

ABSTRACT

Title of Dissertation: **EXAMPLE-AIDED DESIGN:
SUPPORTING VISUALIZATION IDEATION
AND IMPLEMENTATION**

Hannah K. Bako
Doctor of Philosophy, 2025

Dissertation Directed by: Professor Zhicheng Liu
Department of Computer Science

Data visualizations are a powerful tool for understanding data and conveying complex information to audiences with diverse technical backgrounds. However, designers often struggle to choose and craft visual representations that fit their analytical and communicative goals. In practice, when designers cannot develop design ideas, they often use examples of past visualizations to find inspiration or conceptualize the space of possible designs, offering an avenue for designers to understand *what designs are possible* for the data they are working with and *how they can author* these designs. Yet, many existing visualization authoring tools assume designers always have a fully specified design intent, completely overlooking the use of examples as cognitive scaffolds for the visualization design process.

This dissertation systematically investigates example-aided design in data visualization. First, through semi-structured interviews with novice and expert visualization designers, this dissertation studies what, how, and why designers engage in example-aided design. The results of

this interview study reveal designers' motivations and processes for engaging in example-aided design, highlighting gaps between designers' practices and current visualization tools. Specifically, there is a lack of adequate support for retrieving diverse design examples and enabling designers' example adoption strategies (i.e., *select & merge*, *replicate & modify*, *trial & error*). Second, this dissertation presents a two-part controlled experiment to identify and measure the factors that modulate curated examples' influence on design outcomes and investigate the relationship between example browsing patterns and design outcomes. The findings from this experiment show that design outcomes are influenced not only by data schema similarities between examples and the working dataset but also by the timing of example exposure during ideation and the visual complexity of the examples. Furthermore, connections exist between designers' strategies to filter and browse example galleries and the quantity and variety of designs they produce. Finally, towards translating these findings of how designers engage in example-aided design to improve design tools, this dissertation presents Mirny, a web-based tool that supports example-based iterative prototyping of interactive D3 visualizations, enabling the process of replicating and modifying design examples to suit a designer's design and analytical contexts.

By integrating empirical insights with a practical authoring environment, this work advances our understanding of how to harness examples for more expressive visualization creation and lays a foundation for a new avenue of research on example-aided design for next-generation data visualization design tools.

EXAMPLE-AIDED DESIGN: SUPPORTING VISUALIZATION IDEATION
AND IMPLEMENTATION

by

Hannah K. Bako

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2025

Advisory Committee:

Professor Zhicheng Liu, Chair/Advisor
Professor Leilani Battle
Professor Eun Kyoung Choe
Professor Joel Chan
Professor Huaishu Peng
Professor Dave Levin

© Copyright by
Hannah K. Bako
2025

Dedication

In loving memory of Emmanuel K. Bako

May your light continue to shine for the generations to come.

Acknowledgments

This journey began long before I knew who I was or my life purpose. A calling had been established, and today, I sit in the fulfillment of a portion of that promise. All thanks and praise be to God Most High (Jeremiah 1:5).

Dr. Liu, when you took me into your lab, I do not think we would have imagined the twists and turns this dissertation would take. Through it all, your guidance, calmness, and sharp insight steadied and encouraged me through all the chaos. I could not have asked for a better advisor, and I am excited to continue your legacy as I begin my academic career.

I started my PhD with no idea what exactly research truly entailed, and on paper, I had no substantial experience to justify being here. I was simply a girl with a dream and a great deal of perseverance. It is not lost on me that my PhD journey at UMD would not have started without someone taking a chance on me. Dr. Leilani Battle, thank you for taking that chance. Thank you for remaining a steady presence in my academic journey. I will not forget your kindness and the countless advice you gave me over the years. I also want to express my gratitude to Dr. Choe, Dr. Chan, Dr. Peng, and Dr. Levin for your invaluable feedback on this dissertation.

I'm grateful to all my current and ex-labmates Sneha Ghathani, Chen Chen, Kazi Tasnim Zinat, Yuexi Chen, Hyemi Song, Xiaoyu Liu, and Deepthi Raghunandan. You all are a big part of this process. I also want to say thank you to all the students I've worked with on various projects, without your help I would not have even a single page in this dissertation: Anu Faboro, Favor

Nerrise, Alisha Verma, Bissaka Kenah, Mahreen Haider, Xinyi Liu, Grace Ko, Ariya Mehra, Arshnoor Buthani, and John Hooker.

Omolewa, Bunmi, Priscilla, Jane, Tobi, Wheme, Lolade, and Adaobi, thank you for your love and unwavering support. The entire Umu Vibes contingency, when I thought of quitting and casually mentioned it during one of our games night sessions, from that moment, you all stopped calling me Hannah and started calling me “Dr. Bako”. What might have seemed like a small gesture or a joke served as a constant reminder to me of what lay ahead and that giving up was never an option. I’m eternally grateful to you all for this. Kamal, I would be remiss not to dedicate a section to you. You’ve walked with me through every stage of this journey, witnessed the good, the bad, and the ugly, stood by me through the highs and lows, listened to every complaint, brainstormed with me, offered pep talks, and cheered me on every step of the way. I know in my heart that I could not have done this without you.

Lastly, but by far the most important, my family. To my parents and siblings, Henry, Edith, Emmanuel, and Joseph, without you, I would be nowhere. You are the wind beneath my wings. Your support and investment in my life are irreplaceable. I love you all dearly.

Table of Contents

Dedication	ii
Acknowledgements	iii
Table of Contents	v
List of Tables	viii
List of Figures	ix
Chapter 1: Introduction	1
1.1 Thesis Statement	2
1.2 Thesis Organization and Contributing Papers	4
Chapter 2: Background and Related Work	8
2.1 Example Usage in Design	8
2.1.1 Idea Generation Using Examples	8
2.1.2 Example Search and Storage Strategies	10
2.1.3 Example Usage and Influence on Creativity	12
2.2 Visualization Authoring	16
2.2.1 Programming Tools	16
2.2.2 Direct Manipulation Tools	17
2.2.3 Visualization Recommendation Tools	17
2.2.4 Example usage in Visualization Authoring	18
Chapter 3: The Role of Examples in Visualization Design	19
3.1 Introduction	19
3.2 Methods	21
3.2.1 Participants	21
3.2.2 Interviews	23
3.2.3 Analysis	24
3.3 The search for examples	25
3.3.1 Why and When are Examples Needed?	25
3.3.2 What Tools and Resources are Used to Find Examples?	28
3.3.3 How do Designers Search for Examples?	32
3.3.4 What Challenges Arise When Searching for Examples?	35
3.4 Using Examples	36

3.4.1	How do Designers Use Examples?	36
3.4.2	Reusing the Same Example	40
3.4.3	Tools Designers Use	41
3.4.4	Challenges Faced While Using Examples	42
3.5	Cross Cutting Themes and Activities	43
3.5.1	Selecting Examples	44
3.5.2	Curating Examples	48
3.5.3	Design Fixation and Its Effects on Example Use	50
3.5.4	Reflections on Desired Tools	52
3.6	Discussion	53
3.6.1	Limitations	56
3.7	Summary	57
Chapter 4:	Factors that Influence Example-Aided Visualization Design	58
4.1	Introduction	58
4.2	Factors and Outcomes of Interest in This Study	60
4.3	Study Design	61
4.3.1	Experiment Overview	61
4.3.2	Experiment Conditions	63
4.3.3	Examples Corpus and Interface	65
4.3.4	Participants	67
4.3.5	Experiment Procedure	68
4.4	Quantifying Curated Examples and Design Outcomes	70
4.4.1	Coding Visualization Design Components	70
4.4.2	Deriving Properties of Curated Example Set	72
4.5	Measuring Design Outcomes and its Modulators	77
4.5.1	Data Description	77
4.5.2	Influence of Timing and Curated Examples On the Quantity and Variety of Designs	78
4.5.3	Influence of Curated Examples on Idea Transfer	80
4.6	Why and How Designers Incorporate Examples	82
4.6.1	What Example Features Influence Designers?	83
4.6.2	Forms of Idea Transfer	85
4.6.3	Reflecting on Implemented Designs	87
4.7	Discussion	88
4.7.1	Limitations of Our Study	91
4.8	Summary	91
Chapter 5:	Ideation Strategies in Example-Aided Visualization Design	93
5.1	Introduction	93
5.2	Operationalization of Example-Aided Ideation	96
5.3	Analysis Methods	100
5.3.1	Analyzing Temporal Patterns in Browsing Behavior	100
5.3.2	Example Filtering Strategies	102
5.3.3	Example-Aided Ideation Strategies	103

5.4	RQ1: Effect of Timing on Ideation Strategies	104
5.4.1	Filtering Strategies	105
5.4.2	Ideation Strategies	107
5.5	RQ2: Quantity of Designs	110
5.5.1	Cohort Overview	110
5.5.2	Filtering Strategies by Sketch Quantity	111
5.5.3	Ideation Strategies by Sketch Quantity	113
5.6	RQ2: Variety of Sketches	116
5.6.1	Filtering Strategies by Variety of Sketches	117
5.6.2	Ideation Strategies by Variety of Sketches	118
5.7	Discussion	121
5.7.1	Limitations	124
5.8	Summary	124
Chapter 6:	Mirny: An Example-Driven Prototyping Tool for Interactive D3 Visualiza- tions	126
6.1	Introduction	126
6.2	Motivating Usage Scenario for Mirny	129
6.3	Preliminary Study: Lessons from Analyzing D3 Examples Online	131
6.3.1	Design Considerations	134
6.4	Mirny System Overview	135
6.5	Recommending Code Snippets to Make D3 Visualizations Interactive	137
6.5.1	Predicting Which Interactions to Implement	138
6.5.2	MDP Example.	140
6.5.3	Preliminary Model Accuracy Results	143
6.5.4	Predicting Visualization types	143
6.6	Augmenting User Code	144
6.7	Evaluation: User Study Design	149
6.8	Evaluation: Analysis & Results	152
6.8.1	Measuring Session Duration and Design Prototypes	154
6.8.2	Classifying Template Usage and Modification	155
6.8.3	Participant Feedback	157
6.9	Discussion	159
6.10	Summary	163
Chapter 7:	Conclusion	164
7.1	Future Research Directions	165
7.1.1	Exploring the Dynamics of Example-Aided Creative Ideation	165
7.1.2	Advancing Human–AI Co-Design for Visualization Authoring	167
7.1.3	Integrating Example, Domain, and Perceptual Knowledge in Visualiza- tion Design	168
Bibliography		170

List of Tables

4.1	Example selection criteria and definitions	66
5.1	Participants split for quantity of sketches.	111
5.2	Participants split for variety of sketches.	111
5.3	Significant differences across quantity cohorts for tag states. z-scores are shown only for significant results (*).	112
5.4	Comparison of ideation strategies by quantity of sketches.	113
5.5	Statistically significant pairwise posthoc analyses results across quantity cohorts for durations.	114
5.6	Significant differences across all variety cohorts (z-scores shown only for significant results, marked with an asterisk).	117
5.7	Statistically significant pairwise posthoc analyses results across variety cohorts for action durations	119
5.8	Comparison of ideation strategies by variety of sketches	120

