a row in the data table.

7.2.2 Modeling Provenance Attributes: Frequency, Recency

After tracking user interactions, accurately modeling provenance attributes is crucial for understanding analytic behaviors. To model provenance, we generally follow the methodology employed by Lumos [27]. For unit visualizations, we map one interaction with an attribute or record as $+1$ unit of interaction. For aggregate visualizations that show a single value computed from multiple data records (e.g., a bar in a bar chart with an aggregation function such as *mean* or *sum*), we map one interaction with an aggregated entity as $+1/N$ units of interaction for each of the N data records that form the hovered entity. For example, consider a bar chart showing average *IMDB Rating* for different movie *Genre*s; if the user hovers on the bar *Genre*="Action" that represents five action movies (whose mean is encoded as the bar's height), we log each action movie as having $+0.2$ units of interaction.

We also log the timestamp (as milliseconds since epoch) of each interaction. For unit visualizations, we simply map the interaction timestamp to the corresponding attribute or record. For aggregate visualizations, we log the same interaction timestamp for all data records that form the aggregated entity (i.e., many records will have the same timestamp).

From the interaction units and timestamps, we compute two metrics that we refer to as **provenance attributes**: *frequency* and *recency*. We chose to focus on these metrics as they are both relevant to provenance tracking and have been commonly used in visualization research and practice [9, 105, 27, 25, 110]. Figure 7.1 illustrates sample *frequency* and *recency* computations, which are described in the following paragraphs.

Frequency. This provenance attribute computes a frequency score f_x normalized from zero to one, for each data attribute:

$$f_x = \frac{n_x}{\max_{i=1}^{N} n_i}$$

where n_x is the total number of interactions with a data attribute x and $\max_{i=1}^{N} n_i$ is the maximum number of interactions from among all N data attributes. A score of zero implies no interactions (or zero focus) and a score of one implies the most number of interactions (or maximum focus). Like data attributes, we also compute f_x for data records.

Recency. This provenance attribute computes a recency score r_x normalized from zero to one, for each data attribute:

$$r_x = \frac{\text{rank}_N(\max(t_x))}{\sum_{i=1}^{N} n_i}$$

where $\max(t_x)$ is the timestamp of the most recent interaction with a data attribute x, $\text{rank}_N(\max(t_x))$ is its serial order in the overall sequence of interactions across all N data attributes (i.e., the entire analysis history), and $\sum_{i=1}^{N} n_i$ is the total number of interactions across all N data attributes. A score of zero implies no interactions (or zero focus) and a score of one corresponds to the most recent interaction (or focus). Like data attributes, we also compute r_x for data records.

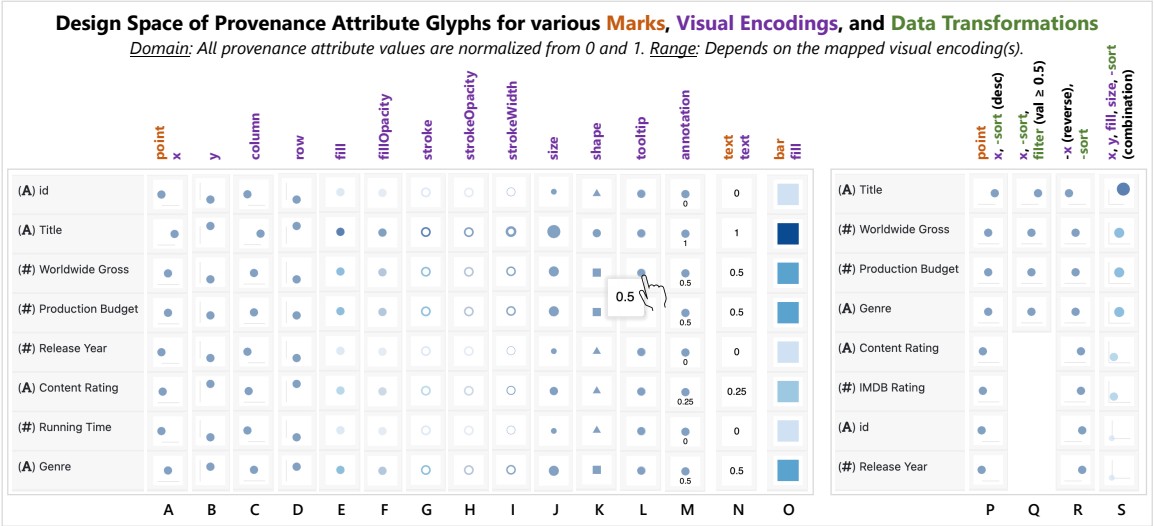


Figure 7.2: Design space of **provenance attribute glyphs** (**A**–**S**) to visualize the values of provenance attributes (normalized from 0 to 1) for data attributes (or records) across different **marks** (**point**, **text**, **bar**), **visual encodings** (**x**, **y**, **column**, **row**, **fill**, **fillOpacity**, **stroke**, **strokeOpacity**, **strokeWidth**, **size**, **shape**, **tooltip**, **annotation**, **text**), and **data transformations** (**sort**, **filter**), including alternate configurations (e.g., **–x** where the range is **reverse**d or **desc**ending sort order) and combinations (e.g., **x** + **y** + **fill** + **size** + **sort**). For instance, for **mark=bar** and **encoding=fill** (**O**): "Title" ■ has the largest value (darkest bar) followed by "Worldwide Gross" ■, "Production Budget" ■, and "Genre" ■; "id" ■, "Release Year" ■, and "Running Time" ■ have the smallest values (lightest bars). Notice the change to the attribute sort order for the right side of the design space (**P**–**S**), compared to the unsorted attributes on the left (**A**–**O**).

7.2.3 Visualizing & Interacting with Provenance during Analysis

Our main goals were to enable users to access the provenance of specific attributes or records and also obtain a visual provenance overview of the entire dataset during analysis. In response, we designed small glyphs called **"provenance attribute glyphs"**, which comprise a mark type (e.g., *point* ○) and one or more visual encodings (e.g., *fill* ●) that encode the provenance attribute values (e.g., ●●● where darker glyphs imply higher values). These glyphs can represent data records within visualizations (e.g., points in a scatterplot), can be displayed alongside data attributes (e.g., in the attribute panel), and can integrate well with sorting and filtering operations. These glyphs can reveal interesting provenance patterns, e.g., if there are more dark than light glyphs (or vice versa), then the user has

interacted disproportionately. Figure 7.2 summarizes this design space (for attributes).

7.2.3.1 Marks and Visual Encodings

Our design space covers three mark types: *point* ○, *bar* ■, and *text* (0.5). Other mark types such as *line* and *area* require at least two values (a start and an end), which make them unsuitable for encoding, and hence are not considered. Each mark type encodes a single value across one or more visual encodings, as described next.

Our design space covers 13 encodings [260]: *x*, *y*, *column*, *row*, *fill*, *fillOpacity*, *stroke*, *strokeOpacity*, *strokeWidth*, *size*, *shape*, *tooltip*, *text*, and *annotation*; *annotation* is a special encoding that adds an extra text mark displaying the encoded value next to the visualization mark, unlike the *text* encoding, that only displays the text (as the visual mark itself).

For instance, Figure 7.2**O** (**mark=bar**, **encoding=fill**) shows that "Title" has the largest value (darkest bar ■), whereas "id", "Release Year", and "Running Time" have the smallest values (lightest bar ■). Figure 7.2**I** (**mark=point** and **encoding=strokeWidth**) shows the glyph ○ for the largest value (thick stroke) and ○ for the smallest value (thin stroke).

7.2.3.2 Data Transformations

In addition to visualizing provenance attributes (marks and encodings), we also enable users to transform (filter and sort) their data by the provenance attributes. This functionality was inspired by DataPilot [4], which similarly lets users sort and filter their data attributes and records based on their quality and usage characteristics.

Sort displays the data attributes or records in order of the encoded provenance attributes. For example, Figure 7.2**P** shows an active descending **–sort** by *frequency*, visualized as **mark=point** and **encoding=x**, illustrating that "Title" had the most interactions followed by "Worldwide Gross", "Production Budget", and so on. Notice the change to the attribute sort order for the right side of the design space (Figure 7.2**P**–**S**), compared to the unsorted

attributes on the left (Figure 7.2**A**–**O**).

Filter displays a subset of data attributes or records based on the criteria provided by the user (e.g., the encoded entity values). For example, Figure 7.2**Q** shows an active **filter** for *frequency* greater than or equal to 0.5, emphasized by **mark=point**, **encoding=x**, and a descending **–sort** (also by *frequency*), resulting in four attributes that match the filter ("Title", "Worldwide Gross", "Production Budget", "Genre").

7.2.3.3 Configurations and Combinations

Our design space enables various configurations and combinations of marks, encodings, and data transformations, affording user agency and control, and promoting accessibility, like the different strategies to debug NL2SQL workflows, as undertaken by the participants of Debug-It-Yourself [26] (chapter 4).

Configurations. Users can configure the range of the encoding scale (e.g., for *size*, high values map to bigger glyphs or vice versa) or sort directions (i.e., ascending or descending). For example, all data attributes to the right side of the design space in Figure 7.2**P**–**S** are **–sort**ed by *frequency* in the descending order (which means attributes at the top have been used more often in the interface). However, **P** maps the glyphs to **x** whereas **R** maps them to **–x** (reverse). Such configurability can support different user preferences and use cases, e.g., visited points could become smaller, nudging users to interact with other points (increase coverage); alternatively, they could become bigger, helping users quickly spot them.

Combinations. Our design space does not limit the user to one configuration at a time. Users can simultaneously utilize one or more visual encoding assignments, as well as filter and sort criteria to realize a wide variety of glyph combinations. For example, Figure 7.2**S** shows a combination of *x*, *y*, *fill*, *size*, and *sort*. Combinations can help reinforce certain interaction patterns and also help tools ship with smart defaults to serve a wider audience,

e.g., using both *fill* and *size* can support both colorblind and non-colorblind users, similar to how figures in articles are often colored but also hatched.

7.3 ProvenanceLens

To study the utility and usage characteristics for provenance attributes during visual data analysis, we developed a system prototype, ProvenanceLens, that allows users to map provenance to visual encodings and data transformations; essentially, users can now interact with provenance in the same way as regular attributes.

7.3.1 User Interface

We developed a system (Figure 7.3) wherein users can utilize provenance attributes to perform visual data analysis (e.g., inspect a dataset, create visualizations, and apply transformations) and also answer questions about their provenance. It comprises seven views:

A Data Attributes. In this view, users upload and configure the dataset (⚙) and see the underlying attributes (or features or columns). Hovering on the information icon ℹ shows the attribute's definition in a tooltip. Clicking on the expand icon ⌄ opens a detailed view with a distribution plot of the attribute's values: an area curve for numerical attributes and a column chart for categorical attributes, both of which show percentage counts corresponding to the attribute quantiles and categories, respectively. Users can sort and filter the attributes using the *recency* and/or *frequency* provenance attributes.

B Mark. This view includes a dropdown to configure the mark type for the visualization. Users can select one of *point*, *bar*, *line*, *area*, or *text* to begin a visualization specification.

C Encodings. This view shows the visual encoding channels. Users select or drag one or more attributes (data and/or provenance) to one of *x*, *y*, *fill*, *fillOpacity*, *stroke*, *strokeOpacity*, *strokeWidth*, *shape*, *row*, *column*, *tooltip*, and/or *text* to complete a visualization spec-

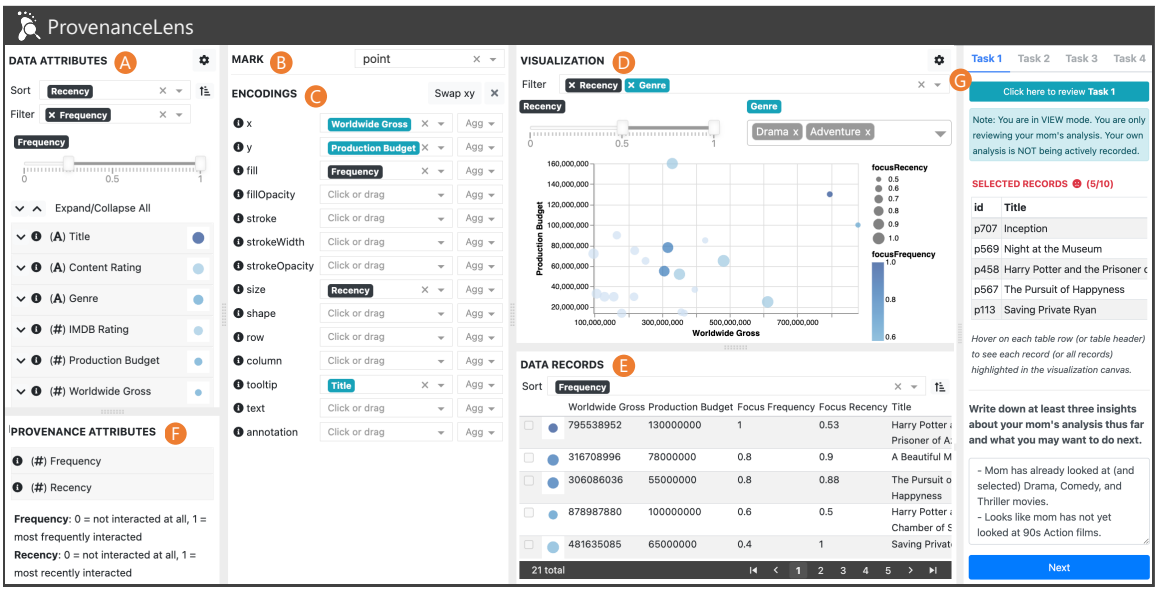


Figure 7.3: The ProvenanceLens user interface consisting of seven views: A the **Data Attributes** view shows the attributes and enables transformation (e.g., sort, filter); B the **Marks** and C **Encodings** views specify the visualization; D the **Visualization** view renders the specified visualization and supports filtering of data records; E the **Data Records** view supports review and transformation (sort) of the data records shown in the visualization; F the **Provenance Attributes** view lists the *recency* and *frequency* attributes; and G the **Tasks** view shows the task instructions and questions, and tracks the user's progress.

ification. An additional encoding, *annotation*, adds a new annotation displaying the value of the encoded entity next to the selected mark.

D **Visualization.** This view renders an interactive visualization based on the selected mark type and activated visual encodings in the "Encodings" view. It also includes a "Filter" drop-zone to filter out data points by attributes (data and/or provenance). A numerical attribute displays a range slider and a categorical attribute displays a multiselect dropdown.

E **Data Records.** This view shows the data bound in the visualization as a paginated data table. If the user hovers on a datapoint in a unit visualization (e.g., a scatterplot), this table filters to only show that data record whereas if the user hovers on an entity in an aggregate visualization (e.g., a bar showing the *mean* value), this table filters to show all data records belonging to the hovered entity (e.g., the bar).

F Provenance Attributes. This view shows the two provenance attributes: *frequency* and *recency*. Like data attributes, users can select or drag these provenance attributes and drop them to the sort and/or filter drop zones in the "Attribute View", the encoding channels in the "Encodings" view, the filter drop zone in the "Visualization" view, or the sort drop zone in the "Data Records" view.

G Tasks. This view has six tabs, one for each task (T1–T6), allowing users to access their task instructions, track their progress, and answer questions via integrated forms.

Configurations. ProvenanceLens can be programmatically configured into three modes: (1) *edit*-only, where provenance is tracked and presented back to the user in real-time, (2) *view*-only, where existing provenance is imported into ProvenanceLens without real-time tracking, and (3) *hybrid*, where provenance is imported, and real-time tracking is enabled.

7.3.2 Example Scenarios

We present two usage scenarios on how ProvenanceLens can enhance visual data analysis for real-time provenance tracking as well as post-analysis review of a user's provenance.

Real-time Provenance Tracking. Assume Mark works for a movie production company and must determine what kinds of movies to make next. They upload the dataset of movies in ProvenanceLens (configured in its *edit*-only mode, i.e. real-time provenance tracking) and begin exploring (by specifying different visualizations, applying relevant filters, and hovering on certain movies). After taking a short break, they wish to revisit their most recently hovered movie, so they create a visualization with a *point* mark type and map the *recency* provenance attribute to the *x* encoding. They immediately find their desired movie (with the highest recency score) at the horizontal axis' rightmost-end. Had they mapped *recency* to another encoding such as *fill* or *size*, it may have taken them some time to accurately determine the darkest or biggest point.

Upon further inspection of their focus on different movies, Mark noticed they have interacted with some movies multiple times and not considered a lot of other movies (i.e., they exhibited less exploration coverage). Mark wants to change this analytic behavior, and thus starts visualizing traces of their interactions in real-time. Because they are color-blind, they avoid the *fill*, *stroke*, *fillOpacity*, and *strokeOpacity* encodings and choose *size* to encode the *recency* provenance attribute. They can now track visited points using their bigger size and continue exploring. However, because visited points get bigger, they are getting drawn to the same points even more; thus, they reverse the *size* range to make the visited points smaller instead. Happy with this configuration, they continue exploring and eventually submit their analysis report to their manager.

Post-Analysis Provenance Review/Audit. In addition to real-time provenance tracking, ProvenanceLens can also *import* an existing log of a user's provenance to facilitate collaboration (e.g., continuing a colleague's analysis), auditing (e.g., inspecting a colleague's analysis), or post-hoc analysis (e.g., reviewing user study logs [92, 261]).

Assume Anya is Mark's manager and is reviewing their previous analysis. They express surprise at Mark's recommendation to make a *Drama* movie next. Wanting to review Mark's analysis, Anya imports Mark's provenance (exported from ProvenanceLens) into ProvenanceLens. They configure ProvenanceLens in its *view*-only mode, i.e., no real-time provenance tracking, to view and interact with Mark's analysis. They make a bar chart with "Genre" on x and the sum of *frequency* on y. This visualization shows Mark's total focus across movie genres. Anya notices that certain bars corresponding to *Action* and *Adventure* movies are really short (i.e., of low *frequency*). They ask Mark to review those genres before making a final recommendation. In this way, Anya was not only able to review Mark's report but also their analysis process, thereby supporting enhanced decision-making.

7.4 Evaluation: Exploratory User Study Using ProvenanceLens as a Design Probe

After establishing the design space based on provenance attributes, we wanted to study a second research question, *"How do people use provenance attributes during visual data analysis?"* For example, do users prefer mapping provenance to color, size, both color and size, or something different, like an axis? If not visual encodings, do users prefer interacting with provenance via data transformations such as sorting or filtering? Lastly, do these preferences change for specific tasks, e.g., while reviewing someone else's analysis history versus doing their own analysis? We designed a decision-making task that involves reviewing and answering questions about both another user's and one's own analytic provenance.

7.4.1 Pilot Studies and Evaluation Considerations

Before finalizing our study design, we explored two alternate designs and also conducted subsequent pilot studies.

Pilot Study 1: Decision-Making. We recruited four Ph.D. students (three years into the program) as our pilot users. We tasked users to explore a dataset of movies and select (1) ten movies (records) satisfying certain criteria, (2) four movie characteristics (attributes) that were important to their analysis, and (3) answer a series of questions about their analysis.

We observed users did not utilize the provenance attributes to track their analysis process and only used them during the subsequent question-answering. We also noticed that answering questions immediately after analysis may not be hard for certain users because the analysis may be quite fresh in their memory. In addition, the idea to make users select important attributes was straightforward and seemed redundant because the task criteria already hinted what attributes to consider.

Pilot Study 2: Analysis Review and Decision-Making. We recruited two other Ph.D. students as our pilot users. In this revised study design, we introduced an initial *review* task

to first make the user explore another user's analytic provenance, write three insights, and then answer questions about the prior analysis process. The original task to select movies followed this task. Our goal was to make the user actively utilize the provenance attributes during their own analysis, and we believed making them answer questions about another user's analysis first would remove the fresh-in-memory aspect from the equation. We also discarded the selection of four movie characteristics. Overall, we noticed this design had a desired positive effect and decided to use it for our eventual user study, as described next.

7.4.2 Participants and Procedure

Participants. We recruited 16 participants from a public university in the U.S. who were pursuing a bachelors (1), masters (11), or doctoral (4) degree in computing or related fields (15) and economics (1). These participants were either enrolled in or alumni of at least one visualization class and self-reported their visualization literacy to be at least 3 on a scale from 1 (novice) to 5 (expert). Demographically, they were in the *18-24* (7) or *25-34* (9) age groups (in years) and of *female* (5), *male* (10), or *preferred not to say* (1) genders.

Study Session. Each study session lasted between 75 and 90 minutes. We compensated each participant with a $15 gift card for their time. We conducted the study remotely over Zoom; the experimenter provided participants access to the study environment by sharing their (the experimenter's) computer screen and granting input control to the participant. After providing consent, participants saw a five-minute video tutorial that demonstrated the features of ProvenanceLens. Participants then performed a practice task on a *dataset of cars* to get acquainted with the UI before starting the actual task.

The actual task was on a dataset of *movies* and lasted about 60 minutes. Participants were not required to think aloud during the task to simulate a realistic work setting (although some participants felt comfortable doing so). During the task, participants' interactions with the system were logged. The study ended with participants completing a

feedback questionnaire and a background questionnaire. Each study session was screen- and audio-recorded for subsequent analysis.

Task and Dataset. We designed the following visual data exploration and decision-making task about a movies dataset:

Imagine your family is planning a month-long vacation to Europe. Going with you are your siblings, parents, grandparents, your uncle and aunt, and their boy (your cousin). Your mom began selecting some movies to pick and watch from for the occasional movie nights. **Her target is to select ten movies to carry to the vacation**.

Because she wanted to ensure a delightful and well-rounded movie night experience for your entire family, **she took the below suggestions and preferences from some of your family members into consideration**.

1. Dad: *"I like thrillers and comedies."*
2. Uncle: *"Let's watch a heartfelt drama."*
3. Cousin: *"I want to re-watch a Harry Potter movie."*
4. Grandpa: *"I would love to watch a 90s' action film."*
5. Aunt: *"I want to watch a hidden gem: a highly-rated movie that didn't do well commercially."*

After selecting five movies, your mother got pulled to do another task, and assigned you to complete it.

Thus, you will complete the following six tasks (T1-T6).

—— *Working with your mom's analytic provenance* ——

T1 **Review:** Review the five movies your mom selected and many others she explored; write three insights.

T2 **Recall:** Answer five objective questions about mom's analysis.

T3 **Visualize:** Create visualizations to answer subjective questions about mom's analysis.

—— *Working with your own analytic provenance* ——

T4 **Analyze:** Select the remaining five movies.

T5 **Recall:** Answer five objective questions about your analysis.

T6 **Visualize:** Create visualizations to answer subjective questions about your analysis.

Task	Que	Description	μAcc. (%age)	μConf. (/7)	μSur. (/7)
T1 (Review)	Q1	Review YOUR MOM's analysis and write three insights.	-	-	-
T2 (Recall)	Q1	Select the movie characteristic(s) that YOUR MOM interacted with the MOST.	100	6.81	-
	Q2	Which was the last (i.e. most recent) movie YOUR MOM interacted with?	93.75	6.81	-
	Q3	Did YOUR MOM ever interact with the movie 'Titanic'?	100	6.75	-
	Q4	Did YOUR MOM interact with at least one movie from all 'Content Rating's?	100	6.69	-
	Q5	Did YOUR MOM interact with all attributes at least once?	100	6.75	-
T3 (Visualize)	Q1	What was the distribution of YOUR MOM's focus across different movie 'Genre's?	100	6.75	-
	Q2	How did YOUR MOM's focus on 'Drama' Movies evolve over time?	100	6.06	-
	Q3	Which were YOUR MOM's most FREQUENTLY interacted movies? Try to show five.	100	6.63	-
T4 (Analyze)	Q1	Select the five remaining movies.	-	-	-
T5 (Recall)	Q1	Select three movie characteristic(s) YOU most RECENTLY interact with.	100	6.88	1.88
	Q2	Which was the first (i.e. earliest) movie YOU interacted with?	93.75	6.25	3.56
	Q3	Did YOU interact with the movie 'Pearl Harbor'?	100	6.81	1.63
	Q4	Which 'Content Rating' category did YOU interact with the LEAST or NONE AT ALL?	100	6.69	4.19
	Q5	Did YOU interact with at least half of the movie characteristics available?	100	6.88	1.94
T6 (Visualize)	Q1	How similar were YOUR interaction patterns for 'Comedy' and 'Thriller' movies?	100	6.38	2.56
	Q2	Which were YOUR most RECENTLY interacted movies? Try to show THREE movies.	100	6.63	2.69
	Q3	Given an opportunity, which (kinds of) movies would YOU like to go back and interact with?	100	6.44	2.38

Figure 7.4: Tasks and summary performance statistics for sixteen participants: **Task** and **Que**stion index, task **Description**, and wherever applicable, average accuracy (μ**Acc.**), average confidence (μ**Conf.**), and average surprise (μ**Sur.**) on a scale from one (low) to seven (high). T1 and T4 were exploratory in nature, hence we did not compute accuracies or ask participants to self-report confidence and surprise scores. Similarly, T2 and T3 were focused on answering questions about mom's analysis, hence we did not ask participants to self-report their surprise scores. Overall, participants performed exceedingly well on all the tasks, achieving high accuracies with high confidence with some moments of surprise.

To help you review and track the movies (records) and their characteristics (attributes) one has *interacted* with how many times and when, our system provides two special provenance attributes: **Frequency** and **Recency**. You can use these in the same way you would use the dataset attributes, i.e., create visualizations by mapping them to visual encodings, filter by them, or sort by them. Note that "*interacted*" refers to a user's interactions in the interface such as hovering on a record to get additional details, mapping dataset attributes and/or provenance attributes to visual encodings (e.g., 'Genre' to *x*), applying a filter (e.g., 'Title'='Titanic') or sort.

7.4.3 Results

In this section, we present general and task-specific findings from the user study along with participant performance on the six tasks (T1-T6) and discuss them in the context of the qualitative feedback from our participants ($P_{1,\ldots,16}$).

7.4.3.1 General Feedback

Participants scored ProvenanceLens 80.94 out of 100 on the system usability scale (SUS [186]), finding it very useful. P_{15} acknowledged it is impossible to remember everything during analysis, calling for *"Tools like Tableau [to] really overlay communities' or experts' interaction stats such as frequency and recency in their system. It is one of those features that only adds value and is not necessarily a hindrance."* P_{14} was impressed that the two provenance attributes could help answer a variety of questions, enabling users to revisit, review, and maybe even recreate someone's history of interactions while doing analysis. P_{11} liked being able to detect subconscious interaction patterns during analysis. P_{16} suggested a potential use case, *"If you are a manager and if you need to review the decision process of your employees, then it is very useful."*

7.4.3.2 Task Performance

Figure 7.4 shows the task-specific breakdown of participants' accuracy and self-reported confidence and surprise.