List of Figures

3.1	The top two images are visualizations created by P7, showing the authors and co-authors of published research funded by NASA and projects they have worked on. The image on the bottom is a snapshot of a futuristic design in a movie that inspired P7.	29
3.2	An example showing how ideas are translated from multiple examples into a single design. Visualization elements like circles with images of race car drivers are selected from example A and merged with paths from the Sankey chart in example B. The participant sketches their design idea in C and includes a note about radial bar charts as another source of inspiration. In the final design, we can identify relevant design elements from the two examples and initial sketch.	37
3.3	An example of reusing a single example to generate multiple data representations. Here, P3 reuses a base map layout (A) to visualize data on building structures in a specific geographic location (B).	40
4.1	A snapshot of the experiment interface provided to participants during the experiment. (A) select tags to filter examples, (B) search for tags with auto-complete suggestions, (C) browse examples in the main panel, and (D) bookmark relevant examples.	64
4.2	A depiction of our experiment protocol focusing on the initial interview. After familiarizing themselves with the scenario description and dataset, participants begin the brainstorming process. Depending on the <i>timing</i> condition, participants are either provided with examples at the start of the process (T_{srt}) or asked to brainstorm ideas before introducing examples(T_{del}).	69
4.3	An illustration of how example properties are calculated for a set of curated examples. Please read 4.4.2 for more details.	71
4.4	Distribution of scores for properties of curated examples based on the different experiment conditions in our study.	76
4.5	Overview of components found in curated examples and design sketches produced by participants. “Other” contains the sum of all components with less than 5% of each group.	78
4.6	An overview of the style-related components found in curated examples and design sketches created by participants.	79
4.7	A depiction of the forms of idea transfer identified in sketches produced by three participants who bookmarked the same example. In design replication, the entire example is copied, whereas, in partial transfer, participants copy either encodings or styles from the example.	83

5.1	A screenshot of the web interface used to instrument the example-aided ideation process. The interface contains a filter panel (A), a search bar (B), and a main view to explore relevant examples (C). Participants can click on “example source” to expand individual examples (F) and toggle (D) to bookmark relevant examples. (E) and (G) are used to review bookmarked examples.	95
5.2	An overview of our operationalization of the operations and actions involved in the example-aided ideation activity.	96
5.3	Frequency distribution across all 16 states by timing condition. The “Blank” state is a state where no tags are selected.	104
5.4	An example of filtering evolution over time. Presented in this figure are the filtering activities of participants in the start condition, demonstrating dominant anchors on data and topic tag groups.	107
5.5	Action frequency and duration across different timing conditions	107
5.6	Transition distribution for start and delayed timing conditions.	108
5.7	Participant distribution across quantity and variety of sketches.	111
5.8	Frequency distribution for the quantity of sketches cohorts.	112
5.9	Frequency in percentage and duration of actions in seconds (logarithmic scale) for each sketch quantity cohort	115
5.10	Action Transition distributions (dist.) for the quantity (row 1) and variety (row 2) cohorts.	116
5.11	Frequency distribution for all 16 states across the low, medium, and high sketch variety cohorts.	118
5.12	Frequency and duration of actions in seconds (logarithmic scale) for each variety cohort.	121
6.1	An example of an interactive visualization designed using Mirny. A user can select a visualization such as a <i>scatterplot</i> to create from the Template Panel (A) . The code responsible for creating the <i>scatterplot</i> is displayed within the Editor (B) and the corresponding visualization is rendered in the Visualization Panel (C) . A list of recommended interactions is displayed to the user in the Recommendation Panel (D) . A user can then choose a recommended interaction such as a <i>Hover</i> and the code responsible for implementing the interaction is added to the interface and highlighted for the user to view and modify (E). Once a user is satisfied with their visualization, they can export the code and visualization using the Controls (F) and (G).	131
6.2	Results from the analysis carried out on the three example corpus [17] (a) presents the frequency of the top visualization types and their corresponding interactions observed in our datasets. Visualizations with 1% and less frequency are captured in the “others” category. (b) shows the number of times different interaction types were implemented for each visualization type in our corpus.	132
6.3	Mirny’s architecture. The Mirny interface captures user requests for interactions and recommendations. Using the Code Fitter, it generates the appropriate D3 specification and renders the results for the user via the Interface. Requests are also sent to the prediction model to receive recommended interactions for the user.	137

6.4	The Mirny recommendation engine models user interaction state using a Markov decision process (or MDP) model. Here is a simplified example of how the Mirny MDP model is structured with two interactions (zoom and hover), explained in subsection 6.5.2.	141
6.5	Results of the LOO cross-validation of the recommendation model per interaction type.	141
6.6	Given a recommended code snippet, e.g., a template for “Hover” interactions in D3, the Mirny system supports <i>automatic code augmentation</i> , where the user’s D3 code is mapped to an Abstract Syntax Tree (AST) that Mirny uses to populate and insert the code snippet. In (A), Mirny detects the “fill” variable in the user’s current code (highlighted in orange). In (B), Mirny uses the “fill” variable to automatically populate the “Hover” template (highlighted in green). Finally, in (C), Mirny inserts the populated “Hover” template into the user’s code at a detected anchor point in the AST (marked with a star).	145
6.7	Count of interactive visualizations produced by participants.	153
6.8	Distribution of time taken to program a single interaction.	153
6.9	Distribution of post-study questionnaire responses.	154
6.10	Examples of visualization designs created by participants during the experiment. (A), (B), (C), and (E) are visualizations of the Cherry Blossoms dataset depicting different attributes in the dataset such as yearly records of January Snow and February Snow (A), day of peak bloom vs January Temperature (B), etc. (D) and (F) are visualizations of attributes from the cars dataset.	159

Chapter 1: Introduction

Visualizations leverage our perceptual and cognitive abilities to interpret visual patterns and translate them into meaningful insight. As such, visualizations play a dual role in data analysis and communication, i.e., exploring and uncovering patterns in complex data during exploratory analysis and translating insights to diverse audiences. Effective visualization design is therefore essential for data analysis and communication, especially in today’s world, where the inaccurate representation of data can easily lead to misinterpretation and misinformation.

Determining an effective visualization design is not straightforward, as it is not always obvious which visual representation is the most appropriate for a designer’s analytical or communicative goals. As a result, visualization designers must carefully explore multiple design candidates and consider a number of factors, such as appropriate visualization heuristics, how the visualization will be consumed, and the target audience’s technical experience. Consequently, navigating the space of appropriate designs is a hard and tedious process, often requiring manual implementation and evaluation of multiple design alternatives before selecting a final design. This process is particularly challenging for designers who may not have the technical know-how or time to realize their designs. We use the term “designers” broadly: while “designers” often refers to individuals who specialize in the profession of design, in this dissertation, this term refers to individuals who create any form of data visualizations for various purposes, including

but not limited to journalists, data analysts, and UX designers.

Towards alleviating these challenges, visualization designers often resort to using “*examples*” of existing visualization designs (e.g, images or code shared on online repositories) as sources of inspiration. These examples not only help designers *find inspiring ideas* but also provide guidance on *how to implement their design ideas*. Research in creativity and design has long recognized examples as vital cognitive scaffolds in creative work [68, 134, 194], yet studies on the use and adoption of examples into the visualization design process are overlooked in visualization literature. As such, there is limited understanding of how examples influence visualization designers, how they integrate examples into their designs, and the challenges designers face during example-aided design.

Furthermore, current visualization tools focus on facilitating the implementation of visualizations from scratch (via programming [36, 184] or direct manipulation [119, 182]) or recommending a limited set of visualizations to designers [64, 99, 217]. These methods assume that a designer always has a specific design intent in mind, overlooking this crucial behavior of repurposing ideas from examples into visualization designs. This over-focus on creating designs from scratch directly contradicts designers’ workflow of creating visualizations, which typically starts from adapting an example to their data. Therefore, there is a need for improved tools to support the exploration and implementation of data visualization design candidates.

1.1 Thesis Statement

This dissertation investigates the process of **example-aided visualization design**, where examples act as stimuli and cognitive scaffolds for generating new visualization designs. Visual-

izations in this context extend beyond common statistical graphs such as bar charts or scatterplots to include the more bespoke (e.g., chord diagrams, Network charts) and unconventional (e.g, data art) designs.

Our focus is on uncovering the creative process underpinning example-aided data visualization design. Thus, the central research question in this dissertation is *How can we **understand**, **measure**, and **support** example-aided design in data visualization authoring?* Towards answering this question, this dissertation makes four core contributions:

1. A qualitative characterization of designers’ motivations, strategies, and challenges in example-aided design (***Understand***).
2. A controlled experiment that quantitatively measures the factors modulating an example’s influence on design outcomes (***Measure***).
3. An analysis of ideation strategies that lead to increased quantity and variety of visualization designs (***Measure***).
4. A design tool that automates example adaptation for iterative visualization prototyping in D3 (***Support***).

Together, these contributions provide a holistic view of the example-aided design process, from the drivers behind example use to the empirical impact of specific example attributes and behaviors and tailored tool support to enhance example exploration and adaptation. By linking in-depth qualitative insights, controlled experimental findings, analyses of designers’ interaction patterns, and the development of an authoring environment, this dissertation uncovers the foundations of example usage in data visualization design, towards establishing a comprehensive framework for example-aided visualization design.

1.2 Thesis Organization and Contributing Papers

Chapter 2: Background and Related Work. This chapter presents a comprehensive overview of existing literature on the idea generation process using examples, example search and curation, and how examples influence creativity. It also discusses core concepts and techniques for visualization authoring tools and recommendation systems, connecting the significance of this work to the literature in all of these fields.

Chapter 3: The Role of Examples in Visualization Design. Although the use of examples by visualization designers is known anecdotally, visualization literature offers no systematic account of such “example-aided design” process. [chapter 3](#) contributes the first formal study of the role examples play in visualization design, uncovering how, why, what, and when designers engage with examples. Using a semi-structured interview of 30 visualization designers, we examine (1) what types of examples designers seek, (2) why they choose particular visualizations, (3) when examples enter their workflow, and (4) how they adapt them—employing strategies such as select & merge, replicate & modify, and trial & error—to inform their own designs. These insights highlight both designers’ current workflows to search for and use examples, and the limitations of existing visualization tools, which lack adequate support for example retrieval and adaptation.

Contributing Papers:

H. K. Bako, X. Liu, L. Battle, and Z. Liu. Understanding how designers find and use data visualization examples. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):1048–1058, 2023

Chapter 4: Factors that Influence Example-Aided Visualization Design. Towards developing

computational means to support example adaptation strategies, it is essential to first understand how example-aided ideation influences design outcomes and what factors may modulate this influence. [chapter 4](#) empirically investigates how example-aided ideation shapes visualization design by running a controlled experiment in which participants created data visualizations with the aid of examples. This experiment controlled when participants were exposed to examples (before or after initial brainstorming), allowing them to browse and curate relevant examples as they sketched designs, thus varying the data-schema similarity, diversity, and quantity of curated examples. Our findings reveal a positive influence of delayed example introduction on visualization design quantity and variety. We also find that a higher dataset similarity between curated examples and the dataset being visualized results in increased idea transfer from examples to generated designs. Furthermore, these results reveal that, in addition to data compatibility, the timing of an example’s introduction and its visual complexity significantly affect the quantity and variety of the designs a designer creates as well as the degree of idea transfer from examples to designs. These findings contribute to ongoing debates on the usefulness of delayed example introduction and introduce dimensions of examples that are relevant and important for data visualization design, challenging the prevailing emphasis on data matching alone for selecting relevant designs.

Contributing Papers:

H. K. Bako, X. Liu, G. Ko, H. Song, L. Battle, and Z. Liu. Unveiling how examples shape visualization design outcomes. *IEEE Transactions on Visualization and Computer Graphics*, 31(1):1137–1147, 2025

Chapter 5: Ideation Strategies in Example-Aided Visualization Design. Analogous to the

factors that modulate an example’s influence on design outcomes are the strategies that designers actually engage in to browse through a gallery of examples. Chapter 5 builds on the empirical insights of [chapter 4](#) by shifting focus from which example properties influence design outcomes to how designers actually engage with examples during ideation. Through event-sequence analysis of timestamped interaction logs from a controlled study with 32 designers, it shows that early exposure to examples encourages broad filtering and exploratory browsing, whereas later introduction drives more targeted, data-focused search. Moreover, strategies combining consistent anchoring, flexible filtering, and detailed inspection yield a higher quantity of designs and greater design variety. By revealing these dynamic ideation patterns, [chapter 5](#) deepens our understanding of the mechanisms behind the effects observed in [chapter 4](#) and also suggests actionable guidelines for design retrieval and authoring tools that adapt to designers’ evolving strategies during creative ideation.

Contributing Papers:

H. K. Bako, K. T. Zinat, S. N. Sakhamuri, and Z. Liu. Patterns of inspiration: Analyzing ideation strategies in example-aided visualization design. *Under Review*, 2025

Chapter 6: An Example-Driven Prototyping Tool for Interactive D3 Visualizations. A key challenge designers face is the lack of support for strategies for adapting ideas from examples to designs in current visualization authoring tools. One of these strategies is to “*replicate & modify*” existing examples, which often involves replicating visualization example code with the data a designer wants to visualize and modifying the resulting chart to suit their design needs. [chapter 6](#) presents Mirny, a web-based environment that builds on the *replicate & modify* strategy identified in [chapter 3](#) to support the iterative generation of interactive D3 visualizations. Mirny

draws from a corpus of 2,500 curated D3 code examples to offer designers dynamic templates that they can iteratively adapt via a GUI interface. Mirny bridges the gap between GUI-based visualization generators and the fine-grained control afforded by programming toolkits like D3.js, demonstrating its utility for both novices, who gain an accessible entry point into code, and experts, who benefit from example-driven code scaffolding. Furthermore, Mirny shows how tooling can operationalize example adaptation strategies identified in this dissertation to support more expressive, efficient visualization authoring.

Contributing Papers:

- H. K. Bako, A. Varma, A. Faboro, M. Haider, F. Nerrise, B. Kenah, and L. Battle. Streamlining visualization authoring in d3 through user-driven templates. In *2022 IEEE Visualization and Visual Analytics (VIS)*, pages 16–20, 2022
- H. K. Bako, A. Varma, A. Faboro, M. Haider, F. Nerrise, B. Kenah, J. P. Dickerson, and L. Battle. User-driven support for visualization prototyping in d3. In *Proceedings of the 28th International Conference on Intelligent User Interfaces*, IUI '23, page 958–972.

Chapter 7: Conclusion and Future Work. Finally, Chapter 7 presents exciting directions for research on example-aided design, and the development of the next generation of co-design tools to enhance productivity and creativity in visualization authoring.

Chapter 2: Background and Related Work

Examples are used to fulfill diverse goals in learning and creation. On one hand, examples serve as a tool for gaining inspiration for new ideas; on the other, examples serve as learning guides to understand how one can accomplish a specified goal. For instance, in educational settings, step-by-step worked examples guide student learning by illustrating problem-solving processes and reducing cognitive load [174, 175, 177]. In creative fields, exposure to example designs fosters creativity by illustrating what is possible [92, 94]. Analogous to this, in software engineering, programming-by-example uses exemplar demonstrations by users to support program synthesis [84] while code examples are used as reference templates to replicate solutions and accelerate implementation [95, 150, 153]. As such, examples serve multiple purposes in the design process: a single example can simultaneously inspire new ideas and offer concrete guidance. This dissertation explores the dual role of examples as both sources of inspiration during design ideation and as structured guidance for visualization Authoring.

2.1 Example Usage in Design

2.1.1 Idea Generation Using Examples

One school of thought defines Creativity as the process through which a person generates ideas. The creative process involves combining past or recently gained knowledge in a novel

way to generate appropriate solutions to a problem [213]. In the past, research has attempted to provide a framework for the creative process [8, 155, 191, 208]. Recently, Warr and O’Niel [213] provided a unified definition of creativity. They argue that creativity itself encompasses the combination of two or more thought processes in a way that is considered new to the mind and in a manner that solves a defined problem. They propose a unified generic model of past research that describes the creative process as three iterative and intertwined phases. These phases include (1) *Problem Preparation* to develop an understanding of domain knowledge, (2) *Idea Generation* where an individual generates new ideas using both past and newly gained knowledge, and (3) *Idea Evaluation* where the ideas generated are judged for their appropriateness as solutions to the defined problem.

In general, idea generation for designers encompasses *Idea-Finding*, or a search for knowledge—new or old—that sparks inspiration for solutions to design problems [155, 213]. While idea generation is an integral part of the creative process, in practice, designers struggle the most with coming up with new design ideas [155]. Smith et al. find that designers often engage in deliberate information retrieval methods—either through search engines or past experience—to understand the problem domain in order to generate ideas [195]. Designers have employed various techniques to aid the idea-generation process. Smith et al. synthesized a list of 55 “active ingredients” that describe the techniques used for idea generation [195]. These techniques were broken down into (1) Strategies that describe means for generating ideas, (2) Tactics that refer to devices for ideation, and (3) Enablers which are the conditions or passive means of promoting idea generation. The most common strategies identified in this work involved deliberate information retrieval methods through search engines or past experience to understand the problem domain in order to generate ideas. For example, designers could form associations between problems and

ideas in memory or past experience to generate ideas. Herring et al. similarly define three activities within the idea-generation process, which involve research, represent, and refine [93]. They also identify 19 idea-generation techniques employed by professional designers. They find that designers seek inspiration through images, text, physical objects, or analogies, etc. [93]. Designers were also found to have a preference for visual artifacts over textual descriptions as sources of inspiration [79].

While extensive research into example use for ideation has been explored in fields like engineering and graphic design, to the best of our knowledge, how *data visualization designers* currently seek out inspiring examples has not been studied. We contribute to this body of knowledge by examining if visualization designers employ similar strategies to generate design ideas.

2.1.2 Example Search and Storage Strategies

Examples provide a visual framework for designers to understand problem domains, build, generate, and validate ideas, and communicate to other designers [92, 121, 141]. For example, knitwear designers use past designs and newly found examples to generate ideas and share trending styles with others [68]. Designers often seek out examples using a variety of tools and systems, such as search engines [92, 156, 157], books and published material [67, 86, 229], and online sharing platforms [92, 128, 141].

Designers tend to use two strategies for finding examples: an *active search* where there is a deliberate search with a “target” example in mind [93]; and a *passive search* where the goal is to browse material to randomly find inspiration [10, 79, 92]. Miller et al. observe that the type of design activity and a designer’s job role influence the strategies and criteria designers

use to find, store, and use examples. For example, web designers use examples to search for known solutions to a problem and to differentiate their designs, whereas industrial designers use examples to understand the domain [141].

While examples clearly influence how ideas are generated, finding relevant examples can be a challenge. The web is a popular resource for examples [92, 141], but designers often struggle to compose the right search queries [93, 202, 215] and search results may not contain any inspiring examples. As a result, designers are often forced to modify their keywords to diversify search results [92, 94, 180]. Past work has also found the opposite: that designers may seek out similar examples, restricting the diversity of their exploration [141]. Once relevant examples are found, designers tend to curate them into personal collections [107] as mood boards, design documents, electronic bookmarks, or image libraries [92, 220].

A few tools aim to help designers find relevant visualization examples. Hoque et al. created a search engine to help users find relevant D3 designs by matching search queries to the visual style and structure of D3 visualizations [97]. Saleh et al. enable similar search capabilities by computing style similarities between low-level visual elements within infographics [181]. IdeaRelate provides users with relevant examples based on their conceptual distance from a user’s design idea [221].

Prior work identifies key strategies that designers use to search and store design examples. In chapter 3, we clarify how these strategies apply specifically to visualization design by investigating how visualization designers structure their search parameters to elicit design examples, as well as how these examples are selected and used—independently or together—to generate novel design ideas.

2.1.3 Example Usage and Influence on Creativity

Extensive research in fields such as psychology, design, and mechanical engineering have highlighted the challenges of finding and extracting ideas from relevant examples [10, 79, 92, 141] and explored how examples affect the ideas designers create [68, 79, 103, 141, 165, 194, 196]. Collectively, these works show that examples significantly affect creativity: some report examples as a source of inspiration [15, 68, 79, 141, 158], while others report constraining effects of examples [56, 103, 103, 116, 133, 164, 194]. Together, these works have identified key factors that modulate the effect that an example has on the ideas designers come up with. In [chapter 4](#) we evaluate how prior findings on these factors apply to the data visualization context.

2.1.3.1 Factors that Modulate an Example’s Influence

Timing of Example Introduction. Much research has explored *when* examples should be introduced in a design process. Smith et al. show that introducing a delay between exposure to examples and design ideation does not reduce conformity (the number of features copied from examples) within designs [196]. These findings support long-term retention of examples [189] and suggest that design conformity may be a result of the unintentional use of memory from prior exposure to examples [133, 135]. However, there is no consensus on when examples should be introduced. Kulkarni et al. suggest early exposure to examples [111], while others advocate for on-demand provision of examples [48, 193]. Yet, another thread of research shows that when designers have had time to brainstorm potential solutions before exposure to examples, they can better integrate examples into their solutions instead of conforming their solutions to examples [144, 194, 202]. Tradeoffs have also been observed on the delayed introduction of examples,

with some works observing Einstellung effect where designers are unwilling to let go of past ideas even in the presence of new examples, ultimately inducing fixation [29, 30, 122]. Our work extends these emerging findings on the timing of example introduction by investigating how the time of exposure to examples may lead to meaningful differences in design outcomes.

Commonness of Examples. The “commonness” of an example, refers to how typical an example is as a solution for a given problem [47, 194]. For instance, in data visualization, bar charts are a *common* solution for comparison tasks where one seeks to observe similarities or differences in values across different data groups. Research finds that focusing on common examples, closely related examples, or examples related to the problem domain limits the space of ideas explored, which may lead to fixation [103]. Expanding the space of explored examples to include solutions slightly distant from the problem domain [54, 74] or even analogical examples [31, 46, 47, 90] has proven more beneficial for idea generation [164], leading to higher quality and novelty of design solutions [194].

Diversity of Examples. Research on example-aided design has demonstrated that showing designers stimuli of diverse types and formats may influence design fixation [58, 194]. Yet these studies rarely define or quantify “diversity” in a systematic way. Nevertheless, the results of prior work show that diverse examples lead to increased novelty and diversity of ideas [22, 152, 192]. However, when it comes to the influence of diverse stimuli on the quality of design ideas generated, the results diverge [105]. Some experiments report that diverse examples improve design quality [98, 199, 226], while others find a negative impact of diversity on the quality of generated design ideas [25, 146].

Quantity of Examples. Compared to other properties, there has been less investigation into how

the number of examples examined by designers influences designs. Marsh et al. investigated how exposure to different numbers of examples ($n = [1, 3, 6, 9]$) may increase conformity [134]. Their results show that more examples lead to more critical features copied from examples to designs. Sio and Kotovsky also find a negative correlation between the number of examples and the novelty of design ideas [194]. However, there is no consensus on the appropriate number of examples to provide as experiment stimuli. Recent studies tend to arbitrarily choose the number of examples from an exploration of the solution space [47, 54, 74], while others allow participants to independently curate examples from the web [141].

Quality of Examples. Literature shows that examples induce conformity for designers [103, 134, 196]. However, the quality of examples determines the degree to which conformity enhances or impairs design outcomes. Janson and Smith found that individuals do not only copy ideas from examples that solve the provided problem but also from examples with obvious design flaws [103]. Chrysikou and Weisberg found that individuals copy inappropriate ideas from examples even when instructions explicitly ask them to avoid these inappropriate ideas [56]. Fu et al. also show that introducing poor examples leads to a decrease in the ability to converge on solutions and the quality of solutions [73]. In these works, “poor” examples are designs with problematic features that do not adequately solve the defined design problem. Assessing the quality and effectiveness of visualization designs is highly subjective [44, 161], dependent on many dimensions beyond just encoding choices, such as the context of use and the intended audience.

2.1.3.2 Metrics for Evaluating Design Outcomes

In previous work, design outcomes are often measured by quantitative metrics such as the quantity and conformity of designs and subjective evaluation of the quality, variety, and novelty of designs. The conformity of designs, as described by Smith et al., is the tendency for designers to copy features of examples into their design ideas [196]. In most studies, conformity is measured by assessing the number of example-related ideas duplicated in final designs [194] and is believed to be an indication of the constraining influence of examples [134, 194, 196].

Subjective measures such as the quality of design outcomes measure the degree to which a design solves the specified problem [43, 47]. In general, the evaluation of design quality is based entirely on metrics decided upon by researchers, such as ease of use [43], cost feasibility, and build time [47, 74]. Similarly, the novelty of designs is based on how unique a generated design is relative to design ideas generated by other participants [6, 176, 194, 202]. Variety, on the other hand, measures the explored space of solution ideas, i.e., the range of ideas produced by a single participant [6, 43, 202].