Accuracy. Participants performed exceedingly well overall, achieving high accuracies during both **recall** tasks (T2, T5) and **visualize** tasks (T3, T6). The mean accuracies were as follows: T2 ($\mu = 98.75\%$), T3 ($\mu = 100\%$), T5 ($\mu = 98.75\%$), T6 ($\mu = 100\%$). We did not compute accuracy for the **review** (T1) and **analyze** (T4) tasks as they were exploratory.

Only two participants ($P_{13,14}$) incorrectly answered one question each. For instance, for T2.Q2 (*"Which was the last (i.e. most recent) movie YOUR MOM interacted with?"*), P_{14} selected the 'point' mark type and assigned "Genre" to *x*, *recency* on *size*, and "Title" on *tooltip*. Then, they applied a *recency* filter of [0.63, 1]. This configuration resulted in four marks stacked on top of each other (unknown to the participant who thought there was only one). The user hovered on it, read *"The Curious Case of Benjamin Button"* in the tooltip, and selected it as the answer. However, this answer was incorrect because *"Saving Private*

Ryan", under this movie, had the higher recency score of 1.0. This error may have been avoided if instead of (or along with) *size*, the user assigned *recency* to *x* or *y*, which would have spaced out the points.

Confidence & Success. On a scale from 1 (low) to 7 (high), participants self-reported very high *confidence* in answering questions during tasks T2 (μ=6.76, M=7), T3 (μ=6.48, M=7), T5 (μ=6.70, M=7), and T6 (μ=6.48, M=7). In addition, while answering questions based on their own analysis, participants self-reported varying *surprise* during tasks T5 (μ=2.64, M=2) and T6 (μ=2.54, M=2). Note that T1 and T4 were exploratory in nature, hence we did not ask participants to self-report confidence and surprise scores. Similarly, T2 and T3 were focused on answering questions about mom's analysis, hence we did not ask participants to self-report their surprise scores.

Fidelity. On a scale from 1 (low) to 5 (high), participants self-reported that overall, all six tasks (T1-T6) caused low *physical demand* (μ=1.44, M=1), *temporal demand* (μ=2.44, M=2), and *frustration* (μ=1.5, M=1), average *mental demand* (μ=3.31, M=3) and *effort* (μ=2.94, M=3), but resulted in high *performance* (μ=4.38, M=4). On average, participants spent around 56 minutes on the study and performed 409 interactions. We avoided seeking feedback after each task to prevent interrupting the user's analysis and demotivating them.

Summary. The high *accuracy*, *confidence*, and *performance* along with average *effort* and *mental demand* suggest that participants were able to effectively use provenance attributes to answer the questions. The high variance in surprise scores suggests that participants were sometimes able to recall their analysis from memory (less surprise) or had an incorrect mental model or recall of their interactions (more surprise).

7.4.3.3 Task-specific Feedback

Reviewing Mom's Previous Analysis (Review: Task T1). Due to the nature of this task, all participants leveraged the provenance attributes and found them to be useful. Participants felt that the provenance attributes could provide *"a record of [mom's] analysis"* (P_{12}), which made it easier to *"track her way of thinking through a quantitative behavior analysis"* (P_3). For P_6, the provenance attributes were only somewhat useful because they were able to understand mom's analysis simply by inspecting her final selections. On the other hand, P_4 argued that the provenance attributes *"gave context for [mom's movie selections],"* which they would have to otherwise guess (P_{12}).

Answering Questions about Both Mom's and One's Own Analysis (Recall, Visualize: Tasks T2, T3, T5, T6). Like T1, due to the nature of the questions, all participants found the provenance attributes to be useful. P_1 noted, *"It was very useful to use provenance attributes as filters and encodings. I would not have been able to answer questions effectively without the visualization ability."* P_{14} noted, *"While looking at and analyzing [mom's] focus, it was hard to keep all the insights in mind, so having it visualized helped understand and hence remember it better."* P_9 found the provenance attributes to be more helpful to answer questions about visualizing focus (T3, T6) than searching for insights (T2, T5), noting that *"I was able to concentrate on specific facets of [mom's] focus at a time [via the visualizations], which felt more organized."*

Selecting Remaining Movies (Analyze: Task T4). Ten participants ($P_{1,2,3,5,6,7,10,11,12,16}$) either did not use the provenance attributes or found them to be less useful. Among these, P_3 was more focused on selecting movies that satisfied the family's constraints and hence was more inclined to check the actual attributes rather than the *frequency* and *recency*. $P_{6,12}$ speculated that the provenance attributes would be more useful if the study lasted longer or was spread across multiple sessions.

On the other hand, six participants ($P_{4,8,9,13,14,15}$) found utility in tracking and reviewing their provenance in real-time. P_{13} actively used the provenance attributes to keep track of their own analysis and not become stuck while making a choice. P_4 realized that after satisfying all family members' movie constraints, they were free to give their own recommendations. They saw they had not interacted with "Production Budget" and "Worldwide Gross" too much, so they created a scatterplot with these attributes, mapped *frequency* and *recency* on *color* and *size*, and utilized the visual scents to analyze unexplored movies and accordingly determine their final movie selections.

7.4.3.4 Reasons for Using the Provenance Attributes

Answer questions when unable to recall analysis behavior. Due to our study design, answering questions was the most common reason to use the provenance attributes. Reviewing analytic behavior as a quantized set of variables (P_3) and using them in visual encodings ($P_{1,14}$) and data transformations (P_1) helped confidently answer questions (P_3).

Verify and build confidence while answering questions. Sometimes participants recalled answers from memory, but using the provenance attributes helped them verify and double-check their choices ($P_{14,16}$) and gain confidence ($P_{12,14}$).

Save time during analysis. Some participants found the *recency* and *frequency*-based data transformations and aggregations particularly useful to generate quick responses to the study questions (P_{11}), directly saving them a lot of time (P_{10}).

Increase awareness, uncover new insights, and improve exploration coverage. When selecting the remaining movies, the provenance attributes helped participants optimize their analysis by increasing awareness (P_4), helping them avoid revisiting the same points ($P_{9,15}$), and facilitating quick decisions (P_{15}). Some participants uncovered new insights ($P_{2,3}$) or unexplored data (P_3), and identified behavioral trends (P_3).

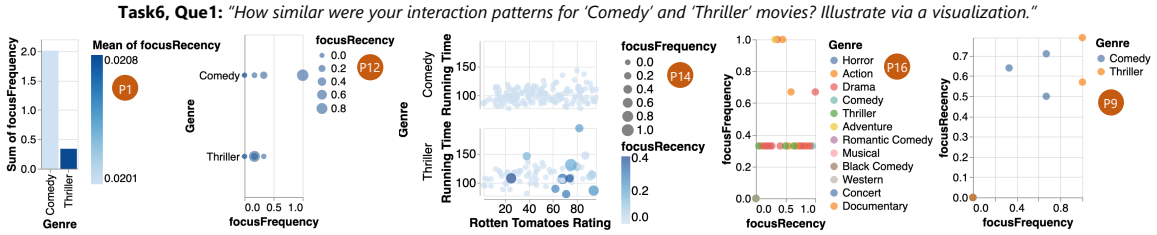


Figure 7.5: Five participants' different strategies to answer the same question, T6.Q1, *"How similar were your interaction patterns for 'Comedy' and 'Thriller' movies? Illustrate via a visualization."* While P_1 created an aggregate bar chart visualizing showing the two "Genre"s on x, total *frequency* along y, and colored by average *recency*, P_{14} created a scatterplot visualization faceted by "Genre", colored by *recency* and sized by *frequency*.

Fun and Feel Good. For some participants, the provenance attributes were easy and fun to use (P_{13}). The real-time changes to the colors or sizes of points made them feel good and feel like they were making progress in the task (P_{13}).

7.4.3.5 Reasons for More or Less Surprise

While answering questions, participants were surprised because (1) the system's answer did not match their recollection of their own analysis, (2) the system's answer did not match their original analytic intention, (3) the system inadequately captured the participant's focus (e.g., interactions with an aggregate visualization gave equal focus to constituent datapoints, which was not agreeable to the user), or (4) the system logged interactions that were perhaps accidental (our threshold to discard mouseovers under ∼250 ms be too low to count towards focus). For example, P_2 noted, *"I found [an interaction] to be a Harry Potter movie, which makes sense, but I had forgotten about having interacting with it first."*

On the flip side, participants were either less surprised or not surprised at all because (1) they actively used the recency and frequency attributes during analysis and thus knew what they were doing and (2) they were able to simply remember their behaviors. P_5 noted, *"I already knew what I had focused on, but far ahead in the future, these provenance attributes might be helpful to provide reasoning for my choices when I might not explicitly remember my thought process."*

7.4.3.6 *Participant Strategies to Answer Questions*

To answer questions, participants often utilized provenance attributes as a combination of encodings, transformations (sort and filter), and subsequent interactions (e.g., tooltip on hover). We observed 71 different strategies. For example, Figure 7.5 shows five participants' different strategies to answer T6.Q1: *"How similar were your interaction patterns for 'Comedy' and 'Thriller' movies?"* While P_1 created an aggregate bar chart showing the two "Genre" categories on *x*, *frequency* on *y*, and average *recency* as *color*, P_{14} used a scatterplot encoding *recency* (color) and *frequency* (size), faceted by "Genre".

Co-occurrence analysis. Figure 7.6 shows the co-occurrence of provenance attributes (*only frequency*, *only recency*, or either) as A combinations of visual encodings compared to B data transformations. In terms of visual encodings, *x* and *y* independently were used most often followed by *fill* (color). Interestingly, *frequency* and/or *recency* were simultaneously mapped to both *x* and *y* 36 times. Between encodings and data transformations, standalone encodings were used most often (131 times) followed by their combination with record filter (*encoding_rFilter*, 48 times) and attribute sort (*encoding_aSort*, 35 times).

7.4.3.7 *Preferences for Utilizing the Provenance Attributes*

Figure 7.7 shows how users mapped data and provenance attributes to visual encodings A , and their general preferences for encodings compared to transformations B.

Visual Encodings. For provenance attributes, *x* was the most preferred encoding followed by *y*, *fill*, *size*, *text*, *fillOpacity*, *row*, *annotation*, and *column*. The encodings *shape*, *stroke*, *strokeOpacity*, and *strokeWidth* were either rarely used or not used at all. Similarly, for data attributes, *x* or *y* were the most preferred encodings. The nominal "Genre" and "Content Rating" attributes were also occasionally mapped to *fill* and *row* encodings. "Title" was predominantly mapped to *tooltip*, followed by *x*, *annotation*, *y*, and *text*.

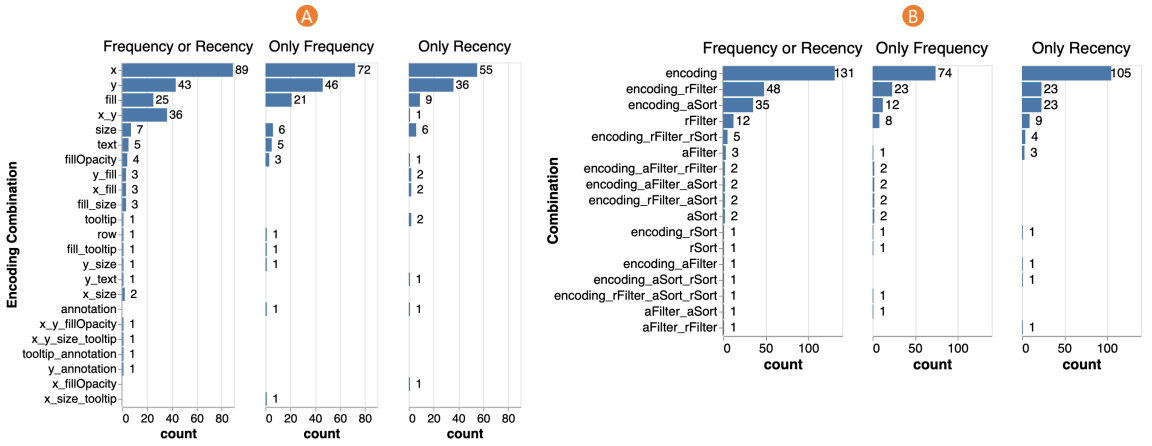


Figure 7.6: Co-occurrence statistics for how users map provenance attributes (*only frequency*, *only recency*, or either) to visual encoding combinations (A), as well as general preferences for visual encodings compared to filtering, and sorting (B). Note that these statistics correspond only to the **recall (T2, T5)** and **visualize (T3, T6)** tasks; we exclude the **review (T1)** and **analyze (T4)** tasks as they were more open-ended in nature.

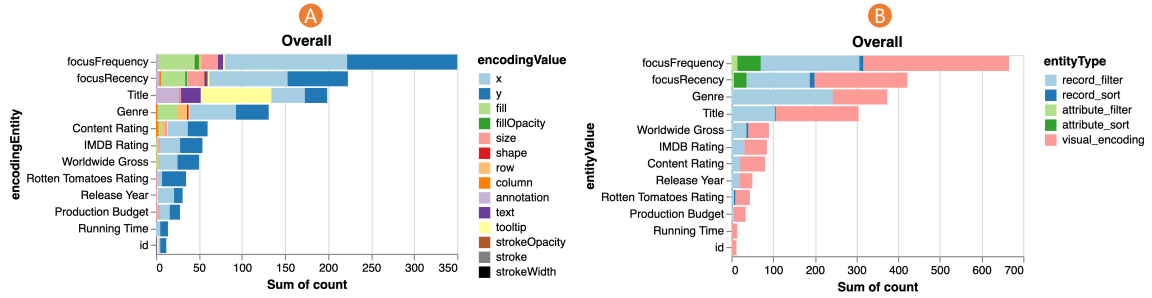


Figure 7.7: User preference when mapping attributes to (A) different visual encodings and (B) using visual encodings in general compared to data transformations.

Visual Encodings or Data Transformations. For provenance as well as data attributes, visual encodings were most preferred followed by filtering records and sorting attributes.

7.4.3.8 Recency or Frequency? What was more Useful?

Three participants ($P_{1,5,14,16}$) mentioned frequency was more useful than recency, in particular to identify movies they had visited multiple times. P_{11} also noted that frequency aided in recognizing options that were still being considered. P_{15} suggested that while frequency was more useful for the short duration of the study, recency might be more valuable in the long term, especially for tasks like auditing or reviewing analysis after days or weeks.

7.4.4 Discussion

7.4.4.1 Making Provenance a Core Aspect of Analysis

Our study revealed that users can use provenance attributes to answer questions with high accuracy and confidence, while also sometimes being surprised. In addition, there were several instances of users requesting new capabilities not currently supported by ProvenanceLens. For example, P_6 suggested an interesting feature to toggle between their own provenance and their mom's during analysis. P_8 requested the ability to "undo" an accidental interaction, hinting towards an ability to directly manipulate/correct their provenance. These feature requests solidify that provenance still has unrealized utility, and should thus be considered as a core element during analysis, thereby calling for visual data analysis tools to inherently support it.

7.4.4.2 Fostering Provenance-driven (Not Data-driven) Analysis.

Many participants visualized provenance by mapping it to visual encodings (e.g., color) to keep track of their ongoing analytic progress (e.g., to avoid revisiting the same points). Some participants also actively sorted and filtered the data attributes and records by provenance attributes to either reduce their search space or look-up what has been previously considered. These are examples of provenance-driven (not data-driven) analysis wherein provenance information is used to steer the analysis process. Similar to how data-driven analysis focuses on using data to guide decision-making and insight generation, provenance-driven analysis can surface the lineage and context of data and interactions to determine next steps and ensure the accuracy, reliability, and interpretability of the analysis. We thus call for tools to support both these analysis paradigms.

7.4.4.3 Integrating Provenance-Tracking & Visual Data Analysis

During our study, participants often used provenance attributes together with data attributes. For example, participants mapped provenance attributes to *x*, *y*, *fill*, and *size*, which are encodings commonly used for data attributes. Furthermore, there were cases where a participant replaced a data attribute (that was mapped to a visual encoding), with a provenance attribute (and vice-versa). In other cases, participants had a preferred way of interacting with analytic provenance (e.g., always mapping it to *fill*) and they consistently used the same strategies during analysis or when answering questions. These behaviors suggest that tools should offer both data and provenance attributes for more flexible workflows.

7.4.4.4 Affording Flexibility in Mapping Provenance to Encodings

Our study revealed that users often visualized provenance on visual encodings such as *x*, *y*, and *tooltip*, something not often seen in existing tools. Furthermore, while performing tasks T2 (Recall) and T4 (Analyze), some participants used multiple visual encodings for the same question in succession, to verify the results. For example, P_4 first mapped *frequency* to *fill*, but found it hard to accurately differentiate between different shades and hues (which can be hard for colorblind users). To verify their takeaways, they mapped *frequency* to *size* instead, but were worried about occlusion, depending on the x/y point positions. Consequently, they applied a double-encoding by also mapping *frequency* to *x*. This flexibility shows that if an encoding is unavailable and another is inferior, then a third encoding can still be effective, underscoring our core goal to afford flexibility in mapping provenance during analysis.

7.4.4.5 Comparing Provenance Encodings and Transformations

During our study, participants regularly mapped provenance to visual encodings or applied data transformations. However, depending on the question type, one technique can be more efficient than the other. Consider T6.Q2 wherein participants were tasked to show **three**

movies that they most recently interacted with. Those who applied a *recency* filter often struggled with this task due to ties in the number of times a movie was interacted with. For example, P_1 asked, *"How do I get exactly three?"* Filtering for top-N movies (records) or movie characteristics (attributes) is less trivial as it is impossible to guess what range will produce the exact number of items. Mapping provenance to positional encodings (e.g., *x*) or filtering based on ordinal *provenance rank* instead of a quantitative *provenance score* can make it easier to spot ties.

7.4.4.6 Supporting Collaboration during Visual Data Analysis

In tasks T1–T3 of our user study, participants had to utilize provenance attributes to review and answer questions about another user's (mom's) analysis. In doing so, we indirectly studied how provenance attributes can facilitate asynchronous collaboration. By analyzing another user's provenance in terms of what they looked at, when, and for how long, the user can not only verify or find flaws in prior analyses but also formulate a starting point or become unstuck [4]. However, overreliance on another user can hamper creativity and result in "herd behavior" [4]. Balancing these two behaviors can result in efficient collaborative analysis and decision-making.

7.5 Limitations and Future Work

7.5.1 Modeling Provenance as an Attribute.

First, mapping *recency* and *frequency* to visual encodings, such as darker and larger points for more recent or frequent data, can foster confirmation bias [262] by leading users to unknowingly and disproportionately prioritize recently or frequently interacted data points. Such behavior is an unfortunate consequence of the recency effect [263] and the frequency effect [264]. As a result, users may potentially overlook less accessed but relevant information. To mitigate this bias, a reversed provenance scale that emphasizes older and less frequent data (as darker and larger) can provide a more balanced view.

Next, our approach to normalize recency and frequency values to a uniform ranked scale from zero to one, instead of displaying absolute values like raw timestamps and interaction counts, can simplify comparisons but sacrifice explainability. For instance, without timestamps, users may not understand the exact sequence or timing of interactions, i.e., whether frequent interactions occurred closely together or were spread out over time. Additionally, users may overlook the original context (i.e., the data attributes and records themselves) or the rationale of the previous user whose provenance they are reviewing. Such lack of clarity can lead to incomplete or biased interpretations during decision-making.

Lastly, provenance attributes can be modeled in multiple ways. For *frequency*, the default *relative* strategy divides the "total interaction units" for an attribute by the maximum value among all attributes; the *absolute* strategy divides each value by the sum of all values; and the *binary* strategy treats zero interactions=0 and at least one interaction=1. For recency, the default *relative* strategy determines the value based on the sequence (or rank) of interactions; the *absolute* strategy determines the value based on the actual time duration between interactions; and the *binary* strategy assigns a value=1 for the most recent interaction and a value=0 for all other interactions. Our conceptualization of provenance attributes seamlessly supports all of these models for subsequent visualization.

7.5.2 Exploratory User Study.

While ProvenanceLens lets users customize the range of provenance attributes (e.g., darker points can be mapped to smaller or larger values), we disabled this functionality to minimize users' cognitive load during the study. Systematically studying this feature, i.e., if one range leads to more unique data discoveries [84] while another leads to more data revisits [27] is future work. Next, we modeled *recency* and *frequency* only based on mouse interactions such as clicks and hovers, which may not be a complete proxy for focus. A similar confusion had also come up during our pilot studies, but we tried to address it by clearly explaining to users how provenance is computed, and ensuring that users become

acquainted with it during the practice. We posit some participants may have been confused by the system's projection of their provenance not aligning with their expectations. Future work may utilize user gaze (e.g., which attribute is the user actually looking more at) to more accurately model focus. Lastly, we currently model focus by equally weighting all interactions but future work can weight recent interactions more than the older ones [110].

7.6 Summary

In this chapter, I described a design space for communicating analytic provenance, by utilizing provenance as an attribute during analysis, mapping it to visual encodings and data transformations. In particular, we utilized provenance as an attribute *during* analysis, tracking both *recency* and *frequency* of user interactions with data. We integrated these provenance attributes into a prototype visual data analysis system, ProvenanceLens, which allows users to track and visualize recency and frequency by mapping them to visual encodings (e.g., color or size) and data transformations (sort or filter). An exploratory study with sixteen users found that provenance attributes can help users accurately and confidently review and answer questions about their analysis, often surprising them and facilitating self-reflection. For details, I refer the reader to the associated publication [11]. Additionally, because provenance information is often the basis for provisioning guidance, this design space sets the foundation for the guidance design space, as described next.

CHAPTER 8

DESIGN SPACE AND PLAYGROUND FOR COMMUNICATING GUIDANCE

Building upon the design space for communicating provenance (chapter 7), in this chapter, I describe a design space for communicating guidance, by introducing the concepts of "wildcards", "states", and "levels", and presenting it through adaptive UI elements, partly achieving **RG3**: *Establish a design space for guidance communication during analysis*. This chapter is based on work under review [12] and is patented by Adobe Research [21].

8.1 Motivation and Background

Recall from the evaluation of "Lumos" (chapter 5) that presenting "interaction traces" (visual scents of user's interactions) in the UI makes users pause and reflect on their analytic behavior during analysis. Many users found this technique to be useful, intuitive, and 'fun'. For example, these users inspected the ex situ interaction traces (i.e., red-green colored attributes in the Distribution Panel, as determined by the AD [111] bias metric), acknowledged that they did indeed over- or underemphasize certain attribute categories/quantiles, and were also able to devise successful mitigation strategies. Essentially, these users applied a 'reverse' filter or removed an existing 'culprit' filter after which their interactions negated their over- or underemphasis, respectively. Another set of users assigned the 'biased' attribute of interest to one or more visual encodings (e.g., X or Y axes) and then relied on the in situ interaction traces (i.e., points colored in shades of white→blue) to perceptually drive their subsequent mitigating interactions. However, unlike these users, some other users expressed concern and confusion, essentially wanting some 'more' and also 'different' type of guidance. For example, upon inspecting the same ex situ interaction traces (red-green colored attributes), these users were unable to comprehend the visual cues to devise a strategy (i.e., next steps) to mitigate the skewed analytic behavior.

Next, even if the users are able to determine next steps, sustaining this process can become cumbersome and frustrating because in trying to *fix* a biased attribute, the user might unknowingly end up biasing another, essentially, *break* their correct analytic behavior with other attributes. For example, one user saw their interactions with different movie *Genres* (Concert, Documentary, and Western) and reflected, *"I should now interact with Drama since that is maximum and these [other bars] are almost nil."* They applied a filter to correct their unintended underemphasis, but after a few interactions found themselves overemphasizing towards Drama movies and had to reverse that filter; this derailed them from making progress towards their main analysis task, which is undesirable. This necessitates guidance to be *explainable* and *actionable*. So I ask (and propose):

1. *What if the UI could present as guidance explanatory natural language (NL)-based* ***analytic behavior facts*** *(inspired from interactive data facts [265]) based on the users' interactions with data (e.g., "You have underemphasized Drama movies.")?*

2. *What if the UI could present as guidance NL-based* ***concrete next steps*** *to mitigate bias (e.g., "Apply the 'Genre=Drama' filter and interact with datapoint P1.")?*

Next, viewing ex situ interaction traces separately in the Distribution Panel (in Lumos) was a design choice to provide a hybrid of *orienting* and *directing* guidance [35, 36] to users: *orienting* because the UI only presented visual scents for the distribution of the user's interactions and the underlying data; and *directing* because these attributes were also ranked with the redder, higher AD metric values at the top to capture the user's attention first for priority mitigation. However, this form of passive guidance was *inconvenient* and *not enough* for some users; *inconvenient* because the panel was positioned to the right of the screen, away from all the analysis action because of which the user has to do a lot of back-and-forth between the two; *not enough* because the user still has to manually examine the *problem(s)* with their analytic behavior and accordingly devise a mitigating *solution*; for example, one user called for "a button to automatically apply a reverse filter [instead of

them having to manually apply it]." An extreme example of guidance is to directly point the user to a certain 'best' datapoint to interact (instead of applying a reverse filter which may result in multiple datapoints). Lastly, while different degrees of guidance can be helpful, what is also desirable is users' preferences, or more broadly, their agency during analysis. What if a user is working with a particular 'analysis view' (e.g., scatterplot) and the computational guidance by the system makes them switch to a completely different view (e.g., barchart with completely different attributes)? We believe this can also derail the analysis, necessitating guidance systems that integrate user preferences into their workflows.

Thus, to provision such *co-adaptive* guidance, we need a system that offers different degrees of guidance and enables seamlessly transitioning between them, both manually driven by the user's preferences and automatically by the system's computations. We hypothesize supporting such co-adaptive, dynamic guidance with a shared agency and control between the user and the system can result in enhanced, efficient, and enjoyable analysis. A recent survey of existing guidance approaches in visualization literature [48] revealed that (1) orienting is the most common degree of guidance followed by directing and then prescribing, and (2) the total number of approaches providing multiple degrees of guidance is very small, and no approach provides all three guidance degrees. So I ask:

1. *How can we design adaptive UI elements that provide different degrees of guidance?*

2. *What if the user is interested in a certain type of guidance (e.g., directing guidance on filters)? How then can the user specify such* ***guidance preferences*** *to the UI? In response, how then can the system adapt its guidance strategy?*

Zhou et al.'s [50] survey paper proposed a four-dimensional design space to characterize how visualization systems recommend content to users during visual data analysis. These dimensions include "Directness" (where content is surfaced relative to user interaction, either in situ or ex situ), "Forcefulness" (the intrusiveness level of recommendations), "Stability" (timing of content updates, such as periodic or event-driven), and "Granularity" (the

atomic unit of recommended content, like a document or dataset entity). This framework aids in designing and evaluating guidance within interactive data visualizations. In this chapter, we propose a complementary design space, that enables a *co-adaptive* guidance dialog between the user and the system, along with a playground system to demonstrate the various configurations and combinations of the design space, as described next.