In data visualization, metrics for evaluating visualization design are often subjective or rely on perceptual guidelines to judge effectiveness [126]. In this dissertation, we are interested in objective metrics that can be derived from explicit properties of visualization components such as visual marks, layouts, and encodings. Accordingly, we adopt established, quantifiable measures from the design literature, such as quantity, variety, and design conformity, to assess both example visualizations and designer-generated designs.

2.1.3.3 Design Fixation

Examples can significantly elevate creativity, and designers have been found to use examples to prime themselves towards thinking of certain ideas [6]. However, examples have also been found to inhibit creativity when designers prematurely fixate on certain ideas, leading to unintentional transfer of ideas that are either wrong or unhelpful to the current problem-solving task [56, 116, 165, 194]. This phenomenon is known as “design fixation” [103, 164]. Parsons et al. find that data visualization professionals employ various techniques to avoid fixation, such as incubating themselves from the problem, seeking external input, and seeking inspiration in existing visualizations [158].

2.2 Visualization Authoring

There are a plethora of tools available to create data visualizations. These tools take three forms: programming-focused tools, direct manipulation tools, and visualization recommenders. Below, we discuss each of these approaches to visualization creation.

2.2.1 Programming Tools

Programmatically creating visualization designs has been a challenging process for most designers. Historically, creating visualization designs required understanding and using low-level graphic libraries such as the Java 2D graphics library to create visualizations [69, 89, 139]. In 2011, Bostock et al. introduced D3, a JavaScript-based language for creating visualization designs in the browser [36]. D3 gained popularity among visualization designers as it supported manipulating the Document Object Model on a web page, creating affordances for more expres-

sive visualization designs and interactions. However, D3 remains notoriously difficult to learn, creating a barrier to its adoption by less technically savvy designers. To overcome the limitations of low-level programming languages, Vega-lite, a high-level grammar for specifying visualizations, was introduced [184, 217]. Built on D3, Vega-lite allows users to provide a specification of the visualization as a JSON object. Programming-based approaches to visualization design remain the most expressive, allowing designers to have more fine-grained control over their designs. Yet, they all still suffer from steep learning curves, which remains a major hindrance for fledgling designers.

2.2.2 Direct Manipulation Tools

The second set of visualization authoring tools focuses on no-code approaches [89, 119, 182]. Some of these tools provide users with a set of chart templates that they can use to create visualizations [137, 162, 206]. Other tools such as Lyra [182], iVisDesigner [171], and Data Illustrator [119] eliminate the need for programming by providing users with a graphical user interface where designs are created in response to designers' interactions with the system. These tools offer flexibility by supporting direct manipulation; however, considerable effort is still required from designers to create a single design, making it challenging to evaluate and iterate on multiple designs.

2.2.3 Visualization Recommendation Tools

Advances in Machine Learning have given rise to a wave of visualization generation tools, which focus on different techniques to recommend appropriate visualization designs to users [64,

99, 123, 127, 143, 217, 218, 222, 227] or generating visualizations in response to natural language (NL) instructions [149, 219]. These tools offer users the ability to quickly generate visualization designs for exploratory visual analysis without worrying about any implementation details. However, the designs generated by these tools are limited to a small set of visualization types, and the designs themselves cannot be modified. By abstracting the implementation details, these tools limit design expressivity and creativity. More recently, research has explored the space of generative design tools that allow users to generate and evaluate design alternatives for glyphs [40] and dashboards [125]. Some have explored the data-agnostic design of visualization prototypes, eliminating the data constraints that designers often face when coming up with visualization designs [201, 207].

2.2.4 Example usage in Visualization Authoring

In visualization authoring, researchers have introduced tools that retrieve or adapt styles from existing charts [87, 166, 231] or infer visualization specifications from visual example sketches [210, 211]. However, many such approaches address only specific chart types or narrow use cases (e.g., infographics), raising questions about their generalization and effectiveness in practice [190]. These gaps underscore the need for broader, systematic studies and comprehensive tool support to understand and leverage example usage across diverse visualization design workflows.

Chapter 3: The Role of Examples in Visualization Design

3.1 Introduction

From knitting to web design to manufacturing, designers across domains rely on examples of past visualization designs as sources of inspiration [92, 96, 141]. It is no surprise that visualization designers often look for relevant examples [24, 158] on galleries like Bl.ocks.org [34] and Observable [35] and sharing communities like Tableau public [4]. Little research has studied how these designers use examples in their work: what do they look for when exploring examples? What strategies do they use? How do they incorporate elements from examples into their own designs? How do examples inspire or hinder creativity? *Understanding the role of examples* in visualization designers' current workflows is crucial to laying the foundation of the example-aided design process, addressing the first goal of this dissertation.

While others have studied the use of examples by designers in the context of engineering [32, 33] and graphic design [68, 79, 92, 141], the findings are not readily transferable to visualization. Engineering design often focuses on the development and performance of physical products, while graphic design is purely concerned with the visual representation of information. Visualization design, on the other hand, focuses on visually representing data so that it is easy to find and interpret patterns [203]. As a result, the design decisions made by visualization designers are both informed and constrained by the underlying data [209]. Furthermore, the metrics to

evaluate a visualization design involve both objective measures (e.g., perceptual accuracy) and subjective criteria (e.g., visual appeal), requiring more balanced considerations in choosing and using examples.

In this chapter, we attempt to address this gap in the visualization literature by investigating *how visualization designers use examples in their current practices*. We conducted 30 semi-structured interviews with 15 students (novice) and 15 professional (expert) visualization designers. We chose to include novice designers in this study to ensure our study reflects the full range of real-world design experiences. By contrasting the approaches of novice designers with those of seasoned practitioners, we can pinpoint practices and challenges unique to early-career designers and also identify findings that generalize across varying skill levels. To understand how designers find and use relevant examples, we asked our participants to describe how they sought out examples, how they determined if an example was useful or not, and how ideas were translated from examples to their final designs. Through qualitative analysis of our interview data, we contribute core themes, patterns, and challenges centered on example-based visualization design activities.

Our analysis focuses on two major activities: *searching* for relevant examples and *utilizing* examples. We find that designers search for examples to not only find design ideas, but to gain contextual information about their data and target audience. Further, designers may adopt different strategies to search for examples depending on their experience, such as performing a targeted search to match a known design prior (i.e., a design idea they have in mind before search), versus an exploratory search to opportunistically identify new and interesting examples. We discuss how these variations in strategy lead to differences in tool selection and usage as well as in the challenges designers encounter while searching for examples.

We find that designers engage in three main strategies when using examples: 1) *select and merge* visual elements such as color schemes and glyph designs from multiple examples, 2) *replicate and modify* an example to suit design goals, and 3) perform *trial and error* on many examples to identify a desirable design. We discuss the tools used to implement the strategies, the role of example reuse in design, and the common issues designers face when using examples. We observe that designers select relevant examples based on their effectiveness and aesthetics, and often examine the trade-off between these two criteria. Finally, we examine existing practices and perspectives around example curation and design fixation, and discuss the challenges our participants face and their visions for ideal example search and utilization environments.

Through our findings, we suggest research opportunities to help visualization designers find new examples to spark their creativity, select useful examples relevant to their target use case, and reuse examples effectively.

3.2 Methods

Our goal in this work is to understand the practices of visualization designers and how they find and utilize design examples. To this end, we conducted semi-structured interviews with both novice and professional visualization designers. In this section, we describe the selection criteria for participants and the study design.

3.2.1 Participants

To fully understand the role of examples in visualization design workflows, we needed to account for all levels of visualization design expertise. We recruited two groups of partici-

pants: students representing novice visualization designers, and professionals with industry experience in designing visualizations. We recruited participants from university and visualization conference mailing lists, online visualization communities (e.g., [r/dataisbeautiful](#)) and Twitter. A demographic survey was used to screen participants. To evaluate participants' experience, we collected data on their occupation, past visualization projects, how often and how long have they been creating visualizations.

We recruited 9 male and 6 female student designers in our study. They self-report as students, and have created visualizations in an academic context (e.g., class projects) but not in a professional setting (e.g., [with] clients). 80% (n=12) of our student designers had between 1 to 4 years experience creating data visualizations, the remaining 3 students had between 3 months to 1 year experience. 53% (n=8) of these students report creating visualizations less than once a month, 33% (n=5) less than once a week and 12% (n=2) create visualizations weekly or daily. Their background ranged from domains like security or robotics to more diverse domains such as journalism and architecture.

We recruited 8 males and 7 females as professional designers for our study. All but one of our professional designers report their job function as designing visualizations for others (e.g., UX designer, data analyst, BI developer). Only one of our participants identified as senior management, working as a manager of a team of data analysts. 86% of our professional participants reported between 3 years to over 5 years of visualization design experience. 80% of professional participants report using data visualizations either daily (n=7) or weekly (n=5). A total of 30 participants (17 males and 13 females) between the ages of 18 to 54 were interviewed. In-depth details of participants' demographics can be found on OSF ¹.

¹ Click to visit our OSF repository

3.2.2 Interviews

We conducted semi-structured interviews, which were held remotely and lasted between 43 to 83 minutes. All interview sessions were recorded and transcribed using Zoom. First, participants were briefed on the study protocol and given time to complete the consent form. Then, we asked open ended questions that encouraged participants to describe past experiences. For example, participants were asked “*Can you describe a situation where you found an example that sparked a new design idea for a visualization?*”. Participants were encouraged to share pictures of visualizations as they described their experiences. The interviews followed a loose script as follows:

1. **How do designers find examples?** To understand how designers found relevant examples, we explored what types of tasks lead to the search for visualization examples, and the resources (including recommendation engines) and strategies used to find them.
2. **How do designers identify relevant examples?** We asked questions to understand what makes an example useful, what features indicate a good example and what practices (if any) do designers engage in to curate and manage collections of examples.
3. **How do designers use examples?** To understand how ideas are transferred from examples to final designs, we investigated how designers use examples in their workflows, what tools designers used to implement their visualizations, and if participants were aware of design fixation and how do they overcome fixation.
4. **What challenges do designers face?** We investigated the challenges designers faced while seeking and using examples. We also sought to explore what designers thought an ideal search and retrieval process would be. To do so, we asked designers to reflect on their

current processes of finding and using examples and describe what an ideal process would look like for them.

Our interview script is shared in our OSF Repository (see footnote 1).

3.2.3 Analysis

In line with existing qualitative analysis approaches [7], each response to an interview question was coded. We used an iterative open coding process where two coders independently coded every interview and then swapped codes with each other. Both coders worked together to correct mistakes in the transcripts generated by Zoom.

Then, each coder reviewed their partner’s codes and updated any codes they agreed with but missed during their own coding. The coders met weekly to resolve disagreements in the codes. One coder consolidated the codes and organized them into clusters based on context. These clusters were subsequently analyzed, refined and consolidated (if needed) into a second layer of categories describing the high-level activities occurring within each cluster. The authors met to discuss these clusters and identify any emergent themes.

In the following sections we discuss these high-level themes across two main activities: the *Search* for examples (section 3.3) and *Utilization* of examples (section 3.4) . In section 3.5, we discuss the cross cutting themes identified in our analysis: 1) the *criteria* to select examples, 2) example *curation* practices, 3) *design fixation*, and 4) participants’ reflections. We use descriptive statistics in parentheses and quotes to support our claims. For the rest of the chapter, we denote professional and student participants using the characters “P” and “S” respectively. Participants discussed multiple sentiments and also engaged in a variety of example usage activities; we report distributions on how many participants *explicitly* discuss each point we observed. The full list of

extracted thematic codes and their corresponding distributions among our participant groups is available on OSF.

3.3 The search for examples

In this section, we discuss our observations on when and why visualization designers search for examples, the approaches taken to find relevant examples, and the challenges faced while searching for examples.

3.3.1 Why and When are Examples Needed?

We find that most designers look for examples when they want to *communicate* using visualizations. Examples are used to not only generate design ideas, but also to gain contextual information about data and the intended audience of the visualizations.

3.3.1.1 Examples are Sought When Designing to Communicate

Whether the audience is clients, colleagues or the general public, visualizations provide an easy medium for designers to explain key ideas and concepts as opposed to displaying raw data [12, 185, 203]. Participants repeatedly discussed using visualizations to *tell a story* about the data and break down interesting insights into a form that can be easily understood by their audience (9 P, 13 S).

So one of my biggest reasons, I would say, is when I'm trying to explain something to someone, I use visualizations because it's easier to do it that way -S13 I do create more of a custom visualization to tell the very precise story that I want to tell. -P9.

While visualizations are a convenient medium for communication, participants struggle with finding an effective way to express ideas to an audience. Examples can provide *guidance* on what visualizations to create to achieve specific communication goals. Using examples, participants can acquire ideas and use them to inform their designs:

These visualizations inform how decisions are made. I think when you need to convey your ideas to someone you start to look for examples and based on that example reference we iterate and it becomes our own visualization idea... -P14

3.3.1.2 Examples are Used to Generate Design Ideas

We observed that participants may search for examples at the start of the design process when brainstorming ideas or trying to find new representations of data. Examples may also be used in the later stages to refine their visualization designs. These sentiments were similarly echoed by engineering and graphic designers in prior work [92].

Finding common visualization designs. Designers may seek out popular examples as inspiration for design ideas. This often occurs during a brainstorming session with friends, teammates, or colleagues (3 P, 9 S). During these sessions, participants may iterate over examples and draw out ideas they would like to include in their designs (5 P, 2 S). These sessions may focus on understanding common visualizations for certain data types or topics (3 P, 7 S). For example:

First I'll try to see what might be the best way of showing a data. And I may type my topic into Google just to see data visualizations examples for these topics. -S9

Finding new visualization ideas. A few participants also distinguished between finding examples while brainstorming ideas and finding examples to avoid what they perceived as “boring”

visualizations. In these cases the goal is to find *unconventional* ways of representing data (4 P, 5 S). A participant recounts searching for such examples while working on new projects:

yeah I do that [search for examples] every now and then, when things are boring, to me, sometimes I just get bored and I want to try something new -P8

Making visualization designs fancier. Some participants described searching for ways to improve design aesthetics after they already had some design priors in mind (4 P, 2 S). In this case the goal is to improve on current designs by making them “fancier”. For example:

I usually look for how to make it look fancier... I think of how can I do this kind of a bar chart and make it more beautiful. At that time I may find examples -P4

3.3.1.3 Examples are Used to Understand What Users Want

Before beginning the ideation process, examples may be used to gain a high-level understanding of the needs of the target audience. Some professional participants described using examples to understand what their users want and how they will eventually use these visualizations (3 P). The contextual information derived from these activities influence the nature of visualizations that participants may search for and eventually implement. P8 describes working with clients to understand design requirements for visualizations. Once high-level goals are understood, P8 searches for examples, presents them to the clients for feedback, and uses that feedback to guide their designs. Similarly, P7 describes using examples provided by clients to guide their search:

If they send me examples—when they have some good examples—I can go find more examples. I do send them something back to see [final designs] to make sure that it’s clear... give me some feedback -P7

One participant emphasized how important it is to understand who the target users of the visualizations are and how they intend to use the visualizations. This contextual information informs the design decisions for the dashboard they eventually create:

I think the most important is to figure out who that user to be and then how they're going to use this dashboard. I think this leads to the main direction for me to design. -P11

3.3.2 What Tools and Resources are Used to Find Examples?

Designers rely on various tools and resources to find visualization examples, such as search engines, portfolios, books, and news articles.

3.3.2.1 Common Sources of Examples

Search engines and image sharing sites. These two types of resources were mentioned the most during the interviews. Participants frequently utilized keywords to search for examples on Google and websites like Pinterest [160], Tableau public [4] and Behance [5] (11 P, 11 S). Participants may favor Google because of the *variety* of designs that can be found. On the other hand, designs from websites like Pinterest and Behance were perceived by some as having higher *quality*:

...because Google, it has everything so whether it's good or bad it has everything, and Pinterest feels like someone already filtered those for you and if they are being collected by someone probably people think they're good, so it's one filter up -S9

Online communities. Participants described browsing through examples on community sharing sites like Subreddits (e.g., r/dataisbeautiful [168]) where they can view and comment on visualizations. These types of resources were often discussed in the context of exploratory search and

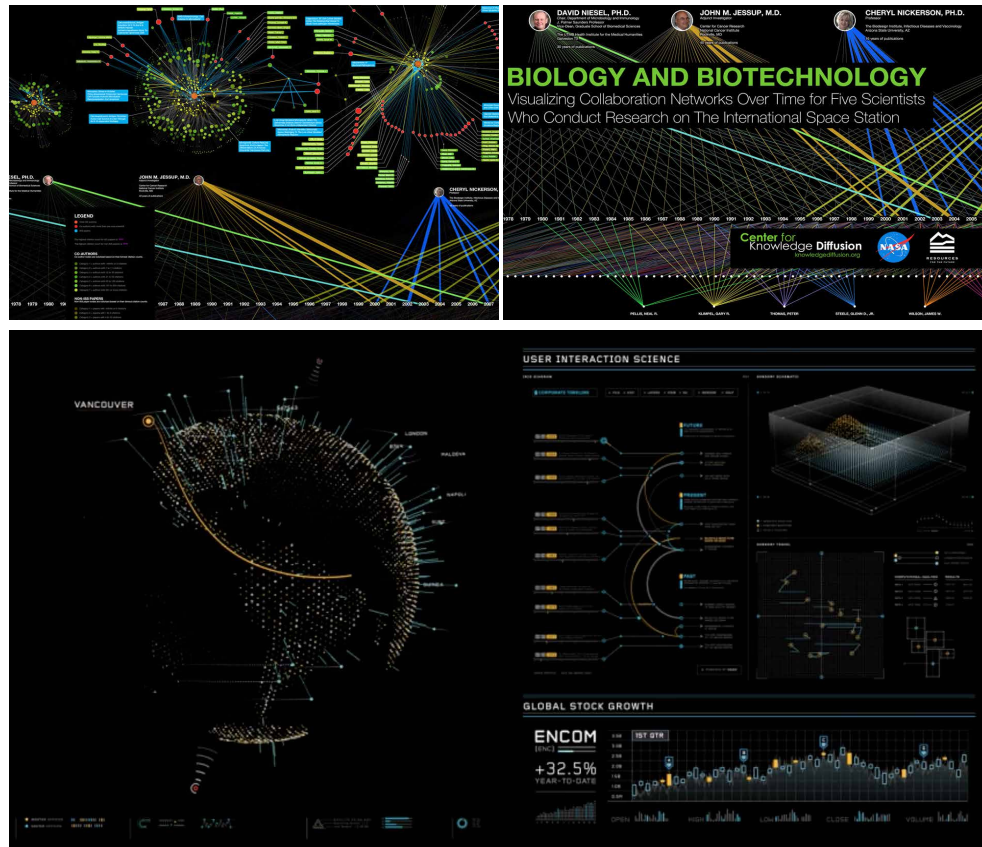


Figure 3.1: The top two images are visualizations created by P7, showing the authors and co-authors of published research funded by NASA and projects they have worked on. The image on the bottom is a snapshot of a futuristic design in a movie that inspired P7.

were frequently engaged in by professional visualization designers (7 P, 2 S). Participants generally found them to be great sources of information that require some filtering to sort through and find relevant examples.

Tableau is a tool that there's a vibrant online community for ... So there's just a lot of information out there you just have to sort through it. Some of its better than others, but that that's helpful. -P6

Published media. Some participants rely on published media, such as news articles, blog posts, and movies, as resources to find examples. Participants described having specific news websites (e.g., New York Times, Washington Post) they frequent to find design examples (4 P, 5 S). Exam-

ples may also be sourced from books, research papers and occasionally training materials (6 P, 7 S). Books may be used when participants are looking for high quality examples. For instance:

...but when we want more solid and accurate information, we kind of went into the library, I think we did go for hard-copy books...and look for visualizations which really connected to what we wanted to do. -P3

Some professional participants also described using portfolios of experts and colleagues published online or on social media sites to find examples (7 P). In general, media resources provide participants with access to new visualization types and can help them gain inspiration, even from abstract concepts. For example, one participant describes getting inspired by a visualization example from a movie (see [Figure 3.1](#)):

When I did this one, I was looking a lot at the futuristic user interfaces that you see in like science fiction movies. I think there was like a tron globe on this one and I liked how it connected with space. -P7

Software documentation. Participants reported using documentation for visualization tools and libraries as a resource for examples (2 P, 5 S). Most visualization tool developers provide galleries of example designs. The main appeal for these resources is the availability of code or demo videos alongside visualizations, which serve as implementation tutorials and resources that can be modified or repurposed.

3.3.2.2 Visualization Recommenders

Recommendation engines such as Tableau’s ShowMe [[127](#)] and Excel’s Recommended Charts [[140](#)] are frequently used to find appropriate visual representations [[127](#),[217](#),[218](#)], and are employed by both professionals in their analytical work and in academia to teach students [[214](#)].

We asked participants about their experiences (if any) using visualization recommenders to come up with designs.

A total of 11 participants report never using recommendation engines (5 P, 6 S). The remaining 19 reported using recommendation engines frequently (7 P, 8 S) or at least once in the past (3 P, 1 S). On the one hand, some participants found recommendation engines to be helpful resources for providing new options and templates, which they could use to start their own designs (5 P, 7 S). These tools also eliminate the burden of implementing visualizations, making them easy to use and appealing to students (1 P, 6 S):

It's easy to try out new visualizations that I, maybe wouldn't have tried before or like they're kind of too hard to do in a quick amount of time, so I think that the Tableau recommendations definitely do help me to look at new visualizations that are not total garbage -S8

On the other hand, a few participants reportedly avoid using recommenders because they do not see value in the recommendations (3 P, 1 S). They tended to be professionals with considerable experience. For instance, P10 who has 3 - 4 years professional experience said:

Recently it's it hasn't been as necessary as it used to be ... I used to use it a good bit back when I was still sort of grasping what you would need to maybe make an area chart... I've just become a little more skilled and knowing what I want once I understand the data. So I think I need less of that direction -P10

Participants also noted specific challenges that may hinder the adoption of recommenders. Participants found that they may not understand why certain visualizations are recommended to them (1 P, 2 S). The variety of charts and styling options included in recommendation engines are limited and could restrict participants (1 P, 2 S). As a result, participants find themselves

frequently making modifications to tweak recommended charts, which may discourage them from using these tools (2 P, 1 S). Finally, some participants find the available interactions (e.g., tooltips) in recommended charts to be too limited (2 S).

3.3.3 How do Designers Search for Examples?

Participants discussed different approaches they took to find examples. We characterize the different ways participants construct search queries to find relevant examples and discuss recurring themes here.

3.3.3.1 Targeted vs Exploratory Search

Visualization designers may discover inspiring examples spontaneously or through deliberate retrieval efforts. Participants reported two main approaches to finding examples: *targeted* and *exploratory* search. Visualization designers may engage in a *targeted* search for examples where the goal is clearly defined (3 P, 4 S). In these situations, there is typically a design prior in mind and a search is initiated to find an example that matches this design prior (5 P, 7 S). For instance:

...but then, if I know what my data looks like, for example, and I know how I need to visualize it, I will usually look for examples [implementation] in the particular tool I'm using to create the visualization. -S8

On the other hand, *exploratory* search occurs when participants skim through examples with no goal in mind, hoping to randomly find an interesting visualization example (8 P, 2 S). In these cases, there is no established design prior and the search process is more open-ended (4 P, 3 S). We find that participants who engaged in this type of search were mostly professionals.