8.2 Design Space for Communicating Guidance

In this section, we describe our proposed design space for guidance communication during visual data analysis. We will first define new concepts and terminologies and then explain each aspect of the design space along with relevant usage scenarios.

1. **Wildcard.** We define a "wildcard" as a *special* attribute that can be assigned to any attribute or record of a tabular dataset. Each wildcard holds values about an analytic entity – such as user's focus (akin provenance attributes [11], described in chapter 7), data quality and usage (akin DataPilot, described in chapter 3), and so on. We utilize these wildcards to model and subsequently present guidance in the UI.

2. **(Wildcard) State.** We associate each wildcard with a contextual temporal perspective called "state" which can take four values: Past, Present, Problem, and Future, covering all relevant aspects of analysis. Each state offers a unique perspective: Past reflects the previous analysis state of the wildcard; Present represents the current state; Problem represents the issue with the current state, or the 'knowledge gap' between the user and the system; and Future represents the system-suggested next steps for the user that overcome the Problem. These states are inspired by Engels' characterization of guidance, in particular the "what" dimension that (also) defines the problem which is decomposed into an "initial state" at the start of analysis and a "goal state" that must be reached [123].

3. **Level.** We define a "level" as the amount of guidance to provision during analysis. It can take five values: Level 1, Level 2, Level 3, None, Adapt. Level 1 is best represented

as *orienting*-like guidance [35], wherein the system provides users with subtle hints to help them focus on relevant data without directing specific actions. Level 2 is best represented as *directing*-like [35], wherein the system provides users with a fixed number of explicit aspects to focus on or steps to pursue next. Level 3 is best represented as *prescribing*-like [35], wherein the system provides users with exactly one 'best' aspect to focus on or step to pursue next. None implies no guidance. Adapt represents any one of the above guidance levels at any given point in time, aiding dynamic (adaptive) transitions between them; this level can be system-determined or user-configured.

For each wildcard, we structure our design space as a 3x4 matrix that balances the guidance levels (vertical axis) with different wildcard states (horizontal axis), as shown in Figure 8.1. The horizontal axis of the design space captures the different states of guidance provided: **Past** (previous state), **Present** (current state), **Problem** (issues in current state), and **Future** (suggested future actions). This horizontal axis guides users in understanding their previous action, their current actions, potential issues with their current actions, and ideal next steps. The vertical axis introduces three guidance levels that explain the same state in progressively different ways, ranging from **Level 1** (to orient the user by offering visual cues to focus on relevant data), **Level 2** (to direct the user to focus on or consider one of top 'N' entities), **Level 3** (to prescribe a single 'best' entity to interact with next).

We believe that together, *wildcards* and their *states* cover all relevant analysis states necessary to provision guidance, and also form a solid basis to intuitively and seamlessly transition between the analysis states. With *levels*, these wildcards and their states can provision different amounts of the same guidance, depending on the analysis needs and the user's preferences. We next describe each cell in the design space matrix along with a relevant use-case. For ease of explanation, we will describe each wildcard state within a guidance level, before moving to the next level; additionally, we will describe the Present state before Past, even though logically the latter occurs before.

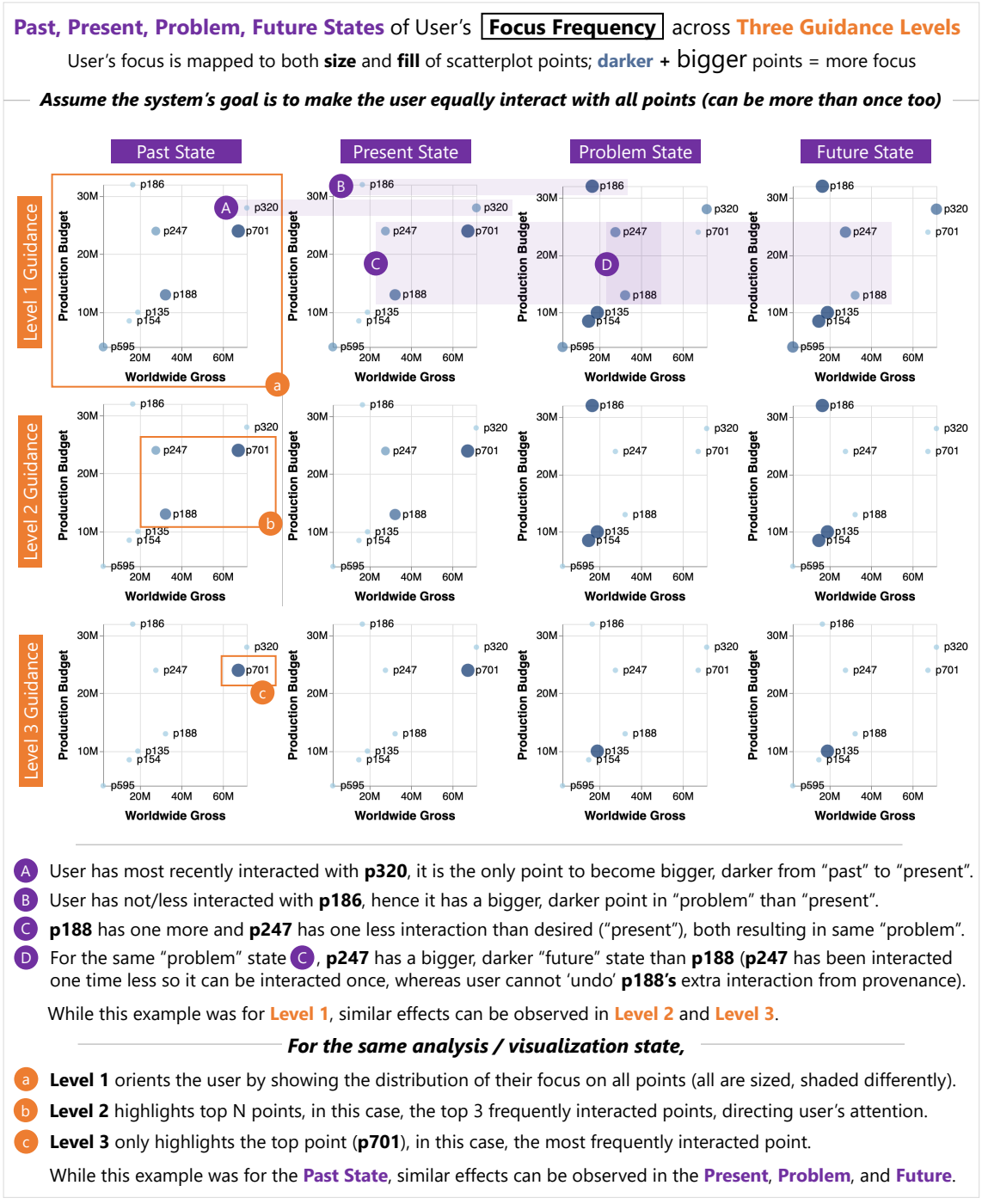


Figure 8.1: Example Application of the proposed design space using a "Focus Frequency" *Wildcard* that tracks the frequency of users' focus on individual datapoints in a scatterplot, as visualized across the four *States* (Past, Present, Problem, Future) and three *Levels* of guidance (Level 1, Level 2, Level 3). Follow the annotations in the figure to understand the state transitions in this case. Additionally, while this example utilizes *fill* and *size* to encode the wildcard, it can also utilize other visual encodings – such as *x*, *y*, *shape* – and data transformations such as *filter* and *sort*, akin ProvenanceLens (chapter 7).

Consider Figure 8.1 which illustrates an example application of this design space using a "Focus Frequency" *Wildcard* that tracks the frequency of users' interactions (focus) on each scatterplot datapoint. Note that, while this example utilizes *fill* and *size* to encode the wildcard, it can also utilize other visual encodings – such as *x*, *y*, *shape* – and data transformations such as *filter* and *sort*, akin ProvenanceLens (chapter 7).

Level 1. This level provides visual cues on the distribution of wildcard values across all datapoints, orienting the user without directing or prescribing next steps, granting complete user agency and control.

Present. This cell reflects the user's current analysis, encoding the frequency of their focus on datapoints using shades of blue and varying sizes, enabling the user to determine their most *frequently* interacted datapoint (by looking for the biggest and darkest datapoint).

Past. This cell reflects the user's previous analysis state, enabling the user to determine their most *recently* interacted datapoint (by finding the 'diff' with the Present state, as shown in Figure 8.1Ⓐ) or, if the system permits, potentially perform an 'undo' operation.

Problem. This cell reflects the problem in the user's present state (or the 'knowledge gap' with the system). Consider the guidance scenario shown in Figure 8.1 wherein the user's goal is to interact with each datapoint equally (not necessarily exactly once). In this case, if a user interacts with one datapoint, then the other uninteracted datapoints automatically have a 'problem'. When this problem is mapped to visual encodings (e.g., *fill*, *size*), darker and bigger points indicate most problematic points (Figure 8.1Ⓑ), nudging users to do something about them (e.g., interact). Note that, depending on the task, there can be different Problem states (and hence, Future states) for the same Present state.

Future. This cell reflects the system recommended future state, or essentially, a kind of *fix* to the Problem state. Note that, in some cases, this Future state *may* be equivalent to the Problem state (because entities with the biggest problems are generally the top choices for immediate future consideration). However, in the guidance scenario illustrated in Figure 8.1, wherein the task is to ensure equal focus on all datapoints, if one datapoint has been interacted exactly one time *more* than the target number of interactions (at that time), and another datapoint has been interacted exactly one time *less* than the target number of interactions, then they both have the same Problem state, as both are off-target by one interaction (Figure 8.1Ⓒ), but different Future states, as the system will prioritize interacting with the less interacted point as it cannot undo the interaction with the more interacted point (Figure 8.1Ⓓ).

Level 2. Compared to Level 1, which highlights all datapoints, Level 2 selectively highlights the top-N datapoints (e.g., most frequently interacted points with high "Focus Frequency"), directing the user to continue focusing on them, while sacrificing some agency and control (as the 'full picture' is no longer available). Alternatively, the system can selectively highlight the bottom-N datapoints (i.e., least frequently interacted points) directing users to focus on or consider them next. Additionally, if multiple datapoints have the same value, one can either highlight all of them (which may exceed 'N') or use a "first come, first served" approach until the desired 'N' is reached, though this may be less objective.

Present. Instead of highlighting all datapoints, the system selectively highlights the top-N most frequently interacted points (high "Focus Frequency"). In doing so, Level 2 guidance provides a quicker, more focused overview of the user's current state, compared to Level 1.

Past. The system selectively highlights the top-N most (or bottom-N least) frequently interacted datapoints until the state before the Present, aiding the user in quickly assessing prior decisions (Figure 8.1b).

Problem. The system highlights the top-N most (or bottom-N least) interacted datapoints, helping the user quickly identify key problematic areas to focus on or (not) and accordingly determine next steps.

Future. The system highlights the bottom-N least frequently interacted datapoints, directing the user to next interact with one of them. Note that in our example, the top-N most frequently interacted datapoints are not highlighted because one cannot 'undo' an 'extra' interaction to meet the desired interaction count (hence only bottom-N are highlighted).

Level 3. Compared to Level 1, which highlights all datapoints and Level 2, which selectively highlights top-N or bottom-N datapoints, Level 3 highlights *exactly one* 'best' datapoint, prescribing the user to focus on it next, entirely sacrificing user agency and control (as information related to other datapoints is unavailable).

Present. The system highlights the most (or least) frequently interacted datapoint thus far. All other datapoints share the same visual properties.

Past. The system highlights the most (or least) frequently interacted datapoint immediately before the Present state.

Problem. The system highlights the most (or least) 'problematic' datapoint in the Present state, nudging the user to assess and find the fix by themselves.

Future. The system highlights its top-choice datapoint to focus on next.

8.3 Lighthouse: A Playground for Demonstrating the Guidance Design Space

We developed a visual data analysis system, Lighthouse, that offers an interactive playground with adaptive guidance-enriched UI elements, that seamlessly transition between

different guidance levels, for each wildcard state. Lighthouse essentially helps users 'experience' the entire guidance design space matrix and is the first VA system of its kind.

8.3.1 User Interface

Figure 8.2 shows the Lighthouse user interface with the following views:

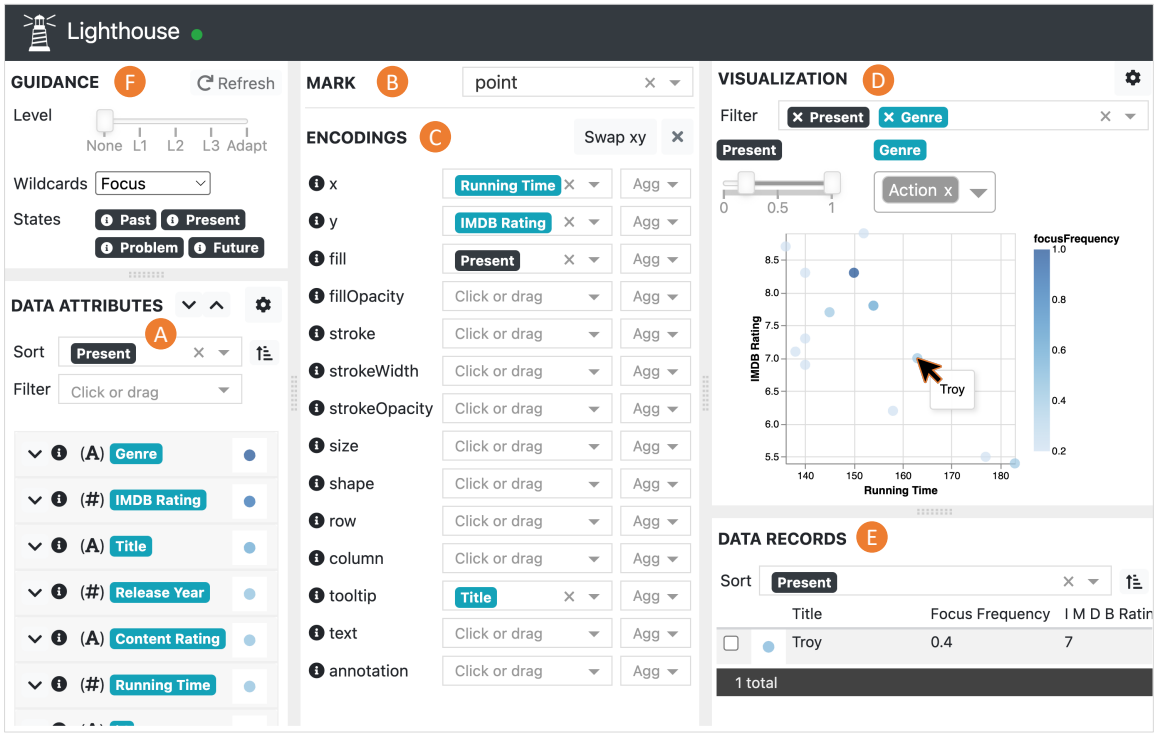


Figure 8.2: The Lighthouse user interface includes traditional visual data analysis functions: **(A) Data Attributes View**, **(B) Marks and Encoding View**, **(C) Visualization Canvas**, **(D) Data Records View**, along with a new **(E) Guidance Panel**.

A Data Attributes. In this view, users upload and configure the dataset (⚙) and see the underlying attributes (or features or columns). Users can sort and filter the attributes using the different *wildcards* (e.g., "Focus") and their *states* (e.g., Present). Hovering on the information icon ℹ shows the attribute's definition in a tooltip. Clicking on the expand icon ⌄ opens a detailed view with a distribution plot of the attribute's values: an area curve for numerical attributes and a column chart for categorical attributes, both of which show percentage counts corresponding to the attribute quantiles and categories, respectively.

B Mark. This view includes a dropdown to configure the mark type for the visualization. Users can select one of *point*, *bar*, *line*, *area*, or *text* to begin a visualization specification.

C Encodings. This view shows the visual encoding channels. Users select or drag one or more data attributes and/or guidance wildcard states to one of *x*, *y*, *fill*, *fillOpacity*, *stroke*, *strokeOpacity*, *strokeWidth*, *shape*, *row*, *column*, *tooltip*, and/or *text* to complete a visualization specification. An additional encoding, *annotation*, adds a new annotation displaying the value of the encoded entity next to the selected mark.

D Visualization. This view renders an interactive visualization based on the selected mark type and activated visual encodings in the "Encodings" view. It also includes a "Filter" drop-zone to filter out datapoints by data attributes and/or wildcard states. A numerical attribute displays a range slider and a categorical attribute displays a multiselect dropdown.

E Data Records. This view shows the data bound in the visualization as a paginated data table. If the user hovers on a datapoint in a unit visualization (e.g., a scatterplot), this table filters to only show that data record whereas if the user hovers on an entity in an aggregate visualization (e.g., a bar showing the *mean* value), this table filters to show all data records belonging to the hovered entity (e.g., the bar).

F Guidance Panel. This view enables users to configure different aspects pertaining to guidance: (1) a slider to adjust the "Level" or the amount of guidance (to one of None, Level 1, Level 2, Level 3, or Adapt); recall that the Adapt mode is a dynamic system-determined guidance level that, as per the intended analysis use-case, can dynamically transition between the other fixed guidance levels; (2) a dropdown to select one or more "Wildcards" (e.g., "Focus" (Frequency), "Data Quality"); and (3) handles for "States" that contain the values corresponding to the wildcard state (Past, Present, Problem, Future), and which can be utilized in the UI as attributes, e.g., mapping to encodings or applying data

transformations. Lighthouse supports multiple wildcards, each with their own states, e.g., one can simultaneously interact with Present + "Data Quality" along with Future + "Focus". For simplicity, we demonstrate the UI with only one wildcard – "Focus" (Frequency).

8.3.2 Guidance Enhancements in the User Interface

Figure 8.3: Attribute Panel showing underlying data distributions ("No" guidance), interaction facts (Level 1), multiple recommendations (Level 2), and one 'best' recommendation (Level 3) for each guidance level, using natural language, visual cues, and calls to action.

Guidance Facts and Recommendations. To address the challenge of determining next steps solely based on visual cues, as also surfaced in the evaluation of Lumos (chapter 5), Lighthouse offers additional guidance via natural language *facts* and *recommendations*. Consider an 'exploration bias' mitigation scenario, wherein the user has exhibited bias against a categorical attribute, e.g., "Content Rating", by overemphasizing "PG-13" movies

(Figure 8.3). Depending on the chosen guidance "Level" – Level 1, Level 2, and Level 3[1] – the system communicates relevant guidance as follows:

Level 1. The system *orients* the user by presenting "interaction facts" – relevant factual information computed from the logs, e.g., *"**86.21%** of your focus has been on values with **Content Rating = [PG-13]**."* Additionally, the user can hover on the "Show me" icon button to visualize the fact in the corresponding distribution plot. Essentially, in addition to the blue area curve (Focus) and the red line (Target), the system uses green rectangle to highlight the region referred to in the fact.

Level 2. In addition to showing Level 1 guidance, the system additionally *directs* the user by suggesting an operation (that will result in multiple next steps to choose from), e.g., *"We suggest you interact **less** with **Content Rating = [PG-13].**"* Additionally, the user can hover on the "Help me" question-mark icon button to preview this suggestion. Upon review, if the user is happy with the suggested operation, they can click the toolbox icon button to execute the operation. The user can then decide which of the remaining datapoints to interact with next.

Level 3. The system *prescribes* a single 'best' next step for the user to perform, e.g., *"Interact with the **R-rated** record **p28**."* Clicking the "Help me" toolbox icon button will apply a filter to show exactly one datapoint (id=28) to interact with next.

Guidance Traces. Lighthouse operationalizes the proposed guidance design space and makes them available in the user interface as "guidance traces" (or visual cues of guidance). Essentially, users can map one or more wildcard "States" to visual encodings and/or apply data transformations (e.g., filter), to visualize and interact with the guidance scents in the Visualization Canvas as well as the glyphs in the Data Attributes and Data Records views, similar to interacting with *provenance attributes* in ProvenanceLens (chapter 7).

[1]The fourth Adapt level represents one of three guidance levels at any given point in time.

Figure 8.2 shows an example scenario wherein the user is interacting with points in a scatterplot of "Running Time" and "IMDB Rating". They have (1) mapped the Present state of the "Focus Frequency" wildcard (frequency of the user's interactions with individual datapoints and attributes) to the *fill* visual encoding, visualizing their present distribution of focus via different shades of blue; (2) filtered out points in the visualization that have Present focus under 0.25 units; and (3) sorted data attributes in the **Data Attributes View** as well as data records in the **Data Records View** based on their Present focus.

Similarly, Figure 8.1 shows an example scenario wherein the user is interacting with points in a scatterplot of "Worldwide Gross" and "Production Budget", and reviewing the Past, Present, Problem, and Future *states* of the "Focus Frequency" *wildcard*, (double) mapped to the fill color and size visual encodings, across all three guidance *levels*.

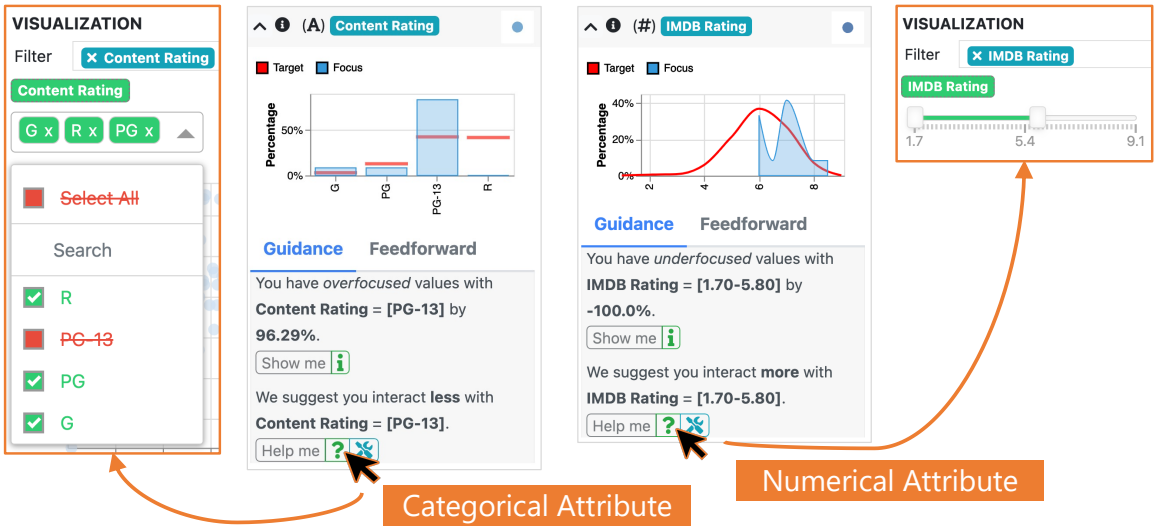


Figure 8.4: **GuidanceWidgets** – enhanced UI controls that guide users about the next operation(s) to perform. Multiselect dropdowns for categorical and range sliders for numerical attributes overlay which options/range to select (in green) or remove (~~red strikethroughs~~).

GuidanceWidgets. Next, Lighthouse introduces GuidanceWidgets – enhanced UI controls that guide users about, e.g., what operation(s) to perform next. Consider an 'exploration bias' mitigation scenario, wherein the user has exhibited bias against a categorical attribute, e.g., "Content Rating", by overemphasizing "PG-13" movies (Figure 8.4). To

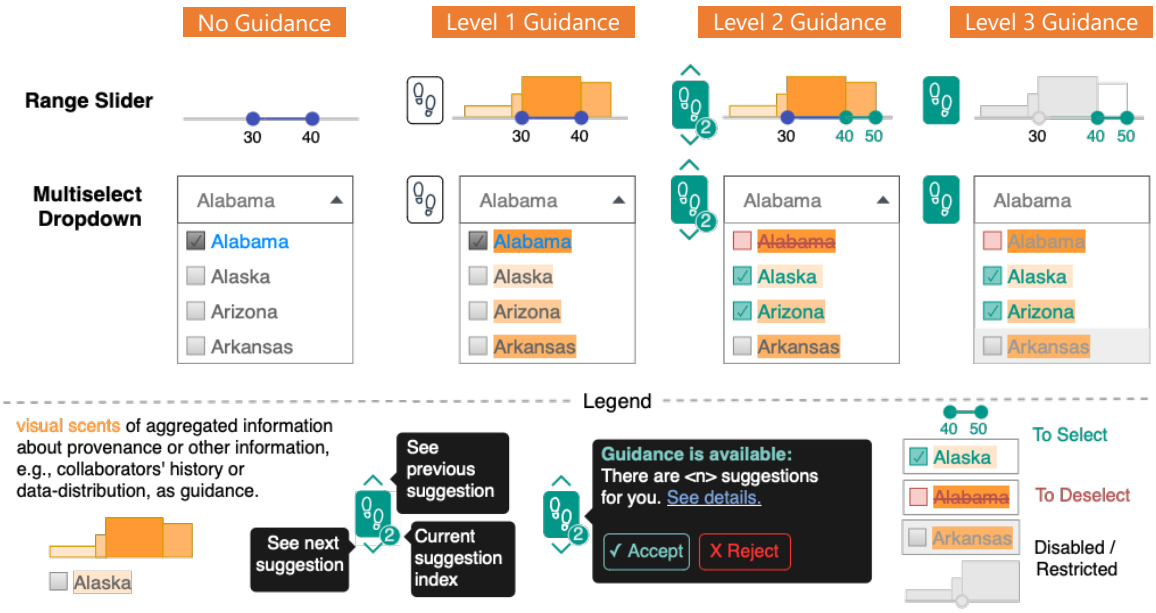


Figure 8.5: **GuidanceWidgets** providing guidance across different *level*s of detail.

mitigate this bias, the system recommends interacting less with "PG-13" movies by applying a 'reverse' filter. The system communicates this via a natural language explanation in the **Data Attributes** view (e.g., "We suggest you interact **less** with **Content Rating = [PG-13]**") to which the user can respond via the "Help me" icon button. Specifically, if the user hovers on the question-mark icon, the system will offer a visual preview of the operation to perform next (e.g., filter out "PG-13" movies). This preview is shown via an enhanced multiselect dropdown wherein green options will be retained, and ~~red and struckthrough~~ options will be filtered out. For numerical attributes (e.g., "IMDB Rating"), the system similarly communicates the filter operation by highlighting the suggested range in a range slider. Additionally, these widgets can overlay visual scents, akin Scented Widgets [85], corresponding to the "Past", "Present", "Problem", and "Future". Lastly, the widgets can also *disable* options/ranges (preventing the user from selecting them) or reorder options based on some computed preference order (as an example of directing guidance). If the user is satisfied with the preview, they can click the toolbox icon, and the system will execute the corresponding operation(s). Figure 8.5 illustrates how these widgets can adapt

to different guidance levels. Through these enhanced UI controls, users receive contextual guidance in situ, and which also ensures a fluid transition through the wildcard states [172].