Participants described engaging in exploratory search to keep abreast with what other people are creating:

...to be honest I'll go out there and see what other people are doing and see how I can apply that to what we've got going on...If I'm bored I just type in some random stuff and I find something that looks interesting to me... -P8

3.3.3.2 Search is an Iterative Process

The search for relevant design examples is an iterative process. Participants report iterating over keywords to retrieve different sets of examples (4 P, 6 S). The search starts with general purpose keywords to find a diverse set of examples (1 P, 6 S) (e.g., “*map related data visualization*” -S9). As interesting examples are found and design concepts become concrete, the search shifts; keywords become more explicit to target specific examples. For instance, participants may refer to the type of chart they want to create (e.g., “*area chart*” -P5). Explicit keywords help to narrow down the scope of search results (2 P, 2 S). Participants may also use visualization libraries or tools to fine-tune the search scope. Visualization tool names are used as keywords to narrow down the results or find implementations; excluding them as keywords increases the variety of retrieved examples (11 P, 12 S).

Usually I'll try to find a couple of examples, three or five that I can compare and then I'll do individual searches on each one usually to get more specific. I know that I definitely don't want to look at these now or like these are definitely very interesting...I'll just kind of start adding words to my search to get more and more specific -S5

3.3.3.3 Search Query Construction

Whether engaging in targeted or exploratory search, we find that our participants emphasize different kinds of parameters, such as dataset characteristics, visualization design, and the visualization task that the examples are used for. We describe the key parameters here.

Search based on dataset characteristics. When asked to give examples of how they phrase their keywords during search, some participants often discussed their keywords in terms of the characteristics of the datasets they were working on (11 P, 6 S). There seems to be a preference among professionals to center their search strategies on dataset characteristics such as data type or domain:

I work in sales marketing so I'll type in like "sales KPIs" or "targeting metrics" or "market share dashboard". So anything that I'm working on, and those are all like standard terms that you would use across any business. -P8

Search based on visualization design and elements. Participants may tailor their keywords to finding a specific type of visualization (e.g., "pie chart" -P1) or visual element (e.g., "Matplotlib error bars" -S12) (3 P, 11 S). Participants may search through examples to find a target design or to identify a common strategy for creating a visualization, which they identify by examining and isolating common design concepts among the examples (5 P, 4 S). These types of keywords may also be used to find corresponding implementations (1 P, 6 S). For instance:

let's say I want a visualization about a map, for example. I would actually search for "map" and google the D3 version [as keywords] and then it kind of pops out right away...and I get their sources [code]. -S7

Search based on a task or goal. While less common, both professional and student participants

may also use the intended visualization task (2 P) or their design goals to search for visualizations (2 P, 2 S). These strategies are exploratory in nature and would often involve general purpose keywords such as “*connectivity city mapping*”-P3.

3.3.4 What Challenges Arise When Searching for Examples?

The challenges often discussed by participants were associated with the inability to find truly relevant examples. First, participants find it hard to identify the right keywords to use to search for examples (2 P, 5 S). As mentioned in section 3.3.3.2, participants often iterate over search keywords. This iteration occurs in part because participants may be unfamiliar with established terminology for their target visualizations. Hence, participants initially resort to trial and error to identify relevant keywords. For instance:

With my first search trial, it may not work. But then I make some changes two or three times and then I get mostly what I want. So it's always difficult to look for the most optimal one, but you know it's pretty easy to get what you want after like two or three shots. -S8

The search results may also be incompatible with participants' datasets (3 P, 2 S) or cluttered with examples that are not useful (2 P, 4 S). This requires additional effort from participants to identify and eliminate these irrelevant examples. One participant said:

When trying to find examples of projection plot, it looks similar to scatter plot, so all the examples I was seeing it was like it just implementations of scatter plot they have like similar name and they look similar but they're actually different things. -S14

Even when the right keywords were identified and relevant visualizations were found, some participants reported having *insufficient experience or knowledge* to understand what a makes a

particular example good (2 P, 1 S) or how to use a good example when they find one (1 P, 1 S).

P2 who uses blog posts to find examples said:

You need to have the base knowledge to know what you're looking for and you need to have the base knowledge to understand the post you're reading, and really dig in to make sure that this example is completely irrelevant or not.

A few participants describe not using examples often (3 P, 2 S). This may be as a result of their experience or lack of time.

From time to time and try and get ideas but honestly I feel like my style is different ... maybe I'm too stuck in my ways and it really is a bandwidth issue too a lot of times I'll do it for like 15 minutes at a time, but I've got work to do. -P11

3.4 Using Examples

Finding relevant examples is just one step in the visualization design process. The next step for visualization designers is to translate the idea(s) they derive from examples into their own design(s). In this section, we discuss strategies participants used to apply ideas inspired by examples, criteria for reuse, and challenges faced when using examples.

3.4.1 How do Designers Use Examples?

We observed that our participants were largely responsible for the design and implementation of their visualizations and described 3 strategies to using examples in visualization designs: (1) *select and merge* features from one or more examples, (2) *replicate* an example then *modify* it, and (3) *trial and error* with multiple examples.

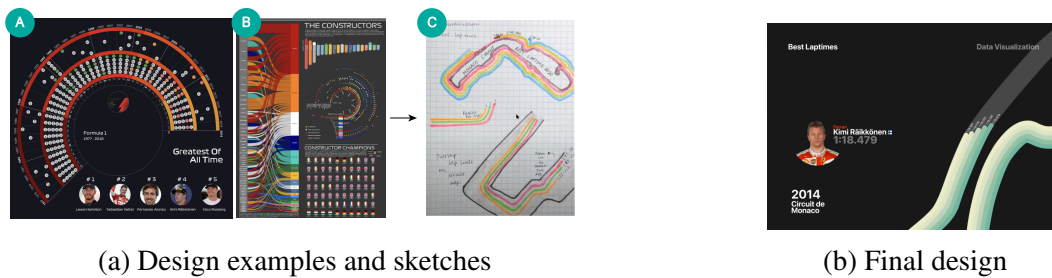


Figure 3.2: An example showing how ideas are translated from multiple examples into a single design. Visualization elements like circles with images of race car drivers are selected from example A and merged with paths from the Sankey chart in example B. The participant sketches their design idea in C and includes a note about radial bar charts as another source of inspiration. In the final design, we can identify relevant design elements from the two examples and initial sketch.

3.4.1.1 Select and Merge

Many participants report using more than one visualization examples to generate design ideas (11 P, 11 S). Using multiple examples allows participants to pick and merge different visual elements that inspire them (6 P, 8 S). For example, S2 describes using examples from a blog post, portfolio and radial bar chart to visualize the lap times of race car drivers, shown in [Figure 3.2](#). We can see the influence of the radial bar chart (A) and sankey diagram (B) in [Figure 3.2a](#) on the representation of the lap times in [Figure 3.2b](#), as well as the influence of example (A) on the annotated image of the driver with the fastest lap time in [Figure 3.2b](#).

One participant described using the select and merge strategy to break down complex visualization ideas into individual components or visual elements. This allowed them to isolate elements in examples, and seek out implementations for these elements which they merge together in their design. A few participants describe combining multiple examples together to form multiple-view visualizations (3 P, 1 S).

3.4.1.2 Replicate and Modify

Examples are sometimes used as a baseline for visualization design. Some participants discuss making copies of interesting examples using their data (*replicate*) and altering the design (*modify*) to fit their design goals. The major reasons for modifying examples involved changing core features of the example (5 P, 7 S) such as adapting a visualization to match their data or introducing interactions to the visualization. P5 describes modifying examples to fit the scope of their dataset:

...so the first modification is that ... I swapped the data content. The data content is different, so your data scope might be different. They [example] might be using very small dataset, for example, if you are using like bubble chart, your x axis or y axis can be super different; they can be very small, but yours might be pretty large -P5

While replication is convenient, some examples do not meet the aesthetics standard a designer may want. Participants discuss making modifications to adjust visual styles and layouts in replicated examples (5 P, 3 S). These changes typically targeted components such as the font, orientation, and color schemes. For professionals, modifications may be aimed at matching designs to the stipulated company branding.

...And I think that goes back to there's not a lot of business examples out there, so you kind of have to "steal" from other contacts and a lot of it isn't that translatable that you can just grab it and go. I think most of the time it would require applying our branding guidelines to something. So that's color palette, fonts that kind of stuff. -P6

3.4.1.3 Trial and Error

While most visualization designers distill ideas from one or more examples, a few would resort to a brute force approach towards using examples. Six participants report trying to implement multiple visualization examples and ideas until they find an acceptable design. Participants resorted to this strategy when they were in situations where they did not know what to create or were stuck in making design decisions (2 P, 1 S). Participants may also use a trial and error approach when working with tools that provide quick access to different visualization designs (e.g., visualization recommenders) (3 S):

What I used to do in tableau is just hit every every button... if I really don't know what to do, I can change the map layers, I can include this check the boxes if I want streets on here or landscapes, like all those different things make it pretty easy to customize -S6

For some designers, trial and error is not limited to recent searches for examples. They may also iterate over design ideas from visualizations remembered from previous design sessions. P9 describes:

To me it's almost like testing a hypothesis, when I come to a dataset I've got some preconceived notion of what I want to tell people... At this point in my career, I will have observed so many different visualizations styles that I have a decent idea of what I want to try first. And then I will try to objectively judge how well I've succeeded there and then I'll start tweaking it and occasionally scrap the whole thing and start over, so it's kind of like trial and error -P9

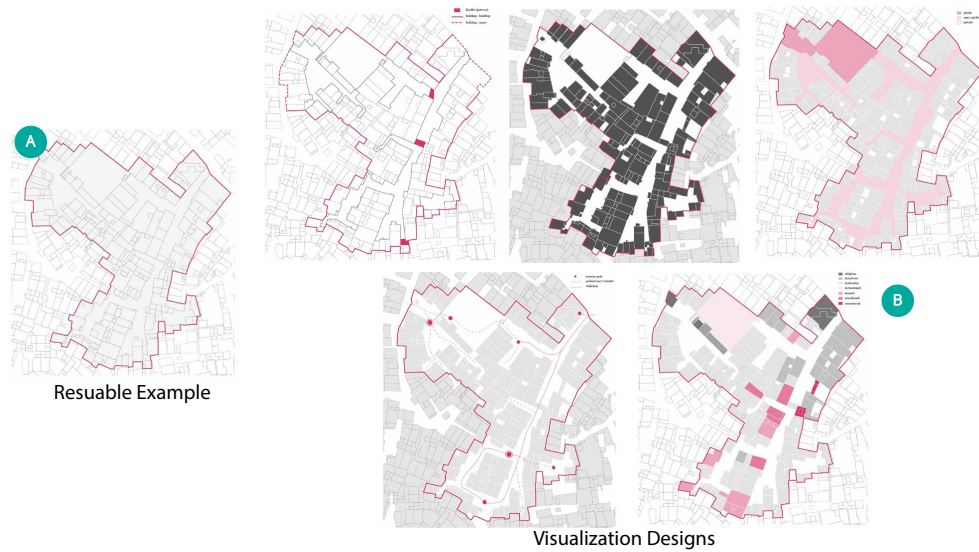


Figure 3.3: An example of reusing a single example to generate multiple data representations. Here, P3 reuses a base map layout (A) to visualize data on building structures in a specific geographic location (B).

3.4.2 Reusing the Same Example

Participants report reusing the same example for multiple visualization designs (10 P, 11 S) when they are engaged in repetitive tasks such as representing different dimensions from the same dataset, or when engaged in creating visualizations that were “standard” for their field. A few participants reuse examples by applying the same design concept to represent different data points. For instance, P3 who is an architect describes using a single map layout example to visualize data on multiple aspects of building structure as seen in [Figure 3.3](#).

Some participants may be inclined to reuse an example because it has been successful at communicating information to their audience in the past (2 P, 3 S).

These are things people have already seen in my own field so it's sort of a common language type of thing. So it's best to just go with the same thing. -S8

Participants may also reuse an example if they have easy access to the code (2 P, 2 S). For example, S10 recalled reusing the concept of a 3D scatter plot from a word cluster to visualize

human skeletons in 3D by visualizing each joint with a point as in a scatter plot. They were even able to reuse the same code. We also find that participants often engage in reuse when the underlying visualization task is similar. For some participants, this may be a result of the type of data they work with. For instance, P5 who is a risk management associate, tends to reuse the same examples and designs because their job is mostly concerned with identifying trends in data.