8.4 Summary

In this chapter, I described a design space for communicating guidance, represented as a 3x4 matrix comprising three guidance *levels* and four *wildcard states*. Guidance "levels" refers to the amount of guidance to provide; inspired by Ceneda et al.'s [35] characterization of guidance degrees, we model five guidance levels: level 1 (orienting), level 2 (directing), level 3 (prescribing), none, and adapt (any of the above). "Wildcards" are special attributes that can be assigned to any attribute or record of a tabular dataset – such as frequency of user's interactions, data quality, and so on. Next, we associate each wildcard with a contextual temporal perspective called "state", which takes four values: past, present, problem, or future. This characterization allows users to be guided about their previous interactions, their current state, potential issues with their current state, and ideal next steps.

Additionally, these wildcards and wildcard states can be utilized as attributes during analysis by mapping them to visual encodings and data transformations (akin the recency and frequency provenance attributes described in chapter 7). For example, consider a user (1) maps their *present* (state) frequency of focus on data points (wildcard) to the color encoding channel and (2) sets the guidance level to *Level 1*. In the resultant visual configuration, they will observe darker points as those that have been focused more frequently than others. Instead, if the user maps the wildcard to the *future* state and chooses *Level 3* guidance, they will observe exactly one dark point that they must interact with next.

We also introduce a visual data analysis system, Lighthouse, that serves as a playground to demonstrate and study the introduced design space. This system utilizes visual cues and NL explanations to communicate different levels of the same guidance via adaptive UI controls. We demonstrate the effectiveness of this system through a series of usage scenarios. For details, I refer to the associated publication [12] (under review) and patent [21].

CHAPTER 9
DEMOCRATIZING GUIDANCE FOR VISUAL ANALYTICS

In this chapter, I describe a library of enhanced user interface (UI) controls, such as sliders and dropdowns, that tracks and dynamically overlays analytic provenance. By showing the user what they have done so far, these widgets can make the user reflect upon their present choices to influence subsequent ones. Additionally, if these widgets are preconfigured to show customized information (e.g., interaction behavior of peers), they can be used to nudge users in specific directions (e.g., interact with previously overlooked aspects). Next, because provenance is often a basis for providing guidance, the provenance-tracking ability of the library can be used to prototype guidance systems. Lastly, this library is open-source, enabling developers to build custom provenance and guidance systems, achieving **RG4**: *Create tools to help developers build custom guidance-enriched systems*. This chapter is based on work published at IEEE VIS 2024 [9].

9.1 Motivation and Background

Analytic provenance is the documented history of data and analytical actions, showing how data was obtained, transformed, and analyzed. In a visualization context, analytic provenance tracks how users interact with visualizations as a representation of their reasoning process [52], which can be helpful for recalling the analysis process, reproducing it, collaborating, and logging for evaluation or meta-analysis [53]. Presenting provenance during analysis has been shown to increase awareness of analytic behaviors [27, 7], increase confidence [86], mitigate biases [25], and result in more unique insights [84, 85].

Prior work has made strides in logging frameworks that help developers capture and store provenance [102, 103, 104, 105]. For example, Trrack [105] is an open-source library to create and track the provenance (history) of interactions in web applications for various

purposes including action recovery, reproducibility, collaboration, and logging; TrrackVis complements Trrack via a customizable provenance visualization front-end [105].

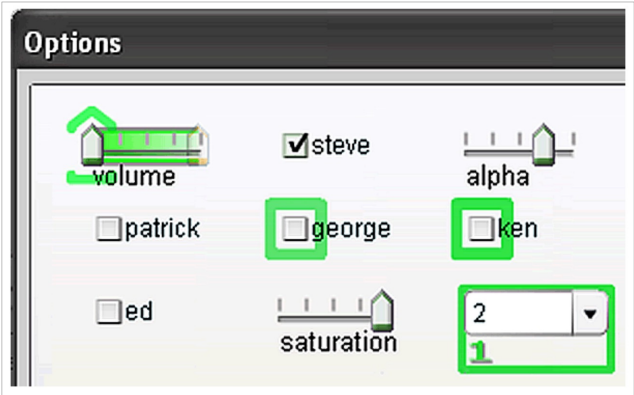


Figure 9.1: Phosphor Objects [266] instantly show and explain state transitions in GUI controls. The slider labeled "volume" was dragged to the left, the two checkboxes corresponding to "george" and "ken" were unchecked, and the combo box was set from 1 to 2.

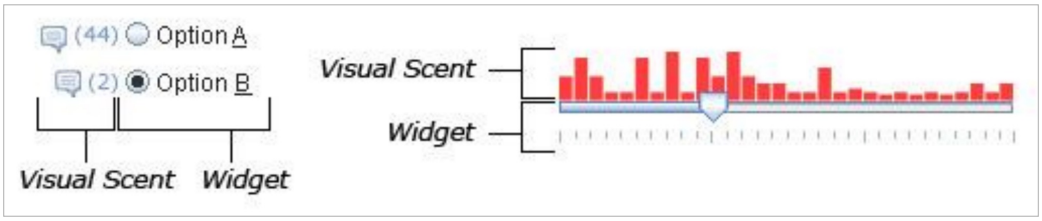


Figure 9.2: Scented Widgets [85] enhance GUI controls with embedded visualizations that facilitate navigation in information spaces. The radio buttons on the left illustrate the number of comments on the two options, whereas the slider on the right is embedded with a histogram showing the distribution of some bound data values.

While logging and analyzing provenance after analysis has immense value, there is a need for libraries that aid developers in integrating provenance directly into visual analytic tools (as opposed to separate tools) in a manner that is consistent with common UI standards. There exist many open-source libraries of UI controls [268] that enhance user interaction and facilitate data input in software applications or websites. By utilizing these libraries, other developers can expedite their development process while ensuring consistency and accessibility across various platforms and devices.

Visualization and HCI researchers have also developed several enhanced UI control libraries. For instance, Phosphor objects [266] instantly show and explain state transi-

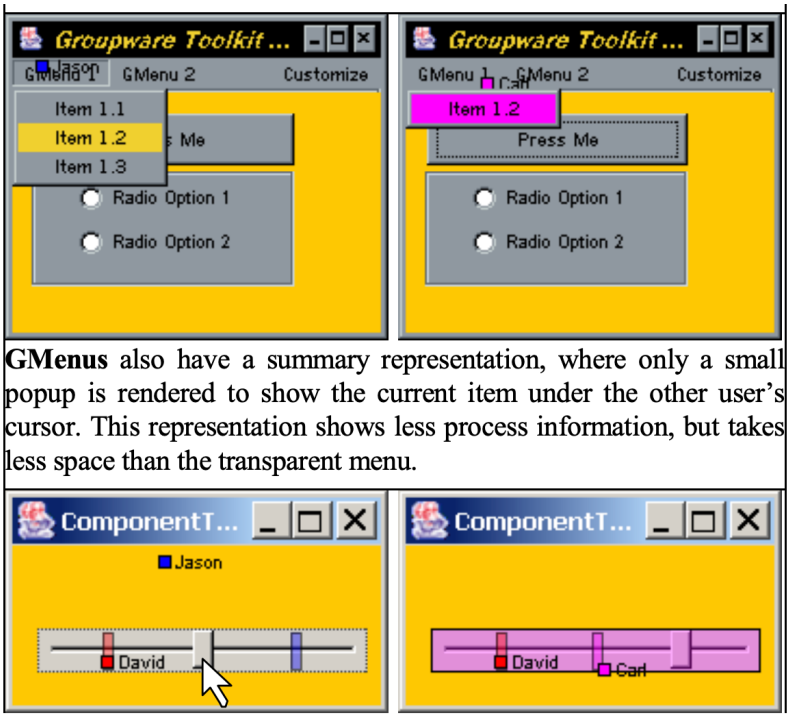


Figure 9.3: Groupware Widget Toolkit [267] with UI components for collecting, distributing, and visualizing group awareness information. Three users: David, Carl, and Jason are simultaneously interacting with the menu items under the 'GMenu', and the single slider.

tions in GUI controls, e.g., manipulating a phosphor slider leaves an afterglow that illustrates how the knob moved (Figure 9.1). Scented Widgets [85] are enhanced GUI controls with embedded visualizations that facilitate navigation in information spaces (Figure 9.2). Groupware Widget Toolkit [267] is a Java toolkit with a broad suite of enhanced UI components for collecting, distributing, and visualizing group awareness information (Figure 9.3). Emotion scents [269] tracks users' emotional reactions while interacting with GUI widgets and visualizes these reactions on the widgets, enhancing the interface for emotional awareness and decision support. DynaVis [270] synthesizes persistent UI widgets in response to an initial natural language (NL)-based visualization editing task, enabling the user to make subsequent modifications by directly interacting with the widgets (instead of re-typing NL).

However, there is no frontend library of UI controls that tracks and presents provenance information *out of the box* during analysis. Tools like TrrackVis (Trrack's [105] frontend library) visualize the logged provenance information graph; however, these visualizations

are available in a separate view/tool, sometimes after analysis. So we asked: *how can developers integrate provenance directly into the user interface of visual data analysis tools?* In response, we built **ProvenanceWidgets**, as described next.

9.2 ProvenanceWidgets

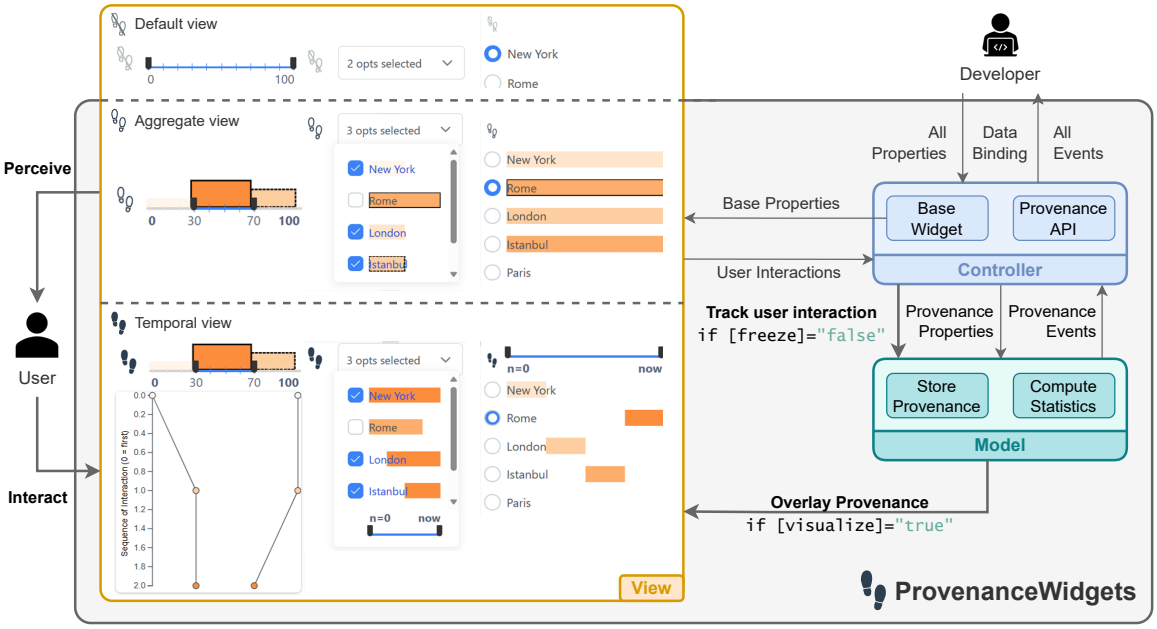


Figure 9.4: Overview of ProvenanceWidgets and the underlying Model-View-Controller-based architecture. The Model stores, computes, and updates the provenance. The View shows how end-users perceive and interact with the widgets. The Controller describes how the Model, View, and developers can interact with ProvenanceWidgets.

ProvenanceWidgets is a JavaScript library of UI control elements that track and dynamically overlay a user's analytic provenance, *out of the box*. We enhanced radio buttons ⦿ ○, checkboxes ☐ ☑, single sliders ⊶, range sliders ⊶⊶, dropdowns ⟶ ⌄, multiselects ☰ ⌄, and input text fields I 🔍 to track how often (frequency) and when (recency) a user interacts with them (e.g., selecting a dropdown option) and present visual overlays showing an aggregated summary as well as a detailed temporal history.

The aggregated summary is presented in a bar chart overlay visualization encoding the frequency (length) and recency (color) of user interactions with the widget. The detailed

temporal history is presented as a timeline visualization, enabling users to access granular information about specific interactions in the past. Below we list our design goals, describe our design process, and architecture and implementation of the library.

9.2.1 Design Goals

We derived seven design goals based on prior provenance visualization tools [105, 85, 266] and our own assessment of the capabilities we aim to support. Our overarching design goal was to *consistently* achieve the underlying goals for all widgets.

- **G1 Log User Interactions on UI controls as provenance.** The library should automatically track relevant user interactions with the UI controls as provenance (e.g., dragging a slider handle or selecting a dropdown option).
- **G2 Compute Aggregated Metrics about Recency and Frequency of Provenance.** The library should process the logged user interactions and compute aggregate summary metrics pertaining to interaction recency and frequency.
- **G3 Dynamically Overlay Provenance on UI controls.** The library should enhance UI controls with a visual overlay of the aggregate summary metrics and an on-demand temporal evolution of the users' analytic provenance.
- **G4 Support Action Recovery.** The library should allow navigating historical analysis states and also updating the current state, both programmatically and by interacting with the provenance visualization overlays.
- **G5 Allow Developer Agency.** Application developers should have the flexibility to tune the default tracking and visualization behavior, including being able to disable it completely. The library should provide an API for the same.
- **G6 Be Framework-Agnostic.** With multiple existing web frameworks (e.g., Angular [175], React [228]), our goal was to make the library integrable into any codebase.
- **G7 Support Meta-Analysis.** The library should support logging and exporting provenance information in a format that is suitable for different kinds of analysis. For

example, ProvenanceWidgets' internal data structure maintains fine-grained logs as well as higher-level computed aggregates.

9.2.2 Design Process

As part of our design process, we first reviewed UI controls and then conducted design exercises to decide efficient visual overlays and associated interactions across all of them.

9.2.2.1 UI Controls Review

To establish a clear understanding of UI controls, we first reviewed their structure, layout, and initial/default values, subsequent values, and associated interaction events.

Structure. All aspects of radio buttons, checkboxes, and (range) sliders are completely visible at all times; whereas, dropdowns, multiselects, and input text fields require an additional click and potential scrolling to bring certain aspects (e.g., options) into focus.

Layouts. Dropdowns, multiselects, and input text fields are oriented *horizontally* with their menus opening vertically (above or below depending on screen position); whereas, radio buttons, checkboxes, and (range) sliders can be oriented *vertically or horizontally*.

Initial/Default Values. Radio buttons, checkboxes, dropdowns, multiselects, and input text fields, can have an uninitialized state with *no (null, empty)* selection(s) or value(s); whereas, (range) sliders must always have *at least one* selection by default.

Subsequent Values. Radio buttons, dropdowns, input text fields, and (range) sliders can have at most *one* selected value; unlike multiselects and checkboxes who can have *multiple*.

Interaction Events. Radio buttons, checkboxes, dropdowns, and multiselects require the user to *click* to (de)select target options. (Range) sliders require the user to *drag* the han-

dle(s) to or directly *click* on the rail at the target value(s). Input text fields require the user to first *type* and then *press the 'Enter' key on keyboards* to mark the typing as complete.

9.2.2.2 Design Exercises and Considerations

Next, we conducted design exercises to explore considerations related to **what** provenance information to show, **where**, **how**, and **when**.

What metrics to log as provenance. We reviewed existing logging frameworks and provenance tools and selected two metrics: frequency and recency of user interactions (**G2**). We chose these metrics for their relevance to provenance tracking, intuitive comprehension, effective visual encoding, and broad applicability across various domains. Furthermore, these metrics can help derive composite metrics such as durations of different widget states and study interaction patterns within and across widgets.

Where to present provenance. We explored on-demand versus always visible visualizations and considered whether they should be juxtaposed against each other, or overlaid or superimposed on the widgets. Then, we discussed the trade-offs of having separate overlays against pushing surrounding elements away to accommodate the visual provenance scents. Inspired by Shneiderman's Mantra [189], we eventually chose to overlay aggregate views in-place (overview) and temporal views separately on demand considering the level of detail in raw interaction data. We designed a tri-state button that would let us toggle between this different views - default, aggregate, and temporal.

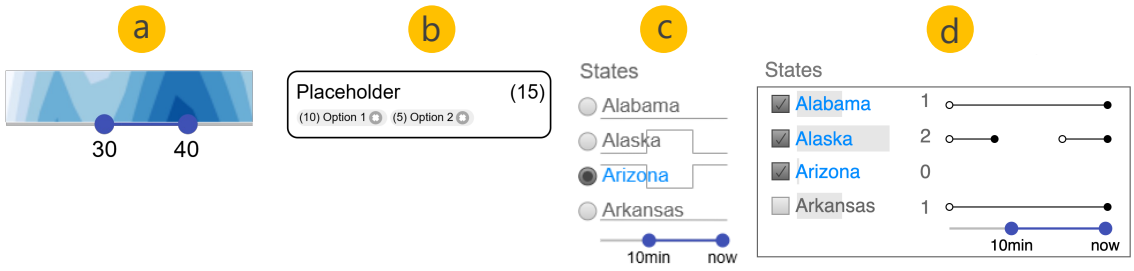


Figure 9.5: Alternate designs: range slider, input text, radio button, checkbox.

How to present provenance. We explored candidate visualization and interaction techniques to overlay and interact with the logged provenance information. We sketched ideas on draw.io [271] and iterated among co-authors over multiple brainstorming sessions. These low-fidelity sketches included considerations for chart types (e.g., bar charts, line charts, and horizon charts), visual encodings (e.g., color, opacity, size), and UI layouts (e.g., panels, overlays). Keeping in mind our overarching goal of ensuring *consistency*, we selected the *bar* mark and *size, color* encodings to encode frequency and recency information (Figure 9.7). Figure 9.5 shows some of our design considerations for sliders, input texts, radio buttons, and checkboxes. For example, we sketched horizon charts in range sliders (Figure 9.5(a)) and chips in input texts (b), but did not implement them because they did not generalize across all widgets. Similarly, stepped line charts in the temporal view (c) seemed occluding and harder to interact with.

When to log provenance. For each widget, we chose to log and perform provenance computations on each interaction event that modifies its *value (or state)*; an event that does not modify a widget's value, such as *mouseover* or *keyup* is not logged. In addition, clicking a historical analytic state in the visualization overlays of ProvenanceWidgets is also considered a new interaction and is also appended to the widgets' provenance.

Additionally, we considered two kinds of logging frequencies – *interaction-based*, capturing every user interaction when it occurs, and *time-based*, capturing snapshots at specific intervals. Finding utility in both, we chose to support both (**G1**, Figure 9.6).

9.2.3 Chosen Designs

Figure 9.7 shows our chosen designs; the base design (default view) of each widget is enhanced by the **Aggregate View** (summary) and a **Temporal View** (detailed history) (**G3**).

When the widget has not been interacted with (i.e., there is no logged provenance), a disabled footprint icon-button is placed next to the UI control. When the widget is

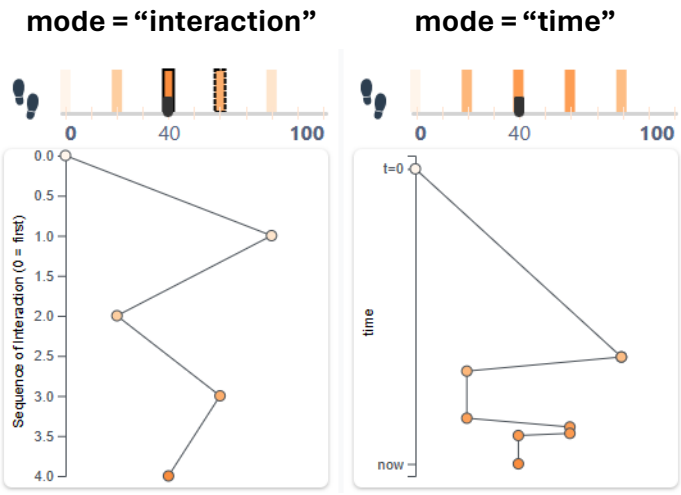


Figure 9.6: ProvenanceWidgets: `[mode]="interaction"` and `[mode]="time"` log interactions every interaction and 1 second (by default), respectively.

interacted with for the first time, this icon-button is enabled, and the widget switches to the Aggregate view, which overlays aggregate provenance information. Clicking the footprint icon toggles between this Aggregate view and the Temporal view, that overlays the temporal history of provenance.

9.2.3.1 Single Slider, Range Slider

Aggregate View. We chose a bar chart overlay that shows previously selected values (slider) or ranges of values (range slider). The frequency of a selection is encoded by height, and the recency of a selection is encoded by color. Taller, darker bars indicate more frequent and recent interactions, respectively. This bar chart is positioned directly above the slider, as in Scented Widgets [85]. Hovering a bar shows a tooltip with contextual information. Clicking a bar updates the slider to the selected value or range.

Temporal View. To visualize the temporal evolution, we chose a popover that is overlaid above or below the slider. Within this popover, we designed a line chart where time is measured along the y-axis, and the slider itself serves as the x-axis. This line chart has one line for a single slider and two lines for a range slider (one for each handle). The line(s)

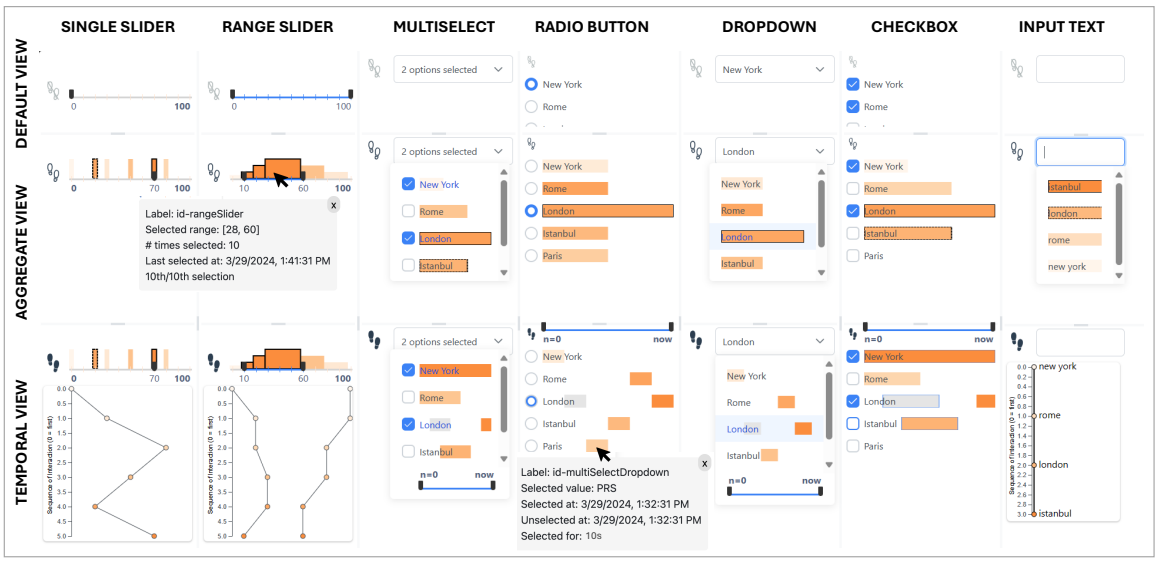


Figure 9.7: ProvenanceWidgets: UI controls (single slider, range slider, multiselect, radio button, dropdown, checkbox, and input text) enhanced with an aggregate summary (Aggregate View) as well as a detailed temporal history (Temporal View) of analytic provenance.

have circular points that represent the exact selections made over time. These points are also colored by the recency of the selections. Hovering a point shows a tooltip with the corresponding value and time of the selection. In addition, clicking a point updates the slider to the selected value or range (**G4**). Lastly, to facilitate navigation, the y-axis can also be brushed to zoom in on more granular, specific time ranges.

9.2.3.2 *Dropdown*, *Multiselect*, *Radio Button*, *Checkbox*

We refer to dropdowns, multiselects, radio buttons, and checkboxes as *selection-type* widgets due to their similar design for visualizing and interacting with provenance information.

Aggregate View. We designed a bar chart and placed it under the options list. An option's selection frequency is encoded by the length of the bar underneath it and the recency is encoded by color. Longer and darker bars indicate higher frequency, recency, respectively. Hovering a bar shows a tooltip with the value, timestamp, frequency, and recency of the selection. Clicking a bar updates the option's selection.

Temporal View. To visualize the temporal evolution, we directly modified the aggregate bar chart unlike that in sliders, where we created a new popover. Each bar represents the time range during which the option was selected. The length of the bar represents the duration of the selection, and the color represents the recency. Longer, darker bars indicate higher frequency, recency, respectively. Hovering a bar shows a tooltip with the corresponding time range. Clicking a bar selects the corresponding option, along with other options that were selected at that point in time. Lastly, to facilitate navigation, there is a horizontal range slider to zoom in on more granular, specific time ranges.

9.2.3.3 Input Text

Aggregate View. We utilized a dropdown list of previously entered values and visualized provenance as a bar chart underneath each list item. The frequency of an input value is encoded by the length of the bar underneath it, and the recency of a selection is encoded by color. Longer, darker bars indicate higher frequency, recency, respectively. Hovering a list item shows a tooltip with the corresponding timestamp, frequency, and recency of the input value. Clicking a list item updates the text input selection to the corresponding value.

Temporal View. To visualize the temporal evolution, we designed an overlay popover above or below the text input. Within this popover, we designed a vertical timeline chart that shows what text input was searched and when. This timeline has circular points that represent the exact search inputs made over time. These points are also colored by the recency of the input searches. Hovering a point shows a tooltip with contextual information. Clicking a point updates the current text input selection to the corresponding value.

9.2.4 Architecture

We define the architecture of ProvenanceWidgets using **MVC** (Model-View-Controller), a software design pattern commonly used to develop GUIs (Figure 9.4), described below.

View: *What the user interacts with* - The View handles all concerns related to the appearance of the widgets, including the base widgets and the overlaid provenance. Internally, we define it almost entirely with Angular templates (HTML) and CSS.

Controller: *What the developer interacts with* - The Controller serves as a hub between the developer, the View, and the Model. Essentially, it wraps the base widget and exposes all of its properties and events, in addition to the ProvenanceWidgets API. It passes on all the base widgets' properties to the View templates, and intercepts all incoming events before re-emitting them for the developers. If not frozen, it relays these events and all provenance-related properties to the Model.

Model: *What we interact with* - The Model stores the raw interaction data received from the Controller, and uses it to compute frequency (how many times a value was input) and recency (how recently a value was input). Once the provenance is updated, the Model can emit it via the Controller as an event for the developers to subscribe to. Then, if visualization is enabled, it updates the View with aggregated summaries of frequency and recency (Aggregate View) or raw temporal history (Temporal View) depending on the active mode.

9.2.5 Implementation

ProvenanceWidgets is implemented using Angular [175] with an extensible API to support flexibility across different systems. To ensure portability across frameworks (e.g., React [228]), we leverage the WebComponents API (**G6**). Below is an overview of the ProvenanceWidgets API with detailed descriptions about the underlying properties (attributes) and events, followed by how each widget can be implemented in applications.

```
provenance?: SliderProvenance|InputTextProvenance|Provenance
mode?: "interaction"|"time"
```

```
<provenance-{slider,dropdown,multiselect,radiobutton,checkbox,
    inputtext}
```

```
    [(provenance)]="provenance"
    (provenanceChange)="function($event)"
    [mode]="mode"
    [freeze]="false"
    [visualize]="true"
    [attr.data-label]="'label'"
/>
```

1. **provenance**: Information about users' interaction history that is recorded and computed by the widget. While each widget has a unique provenance structure, they all record an array of objects with the selected/input value and a timestamp. This property can be used to initialize, restore, modify, and export (**G7**) the provenance of a widget.
2. **provenanceChange**: An event that is triggered whenever the user interacts with the widget such that its *value* (and hence provenance) changes. For example, *clicking* a radiobutton option or *dragging* a range slider handle constitute a valid event; however, *keyup* or *mouseover* events do not contribute to the provenance.
3. **mode**: This property configures the provenance logging frequency (Figure 9.6). When 'mode' is set to *"interaction"*, the widget logs every user interaction and accordingly recomputes provenance metrics and updates the subsequent visualizations. When 'mode' is set to *"time"*, the widget logs interactions every 't' seconds (t=1 second by default) and accordingly updates everything downstream.
4. **freeze**: A property to stop logging interactions with the widget. When 'freeze' is set to true, the widget will not record any new interactions, and existing visualizations will not be updated. When 'freeze' is set to false, the widget will continue recording and visualizing the provenance from the last recorded interaction (**G4**).
5. **visualize**: A boolean property to toggle the visibility of the provenance overlays. This property can be used (with '**freeze**'=true) to completely disable provenance (**G4, G5**).
6. **data-label**: An attribute to pass additional context (e.g., display a "label" in the tooltip).

Understanding code snippets and notations. We describe the ProvenanceWidgets API using TypeScript and Angular's data binding syntax, categorized based on data flow:

1. *From source to view (property binding).* `[property]="expression"` binds the value from the expression to the property. Can be also used to bind class and style properties, and `data-*` attributes.
2. *From view to source (event binding).* `(event)="function($event)"` executes the bound `function` with the `$event` object emitted by the `event`.
3. *In both ways (two-way binding).* `[(property)]="expression"` binds the value from the expression to the property, and vice versa. It is syntactic sugar for combining property and event binding. For example, `[(provenance)]` is syntactic sugar for `[provenance]` and `(provenanceChange)`.

9.2.5.1 Single Slider, Range Slider

A slider allows users to select a numeric value from a given range. Traditionally defined as <input type="range"> in HTML, these elements only allow for a single value to be selected. We also support Range Sliders, which permit selection of a range of values.

```
value: number = 0
highValue?: number = 0 // Omit for Single Slider
handleChange(event: ChangeContext) {
    value = event.value
    highValue = event.highValue
}
options: Options = { floor: 0, ceil: 100, step: 1 }
```

```
<provenance-slider
    [options]="options"
    [value]="value"
    [highValue]="highValue"
    (selectedChange)="handleChange($event)" />
```

A ProvenanceWidgets Slider extends @angular-slider/ngx-slider's SliderComponent and exposes an additional selectedChange event which is triggered with each interaction.

9.2.5.2 *Text Input*

A Text Input allows users to enter text, numbers, and symbols. Traditionally defined as <input type="text"> in HTML, these elements create a single-line text input field.

```
value: string = ''
```

```
<provenance-inputtext [(value)]="value" />
```

A ProvenanceWidgets Text Input extends PrimeNG's AutoComplete component and exposes an additional valueChange event which is triggered when the input value changes.

9.2.5.3 *Dropdown*

A Dropdown allows users to select a single value from a list of options. In HTML, these elements are defined using the <select> tag and a list of <option> tags nested within it.

```
type Option = { label: string, value: string }
options: Option[] = []
selected?: Option
```

```
<provenance-dropdown
    [options]="options"
    optionLabel="label"
    dataKey="value"
    [(selected)]="selected"
/>
```

A ProvenanceWidgets Dropdown extends PrimeNG's Dropdown component and exposes its options attribute to allow developers to provide their list of options.

9.2.5.4 *Multiselect*

A Multiselect input allows users to select multiple values from a list of options. In HTML, these are defined in the same way as Dropdowns (<select>), but with the multiple attribute set to true. However, unlike Dropdowns, a Multiselect input renders a list of options and requires the user to hold down the control key while clicking to select multiple options.

```
selected?: Option[]
```

```
<provenance-multiselect
    [options]="options"
    optionLabel="label"
    dataKey="value"
    [(selected)]="selected"
/>
```

A ProvenanceWidgets Multiselect extends PrimeNG's MultiSelect component and exposes its options attribute to allow developers to provide their list of options. Unlike a traditional multiselect input, this widget renders in a Dropdown-like manner and does not require users to hold down any keys to select multiple options.

9.2.5.5 *Radio Button*

A Radiobutton allows users to select a single value from a list of options. In HTML, these are defined using the <input type="radio"> tag, and all radio buttons with the same name attribute are grouped together.

```
selected?: string
```

```
<provenance-radiobutton
    [data]="options"
    [(selected)]="selected"
/>
```

A ProvenanceWidgets Radio Button extends PrimeNG's RadioButton component. However, unlike traditional Radio Buttons, this widget represents a group of vertically aligned self-contained radio buttons. It exposes a data attribute, which allows developers to provide their list of options instead of having to define each radio button individually.

9.2.5.6 *Checkbox* ☐☑

A Checkbox allows users to select or deselect a single value. Checkboxes can be standalone, or grouped together with the same name attribute. In HTML, these are defined using the <input type="checkbox"> tag.

```
selected ?: string[]
```

```
<provenance-checkbox
    [data]="options"
    [(selected)]="selected"
/>
```

A ProvenanceWidgets Checkbox widget extends PrimeNG's Checkbox component. Like the Radio Button widget, this widget exposes a data attribute, which allows developers to provide their list of options. All *selection-type* widgets expose a selected attribute, that allows developers to provide an initial selection or override the current selection, and a selectedChange event, triggered when the selection changes.

9.2.6 Example Usage Scenarios

Track Interactions from a Widget to a Visualization. ProvenanceWidgets can help users track what charts they make (visualization specification) and what filters they apply (data transformations). Consider Figure 9.8 that shows a scatterplot visualization of two attributes: "Year" and "Life Expectancy" along with corresponding single slider and range slider ProvenanceWidgets. As the user drags the slider handle(s): "Year": 1970 → 1990 and "Life Expectancy": [40, 80] → [70.2, 80], the scatterplot updates and also the orange

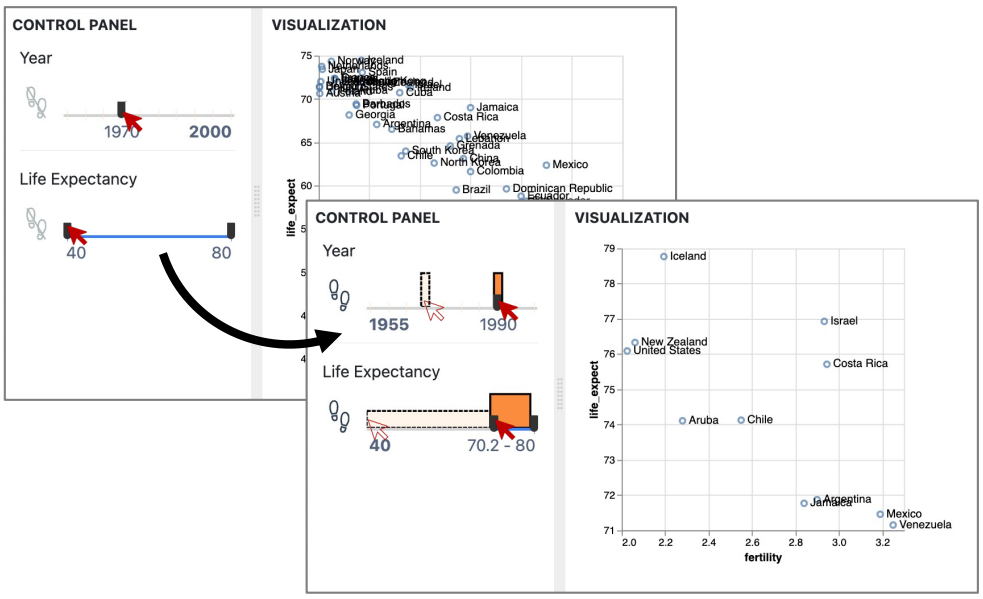


Figure 9.8: Using ProvenanceWidgets, facilitate and also visualize interactions that specify or transform a visualization.

provenance overlays become visible. In this way, the user can utilize ProvenanceWidgets to track already explored data ranges, potentially informing subsequent explorations.

```
const { view } = await embed("spec.vg.json", ...)
const slider= document.createElement("web-provenance-slider");
slider.value = 0;
slider.addEventListener("selectedChange", e => {
    view.signal("slider", e.detail.value).runAsync()
})
```

In the above listing, the developer consumes ProvenanceWidgets as Web Components and binds properties and events in JavaScript. They subscribe to `"selectedChange"` to update the embedded Vega chart [260].

Track Interactions from a Visualization to a Widget. Because not all user interactions happen via UI controls, ProvenanceWidgets can be externally updated when user interactions happen elsewhere, e.g., in the visualization. Consider Figure 9.9 that shows a scatterplot visualization of two attributes and corresponding ProvenanceWidgets range sliders for

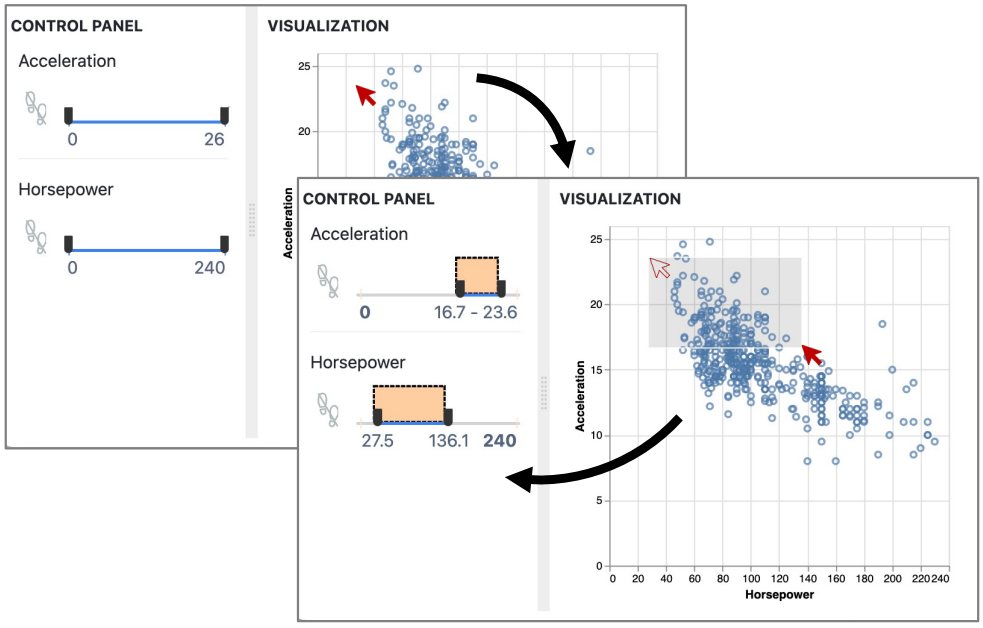


Figure 9.9: Track and visualize interactions that occur within a visualization (e.g., brushing) directly via ProvenanceWidgets.

"Acceleration" and "Horsepower". As the user performs a brush interaction in the visualization, selecting a subset of points within a specific range ("Horsepower": [27.5, 136.1] and "Acceleration": [16.7, 23.6]), the corresponding range sliders can update to show this range. In this way, the user can utilize ProvenanceWidgets to track what data ranges they have already explored, potentially informing subsequent explorations.

```
visBrushed(brush_extent) {
    acc.provenance["data"] = [{ ..., "value": brush_extent }]
        // [16.7. 23.6];
    acc.provenance["revalidate"] = true;
}
```

```
<provenance-slider [(provenance)]="acc.provenance" />
```

In the above listing, the developer subscribes to the visualization's brush event via `"visBrushed()"` and updates the `"provenance"` of the "Acceleration" (`"acc"`) range slider.

9.3 Replicating Prior UI Control Libraries Using ProvenanceWidgets

We utilized ProvenanceWidgets to replicate three prior works in improving social navigation cues (Scented Widgets [85]), explaining transitions in the user interface (Phosphor Objects [266]), and dynamic querying-based [272, 273] or direct manipulation-based [274] interactions with a visualization system (Dynamic Query Widgets).

9.3.1 Scented Widgets.

Recall Scented Widgets [85] enhance UI controls via embedded visualizations of some precomputed metric to facilitate navigation. ProvenanceWidgets can be configured to recreate these widgets by showing static information about (1) social navigation, e.g., number of times each radio button option was chosen across multiple users and when (Figure 9.10A–shades of orange) or (2) data distribution, e.g., distribution of values for that column in the underlying dataset (Figure 9.10A–blue). To realize the range slider in Figure 9.10A–shades of orange, the developer can program the widget in the following way:

```
historical_usage_logs = {
  "revalidate": true,
  "data": [
            {"value": [100, 160], "timestamp": _},
            {"value": [100, 160], "timestamp": _},
            {"value": [160, 200], "timestamp": _},
            ...
        ]
}
```

```
<provenance-slider
    [freeze]="true"
    [(provenance)]="historical_usage_logs"
/>
```

In the above listing, the developer passes the `"historical_usage_logs"` information in the format of interaction logs, which is then mapped to the `[(provenance)]` property. The `[freeze]="true"` property will ensure the widgets don't update in real-time in spite of user interactions.

9.3.2 Phosphor Objects.

Recall Phosphor objects [266] track user interactions with UI controls in real-time and leave visual scents of the most recent and second most recent interaction. ProvenanceWidgets can be configured to recreate Phosphor objects by limiting the *recency of interaction* mapping to the color encoding channel to just include the two most recent interactions (the current and the previous interaction). That way, every interaction will leave behind a single visual trace (e.g., light green bar) corresponding to the previous value. This ability to visualize the present and previous state is akin to the "Present" and "Past" wildcard states in the design space for guidance, helping users *undo* or *review* the previous analysis state, introduced in chapter 8. To realize the single slider in Figure 9.10B, the developer can program the widget in the following way:

```
widgetUpdated() {
    if (provenance) provenance = {
        "revalidate": true,
        "data": provenance["data"].slice(-2)
    }
}
```

```
<provenance-slider
    [(provenance)]="provenance"
    (provenanceChange)="widgetUpdated()"
/>
```

In the above listing, when a widget is interacted with (`"widgetUpdated()"`), the

developer slices the `"provenance"` array to only keep the two most recent interactions and then issues the `revalidate` command to recompute the model and update the view.

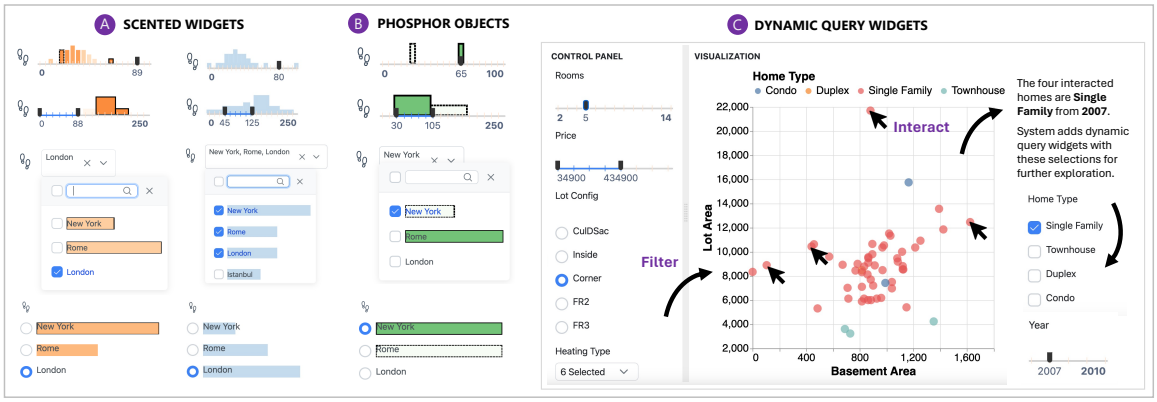


Figure 9.10: ProvenanceWidgets can be configured to (re)create the core functionalities of (a) Scented Widgets, (b) Phosphor Objects, and (c) Dynamic Query Widgets. Scented Widgets enhance UI controls via embedded visualizations of some pre-computed metric, e.g., visit frequency and recency (in shades of orange) or data distribution (in blue) to facilitate navigation. Phosphor objects track user interactions with UI controls in real-time and leave visual scents of the most recent (dark green) and second most recent (light green) interaction. Dynamic Query Widgets are UI controls that continuously update a visualization and/or its underlying data as the user adjusts them. ProvenanceWidgets can facilitate creating a dynamic query [272] to lookup affordable ("Price" $<$ \$500k) houses with five "Rooms" and "Lot Config"=*Corner* and then update the visualization. Alternatively, these widgets can also be created on the fly, e.g., if a user interacts with "Home Type"=*Single Family* and "Year"=*2007* houses, the system can add new query widgets for "Home Type" and "Year" to generalize the user's selection [274] and facilitate future exploration.

9.3.3 Dynamic Query Widgets.

Dynamic querying. Shneidermann [272] introduced the notion of dynamic queries to continuously update the data that is filtered from the database and visualized. These queries ideally work instantly as the user adjusts UI controls such as sliders to form simple queries or to find patterns or exceptions. Williamson et al. [273] then evaluated this approach in a real-estate system called HomeFinder. Figure 9.10C shows how ProvenanceWidgets can create HomeFinder. The developer can program the "Rooms" single slider as follows:

```
1 // Apply the filter and update the visualization
2 widgetUpdated(model) {}
```

```
<provenance-slider
    [visualize]="false"
    [freeze]="true"
    (selectedChange)="widgetUpdated($event.value)"
/>
```

In the above listing, the widget is initialized with `[freeze]`="true" and `[visualize]`="false", disabling logging and overlays. When a widget is interacted with (`"widgetUpdated()"`), the developer can access the new model, filter the data, and update the visualization.

Direct manipulation. Heer et al. [274] introduced direct manipulation techniques that couple declarative selection queries with a query relaxation engine, enabling users to interactively generalize their selections using dynamically generated query widgets. For example, if a user's selections on a housing dataset only include "Home Type"=*Single Family* and "Year"=*2007*, then two dynamic query widgets are created: a checkbox group for "Home Type" with the *Single Family* option checked; and a single slider for "Year", preset to *2007*. Figure 9.10C shows how ProvenanceWidgets can support dynamic query widgets created via direct manipulation. To realize the "Year" single slider, the developer can program the widget as follows:

```
selectedYear = 2007;
showWidget = true;
```

```
<provenance-slider
    *ngIf="showWidget"
    [visualize]="false"
    [freeze]="true"
    [selected]="selectedYear"
/>
```

In the above listing, the widget is created (or made visible) by `*ngIf`="showWidget"

and initialized with `[selected]`="selectedYear" (the output of the generalized selection algorithm). `[freeze]` and `[visualize]` are still set to "true" and "false", respectively.

9.4 Evaluation 1: Cognitive Dimensions of Notation and ProvenanceWidgets

In this section, we describe findings from an author-led assessment of our library, from a developer standpoint, based on the Cognitive Dimensions of Notation [275], a framework of heuristics commonly used to assess the effectiveness of notational systems (e.g., visualization grammars and toolkits). Of the 14 cognitive dimensions, we select a relevant subset for comparing our work with existing tools.

Consistency: *Similar semantics are expressed in similar syntactic forms* – ProvenanceWidgets exposes a common set of provenance-related properties and events, which behave consistently across all underlying widgets. The notation is also consistent with the base libraries it inherits from, as well as consistent across different JavaScript frameworks when used as Web Components.

Diffuseness: *Verbosity of language* and **Hard Mental Operations:** *high demand on cognitive resources* – Since provenance is built into the widgets, developers can directly use components from the underlying libraries to create provenance-aware widgets. Even advanced use cases such as persisting, restoring, or modifying the provenance only require minimal code and cognitive demand. This is in contrast to existing provenance systems such as Trrack and TrrackVis [105], which require developers to set up states, actions, event listeners, and other components to capture and visualize provenance.

Viscosity: *Difficulty of making changes* – The widgets have a low viscosity for primitive attributes, but a high viscosity for complex attributes. For example, developers can easily add labels and toggle provenance tracking and visualization. However, changing the options and provenance data structures requires more effort.

9.5 Evaluation 2: Developer Case Studies Using ProvenanceWidgets

We present case studies with developers who used ProvenanceWidgets to build a custom web-based application. Our aim was to evaluate the effectiveness of ProvenanceWidgets for building provenance-based visual data analysis systems, and to understand developers' experiences working with it, including installation, configuration, and customization.

9.5.1 Participants and Procedure

Participants. We recruited four developers (P_{1-4}) – 2 men and 2 women in the 18-24 (1) and 25-34 (3) age groups, and well versed in front-end web development and analysis.

Task. We tasked participants to:

> Develop a "Pokemon Explorer" visualization system for a Pokemon Fan Club, to help member fans visually explore Pokemon names and stats to pick their dream team. The visualization system should consist of a visualization, and UI controls, that help specify the visualization (e.g., map variables to visual encodings) and/or beautify it (e.g., modify font styles and color schemes). The Club wishes to track fans' interaction behaviors as they explore the data, hence you must use ProvenanceWidgets as your UI controls to help track and visualize each user's provenance.
>
> In addition, try to capture relevant user interactions from other, non-ProvenanceWidgets places in your application, and manually update ProvenanceWidgets. For example, brushing within a scatterplot visualization should log the brushed extents on either axis and append them to the provenance data structures of the corresponding attribute filters.

Dataset. We used a dataset of 802 pokemon [276] comprising nine quantitative variables (*Height_m*, *Weight_kg*, *HP*, *Speed*, *Attack*, *Special Attack*, *Defense*, *Special Defense*, *Happiness*), five nominal variables (*Classification*, *Name*, *Primary Type*, *Secondary Type*, *Is Legendary*), and two ordinal variables (*Pokedex Number*, *Generation*). This variety of variables enables developers to use different widgets.

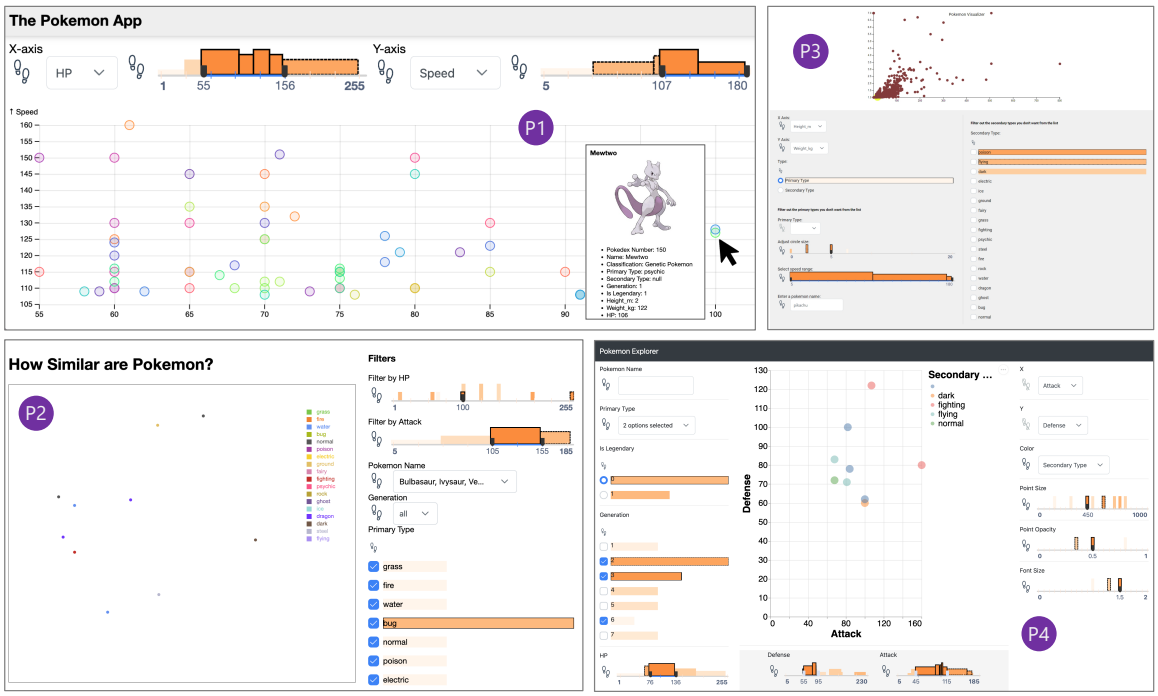


Figure 9.11: Pokemon Explorer applications developed by our participants. Everyone created a scatterplot-based visualization system using ProvenanceWidgets to apply filters ($P_{1,2,3,4}$), specify visual encodings: *xy* ($P_{1,2,4}$), *color* ($P_{1,4}$), and/or adjust styling ($P_{3,4}$).

Logistics. We first conducted a 30-minute onboarding interview over Zoom, during which we sought consent from participants and introduced them to ProvenanceWidgets and the study task. We also asked them their preference between the Angular components and the WebComponent versions of the library. Accordingly, we shared with them a Github repository comprising task and installation instructions, API documentation, and starter code. Next, we gave participants up to one week to complete the task. During this week, we asked them to document their experience (e.g., bugs, happy moments) working with the widgets in a FEEDBACK.md file. If and when stuck, we asked participants to create GitHub issues or directly email the study administrators. Finally, we conducted a 30-minute debriefing interview wherein we reviewed the participants' visualization systems, source code, and feedback notes. We compensated each participant with a $25 gift card.

Analysis. We manually transcribed the audio recordings and feedback, divided them into smaller sections, and applied open coding [185], specifically, constant comparison and theoretical sampling [277]. Next, we briefly describe the participants' applications, development experience, and feedback on the widgets, including future enhancements.