3.4.3 Tools Designers Use

Most participants discussed using visual analysis tools like Tableau to create visualizations. Other tools mentioned include direct manipulation and sketching tools (e.g., Adobe Illustrator, and visualization libraries such as D3.js in JavaScript or Matplotlib in Python). A few participants mentioned using specialized tools to create custom visualizations. For example, S5 uses ArcGIS to develop story maps. A detailed breakdown of the distribution of tools used by the participants can be found on OSF ².

Participants generally felt supported by visualization tools (10 P, 8 S) as they help expand their visualization design skills, provide guidance on how to create visualizations, and allow users to customize their visualization designs. Participants reiterate that each tool plays a specific role in the design process. For instance, Tableau reduces effort for conventional designs and allows quick exploration of visual designs. On the other hand, tools like D3 are more useful when there is a specific design in mind. Participants may also sketch out design ideas on paper or using tools like MS Paint before implementing their designs (6 P, 8 S).

²Click to visit our OSF repository.

3.4.4 Challenges Faced While Using Examples

So far we have discussed how participants re-purpose ideas from examples and the tools used to implement their designs. However, the process is not always easy. We summarize the main challenges that designers face when using visualization examples.

Several participants (mostly professionals) complained about how much time they had to spend modifying examples to implement their design ideas (7 P, 3 S). These complaints focused on the significant time needed to tweak replicated examples. A participant comments:

I think the big challenge is that it's not fun right, it's fun to have a good quality output and it's fun to have the moment where you understand that your visualization is going to work for what you intended to work for. But it's the formatting, the layering, the time spent on all the little things that maybe you're the only person who's ever going to notice. You know that can be a bit of a slog. -P9

Another challenge is related to operationalizing desired changes using specific tools. For example, participants may have difficulties translating desired changes to specific components that need to be modified in the target example (4 P, 3 S). Some participants complained about not knowing how to use tool interfaces, or they had troubles understanding the documentation and tutorials (5 P, 1 S). Tools often require too much manipulation to implement changes (2 P, 4 S), and the designers may have to use multiple tools that may not be compatible (1 P, 4 S). Additionally, programming based tools often required significant effort (3 P, 6 S) and version changes may make examples created with previous versions no longer functional (2 S). One participant said:

Now, regarding the fine tuning that's the trickiest part because that's what requires knowing

where exactly to change to make it suitable...For that, I just go to the documentation to see if I can find what I'm looking for. Sometimes it doesn't work, because it is not easy to read. So I just end up spending some of my time on stack overflow or general forums on how to get certain things done. -S13

A few participants acknowledged the role that their experiences played in responding to challenges faced while implementing design ideas (5 P, 3 S). Participants thought that understanding what and how to make modifications became easier as their experience and understanding of the design process and tools increased. A few participants describe the process as learning on the job. One participant (P5) in particular, reflected on the nature of strategies they employed to create visualizations. For this participant, as a novice, it was easier to just replicate examples; as they became more mature in the field, they found themselves graduating from copying ideas to devising their own ways to improve designs as they came across examples. Other participants had similar trajectories, for example P9 comments:

I will say you know earlier in my career learning to use the tool effectively, learning how to make all those adjustments Without digging through endless menus or options or stuff like that, that is a technical challenge I don't really have anymore. -P9

3.5 Cross Cutting Themes and Activities

Our analysis revealed three themes that cut across the two main design activities (search and utilize): (1) criteria for determining if an example is useful, (2) practices and perspectives for curating examples, and (3) factors inducing design fixations and overcoming fixation.

3.5.1 Selecting Examples

Here, we discuss factors influencing how professionals search for and select visualizations and key criteria for selecting useful examples.

3.5.1.1 The Role of the Target Audience in Search and Selection

Most of our professional participants described their job roles as primarily involving creating visualizations for “clients”. These clients were typically colleagues, management, or business stakeholders who were often described as being less visualization savvy. We observed that professional designers often discussed how much the backgrounds, needs, and interests of their target audience influenced the types of visualizations they searched for and selected.

For example, participants have to consider how clients would perceive the visualizations (2 P) and the context in which the visualization will be used (3 P) when making design decisions. Some professional designers restricted themselves to using to common visualizations such as bar charts in order to accommodate the needs of their clients, even if they would prefer to try out more sophisticated designs (4 P):

...So while I have created a chord diagram for fun, in the professional world I would never share that to anyone to transmit a message. I would share it only to make them impressed with how well I can visualize ... if I were to put it [chord diagram] in front of someone, I'm going to have to spend a long time explaining to that person what the chord diagram is and they're going to have to spend a long time studying it ... and there is almost always a simpler way to represent what's going on. -P9

Participants would also stick to using charts that their clients are already familiar with to avoid

presenting charts they may not understand (5 P).

Accommodating participants' requests for specific designs can be particularly taxing for designers as they have to prioritize these requests even if they may not necessarily think they are good designs. Participant P9 recalls clients requesting designs that were not always feasible or appropriate and, in some cases, having to "...talk people out of bad designs". Ultimately, these considerations may leave participants feeling limited on the amount of control they have over design choices. While some participants consider this limitation as part of the job, others find ways to introduce interesting designs into their visualizations.

...although most of our clients like things simple, like a bar chart for example, or tables...you have to find a way of being creative with the things they give you. So instead of showing just a bar chart of this year's sales, you can show last year sales overlaid behind it, instead of just a pie chart use a donut chart with KPIs in the middle. -P11

3.5.1.2 Criteria for Selecting Inspiring Examples

During our interviews we sought to understand what signals participants used to determine how useful a design example was. We identified six criteria that participants use to select useful design examples. While we discuss these criteria individually, participants often considered multiple criteria simultaneously when evaluating the usefulness of an example. The weights placed on each criteria were subjective.

The effectiveness of a visualization. Most participants (11 P, 7 S) agreed that the most useful examples are effective at communicating the right information to their audience i.e., the example is easy to understand. Participants often stressed the importance of making sure that visualizations were easy to interpret and presented the message as simply as possible. These participants were

often less concerned with how well the design adhered to visualization rules like the appropriateness of data encodings. On the contrary, data encodings were generally an *implicit* consideration for useful examples and only mattered if the rules of thumb for data encodings were violated.

When I started doing data visualization, there was a lot of emphasis on Edward Tufte versus Nigel Holmes rules [where Tufte placed emphasis on visual processing while Holmes was more concerned with making visualizations memorable]...So I'm no longer interested in "this is the Platonic ideal of visualization" and more interested in what are people going to see, or respond positively to... -P6

The aesthetics of a visualization. Several participants mentioned visual elements such as visual appeal and color schemes as criteria for determining if an example was useful (14 P, 6 S). Surprisingly, participants often held aesthetics in the same regard as the effectiveness of an example, mentioning the two criteria together. The aesthetics criterion may include layout and fonts used in a visualization, how appealing a visualization looks, or the use of annotations or colors to provide contextual information. For instance,

100% the color scheme is the first thing that comes to my mind, if they have like the nicest colors on there, like different shades of purple is something that I really gravitate towards for some reason, as soon as I like that [example], it's like a fish bait, then [the second point] it's can I quickly understand it? -P11

Three participants discuss having to find some balance between the aesthetics and effectiveness of visualizations (1 P, 2 S). They admit that the most aesthetically pleasing examples may not be the most effective at communication, indicating some trade-offs between the two criteria.

I also try to balance it out like it's not all visual aesthetic and the data doesn't correspond to the scale and stuff. So I'm trying to like balance at all -S2

The similarity between design priors and examples. Some participants evaluate how similar an example is to the design idea they have in mind as a criterion for selection. In these cases, participants select an example only if it matches their idea or improves on their original design (3 P, 4 S). Occasionally, visualizations that provide improvements are described as unique since they are not visualizations that participants have encountered before (4 P, 4 S). One participant said:

For me it would be how unique the design is. I try to compare it with my previous work. At times I compare it [example] with what I really wanted at that moment. Most time I focus on how it improves from what I want -P1

The ease of integration with data and tools. We find that some participants were concerned with how compatible examples were with the dataset they intend to visualize and how much effort would be required to implement the example. These participants were more likely to select examples that had similar data structures (1 P, 6 S) and were easy to implement (1 P, 5 S). One participant comments:

So communicating the data is more important than being able to create it easily, but there is a threshold. If it's too difficult, then I'll start thinking is there some other way I can spend my time... -S1

The use of interactions and animation in examples A few participants discussed using the presence of interactions and animations as criteria for selecting an example (3 P, 4 S). Participant P10 notes that interactions can be combined with complex charts to help users understand examples and tell a moving story. Another participant said:

I will say, the one thing I love seeing is like somebody using like actions, in a creative way

... I just like people who get creative with it [dashboard] in a way that's still really easy to use...because they're [clients] always going to ask another question and then another question, so you want to make sure you have the ability to get to all four or five level questions within your one dashboard either by drill downs or by filtering or by whatever...a simple overview is never enough. -P8

How examples support inclusiveness Three participants discussed criteria for example selection that focus on evaluating how the design choices in the example support people who may be colorblind or have ADHD (Attention Deficit Hyperactivity Disorder). Participant P8 reports that since they have ADHD, they tend to disregard examples with designs that have too much going as these examples “*wreck my brain*” and make it hard to concentrate on the key ideas within the examples. Two participants (P5 and S9) consider how the colors used in visualization may be perceived by users who are colorblind.

So I feel like sometimes some of the examples are not very like color blind friendly...So sometimes I may look for something that's like you know not completely based on the color like sometimes if they have like patterns, but you know, sometimes, if we add more patterns, it will make the graphic looks very busy for normal people -S9

3.5.2 Curating Examples

We explored designers' example curation practices to assess how curated examples may be used in the design ideation process. Similar to past research, we also find that some participants regularly save examples for future use while others do not curate examples [92]. We discuss the emergent patterns for example curation below.

3.5.2.1 Curated Examples are Rarely Revisited

Several participants regularly save useful visualization examples found during search. These collections take many forms such as digital file folders (4 P, 3 S), web browser bookmarks (4 P, 3 S), links in text files (3 P, 2 S) or liked examples on social platforms (e.g., Pinterest) (1 P, 2 S). Due to the volume of curated examples, these participants labelled collections either by visualization type (e.g., bar charts) or the topic of the project they were working on. This labelling strategy helped participants organize their collections to simplify example retrieval.

We observe that only 3 participants reported routinely reviewing these collections to either organize or retrieve examples after they save them. Most participants who curated examples said that they never returned to these collections after creating them (3 P, 1 S), or returned to them only if they remember something exists (3 P).

If I see something I will usually like bookmark it, whether it's in my browser or my reading list or if I'm on Reddit. The problem is I don't ever go back to my bookmarks. -S8

Participants struggled with managing their collections. These challenges can be attributed to collections being messy, requiring designers to manually search through them. For example P15 recalls clicking through a list of saved web addresses to find an example they had found before. One participant noted that due to the volatility of web pages on the internet, some bookmarked pages may not be accessible when they return to them. Finally, managing saved examples is a time consuming task that may dissuade designers from saving them in the first place. These findings echo observations from existing work in design, which demonstrate that designers express a desire to curate and manage a collection of design artifacts, but this process is often hindered by the difficulty of example curation and reuse [62, 101, 179, 188].

3.5.2.2 Interesting Examples Will Be Remembered

Roughly half of our participants said they do not curate examples (6 P, 10 S). Some participants found curation unnecessary because they could easily remember where they came across compelling examples (4 P, 3 S), or they prefer to implement these examples immediately. We find that participants who expressed this sentiment had a median of 3 to 4 years of visualization experience. Some participants often rely on examples they have seen in the past to generate new visualization ideas (3 P, 2 S). One participant said:

No, I'm either making a visualization for a purpose or not. I mean certainly I might store away in my head the fact that some kind of visualization exists but I won't save it -S12

3.5.3 Design Fixation and Its Effects on Example Use

We sought to understand if designers were aware of design fixation and how they avoid getting fixated on design ideas while searching for and using examples. Inline with the protocol used in past work on design fixation [57, 158], we first asked participants if they were familiar with design fixation and then explained the concept using Jansson and Smiths' experiments [103] as an example of design fixation and its effects. We present our observations on factors that can induce design fixation and strategies that participants use to overcome this fixation.