9.5.2 Results and Discussion

Developed Applications. Figure 9.11 shows four applications developed by our participants using Angular ($P_{1,4}$) and Web components ($P_{2,3}$). All participants created a scatterplot-based system using ProvenanceWidgets to apply filters ($P_{1,2,3,4}$), specify visual encodings: *xy* ($P_{1,2,4}$), *color* ($P_{1,4}$), and/or adjust styling ($P_{3,4}$). P_2's scatterplot visualized the output of a UMAP [278] dimensionality reduction algorithm that groups more similar pokemon to be closer to each other. P_4 wanted to be able to export the visualizations, hence they provided additional options to configure the font size, point size, and point opacity. Only P_4 attempted the bonus task, to capture user interactions externally (via brushing in the visualization) and manually update the provenance on the relevant widget.

Developer Experience. All four developers found ProvenanceWidgets to be useful, commending its built-in capability to track and visualize provenance. P_1 said, *"I was initially very surprised and actually very excited like, wow, it's very well-made, doesn't really break. That's all you can ask for in any library like this."* P_3 particularly found the widgets to be 'self-explanatory', especially for Angular and Javascript developers. P_4 appreciated the consistent design of the widgets and the ability to externally modify the provenance.

In terms of overall development effort, P_1 took approximately 7 hours whereas $P_{2,3,4}$ took 3-4 hours to set up their visualization system and integrate ProvenanceWidgets. While P_1 found the study to be *"time-consuming"* but *"really fun"*, $P_{2,3,4}$ found the amount of time and effort to be appropriate. P_2 did not find a very steep learning curve and said, *"'Intuitive' will be a very good word to describe it."*

Development Strategies. Participants found the documentation ($P_{1,2,3}$) and starter code ($P_{2,3}$) helpful. P_2 said, *"The sample code to start with was just very helpful because I pretty much just modified it to suit my use-case and it just worked. This single-handedly cut the amount of time I spent in half."* $P_{2,3}$ requested adding more advanced examples in the eventual documentation. P_1 accessed the original PrimeNG [279] and ngx-slider [280] documentations to try and customize the dropdown options and slider options, respectively. $P_{2,4}$ requested alternate widget layouts, e.g., horizontally laid out radio buttons and checkboxes (P_2) and vertically laid out sliders (P_4). These customizations and configurations are currently restricted and unsupported, respectively, because they would conflict with the provenance overlay implementation. A takeaway for us is to acknowledge these limitations in the library documentation and include a roadmap for future features.

9.6 Limitations and Future Work

ProvenanceWidgets may require developers to understand certain core concepts of its dependencies (PrimeNG [279], ngx-slider [280], and Angular [175]), which might lead to issues and limitations, necessitating workarounds. For example, customizing the option templates in the dropdowns and re-orienting the sliders, radio buttons, and checkboxes is currently restricted as it conflicts with the provenance overlays. Future work is planned to ensure ProvenanceWidgets inherits all base library features.

Next, ProvenanceWidgets also inherits the inherent limitations of standard UI controls pertaining to scalability and usability. For instance, sliders and dropdowns often struggle with large ranges and numerous options, respectively. As a workaround, developers can increase a slider's step size (reducing the number of selectable values), improving usability but sacrificing precision; and dropdown options can be reordered or filtered based on frequency or recency to ensure already interacted options are always visible and accessible.

Lastly, visualizations often involve multidimensional interactions like brushing and linking [281] or smart brushing [282], wherein multiple attributes get modified in the same

interaction. ProvenanceWidgets can currently visualize such provenance independently on each widget (as shown in Figure 9.9), leading to potential misrepresentation and information loss. Future work is planned to track and visualize provenance across multiple widgets.

9.7 Summary

In this chapter, I described ProvenanceWidgets, an open-source JavaScript library of UI control elements that track and dynamically overlay a user's analytic provenance, *out of the box*. This library includes enhanced implementations of radio buttons ⦿○, checkboxes ☐☑, single sliders —○—, range sliders —○=○—, dropdowns ☰—⌄, multiselects ☰⌄, and input text fields I🔍 that track how often (frequency) and when (recency) a user interacts with them (e.g., selecting a dropdown option) and present visual overlays showing an aggregated summary as well as a detailed temporal history. By showing the user what they have done so far, these widgets can make the user reflect upon their present choices to influence subsequent ones. Additionally, if these widgets are preconfigured to show customized information (e.g., interaction behavior of peers), they can be used to nudge users in specific directions (e.g., interact with previously overlooked aspects).

ProvenanceWidgets is available as open-source software at **https://ProvenanceWidgets.github.io**, enabling developers to integrate provenance-tracking into their systems. Additionally, because provenance is often a basis for providing guidance, the provenance-tracking ability of the library can be used to prototype guidance systems. The library is built using Angular but is universally compatible across different frameworks through Web Components. The library is also highly customizable, allowing developers to realize a variety of configurations such as setting the logging frequency or initializing with an existing provenance log. Using ProvenanceWidgets, we recreated three prior libraries: (1) Scented Widgets [85], (2) Phosphor objects [266], and (3) Dynamic Query Widgets [272]. Case studies with four developers revealed the effectiveness of ProvenanceWidgets to build custom applications. For details, I refer the reader to the associated publication [9].

CHAPTER 10
REFLECTIONS AND FUTURE WORK

In this chapter, I reflect on the works in this dissertation, raise some new questions, and highlight future opportunities for visualization and human-computer interaction research.

How does the "source" of guidance influence its utilization during data analysis?

In all works described so far, guidance was provisioned based on statistical computations, be it the data quality and usage metrics in DataPilot and DataCockpit (chapter 3), the 'smart' sample testing database in DIY (chapter 4), 'bias' metrics [111] in Lumos (chapter 5) and BiasBuzz (chapter 6), and the interaction frequency and recency metrics in ProvenanceLens (chapter 7) and ProvenanceWidgets (chapter 9). Guidance was not provisioned from other entities such as human (non-) experts or artificial intelligence (AI) models.

Literature on guidance has thus far focused on the important dimensions of "why," "how," "what," and "when" in guided interactions [123, 42]. What about a new dimension–"from whom"–focusing on the source of guidance–such as humans or AI? Today, guidance is already being sought from human experts (e.g., an expert analyst or consultant) or groups of peers (e.g., via community forums such as Stack Overflow [283]). Recently, there has been a growing expectation for guidance to come from AI [284, 285, 286], even though this guidance must itself rely on data from experts or groups of humans, or other systems, to train models and align recommendations with user preferences. Prior work has also emphasized and compared seeking guidance from experts and peers [287, 288, 289]. Studying this dimension is thus important because the effect of *source-attribution* in providing guidance carries significant implications for offering effective guidance systems.

We have already conducted a preliminary investigation into this new dimension (publication currently under review [10]), focusing on how users' perception and utility of guid-

ance coming from a particular source impacts their performance during data preparation. In particular, we conducted a between-subjects study with five conditions: guidance from an (1) *AI* model, (2) human *expert*, (3) *group* of analysts, (4) *unattributed* guidance (without source attribution), and (5) a *control* (no-guidance) condition. This design allowed us to compare source-specific guidance effects against both unattributed guidance and no guidance. We built a custom data preparation tool for our study, where users selected relevant attributes from an unfamiliar dataset. Depending on the condition, users could request guidance up to ten times, presented as attribute suggestions (guided attributes). To ensure internal validity, we controlled guidance quality by providing seven relevant and three irrelevant attribute suggestions, randomly selected, to measure source effects.

We found that: (i) guidance benefits users during analysis, with varying effects across sources; (ii) guidance use shifts across analysis stages; (iii) users verify guidance differently based on the source; (iv) initial perceptions of source-attributed guidance are lower, but scores improve post-task, highlighting source-attribution subtleties; (v) while users report more post-task regret with AI guidance, they also experience increased confidence from it, suggesting a nuanced role for AI. While future efforts are needed to fully understand the impact of the source of guidance, our findings suggest systems should be transparent about it and design guardrails to prevent users' pre-conceptions and/or misconceptions about a particular guidance source from driving their analysis. Additionally, this work suggests potential "side effects" of guidance that may impact analysis, as described next.

Are there any "side effects" of relying too much or too little on guidance?

Today, AI assistance and automation is increasingly being integrated into data-intensive tasks across a variety of domains. These approaches promise new ways to support analysts, enabling rapid and enhanced sensemaking by offloading routine but computationally heavy tasks to machines. However, this shift raises new, important questions: Can people become too reliant on these technologies, to an extent they (wrongly) assume all guidance received

to be 'good'? What if people become too skeptical and (also, wrongly) stop accepting it, even if it is 'good'? Also, what if guidance has other detrimental effects to users' cognition and sensemaking processes, e.g., lead to fewer insights? It is thus important to balance human action, AI automation, and human reliance on AI automation.

Overreliance on automation has shown detrimental effects in other domains, including reduced task performance and eroded trust when users develop incorrect mental models of AI systems; in fact, people's reliance on AI has been shown to depend on various contextual factors, such as their AI literacy [290], domain expertise [291], and amount of feedback [292]. Furthermore, several metrics have been proposed that measure people's reliance on AI guidance, quantifying people's "agreement" and "disagreement" with AI recommendations [293], people's "acceptance" of incorrect AI recommendations [294], people's "change" in behavior based on AI recommendations [295, 286], and people's propensity to "delegate" eventual decision-making to AI [296].

So I ask: How can over- and underreliance on AI be modeled and measured during analysis, and what implications do they have on the design of *responsible* analysis tools? Additionally, what negative effects might overreliance on AI guidance introduce to human-centered processes such as insight discovery and knowledge generation? Addressing these questions is crucial to ensure guidance tools support (and not hinder) the analysis process.

How can systems "naturally" learn about the user and their intentions?

In DataPilot and DataCockpit (chapter 3), guidance was static, i.e., the data quality and usage insights were pre-computed and remained unchanged as the user interacted during analysis; whereas in Lumos (chapter 5), BiasBuzz (chapter 6), and ProvenanceLens (chapter 7), the system computed guidance in real-time as the user's focus evolved during analysis (using mouseover interactions as a proxy). Similarly, ProvenanceWidgets (chapter 9) tracked user interactions that changed a UI control's state, e.g., selecting a dropdown option, dragging a range slider handle, to compute frequency and recency.

As guidance systems evolve, they have the potential to leverage a variety of interaction signals beyond traditional mouse and keyboard inputs to better understand users and anticipate their needs. For instance, eye-gaze tracking [66] could reveal focal points and attention spans, while facial expression analysis [68] might help systems interpret user emotions and adjust accordingly. Hand-gesture detection [68] and touch data [65] could add layers of physical engagement, allowing systems to interpret more specific user intentions in real time. Audio inputs, such as speech [65], could enable the system to infer sentiment or even respond to voice commands. With IoT-enabled devices such as smartwatches [297], the system might capture contextual cues like movement or physical state, providing deeper insight into the user's environment and behaviors. By integrating these multimodal signals, I envision future guidance systems to elicit user inputs as 'naturally' as possible, so the user can focus on the analysis task-at-hand.

How can users "better contribute" to a co-adaptive guidance dialog with the system?

Guidance systems are effective when the knowledge gap between the user and the system is continuously minimized as the analysis progresses. For this to occur, both the system and the user must be 'brought and kept on the same page', realizing an effective co-adaptive guidance dialog between the two. Users can participate and contribute to this dialog by providing *feedback* to the system-generated guidance during analysis and by providing *feedforward* into the system which can also occur even before analysis begins.

User feedback can take various forms, including explicit responses via accept/reject buttons, less certain responses via like/dislike or upvote/downvote rating options, timing adjustments (e.g., mute/snooze), or user expressions of less/non-interest (e.g., the "Just Cleaning up" button on Netflix [240]), or no response (i.e. ignorance of the system-generated guidance). Additionally, implicit signals can be derived from interactions with visualizations, such as via "visualization by demonstration" [298] or "semantic interaction" [79].

Regarding user feedforward, Lumos (chapter 5) allows users to configure three types

of expected interaction behaviors–proportional, equal, and custom–for the system to accordingly identify exploration biases. Unlike user feedback, UI affordances for eliciting feedforward are often bespoke and task-specific, making standardization challenging. Nevertheless, there is potential to develop reusable components for common tasks.

This leads to the overall question: What is the design space for users to provide feedback and feedforward to guidance from a system? And in response, how can the system adapt to these inputs from the user to continue a truly co-adaptive guidance process?

Why just visual? How about "multimodal guidance communication" during analysis?

In DataPilot and DataCockpit (chapter 3), DIY (chapter 4), and Lumos (chapter 5), we primarily relied on visual means to communicate guidance. While the evaluations suggested this approach can effectively engage most users, it may not fully exploit the potential of multimodal guidance. We explored multimodality in BiasBuzz (chapter 6), wherein we incorporated haptic feedback alongside visual cues. BiasBuzz' evaluation revealed mixed responses; participants found that the haptic mouse vibrations can be useful in capturing attention instantly, but can also be distracting and disturbing, suggesting careful consideration is needed to balance the pros and cons.

Going beyond visual and haptic feedback, one can integrate audio cues, ambient light, or other sensory modalities to communicate guidance. Another modality for communicating guidance is via augmented reality (AR) and virtual reality (VR) environments. Each modality has unique strengths and weaknesses, which can differently influence users' cognitive load and sensemaking processes during analysis. By systematically studying these modalities–both in isolation and in combination with one another–we may be able to design truly accessible, enjoyable, and effective multimodal guidance systems for users.

"When" to provide guidance and for "how long"?

Timing and duration of guidance are crucial in designing effective guidance systems. For example, Microsoft PowerPoint [299] can sometimes be annoying with its slide design guidance, especially without the user explicitly requesting for it. Additionally, unwelcome or unexpected prompts on websites–like cookie consent or newsletter popups–can detract the user from the main content, highlighting the importance of not just guiding users but doing so tactfully. In all works described in this dissertation, guidance was either pre-computed and always available to user, or pre-computed but only available on user request, or computed and updated in real-time with every user interaction/operation. In Lotse [47] – the open-source Python library to design custom guidance strategies – guidance is orchestrated through an 'inference loop' and a 'guidance loop'; in the inference loop, the system determines which strategies are currently active and should potentially generate suggestions; this loop runs every 30 seconds by default, whereas the guidance loop is triggered every second; together, both these loops determine if and what guidance must be provisioned. But I ask: Are these the only/best strategies to time the guidance?

Determining when and how often to offer guidance may be a balancing act. It could be delivered continuously, on demand (allowing the user to request it only when needed), periodically (triggered at set intervals), or randomly, based on usage patterns and user engagement. Systems might also explore context-aware guidance that considers the user state and task complexity, appearing only when it has inferred the user could benefit from it. Once guidance is presented, its duration should respect the user's workflow, ensuring it doesn't become a distraction if too prolonged or cause confusion if too brief. If the user responds to the guidance, the system must carefully judge when to reintroduce suggestions to avoid interrupting the user's independent analysis process. Thoughtful timing and appropriate duration could make guidance feel less like an imposition and more like an adaptive, supportive layer that enhances rather than disrupts the user's analytical journey.

How to conduct effective evaluations of guidance systems?

Designing and evaluating guidance systems is challenging due to the difficulty of capturing authentic, meaningful feedback from users, even if it is captured 'naturally' via interactions, gestures, and expressions during analysis or 'directly' via feedback forms before/after the study. Short user study tasks, while manageable in a research setting, may fail to capture the depth of how users truly perceive and engage with guidance. For example, while conducting pilot studies using ProvenanceLens (chapter 7), our participants ignored or underutilized the provisioned guidance, as the study setup did not require repeated or prolonged interactions, and users could just rely on their memory to successfully complete the task(s). This observation made us refine our actual task to encourage the intended interaction behavior. However, even such refinements may fall short of simulating real, complex analytical workflows. So I ask: How can user study designs for evaluating systems realistically reflect complex analytical workflows? Would longer or multi-session studies (e.g., MILCs [300]) provide more credible insights into how guidance is perceived and utilized?

I believe determining appropriate metrics and methodologies for evaluating guidance systems poses some challenges. For example, Wall et al.'s [111] 'bias' metrics, that we utilized in Lumos (chapter 5), do not consider the recency of interactions when detecting biases. In ProvenanceLens (chapter 7), we considered both frequency and recency of interactions but did not weight recent interactions more. Quantifiable metrics aside, subjective measures such as users' self-reported trust and reliance on the guidance, the kind we administered in our crowdsourced study to measure the effect of the guidance source (e.g., human or AI) [10], are also important signals. Lastly, what about metrics that authors can self-evaluate their systems against? For example, Ceneda et al. [41] proposed five qualities for evaluating guidance systems in terms of how accessible, reliable, context-aware, user-adjustable, and minimally intrusive they are. With so many nuanced aspects, I call for a comprehensive methodology to ensure internal and/or external validity of the evaluations.

How can we continue to empower others to build their own guidance systems?

Building guidance-enriched systems can be challenging, as it requires developers to integrate frontend and/or backend components while addressing complex questions related to the domain ("what"), timing ("when"), means ("how"), and objectives ("why") of guidance. This necessitates open-source tools that enable developers to focus on determining the guidance-related strategy and designing the user experience, without being bogged down by low-level implementation details. While Lotse's Python library and declarative grammar [47] have made it easier, there remains a significant gap in tooling, especially in frontend user interfaces that enable real-time interactions between users and the system.

Trrack [105] and ProvenanceWidgets [9] (chapter 9) took a first step offering re-usable guidance components for tracking and visualizing analytic provenance in the frontend. However, these tools need to evolve further, supporting customizable, bidirectional interactions that allow users to both receive and respond to guidance. Beyond the technical benefits, fostering an open-source ecosystem around these tools can encourage community-driven innovation, where developers share insights, extend functionality, and refine best practices. Such an ecosystem would also prioritize scalability and interoperability, ensuring tools work across diverse domains and smoothly integrate into existing workflows. Lastly, to truly empower developers, the ecosystem must also offer comprehensive documentation, educational resources, and testing and evaluation frameworks that allow for systematic benchmarking of guidance strategies. I believe building this ecosystem can significantly lower the entry barrier for creating powerful yet customizable guidance systems.

CHAPTER 11
FINAL THOUGHTS

Designing, Developing, and Democratizing Guidance for Visual Analytics. Through this dissertation, I validated that, *"Facilitating co-adaptive guidance in mixed-initiative user interfaces, wherein the user and the system learn from and take initiatives on behalf of each other, enhances human-data interaction experiences as well as analytic processes and outcomes, while promoting the design of new tools that broaden access for researchers, developers, and practitioners alike."* In doing so, I made the following contributions to visualization and human-computer interaction literature, with complementary contributions to database, ubiquitous computing, deep learning, and artificial intelligence literature:

Design, Implementation, and Evaluation of Techniques and Systems

- A data preparation system that presents data quality & usage metrics to guide users in selecting effective subsets from large, unfamiliar datasets (DataPilot [4], chapter 3).
- A mixed-initiative visual data analysis system, that presents real-time visual traces of a user's interactions ("interaction traces"), to increase awareness of biased analytic behaviors against configurable target analytic behaviors (Lumos [27], chapter 5).
- A mixed-initiative system that provides multimodal guidance (visual + haptic) to mitigate biased analytic behaviors during data analysis (BiasBuzz [7], chapter 6).

Design, Implementation, and Evaluation of System Test-beds / Playgrounds

- A question-answering system, integrated with an interactive, self-service debugging view, to help users debug natural language to SQL workflows (DIY [26], chapter 4).
- A visual data analysis system as a test-bed for demonstrating (and studying) the design spaces for provenance communication (ProvenanceLens [11], chapter 7).

- A visual data analysis system as a test-bed for demonstrating (and studying) the design spaces for guidance communication (Lighthouse [12], chapter 8).

Empirical Evaluations

- A series of in-lab and crowdsourced studies to understand how human biases (e.g., gender) impact the way people make decisions during analysis (Left, Right, and Gender [25], section 5.4). We found some evidence that "interaction traces" can increase awareness of unconscious biases, but additional confirmatory studies are needed.
- A crowdsourced study [10] to understand how the source of guidance–such as AI model or human expert–impacts people's perception and usage of guidance during analysis (chapter 10). We found that the source of guidance matters to users, but not in a manner that matches received wisdom; users utilize guidance differently, expressing varying levels of regret, despite receiving guidance of similar quality.

Design Spaces

- A design space for communicating analytic provenance by modeling it as an attribute, and mapping it to visual encodings and data transformations during analysis (ProvenanceLens [11], chapter 7).
- A design space for communicating guidance by modeling it as a *state*-space (past, present, problem, future) and presenting different *levels* (e.g., 1, 2, 3) via adaptive UI elements–visualizations, UI controls, external panels (Lighthouse [12], chapter 8).

Open-Source Libraries and Toolkits

- A Python toolkit that helps developers compute data quality and usage information from data lakes, along with a companion data visualization system to guide database administrators to navigate and monitor data lakes (DataCockpit [5], chapter 3).
- A JavaScript library of UI controls that helps developers prototype custom web applications with provenance-tracking (ProvenanceWidgets [9], chapter 9).

In closing, I would just like to say that having journeyed through this forest that is visualization and HCI research, I take pride in having enjoyed the fruits of existing trees, nurtured new saplings, and sown fresh seeds, with the hope of seeing them continue to thrive to their potential. Along the way, I met many wanderers, each on their own unique journey but sharing a similar pursuit. I learned and grew from them, and perhaps even shared lessons of my own. While all of this brings me joy, I remain unsatisfied, as this forest needs constant care to sustain its current growth and more plantations to support future generations. Also, as in life, times will often be hard, but I will remember – and hope others will reflect on – *"Life is a constant dance between one's* ***desire*** *and* ***destiny****; embrace it."*

REFERENCES

[1] A. P. J. A. Kalam and A. Tiwari, *Wings of Fire: An Autobiography*. Universities Press, 1999.

[2] R. C. Basole, A. Qamar, B. Pal, M. Corral, M. Meinhart, and A. Narechania, "Understanding Failure Mode Effect Analysis Data Using Interactive Visual Analytics," *IEEE Computer Graphics and Applications*, vol. 39, no. 6, pp. 17–26, 2019.

[3] A. Narechania, A. Karduni, R. Wesslen, and E. Wall, "vitaLITy: Promoting Serendipitous Discovery of Academic Literature with Transformers & Visual Analytics," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 1, pp. 486–496, 2022.

[4] A. Narechania *et al.*, "DataPilot: Utilizing Quality and Usage Information for Subset Selection during Visual Data Preparation," in *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, 2023.

[5] A. Narechania *et al.*, "DataCockpit: A Toolkit for Data Lake Navigation and Monitoring Utilizing Quality and Usage Information," in *2023 IEEE International Conference on Big Data (BigData)*, 2023, pp. 5305–5310.

[6] A. Narechania, A. Endert, and C. Andris, "Resiliency: A Consensus Data Binning Method (Short Paper)," in *12th International Conference on Geographic Information Science (GIScience 2023)*, Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2023.

[7] J. R. Paden, A. Narechania, and A. Endert, "BiasBuzz: Combining Visual Guidance with Haptic Feedback to Increase Awareness of Analytic Behavior during Visual Data Analysis," in *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, 2024, pp. 1–7.

[8] S. Sah, R. Mitra, A. Narechania, A. Endert, J. Stasko, and W. Dou, "Generating Analytic Specifications for Data Visualization from Natural Language Queries using Large Language Models," *arXiv*, 2024, IEEE Visualization Conference (NLVIZ Workshop).

[9] A. Narechania, K. Odak, M. El-Assady, and A. Endert, "ProvenanceWidgets: A Library of UI Control Elements to Track and Dynamically Overlay Analytic Provenance," *IEEE Transactions on Visualization and Computer Graphics*, 2024.

[10] A. Narechania, A. Endert, and A. Sinha, "Guidance Source Matters: How Guidance from AI, Expert, or a Group of Analysts Impacts Visual Data Preparation and Analysis," under review.

[11] A. Narechania, S. Guo, E. Koh, A. Endert, and J. Hoffswell, "Utilizing Provenance as an Attribute during Visual Data Analysis Promotes Self-Reflection: A Design Probe with ProvenanceLens," under review.

[12] A. Narechania, S. Guo, E. Koh, A. Endert, and J. Hoffswell, "Lighthouse: A Design Space for Guidance Communication during Visual Data Analysis," under review.

[13] A. Narechania, A. Qamar, and A. Endert, "SafetyLens: Visual Data Analysis of Functional Safety of Vehicles," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 2, pp. 1688–1697, 2021.

[14] A. Narechania, A. Endert, and C. Andris, "Behind the Maps: An Interview Study with Cartographers and GIS Experts," under review.

[15] G. A. Ramos, A. Fourney, B. Lee, and A. A. Narechania, *Natural language query processing and debugging*, US Patent 12,008,043, 2024.

[16] A. A. Narechania *et al.*, *Data Selection based on Consumption and Quality Metrics for Attributes and Records of a Dataset*, US Patent App. 17/693,799, 2023.

[17] A. A. Narechania *et al.*, *Interactive Tree Representing Attribute Quality or Consumption Metrics for Data Ingestion and Other Applications*, US Patent App. 17/693,778, 2023.

[18] A. A. Narechania *et al.*, *Dashboard for Monitoring Current and Historical Consumption and Quality Metrics for Attributes and Records of a Dataset*, US Patent App. 17/693,811, 2023.

[19] A. Narechania *et al.*, *Data Identification and Extraction from Unstructured Documents*, US Patent App. 18/185,547, 2024.

[20] A. Narechania *et al.*, *Data Extraction and Analysis from Unstructured Documents*, US Patent App. 18/482,754, 2024.

[21] A. Narechania, J. E. Hoffswell, S. Guo, E. Koh, and P. Bhutani, *Adaptive Dynamic Guidance in Data Analysis Tools*, US Patent App., 2024.

[22] A. Narechania, A. Srinivasan, and J. Stasko, "NL4DV: A Toolkit for Generating Analytic Specifications for Data Visualization from Natural Language Queries," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 2, pp. 369–379, 2021.

[23] P. Dintyala, A. Narechania, and J. Arulraj, "SQLCheck: Automated Detection and Diagnosis of SQL Anti-Patterns," in *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, 2020, pp. 2331–2345.

[24] A. Ghosh, D. Bansod, A. Narechania, P. Dintyala, S. Timurturkan, and J. Arulraj, "Interactive Demonstration of SQLCheck," *Proceedings of the VLDB Endowment*, vol. 14, no. 12, pp. 2779–2782, 2021.

[25] E. Wall, A. Narechania, A. Coscia, J. Paden, and A. Endert, "Left, Right, and Gender: Exploring Interaction Traces to Mitigate Human Biases," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 1, pp. 966–975, 2022.

[26] A. Narechania, A. Fourney, B. Lee, and G. Ramos, "DIY: Assessing the Correctness of Natural Language to SQL Systems," in *Proceedings of the 26th International Conference on Intelligent User Interfaces*, 2021, pp. 597–607.

[27] A. Narechania, A. Coscia, E. Wall, and A. Endert, "Lumos: Increasing Awareness of Analytic Behavior during Visual Data Analysis," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 1, pp. 1009–1018, 2022.

[28] R. Mitra, A. Narechania, A. Endert, and J. Stasko, "Facilitating Conversational Interaction in Natural Language Interfaces for Visualization," in *2022 IEEE Visualization Conference (VIS)*, 2022.

[29] T. M. Green, W. Ribarsky, and B. Fisher, "Visual Analytics for Complex Concepts using a Human Cognition Model," in *2008 IEEE Symposium on Visual Analytics Science and Technology*, 2008, pp. 91–98.

[30] D. Keim, G. Andrienko, J.-D. Fekete, C. Görg, J. Kohlhammer, and G. Melançon, "Visual Analytics: Definition, Process, and Challenges," in *Information Visualization*, Springer, 2008, pp. 154–175.

[31] B. Shneiderman, C. Plaisant, M. S. Cohen, S. Jacobs, N. Elmqvist, and N. Diakopoulos, *Designing the User Interface: Strategies for Effective Human-computer Interaction*. Pearson, 2016.

[32] E. Horvitz, "Principles of Mixed-Initiative User Interfaces," in *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, 1999, pp. 159–166.

[33] Endert, Alex and Hossain, M Shahriar and Ramakrishnan, Naren and North, Chris and Fiaux, Patrick and Andrews, Christopher, "The Human is the Loop: New Directions for Visual Analytics," *Journal of Intelligent Information Systems*, vol. 43, no. 3, pp. 411–435, 2014.

[34] A. Endert, P. Fiaux, and C. North, "Semantic Interaction for Visual Text Analytics," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2012, pp. 473–482.

[35] D. Ceneda *et al.*, "Characterizing Guidance in Visual Analytics," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 111–120, 2017.

[36] D. Ceneda *et al.*, "Amending the Characterization of Guidance in Visual Analytics," *arXiv*, 2017.

[37] C. Collins *et al.*, "Guidance in the human-machine analytics process," *Visual Informatics*, vol. 2, no. 3, pp. 166–180, 2018.

[38] Microsoft Corporation, *Clippy, the Office Assistant*, Included in Microsoft Office 97, 1997.

[39] U. Shakir, "Tesla's Optimus bot makes a scene at the robotaxi event," *The Verge*, 2024.

[40] H.-J. Schulz, M. Streit, T. May, and C. Tominski, "Towards a Characterization of Guidance in Visualization," in *Poster at IEEE Conference on Information Visualization (InfoVis)*, 2013.

[41] D. Ceneda *et al.*, "Guide Me in Analysis: A Framework for Guidance Designers," *Computer Graphics Forum*, vol. 39, no. 6, pp. 269–288, 2020.

[42] I. Pérez-Messina, D. Ceneda, M. El-Assady, S. Miksch, and F. Sperrle, "A Typology of Guidance Tasks in Mixed-Initiative Visual Analytics Environments," in *Computer Graphics Forum*, Wiley Online Library, vol. 41, 2022, pp. 465–476.

[43] D. Sacha, A. Stoffel, F. Stoffel, B. C. Kwon, G. Ellis, and D. A. Keim, "Knowledge Generation Model for Visual Analytics," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 1604–1613, 2014.

[44] F. Sperrle, A. Jeitler, J. Bernard, D. Keim, and M. El-Assady, "Co-Adaptive Visual Data Analysis and Guidance Processes," *Computers & Graphics*, vol. 100, pp. 93–105, 2021.

[45] F. Sperrle, H. Schäfer, D. Keim, and M. El-Assady, "Learning Contextualized User Preferences for Co-Adaptive Guidance in Mixed-Initiative Topic Model Refinement," in *Computer Graphics Forum*, Wiley Online Library, vol. 40, 2021, pp. 215–226.

[46] F. Sperrle, M. El-Assady, A. Arleo, and D. Ceneda, "A Wizard of Oz Study of Guidance Strategies and Dynamics," *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–15, 2024.

[47] F. Sperrle, D. Ceneda, and M. El-Assady, "Lotse: A Practical Framework for Guidance in Visual Analytics," *IEEE Transactions on Visualization and Computer Graphics*, vol. 29, no. 1, pp. 1124–1134, 2023.

[48] D. Ceneda, T. Gschwandtner, and S. Miksch, "A Review of Guidance Approaches in Visual Data Analysis: A Multifocal Perspective," in *Computer Graphics Forum*, Wiley Online Library, vol. 38, 2019, pp. 861–879.

[49] K. Xu, A. Ottley, C. Walchshofer, M. Streit, R. Chang, and J. Wenskovitch, "Survey on the Analysis of User Interactions and Visualization Provenance," *Computer Graphics Forum*, vol. 39, no. 3, pp. 757–783, 2020.

[50] Z. Zhou, W. Wang, M. Guo, Y. Wang, and D. Gotz, "A Design Space for Surfacing Content Recommendations in Visual Analytic Platforms," *IEEE Transactions on Visualization and Computer Graphics*, vol. 29, no. 1, pp. 84–94, 2023.

[51] W. A. Pike, J. Stasko, R. Chang, and T. A. O'Connell, "The Science of Interaction," *Information Visualization*, vol. 8, no. 4, pp. 263–274, 2009. eprint: https://doi.org/10.1057/ivs.2009.22.

[52] C. North *et al.*, "Analytic Provenance: Process + Interaction + Insight," in *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, Association for Computing Machinery, 2011, pp. 33–36.

[53] E. D. Ragan, A. Endert, J. Sanyal, and J. Chen, "Characterizing Provenance in Visualization and Data Analysis: An Organizational Framework of Provenance Types and Purposes," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 31–40, 2016.

[54] Buneman, Peter, "Characterizing Data Provenance," in *Advances in Databases: 17th British National Conference on Databases*, Springer, 2000.

[55] M. Card, *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann, 1999.

[56] M. Hegarty, "Diagrams in the Mind and in the World: Relations between Internal and External visualizations," in *International Conference on Theory and Application of Diagrams*, Springer, 2004, pp. 1–13.

[57] W. I. D. Mining, *Introduction to Data Mining*. Springer, 2006.

[58] K. A. Cook and J. J. Thomas, "Illuminating the Path: The Research and Development Agenda for Visual Analytics," Pacific Northwest National Lab.(PNNL), Richland, WA (United States), Tech. Rep., 2005.

[59] P. Pirolli and S. Card, "The Sensemaking Process and Leverage Points for Analyst Technology as Identified Through Cognitive Task Analysis," in *Proceedings of International Conference on Intelligence Analysis*, McLean, VA, USA, vol. 5, 2005, pp. 2–4.

[60] G. Klein, B. Moon, and R. Hoffman, "Making Sense of Sensemaking 2: A Macrocognitive Model," *IEEE Intelligent Systems*, vol. 21, no. 5, pp. 88–92, 2006.

[61] D. A. Keim, F. Mansmann, J. Schneidewind, J. Thomas, and H. Ziegler, "Visual Analytics: Scope and Challenges," in *Visual Data Mining*, Springer, 2008, pp. 76–90.

[62] P. Booth, N. Gibbins, and S. Galanis, "Towards a Theory of Analytical Behaviour: A Model of Decision-Making in Visual Analytics," in *Proceedings of the 52nd Hawaii International Conference on System Sciences*, 2019.

[63] W. Fikkert, M. D'Ambros, T. Bierz, and T. Jankun-Kelly, "Interacting with Visualizations," in *Human-Centered Visualization Environments: GI-Dagstuhl Research Seminar, Dagstuhl Castle, Germany, March 5-8, 2006, Revised Lectures*, Springer, 2007, pp. 77–162.

[64] M. O. Ward, G. Grinstein, and D. Keim, *Interactive Data Visualization: Foundations, Techniques, and Applications*. AK Peters/CRC Press, 2010.

[65] A. Saktheeswaran, A. Srinivasan, and J. Stasko, "Touch? Speech? or Touch and Speech? Investigating Multimodal Interaction for Visual Network Exploration and Analysis," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 6, pp. 2168–2179, 2020.

[66] A. Srinivasan, J. Ellemose, P. W. Butcher, P. D. Ritsos, and N. Elmqvist, "Attention-Aware Visualization: Tracking and Responding to User Perception Over Time," *arXiv*, 2024.

[67] J. Thompson, A. Srinivasan, and J. Stasko, "Tangraphe: Interactive Exploration of Network Visualizations using Single Hand, Multi-touch Gestures," in *Proceedings of the 2018 International Conference on Advanced Visual Interfaces*, 2018, pp. 1–5.

[68] D. McDuff, R. e. Kaliouby, K. Kassam, and R. Picard, "Acume: A New Visualization Tool for Understanding Facial Expression and Gesture Data," in *2011 IEEE International Conference on Automatic Face & Gesture Recognition (FG)*, 2011, pp. 591–596.

[69] M. Pohl, M. Smuc, and E. Mayr, "The User Puzzle—Explaining the Interaction with Visual Analytics Systems," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 12, pp. 2908–2916, 2012.

[70] E. T. Brown *et al.*, "Finding Waldo: Learning about Users from their Interactions," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 1663–1672, 2014.

[71] G. A. Miller, "The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information," *Psychological Review*, vol. 101, no. 2, p. 343, 1994.

[72] Z. Liu and J. Heer, "The Effects of Interactive Latency on Exploratory Visual Analysis," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 2122–2131, 2014.

[73] J. Perry, C. D. Janneck, C. Umoja, and W. M. Pottenger, "Supporting Cognitive Models of Sensemaking in Analytics Systems," *DIMACS*, 2009.

[74] P. H. Nguyen, K. Xu, A. Bardill, B. Salman, K. Herd, and B. W. Wong, "SenseMap: Supporting Browser-based Online Sensemaking through Analytic Provenance," in *IEEE VAST*, 2016.

[75] K. Madanagopal, E. D. Ragan, and P. Benjamin, "Analytic Provenance in Practice: The Role of Provenance in Real-World Visualization and Data Analysis Environments," *IEEE Computer Graphics and Applications*, vol. 39, no. 6, pp. 30–45, 2019.

[76] Z. Bylinskii *et al.*, "Learning Visual Importance for Graphic Designs and Data Visualizations," in *ACM UIST*, 2017.

[77] S. Gomez and D. Laidlaw, "Modeling Task Performance for a Crowd of Users from Interaction Histories," in *ACM CHI*, 2012.

[78] A. Walch, M. Schwärzler, C. Luksch, E. Eisemann, and T. Gschwandtner, "Lightguider: Guiding interactive lighting design using suggestions, provenance, and quality visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 1, pp. 569–578, 2020.

[79] A. Endert, P. Fiaux, and C. North, "Semantic Interaction for Sensemaking: Inferring Analytical Reasoning for Model Steering," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 12, pp. 2879–2888, 2012.

[80] L. Bavoil *et al.*, "Vistrails: Enabling Interactive Multiple-View Visualizations," in *VIS 05. IEEE Visualization, 2005.*, IEEE, 2005, pp. 135–142.

[81] Y. B. Shrinivasan, D. Gotz, and J. Lu, "Connecting the Dots in Visual Analysis," in *IEEE VAST*, 2009.

[82] Y. Chen, S. Barlowe, and J. Yang, "Click2annotate: Automated insight externalization with rich semantics," in *2010 IEEE Symposium on Visual Analytics Science and Technology*, IEEE, 2010, pp. 155–162.

[83] S. Gratzl, A. Lex, N. Gehlenborg, N. Cosgrove, and M. Streit, "From Visual Exploration to Storytelling and Back Again," in *Computer Graphics Forum*, 2016.

[84] M. Feng, C. Deng, E. M. Peck, and L. Harrison, "HindSight: Encouraging Exploration through Direct Encoding of Personal Interaction History," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 351–360, 2017.

[85] W. Willett, J. Heer, and M. Agrawala, "Scented widgets: Improving Navigation Cues with Embedded Visualizations," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1129–1136, 2007.

[86] J. E. Block, S. Esmaeili, E. D. Ragan, J. R. Goodall, and G. D. Richardson, "The Influence of Visual Provenance Representations on Strategies in a Collaborative Hand-off Data Analysis Scenario," *IEEE Transactions on Visualization and Computer Graphics*, vol. 29, no. 1, pp. 1113–1123, 2023.

[87] C. Dunne, N. Henry Riche, B. Lee, R. Metoyer, and G. Robertson, "GraphTrail: Analyzing Large Multivariate, Heterogeneous Networks while Supporting Exploration History," in *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, 2012, pp. 1663–1672.

[88] A. Skopik and C. Gutwin, "Improving Revisitation in Fisheye Views with Visit Wear," in *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, 2005, pp. 771–780.

[89] A. Wattenberger, *Footsteps for VS Code*, https://marketplace.visualstudio.com/items?itemName=Wattenberger.footsteps, 2021.

[90] K. Gadhave, Z. Cutler, and A. Lex, "Persist: Persistent and Reusable Interactions in Computational Notebooks," in *Computer Graphics Forum*, 2024.

[91] K. Eckelt, K. Gadhave, A. Lex, and M. Streit, "Loops: Leveraging Provenance and Visualization to Support Exploratory Data Analysis in Notebooks," *OSF Preprint*, 2023.

[92] Y. Ding *et al.*, "reVISit: Supporting Scalable Evaluation of Interactive Visualizations," in *2023 IEEE Visualization and Visual Analytics (VIS)*, 2023, pp. 31–35.

[93] T. Ellkvist, D. Koop, E. W. Anderson, J. Freire, and C. Silva, "Using Provenance to Support Real-Time Collaborative Design of Workflows," in *Provenance and Annotation of Data and Processes*, 2008.

[94] A. Sarvghad and M. Tory, "Exploiting Analysis History to Support Collaborative Data Analysis," in *Proceedings of the 41st Graphics Interface Conference*, 2015, pp. 123–130.

[95] S. K. Badam, Z. Zeng, E. Wall, A. Endert, and N. Elmqvist, "Supporting Team-First Visual Analytics through Group Activity Representations.."

[96] *Google Analytics*, https://marketingplatform.google.com/about/analytics.

[97] *Mixpanel*, https://mixpanel.com.

[98] A. Drachen, "Behavioral Telemetry in Games User Research," *Game User Experience Evaluation*, pp. 135–165, 2015.

[99] T. C. Kohwalter, L. G. P. Murta, and E. W. G. Clua, "Capturing Game Telemetry with Provenance," in *2017 16th Brazilian Symposium on Computer Games and Digital Entertainment (SBGames)*, IEEE, 2017, pp. 66–75.

[100] C.-U. Lim and D. F. Harrell, "Toward Telemetry-driven Analytics for Understanding Players and their Avatars in Videogames," in *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*, 2015, pp. 1175–1180.

[101] P. Cowley, L. Nowell, and J. Scholtz, "Glass Box: An Instrumented Infrastructure for Supporting Human Interaction with Information," in *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*, 2005, pp. 296c–296c.

[102] S. P. Callahan, J. Freire, E. Santos, C. E. Scheidegger, C. T. Silva, and H. T. Vo, "VisTrails: Visualization meets Data Management," in *Proceedings of the 2006 ACM SIGMOD international conference on Management of data*, 2006, pp. 745–747.

[103] W. Aigner, S. Hoffmann, and A. Rind, "EvalBench: A Software Library for Visualization Evaluation," in *Computer Graphics Forum*, Wiley Online Library, vol. 32, 2013, pp. 41–50.

[104] M. Okoe and R. Jianu, "GraphUnit: Evaluating Interactive Graph Visualizations Using Crowdsourcing," in *Computer Graphics Forum*, Wiley Online Library, vol. 34, 2015, pp. 451–460.

[105] Z. Cutler, K. Gadhave, and A. Lex, "Trrack: A Library for Provenance-Tracking in Web-Based Visualizations," in *2020 IEEE Visualization Conference (VIS)*, 2020, pp. 116–120.

[106] *Hotjar*, https://www.hotjar.com.

[107] M. Feng, E. Peck, and L. Harrison, "Patterns and Pace: Quantifying Diverse Exploration Behavior with Visualizations on the Web," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 1, pp. 501–511, 2019.

[108] A. Ottley, R. Garnett, and R. Wan, "Follow The Clicks: Learning and Anticipating Mouse Interactions During Exploratory Data Analysis," in *Computer Graphics Forum*, Wiley Online Library, vol. 38, 2019, pp. 41–52.

[109] D. Gotz, S. Sun, and N. Cao, "Adaptive Contextualization: Combating Bias During High-Dimensional Visualization and Data Selection," in *Proceedings of the 21st International Conference on Intelligent User Interfaces*, 2016, pp. 85–95.

[110] Z. Zhou, X. Wen, Y. Wang, and D. Gotz, "Modeling and Leveraging Analytic Focus During Exploratory Visual Analysis," in *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: Association for Computing Machinery, 2021, ISBN: 9781450380966.

[111] E. Wall, L. M. Blaha, L. Franklin, and A. Endert, "Warning, Bias May Occur: A Proposed Approach to Detecting Cognitive Bias in Interactive Visual Analytics," in *2017 IEEE Conference on Visual Analytics Science and Technology (VAST)*, 2017, pp. 104–115.

[112] J. Heer, J. Mackinlay, C. Stolte, and M. Agrawala, "Graphical Histories for Visualization: Supporting Analysis, Communication, and Evaluation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 6, pp. 1189–1196, 2008.

[113] S. Kaasten, S. Greenberg, and C. Edwards, "How People Recognise Previously Seen Web Pages from Titles, URLs and Thumbnails," in *HCI*, 2002.

[114] W. C. Hill, J. D. Hollan, D. Wroblewski, and T. McCandless, "Edit Wear and Read Wear," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1992, pp. 3–9.

[115] J. Alexander, A. Cockburn, S. Fitchett, C. Gutwin, and S. Greenberg, "Revisiting Read Wear: Analysis, Design, and Evaluation of a Footprints Scrollbar," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2009, pp. 1665–1674.

[116] W. Oliveira, L. M. Ambrósio, R. Braga, V. Ströele, J. M. David, and F. Campos, "A Framework for Provenance Analysis and Visualization," *Procedia Computer Science*, 2017.

[117] T. Kohwalter, T. Oliveira, J. Freire, E. Clua, and L. Murta, "Prov Viewer: A graph-based visualization tool for interactive exploration of provenance data," in *IPAW*, 2016.

[118] P. Chen, B. Plale, Y.-W. Cheah, D. Ghoshal, S. Jensen, and Y. Luo, "Visualization of Network Data Provenance," in *HiPC*, 2012.

[119] C. Dictionary, "guidance meaning: 1. help and advice about how to do something or about how to deal with problems connected with your work, education, or personal relationships.," *Cambridge Dictionary*, 2022.

[120] S. L. Smith and J. N. Mosier, *Guidelines for Designing User Interface Software*. Citeseer, 1986.

[121] A. Dix, J. Finlay, G. D. Abowd, and R. Beale, *Human-Computer Interaction*. Pearson Education, 2003.

[122] J. J. Thomas, *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. IEEE Computer Society, 2005.

[123] R. Engels, "Planning tasks for Knowledge Discovery in Databases; Performing Task-Oriented User-Guidance," in *KDD*, 1996, pp. 170–175.

[124] D. Ceneda, T. Gschwandtner, T. May, S. Miksch, M. Streit, and C. Tominski, "Guidance or No Guidance? A Decision Tree Can Help," in *EuroVA@ EuroVis*, 2018, pp. 19–23.

[125] M. Brehmer and T. Munzner, "A Multi-Level Typology of Abstract Visualization Tasks," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2376–2385, 2013.

[126] J. J. Van Wijk, "Views on Visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 4, pp. 421–432, 2006.

[127] F. Sperrle, A. V. Jeitler, J. Bernard, D. A. Keim, and M. El-Assady, "Learning and Teaching in Co-Adaptive Guidance for Mixed-Initiative Visual Analytics," in *EuroVis Workshop on Visual Analytics (EuroVA)*, 2020, pp. 61–65.

[128] A. Treisman, "Preattentive Processing in Vision," *Computer Vision, Graphics, and Image Processing*, vol. 31, no. 2, pp. 156–177, 1985.

[129] J. Choo, H. Lee, J. Kihm, and H. Park, "iVisClassifier: An Interactive Visual Analytics System for Classification Based on Supervised Dimension Reduction," in *2010 IEEE Symposium on Visual Analytics Science and Technology*, IEEE, 2010, pp. 27–34.

[130] J. Johansson, P. Ljung, M. Jern, and M. Cooper, "Revealing Structure within Clustered Parallel Coordinates Displays," in *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005*, IEEE, 2005, pp. 125–132.

[131] T. May, M. Steiger, J. Davey, and J. Kohlhammer, "Using Signposts for Navigation in Large Graphs," in *Computer Graphics Forum*, Wiley Online Library, vol. 31, 2012, pp. 985–994.

[132] 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*, 2011, pp. 3363–3372.

[133] S. Kandel, R. Parikh, A. Paepcke, J. M. Hellerstein, and J. Heer, "Profiler: Integrated Statistical Analysis and Visualization for Data Quality Assessment," in *Proceedings of the International Working Conference on Advanced Visual Interfaces*, ser. AVI '12, Capri Island, Italy: Association for Computing Machinery, 2012, pp. 547–554, ISBN: 9781450312875.

[134] T. May, A. Bannach, J. Davey, T. Ruppert, and J. Kohlhammer, "Guiding feature subset selection with an interactive visualization," in *2011 IEEE Conference on Visual Analytics Science and Technology (VAST)*, IEEE, 2011, pp. 111–120.

[135] K. Wongsuphasawat *et al.*, "Voyager 2: Augmenting Visual Analysis with Partial View Specifications," in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, 2017, pp. 2648–2659.

[136] F. Bouali, A. Guettala, and G. Venturini, "VizAssist: an interactive user assistant for visual data mining," *The Visual Computer*, vol. 32, no. 11, pp. 1447–1463, 2016.

[137] I. Fujishiro, Y. Takeshima, Y. Ichikawa, and K. Nakamura, "GADGET: Goal-Oriented Application Design Guidance for Modular Visualization Environments," in *Proceedings. Visualization'97 (Cat. No. 97CB36155)*, IEEE, 1997, pp. 245–252.

[138] C. Y. Ip and A. Varshney, "Saliency-assisted navigation of very large landscape images," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 1737–1746, 2011.

[139] T. Munzner, "A Nested Model for Visualization Design and Validation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 6, pp. 921–928, 2009.

[140] S. Miksch and W. Aigner, "A Matter of Time: Applying a Data-Users-Tasks Design Triangle to Visual Analytics of Time-Oriented Data," *Computers & Graphics*, vol. 38, pp. 286–290, 2014.

[141] M. Sedlmair, M. Meyer, and T. Munzner, "Design Study Methodology: Reflections from the Trenches and the Stacks," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 12, pp. 2431–2440, 2012.

[142] M. Sedlmair, M. Meyer, and T. Munzner, "Design Study Methodology: Reflections from the Trenches and the Stacks," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 12, pp. 2431–2440, 2012.

[143] H. Lam, E. Bertini, P. Isenberg, C. Plaisant, and S. Carpendale, "Empirical Studies in Information Visualization: Seven Scenarios," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 9, pp. 1520–1536, 2012.

[144] L. Gitelman, *"Raw Data" is an Oxymoron*. MIT press, 2013.

[145] D. Pyle, *Data Preparation for Data Mining*. Morgan Kaufmann, 1999.

[146] M. Farid, A. Roatis, I. F. Ilyas, H.-F. Hoffmann, and X. Chu, "CLAMS: Bringing Quality to Data Lakes," ser. SIGMOD '16, San Francisco, California, USA: Association for Computing Machinery, 2016, pp. 2089–2092, ISBN: 9781450335317.

[147] D. Deng *et al.*, "The Data Civilizer System.," in *CIDR*, 2017.

[148] R. C. Fernandez, Z. Abedjan, F. Koko, G. Yuan, S. Madden, and M. Stonebraker, "Aurum: A Data Discovery System," in *2018 IEEE 34th International Conference on Data Engineering (ICDE)*, IEEE, 2018, pp. 1001–1012.

[149] I. K. Fodor, "A Survey of Dimension Reduction Techniques," Lawrence Livermore National Lab., CA (US), Tech. Rep., 2002.

[150] G. H. John, R. Kohavi, and K. Pfleger, "Irrelevant Features and the Subset Selection Problem," in *Machine Learning Proceedings 1994*, W. W. Cohen and H. Hirsh, Eds., San Francisco (CA): Morgan Kaufmann, 1994, pp. 121–129, ISBN: 978-1-55860-335-6.

[151] P. Pudil and J. Novovičová, "Novel Methods for Feature Subset Selection with Respect to Problem Knowledge," in *Feature extraction, construction and selection*, Springer, 1998, pp. 101–116.

[152] A. Jović, K. Brkić, and N. Bogunović, "A review of feature selection methods with applications," in *2015 38th International Convention on Information and Commu-*

nication Technology, Electronics and Microelectronics (MIPRO), 2015, pp. 1200–1205.

[153] A. C. Tynan and J. Drayton, "Market Segmentation," *Journal of Marketing Management*, vol. 2, no. 3, pp. 301–335, 1987.

[154] N. R. Council *et al.*, *How People Learn: Brain, Mind, Experience, and School: Expanded edition*. National Academies Press, 2000.

[155] C. Fürber, "Semantic Technologies," in *Data Quality Management with Semantic Technologies*, Springer, 2016, pp. 56–68.

[156] B. Otto, K. M. Hüner, and H. Österle, "Identification of Business Oriented Data Quality Metrics.," in *ICIQ*, 2009, pp. 122–134.

[157] T. N. Herzog, F. J. Scheuren, and W. E. Winkler, "What is Data Quality and Why Should We Care?" In *Data Quality and Record Linkage Techniques*, Springer, 2007, pp. 7–15.

[158] M. Fleckenstein, L. Fellows, and K. Ferrante, "Data Quality," in *Modern Data Strategy*, Springer, 2018.

[159] R. Mahanti, "Data, Data Quality, and Cost of Poor Data Quality," in *Data Quality: Dimensions, Measurement, Strategy, Management, and Governance*, Quality Press, 2019.

[160] L. L. Pipino, Y. W. Lee, and R. Y. Wang, "Data Quality Assessment," *Commun. ACM*, vol. 45, no. 4, pp. 211–218, Apr. 2002.

[161] S. Medić, B. Karlović, and Z. Cindrić, "New Standard ISO 9001: 2015 and its Effect on Organisations," *Interdisciplinary Description of Complex Systems: INDECS*, vol. 14, no. 2, pp. 188–193, 2016.

[162] W. L. Chang, A. Roy, M. Underwood, *et al.*, "NIST Big Data Interoperability Framework: Volume 4, Security and Privacy," 2019.

[163] B. Stein and A. Morrison, "Data Lakes and the Promise of Unsiloed Data," *PricewaterhouseCooper, Technology Forecast: Rethinking integration*, 2014.

[164] T. Nagle, T. C. Redman, and D. Sammon, "Only 3% of Companies' Data Meets Basic Quality Standards," *Harvard Business Review*, vol. 95, no. 5, pp. 2–5, 2017.

[165] N. Sambasivan, S. Kapania, H. Highfill, D. Akrong, P. Paritosh, and L. M. Aroyo, ""Everyone wants to do the model work, not the data work": Data Cascades in

High-Stakes AI," in *proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 2021, pp. 1–15.

[166] 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*, vol. 16, no. 3, pp. 232–256, 2017.

[167] J. Almahmoud, R. DeLine, and S. M. Drucker, "How Teams Communicate about the Quality of ML Models: A Case Study at an International Technology Company," *Proceedings of the ACM on Human-Computer Interaction*, vol. 5, no. GROUP, pp. 1–24, 2021.

[168] M. Kim, T. Zimmermann, R. DeLine, and A. Begel, "Data Scientists in Software Teams: State of the Art and Challenges," *IEEE Transactions on Software Engineering*, vol. 44, no. 11, pp. 1024–1038, 2017.

[169] A. X. Zhang, M. Muller, and D. Wang, "How do Data Science Workers Collaborate? Roles, Workflows, and Tools," *Proceedings of the ACM on Human-Computer Interaction*, vol. 4, no. CSCW1, pp. 1–23, 2020.

[170] I. Drosos, T. Barik, P. J. Guo, R. DeLine, and S. Gulwani, "Wrex: A Unified Programming-by-Example Interaction for Synthesizing Readable Code for Data Scientists," in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, ser. CHI '20, Honolulu, HI, USA: Association for Computing Machinery, 2020, pp. 1–12, ISBN: 9781450367080.

[171] L. Koesten, E. Kacprzak, J. Tennison, and E. Simperl, "Collaborative Practices with Structured Data: Do Tools Support What Users Need?" In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 2019, pp. 1–14.

[172] N. Elmqvist, A. V. Moere, H.-C. Jetter, D. Cernea, H. Reiterer, and T. Jankun-Kelly, "Fluid Interaction for Information Visualization," *Information Visualization*, vol. 10, no. 4, pp. 327–340, 2011.

[173] R. Vaziri, M. Mohsenzadeh, and J. Habibi, "Measuring data quality with weighted metrics," *Total Quality Management & Business Excellence*, vol. 30, no. 5-6, pp. 708–720, 2019.

[174] Microsoft Corporation, *LinkedIn*, 2022.

[175] Google Inc., *Angular*, 2022.

[176] Python Software Foundation, *Python*, 2022.

[177] R. Richards, "Representational State Transfer (REST)," in *Pro PHP XML and Web Services*, Springer, 2006, pp. 633–672.

[178] I. Fette and A. Melnikov, "The Websocket Protocol," Tech. Rep., 2011.

[179] J. Lancar, *Luma*, https://github.com/adobe/experience-platform-dsw-reference/blob/master/datasets/luma/luma_post_extended.csv, 2022.

[180] Tableau, *Tableau*, 2022.

[181] Trifacta, *Wrangler*, 2022.

[182] Microsoft Corporation, *Teams*, 2022.

[183] D. De Cock, "Ames, Iowa: Alternative to the Boston Housing Data as an End of Semester Regression Project," *Journal of Statistics Education*, vol. 19, no. 3, 2011.

[184] G. Gaffney, "Affinity Diagramming," *Retrieved January*, vol. 3, p. 2013, 1999.

[185] R. E. Boyatzis, *Transforming Qualitative Information: Thematic Analysis and Code Development*. SAGE, 1998.

[186] J. Brooke, "SUS: A Quick and Dirty Usability Scale," *Usability Evaluation in Industry*, vol. 189, no. 194, pp. 4–7, 1996.

[187] L. M. Friedman, C. D. Furberg, and D. L. DeMets, "Data Collection and Quality Control," in *Fundamentals of Clinical Trials*, Springer, 2010.

[188] M. A. Schmuckler, "What is Ecological Validity? A Dimensional Analysis," *Infancy*, vol. 2, no. 4, pp. 419–436, 2001.

[189] B. Shneiderman, "The Eyes Have It: A Task By Data Type Taxonomy for Information Visualizations," in *Proceedings 1996 IEEE Symposium on Visual Languages*, 1996, pp. 336–343.

[190] M. Ibrahim and R. Ahmad, "Class diagram extraction from textual requirements using Natural language processing (NLP) techniques," in *2010 Second International Conference on Computer Research and Development*, IEEE, 2010, pp. 200–204.

[191] E. Kaufmann, A. Bernstein, and L. Fischer, "NLP-Reduce: A "naive" but Domain-independent Natural Language Interface for Querying Ontologies," in *4th European Semantic Web Conference ESWC*, 2007, pp. 1–2.

[192] B. J. Grosz, D. E. Appelt, P. A. Martin, and F. C. Pereira, "TEAM: An Experiment in the Design of Transportable Natural-Language Interfaces," *Artificial Intelligence*, vol. 32, no. 2, pp. 173–243, 1987.

[193] J. Herzig, P. K. Nowak, T. Müller, F. Piccinno, and J. M. Eisenschlos, "TaPas: Weakly Supervised Table Parsing via Pre-training," *arXiv*, 2020.

[194] P. He, Y. Mao, K. Chakrabarti, and W. Chen, "X-SQL: Reinforce schema representation with context," *arXiv*, 2019.

[195] C. Wang *et al.*, "Robust Text-to-SQL Generation with Execution-Guided Decoding," *arXiv*, 2018.

[196] V. Zhong, C. Xiong, and R. Socher, "Seq2sql: Generating structured queries from natural language using reinforcement learning," *CoRR*, vol. abs/1709.00103, 2017.

[197] P. Pasupat and P. Liang, "Compositional Semantic Parsing on Semi-Structured Tables," in *Proceedings of IJCNLP*, ACL, 2015, pp. 1470–1480.

[198] F. Li and H. V. Jagadish, "NaLIR: An Interactive Natural Language Interface for Querying Relational Databases," in *Proceedings of the ACM SIGMOD*, 2014, pp. 709–712.

[199] L. Blunschi, C. Jossen, D. Kossmann, M. Mori, and K. Stockinger, "SODA: Generating SQL for Business Users," *Proceedings of the VLDB Endowment*, vol. 5, no. 10, pp. 932–943, 2012.

[200] A.-M. Popescu, O. Etzioni, and H. Kautz, "Towards a Theory of Natural Language Interfaces to Databases," in *Proceedings of IUI*, ACM, 2003, pp. 149–157.

[201] D. H. Kim, E. Hoque, and M. Agrawala, "Answering Questions about Charts and Generating Visual Explanations," in *Proceedings of ACM CHI*, 2020, pp. 1–13.

[202] V. Setlur, M. Tory, and A. Djalali, "Inferencing Underspecified Natural Language Utterances in Visual Analysis," in *Proceedings of ACM IUI*, 2019, pp. 40–51.

[203] B. Yu and C. T. Silva, "FlowSense: A Natural Language Interface for Visual Data Exploration within a Dataflow System," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 1, pp. 1–11, 2020.

[204] A. Srinivasan and J. Stasko, "Orko: Facilitating Multimodal Interaction for Visual Exploration and Analysis of Networks," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 1, pp. 511–521, 2018.

[205] J.-F. Kassel and M. Rohs, "Valletto: A Multimodal Interface for Ubiquitous Visual Analytics," in *ACM CHI '18 Extended Abstracts*, 2018.

[206] V. Setlur, S. E. Battersby, M. Tory, R. Gossweiler, and A. X. Chang, "Eviza: A Natural Language Interface for Visual Analysis," in *Proceedings of ACM UIST*, 2016, pp. 365–377.

[207] T. Gao, M. Dontcheva, E. Adar, Z. Liu, and K. G. Karahalios, "DataTone: Managing Ambiguity in Natural Language Interfaces for Data Visualization," in *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, 2015, pp. 489–500.

[208] Y. Sun, J. Leigh, A. Johnson, and S. Lee, "Articulate: A Semi-automated Model for Translating Natural Language Queries into Meaningful Visualizations," in *Proceedings of the International Symposium on Smart Graphics*, 2010, pp. 184–195.

[209] A. Elgohary, S. Hosseini, and A. H. Awadallah, "Speak to your Parser: Interactive Text-to-SQL with Natural Language Feedback," *arXiv*, 2020.

[210] Y. Su, A. H. Awadallah, M. Khabsa, P. Pantel, M. Gamon, and M. Encarnacion, "Building Natural Language Interfaces to Web APIs," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 2017, pp. 177–186.

[211] A. Kokkalis, P. Vagenas, A. Zervakis, A. Simitsis, G. Koutrika, and Y. Ioannidis, "Logos: A System for Translating Queries into Narratives," in *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, 2012, pp. 673–676.

[212] G. Koutrika, A. Simitsis, and Y. E. Ioannidis, "Explaining Structured Queries in Natural Language," in *2010 IEEE 26th International Conference on Data Engineering (ICDE 2010)*, 2010, pp. 333–344.

[213] A. Simitsis and Y. Ioannidis, "DBMSs Should Talk Back Too," *arXiv*, 2009.

[214] A. Leventidis, J. Zhang, C. Dunne, W. Gatterbauer, H. Jagadish, and M. Riedewald, "QueryVis: Logic-based Diagrams help Users Understand Complicated SQL Queries Faster," in *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, 2020, pp. 2303–2318.

[215] J. Berant, D. Deutch, A. Globerson, T. Milo, and T. Wolfson, "Explaining Queries over Web Tables to Non-Experts," in *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, IEEE, 2019, pp. 1570–1573.

[216] S. Bergamaschi, F. Guerra, M. Interlandi, R. Trillo Lado, Y. Velegrakis, *et al.*, "QUEST: A Keyword Search System for Relational Data based on Semantic and Machine Learning Techniques," 2013.

[217] J. Danaparamita and W. Gatterbauer, "QueryViz: Helping Users Understand SQL Queries and their Patterns," in *Proceedings of the 14th International Conference on Extending Database Technology*, 2011, pp. 558–561.

[218] Y. Su, A. Hassan Awadallah, M. Wang, and R. W. White, "Natural Language Interfaces with Fine-Grained User Interaction: A Case Study on Web APIs," in *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, 2018, pp. 855–864.

[219] T. Yu *et al.*, "Spider: A Large-Scale Human-Labeled Dataset for Complex and Cross-Domain Semantic Parsing and Text-to-SQL Task," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018.

[220] T. Yu *et al.*, "SParC: Cross-Domain Semantic Parsing in Context," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

[221] T. Yu *et al.*, "CoSQL: A Conversational Text-to-SQL Challenge Towards Cross-Domain Natural Language Interfaces to Databases," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, 2019.

[222] K. Lin, B. Bogin, M. Neumann, J. Berant, and M. Gardner, "Grammar-based Neural Text-to-SQL Generation," *arXiv*, 2019.

[223] R. Zhang *et al.*, "Editing-Based SQL Query Generation for Cross-Domain Context-Dependent Questions," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, 2019.

[224] J. Guo *et al.*, "Towards Complex Text-to-SQL in Cross-Domain Database with Intermediate Representation," 2019.

[225] B. Wang, R. Shin, X. Liu, O. Polozov, and M. Richardson, "RAT-SQL: Relation-Aware Schema Encoding and Linking for Text-to-SQL Parsers," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online: Association for Computational Linguistics, Jul. 2020, pp. 7567–7578.

[226] T. Yu *et al.*, "SyntaxSQLNet: Syntax Tree Networks for Complex and Cross-Domain Text-to-SQL Task," 2018.

[227] X. Xu, C. Liu, and D. Song, "SQLNet: Generating Structured Queries From Natural Language Without Reinforcement Learning," *arXiv*, 2017.

[228] *React*, https://reactjs.org.

[229] O. Lojkine et. al., *SQL.js*, https://sql.js.org, 2019.

[230] J. Brooke, "SUS: A Retrospective," *Journal of Usability Studies*, vol. 8, no. 2, pp. 29–40, 2013.

[231] J. Feasel, *SQL Fiddle*, http://sqlfiddle.com/, accessed 2021-01-01, 2021.

[232] W3Schools, *SQL Tryit Editor v1.6*, https://www.w3schools.com/sql/trysql.asp?filename=trysql_select_all, accessed 2020-12-29.

[233] J. Angwin, J. Larson, L. Kirchner, and S. Mattu, "Machine Bias," *ProPublica*, 2016.

[234] G. Gigerenzer, "Fast and Frugal Heuristics: The Tools of Bounded Rationality," *Blackwell Handbook of Judgment and Decision-making*, vol. 62, p. 88, 2004.

[235] E. Wall, L. M. Blaha, C. Paul, and A. Endert, "A Formative Study of Interactive Bias Metrics in Visual Analytics Using Anchoring Bias," *Proceedings of the 17th IFIP TC 13 International Conference on Human-Computer Interaction (INTERACT'19)*, 2019.

[236] I. Cho, R. Wesslen, A. Karduni, S. Santhanam, S. Shaikh, and W. Dou, "The Anchoring Effect in Decision-Making with Visual Analytics," *IEEE Conference on Visual Analytics Science and Technology (VAST)*, 2017.

[237] E. Dimara, A. Bezerianos, and P. Dragicevic, "The Attraction Effect in Information Visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 471–480, 2017.

[238] A. C. Valdez, M. Ziefle, and M. Sedlmair, "Priming and Anchoring Effects in Visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 1, pp. 584–594, 2018.

[239] A. Sarvghad, M. Tory, and N. Mahyar, "Visualizing Dimension Coverage to Support Exploratory Analysis," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 21–30, 2017.

[240] Netflix, Inc., *Netflix*, https://netflix.com, 1997.

[241] M. Allen, D. Poggiali, K. Whitaker, T. R. Marshall, and R. A. Kievit, "Raincloud plots: a multi-platform tool for robust data visualization," *Wellcome Open Research*, 2019.

[242] H. Ishii and B. Ullmer, "Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms," in *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, 1997, pp. 234–241.

[243] V. Hayward, O. R. Astley, M. Cruz-Hernandez, D. Grant, and G. Robles-De-La-Torre, "Haptic interfaces and devices," *Sensor review*, vol. 24, no. 1, pp. 16–29, 2004.

[244] S. Paneels and J. C. Roberts, "Review of Designs for Haptic Data Visualization," *IEEE Transactions on Haptics*, vol. 3, no. 2, pp. 119–137, 2009.

[245] S. Scheggi, A. Talarico, and D. Prattichizzo, "A remote guidance system for blind and visually impaired people via vibrotactile haptic feedback," in *22nd Mediterranean Conference on Control and Automation*, IEEE, 2014.

[246] L. Bonanni and C. Vaucelle, "A Framework for Haptic Psycho-Therapy," *Depression and Anxiety*, vol. 12, p. 24, 2006.

[247] G. Changeon, D. Graeff, M. Anastassova, and J. Lozada, "Tactile Emotions: A Vibrotactile Tactile Gamepad for Transmitting Emotional Messages to Children with Autism," in *Haptics: Perception, Devices, Mobility, and Communication: International Conference, EuroHaptics 2012, Tampere, Finland, June 13-15, 2012. Proceedings, Part I*, Springer, 2012, pp. 79–90.

[248] Y. Do, L. T. Hoang, J. W. Park, G. Abowd, and S. Das, "Spidey Sense: Designing Wrist-Mounted Affective Haptics for Communicating Cybersecurity Warnings," in *Designing Interactive Systems Conference 2021*, ACM, 2021.

[249] S. Ooms, M. Lee, P. Ceasar, and A. Ali, "FeelTheNews: Augmenting Affective Perceptions of News Videos with Thermal and Vibrotactile Stimulation," in *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems*, ACM, 2023.

[250] M. Akamatsu, S. Sato, and I. S. MacKenzie, "Multimodal Mouse: A Mouse-Type Device with Tactile and Force Display," *Presence: Teleoperators and Virtual Environments*, vol. 3, no. 1, pp. 73–80, 1994.

[251] G. Yun, M. Mun, J. Lee, D.-G. Kim, H. Z. Tan, and S. Choi, "Generating Real-Time, Selective, and Multimodal Haptic Effects from Sound for Gaming Experience Enhancement," in *ACM CHI*, 2023.

[252] K.-U. Kyung, H. Choi, K. Dong-Soo, and S. Seung-Woo, "Interactive Mouse Systems Providing Haptic Feedback During the Exploration in Virtual Environment," in *2004 International Symposium on Computer and Information Sciences*, ACM, 2004, pp. 136–146.

[253] J. Terry and H. S. Hsiao, "Tactile feedback in a computer mouse," in *14th Annual Northeast Bioengineering Conference*, IEEE, 1988.

[254] W. Han and H.-J. Schulz, "Exploring Vibrotactile Cues for Interactive Guidance in Data Visualization," in *2020 International Symposium on Visual Information Communication and Interaction (VINCI)*, ACM, 2020, pp. 1–10.

[255] SteelSeries, *SteelSeries Gamesense SDK*, Jun. 2015.

[256] B. Saket, C. Prasojo, Y. Huang, and S. Zhao, "Designing an Effective Vibration-Based Notification Interface for Mobile Phones," in *Proceedings of the 2013 conference on Computer supported cooperative work*, 2013, pp. 149–1504.

[257] M. Obrist, S. Subramanian, E. Gatti, B. Long, and T. Carter, "Emotions Mediated Through Mid-Air Haptics," in *Proceedings of the 33rd annual ACM Conference on Human Factors in Computing Systems*, 2015, pp. 2053–2062.

[258] D. Wang, K. Ohnishi, and W. Xu, "Multimodal Haptic Display for Virtual Reality: A Survey," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 1, pp. 610–623, 2019.

[259] M. A. Baumann, K. E. MacLean, T. W. Hazelton, and A. McKay, "Emulating Human Attention-Getting Practices with Wearable Haptics," in *2010 IEEE Haptics Symposium*, IEEE, 2010, pp. 149–156.

[260] A. Satyanarayan, D. Moritz, K. Wongsuphasawat, and J. Heer, "Vega-Lite: A Grammar of Interactive Graphics," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 341–350, 2017.

[261] C. Nobre, D. Wootton, Z. Cutler, L. Harrison, H. Pfister, and A. Lex, "reVISit: Looking under the hood of interactive visualization studies," in *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 2021, pp. 1–13.

[262] R. S. Nickerson, "Confirmation Bias: A Ubiquitous Phenomenon in Many Guises," *Review of General Psychology*, 1998.

[263] R. C. Atkinson and R. M. Shiffrin, "Human Memory: A Proposed System and its Control Processes," in *Psychology of Learning and Motivation*, 1968.

[264] A. Tversky and D. Kahneman, "Availability: A Heuristic for Judging Frequency and Probability," *Cognitive Psychology*, 1973.

[265] 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, vol. 25, no. 1, pp. 672–681, 2019.

[266] P. Baudisch *et al.*, "Phosphor: Explaining Transitions in the User Interface Using Afterglow Effects," in *Proceedings of the 19th annual ACM symposium on User Interface Software and Technology*, 2006, pp. 169–178.

[267] J. Hill and C. Gutwin, "Awareness Support in a Groupware Widget Toolkit," in *Proceedings of the 2003 ACM International Conference on Supporting Group Work*, 2003, pp. 258–267.

[268] *Bootstrap*, https://getbootstrap.com, 2024.

[269] D. Cernea, C. Weber, A. Ebert, and A. Kerren, "Emotion Scents: A Method of Representing User Emotions on GUI Widgets," in *Visualization and Data Analysis 2013*, SPIE, vol. 8654, 2013, pp. 168–181.

[270] P. Vaithilingam, E. L. Glassman, J. P. Inala, and C. Wang, "DynaVis: Dynamically Synthesized UI Widgets for Visualization Editing," in *Proceedings of the CHI Conference on Human Factors in Computing Systems*, ser. CHI '24, New York, NY, USA: Association for Computing Machinery, 2024.

[271] *Draw.io*, https://www.draw.io.

[272] B. Shneiderman, "Dynamic Queries for Visual Information Seeking," *IEEE software*, vol. 11, no. 6, pp. 70–77, 1994.

[273] C. Williamson and B. Shneiderman, "The Dynamic HomeFinder: Evaluating Dynamic Queries in a Real-Estate Information Exploration System," in *Proceedings of the 15th annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1992, pp. 338–346.

[274] J. Heer, M. Agrawala, and W. Willett, "Generalized Selection via Interactive Query Relaxation," in *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, 2008, pp. 959–968.

[275] A. F. Blackwell *et al.*, "Cognitive Dimensions of Notations: Design Tools for Cognitive Technology," in *Cognitive Technology: Instruments of Mind: 4th International Conference, CT 2001 Coventry, UK, August 6–9, 2001 Proceedings*, Springer, 2001, pp. 325–341.

[276] *Serebii.net*, http://serebii.net.

[277] A. Strauss and J. Corbin, *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*. Sage Publications, 1998.

[278] L. McInnes, J. Healy, and J. Melville, "UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction," *arXiv*, 2018.

[279] *PrimeNG*, https://www.primefaces.org/primeng, 2024.

[280] *ngx-slider*, https://github.com/angular-slider/ngx-slider, 2018.

[281] D. A. Keim, "Information Visualization and Visual Data Mining," *IEEE Transactions on Visualization and Computer Graphics*, vol. 8, no. 1, pp. 1–8, 2002.

[282] R. C. Roberts, R. S. Laramee, G. A. Smith, P. Brookes, and T. D'Cruze, "Smart Brushing for Parallel Coordinates," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 3, pp. 1575–1590, 2019.

[283] Prosus, *StackOverflow*, https://stackoverflow.com, 2008.

[284] K. Vodrahalli, R. Daneshjou, T. Gerstenberg, and J. Zou, "Do Humans Trust Advice More if it Comes from AI? An Analysis of Human-AI Interactions," in *Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society*, 2022, pp. 763–777.

[285] L. Chong, G. Zhang, K. Goucher-Lambert, K. Kotovsky, and J. Cagan, "Human confidence in artificial intelligence and in themselves: The evolution and impact of confidence on adoption of AI advice," *Computers in Human Behavior*, vol. 127, p. 107 018, 2022.

[286] J. M. Logg, J. A. Minson, and D. A. Moore, "Algorithm appreciation: People prefer algorithmic to human judgment," *Organizational Behavior and Human Decision Processes*, vol. 151, pp. 90–103, 2019.

[287] I. Yaniv, "Receiving other people's advice: Influence and benefit," *Organizational Behavior and Human Decision processes*, vol. 93, no. 1, pp. 1–13, 2004.

[288] E. K. Lai, "Expert Advice for Amateurs," *Journal of Economic Behavior & Organization*, vol. 103, pp. 1–16, 2014.

[289] D. Läpple and B. L. Barham, "How do learning ability, advice from experts and peers shape decision making?" *Journal of Behavioral and Experimental Economics*, vol. 80, pp. 92–107, 2019.

[290] M. Jacobs, M. F. Pradier, T. H. McCoy Jr, R. H. Perlis, F. Doshi-Velez, and K. Z. Gajos, "How machine-learning recommendations influence clinician treatment selections: The example of antidepressant selection," *Translational Psychiatry*, vol. 11, no. 1, p. 108, 2021.

[291] S. Gaube *et al.*, "Do as AI say: susceptibility in deployment of clinical decision-aids," *NPJ Digital Medicine*, vol. 4, no. 1, p. 31, 2021.

[292] K. Z. Gajos and L. Mamykina, "Do People Engage Cognitively with AI? Impact of AI Assistance on Incidental Learning," in *Proceedings of the 27th International Conference on Intelligent User Interfaces*, 2022, pp. 794–806.

[293] Z. Lu and M. Yin, "Human Reliance on Machine Learning Models when Performance Feedback is Limited: Heuristics and Risks," in *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 2021, pp. 1–16.

[294] Z. Buçinca, M. B. Malaya, and K. Z. Gajos, "To Trust or to Think: Cognitive Forcing Functions Can Reduce Overreliance on AI in AI-assisted Decision-making," *Proceedings of the ACM on Human-Computer Interaction*, vol. 5, no. CSCW1, 2021.

[295] A. Kim, M. Yang, and J. Zhang, "When Algorithms Err: Differential Impact of Early vs. Late Errors on Users' Reliance on Algorithms," *ACM Transactions on Computer-Human Interaction*, vol. 30, no. 1, pp. 1–36, 2023.

[296] C.-W. Chiang and M. Yin, "You'd Better Stop! Understanding Human Reliance on Machine Learning Models under Covariate Shift," in *Proceedings of the 13th ACM Web Science Conference 2021*, 2021, pp. 120–129.

[297] S. Kashanj, X. Wang, and C. Perin, "Visualizations on Smart Watches while Running: It Actually Helps!" In *2024 IEEE Visualization Conference (VIS)*, 2024.

[298] B. Saket, H. Kim, E. T. Brown, and A. Endert, "Visualization by Demonstration: An Interaction Paradigm for Visual Data Exploration," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 331–340, 2017.

[299] Microsoft Corporation, *PowerPoint*, https://www.microsoft.com/en-us/microsoft-365/powerpoint, 1987.

[300] B. Shneiderman and C. Plaisant, "Strategies for Evaluating Information Visualization Tools: Multi-dimensional In-depth Long-term Case Studies," ser. BELIV '06, 2006, pp. 1–7.

VITA

Arpit Narechania was born on June 15, 1993, in Mumbai to parents Ajay and Rita. He went to elementary school at Bombay Cambridge Gurukul, middle and high school at C.N.M. School, and junior college at Sathaye Junior College, all in Mumbai. In 2015, he received his undergraduate degree in Mechanical Engineering with a minor in Device Materials from Indian Institute of Technology (IIT) Mandi, located in Kamand. During his undergraduate years (2011–2015), he also interned at Mercedes Benz Research and Development India (MBRDI) in Bengaluru, GEMTECH in Mumbai, and Genius Electronics in Mumbai. In 2014, he studied production engineering for one semester as an exchange student at RWTH Aachen University in Germany. Between 2015 and 2018, Arpit worked as a software engineer $\rightarrow$ full-stack data scientist at Khosla Labs, a start-up incubator in Bengaluru. During this period, he was also a founding member of multiple startups (LoansApp, RaveAnalytics, IntuitionAI), studied courses at the Indian Institute of Science Bengaluru (IISc), and volunteered with eGovernments Foundation and with Novopay Solutions (now Trustt). He also volunteered with the Commissioner and Director of Municipal Administration (CDMA), State Government of Andhra Pradesh and the Telecom Regulatory Authority of India (TRAI), Government of India. In 2018, Arpit moved to the U.S. to pursue graduate school, and in 2024, he received his Ph.D. in Computer Science from The Georgia Institute of Technology (GaTech) in Atlanta. During his Ph.D. years, he also co-founded an ed-tech startup (Lumovia), worked with technology leaders including Mu Sigma, Microsoft Research, Adobe Research, and Ford Motor Company through consulting, internships, and research assistantships. Arpit is currently living in Atlanta with his wife, Nupur. Beginning 2025, he will be a tenure-track assistant professor in the Computer Science and Engineering department at The Hong Kong University of Science and Technology (HKUST). Get to know more about Arpit on his website: https://arpitnarechania.github.io.