3.5.3.1 Factors that Induce Fixation

Participants often noticed fixation when they relied too much on past work, made assumptions about what visualizations were industry standards, or were overly focused on examples found during search.

Participants recall fixating on designs they had previously created which were well received by clients. Participants often assumed that these designs were safe and would often continue to use them for different projects over an extended period of time (3 P, 4 S). Participants may also fixate on ideas they perceive as standard visualizations that are accepted by their industry or community (3 P, 7 S).

...only because I see certain visualizations being used in a certain way and people responding to those visualizations, so I get attached to those designs alone and it does have a big impact on me, maybe not consciously -S13

This perception may also be induced by the types of examples that participants find while searching for examples. Some participants may commit to designs that repeatedly appear in search results, as they assume that such examples must be the standard or the best way to represent the information they want (2 P, 4 S). For a few participants, fixation could be caused by requirements from superiors which may make it difficult to balance new design ideas with job expectations (2 P) and cause them to commit to inappropriate ideas (2 S).

3.5.3.2 Examples and Feedback Can Help Avoid Fixation

Participants often tried to avoid fixation by changing their search terms to use simpler words, or adding new words such as “fancy” to change the types of examples they are seeing (4 P, 4 S). Others may consider completely changing the tool they use to search for examples. For instance, S2 mentioned switching from Behance to Google when they would find themselves seeing the same types of design examples.

Participants may connect with colleagues, teammates and clients to talk through ideas they

are fixated on, in order to get a different perspective that may guide them to alternative designs (3 P, 7 S). Feedback may be obtained from evaluations of implemented designs or through conversations with people (5 P, 10 S). For example:

...I just think out loud to my co-authors “hey how about this” and just kind of describe my ideas. And I will look at their expression or excitement, and that gives me good feedback. Even if they don’t tell me which one is a good direction to go in -S13

3.5.4 Reflections on Desired Tools

We asked participants to reflect on their current process of finding and applying examples and to describe ideas for ideal tools and resources they would like to use instead. We summarize their ideas below.

Increased search modalities. Participants frequently described a desire for tools that support more search modalities (1 P, 5 S). These ranged from booklets of examples and corresponding implementations, to tools that could allow search in design centered domains, to find examples outside of data visualization. Three participants (S1, P9 and P3) also mentioned tools that support search for examples based on uploaded datasets. P9 described tools that can transform the underlying dataset into a format that is compatible with target examples. Designers also want to specify search criteria through examples (1 P, 4 S), design specifications (1 P, 2 S) or natural language descriptions (3 S).

Automation to aid, not replace design decisions. Regarding tools that recommend charts to meet the specified design criteria, four participants (2 P, 2 S) expressed concerns about balancing how much control they would have over the design decisions with the level of automation in these

tools. For these participants, automation should only be used to augment their skills by providing designs they would not have found themselves and the ability to rapidly evaluate multiple designs.

The final say on what design is used should be left to the designer.

I need to think about the audience and the context of the data as well, so that part I think I'll do it [make decisions] myself -S13

3.6 Discussion

Our findings highlight interesting patterns of example search and usage that are unique to visualization design as well as patterns that echo observations in other design domains [92, 141]. Unlike designers who specialize in preparing and handing off visual concepts to developers [209], our participants are responsible for designing *and* implementing their visualizations. As a result, data is an important consideration for these designers throughout the design process. Visualization designers have to take into account the constraints imposed by data, the needs of their target audience, and the effectiveness of designs as they search for and use examples. This challenge is also magnified by external factors such as job expectations that can influence their design choices.

We find that experience plays a huge factor in how designers search for and use examples. Professional designers heavily rely on their experience to determine what visualizations they need to create and are more interested in seeing variety in visualization design. As a result, professionals mostly engage in exploratory search and are more likely to focus on examples that are both effective at communicating insights and aesthetically pleasing. On the other hand, students are limited by their knowledge of visualizations and programming languages. Consequently, students are more concerned with the ease of integrating examples into their current workflows and

often opt to replicate and modify examples or iterate over different designs before settling on an optimal design. Our participants acknowledge that these challenges diminish as experience increases. We reflect on the practices and challenges faced by designers and discuss research opportunities, for example, retrieval and visualization authoring.

Enhance visualization retrieval: more criteria, modalities and diversity. Our results reveal three main criteria used by designers to search for examples: 1) dataset characteristics, 2) visualization features and styles, and 3) visualization tasks. However, existing visualization retrieval systems only support search using specifications of visualization features and styles [97, 181]. Consequently, designers often fall back on general-purpose search engines like Google, which lack the specialized filtering and retrieval capabilities needed for visualization examples; echoing shortcomings observed in other design domains [200]. To support the needs of designers, visualization retrieval systems and example gallery designers need to support more searching criteria based on data semantics and tasks. Additionally, these tools need to go beyond using specifications as search parameters to accommodate more modalities, such as natural language descriptions and visual examples. Such tools will enable visualization designers expand their example search space.

Our results also suggest that a tool that supports the serendipitous discovery of new and interesting visualizations may be preferred by experienced users. However, current retrieval tools are optimized to find results that match search queries, potentially missing less relevant yet useful examples. This presents an opportunity for the visualization community to explore how to support serendipity in exploratory search for examples. Specifically, research has to identify how to balance variety in search results with the quality and relevance of visualization examples [76,

141, 197].

Understand and support example-based design ideation. Examples are useful for conceptualizing design ideas for visualizations. However, there is a risk that designers could focus on examples that lead to ineffective designs. Design fixation does not necessarily happen because of inherently bad designs, it may result from multiple factors such as a mismatch between examples and tasks, or inappropriate perceptions of the examples. For instance, student designers might fixate on a specific design if they thought it was favored by their community or someone with greater authority (e.g., a professor). These observations entail the need to understand, measure, and model the effects of examples on ideation outcomes. The results will be useful for mitigating fixation and steering ideation practices towards better designs.

Balance multiple design objectives and factors. Research has explored automated generation (e.g. diatoms [40]) and recommendations of baseline designs to build on. However, analysts and designers desire effective *and* captivating visualizations [39, 225]. Thus, these tools must balance the effectiveness and aesthetics of visualizations, and provide transparency into why the designs are generated or recommended. These tools can also be extended to support direct editing and parallel prototyping, so that users can effortlessly explore and compare multiple design alternatives [65]. Furthermore, research should explore how and to what extent the socio-cultural factors identified in our work (e.g., expectations of clients and superiors) influence design choices, and deliver better tools to facilitate design feedback and iteration.

Author visualizations with examples as inputs. Existing visualization authoring tools do not adequately support any of the example reuse strategies that we have observed (select and merge, replicate and modify, trial and error). In particular, it is not possible to transfer features between

multiple visualization designs, nor are there easy ways to import an example and modify it directly. Some visualization authoring tools offer support for replicating and modifying example templates for visualization designs [59, 87, 137]. However, these systems are limited in the type and format of examples they can support. How to easily turn any example into a usable template is a promising research direction. In the software engineering domain, existing literature has explored how to support the repurposing or remixing of existing examples for new design contexts [37, 38, 63, 91, 142]. Future work on visualization authoring tools needs to explore how to support similar affordances for the seamless transfer of design elements from an existing example to new visualizations. One possible approach is to extract visual properties such as color, axis design, and glyphs from examples and apply these properties to specified attributes. To support such features, systems first need to understand how to automatically isolate and infer the role, organization, and constraints of different visual elements from an exemplar image.

3.6.1 Limitations

Our work is intended to provide the much needed context on how visualization designers use examples as sources of inspiration. Like many other interview studies, our participants may not reflect the overall population of visualization designers. Specifically, our participants were recruited from institutions of higher learning and visualization tool communities. Consequently, the experience of our participants is slightly more advanced and analytical in nature, which may not reflect the experience of casual and non-technical visualization designers.

3.7 Summary

In this chapter, we sought to understand how visualization designers find and use examples to generate and implement visualizations. Through an interview study with 30 visualization designers, we observed two main approaches to the search for examples and three strategies that designers use to translate ideas from examples to their visualization designs. We also uncovered cross-cutting themes related to the selection and curation of examples, as well as the challenges designers face in finding and using relevant examples.

Our findings highlight a lack of support in current visualization tools for common example search and implementation strategies used by designers. Understanding these real-world workflows, the different considerations, and the mental models designers have provides a deeper understanding of designers' example usage practices and lays the groundwork for a systematic exploration into the process of example-aided design.

Chapter 4: Factors that Influence Example-Aided Visualization Design

4.1 Introduction

In the previous chapter, we established the example usage practices of data visualization designers. However, the dynamics of example-aided design and its measurable impact on design outcomes remain unclear. Previous studies have identified several factors that modulate the influence of examples on design outcomes, including the *quantity*, *quality*, and *diversity* of examples [47, 159, 194], and the *timing* of example introduction during design ideation [134, 144]. Yet, these prior studies have conflicting results (see Chapter 2.1.3 for discussion) and visualization design presents unique challenges and complexities that are not shared by other disciplines: visualization designs are constrained by the data to be presented, and designers struggle to anticipate how real data may impact visual forms, constrain design choices, and lead to unexpected edge cases [209]. Thus, there is a need for empirical data that sheds light on the influence of examples in the context of data visualization. Specifically, we need to assess *how factors identified in prior studies, alongside those unique to visualization, modulate the influence of examples on design outcomes*. Additionally, we need to understand *how designers decide which aspects of examples to incorporate into their designs*.

Based on these considerations, this chapter contributes an exploratory study with 32 visualization designers to understand the influence of examples (e.g., images of existing charts) in data

visualization design. The study loosely follows the protocol of Smith et al.’s seminal work evaluating the influence of examples [196], where designers are given a design task with accompanying examples and asked to brainstorm design ideas. However, we make some strategic adaptations to our study design. First we use a faceted browsing interface for participants to explore and curate relevant examples, allowing us to examine multiple factors (e.g., quantity, commonness, etc.) in the same study. Secondly, we include a think-aloud protocol to capture deep contextual information about designers’ thought processes during design ideation (details on our study design are discussed in [section 4.3](#)). Our analysis focuses on quantifying the features of an example and using this information to generate metrics that describe the collective properties of example sets curated by participants ([section 4.4](#)). We use quantitative methods to measure how properties of curated example sets influence objectively measurable design outcomes like the number and variety of designs and ideas copied from examples ([section 4.5](#)). Finally, we perform qualitative coding to gain insights into designers’ rationale for selecting examples ([section 4.6](#)).

Our results show that 1) the timing of example introduction potentially influences the types of examples designers curate and the number and variety of designs produced. 2) a higher similarity between the data schema in examples and the data schema a designer wants to present, leads to more ideas transferred from examples into designs. These results are supported by our qualitative evaluations highlighting *two forms of idea transfer, i.e., partial transfer and design replication*. Finally, we identify salient factors that attract designers to use examples or deter them from doing so, such as the underlying visual tasks (e.g., comparison, distribution), visual composition, design complexity, and the similarity between examples and prior design ideas. Our findings shed light on results from prior work and introduce new factors that influence design outcomes, giving rise to guidance on how designers may incorporate examples into the design process and discussions

on advancing the use of examples for visualization design.

4.2 Factors and Outcomes of Interest in This Study

Examples directly influence the creativity of designers [134] and can be intentionally or unintentionally transferred into designs when people perform creative tasks [133, 135, 196, 212]. Sio and Kotovsky identify several factors that modulate the influence examples have on creativity [194], including (1) the *commonness* of an example which examines how familiar the example is to the problem domain [47, 159, 194]; (2) The *Timing* of the introduction of examples [111, 144, 193] (3) *Number* of examples used in the design process [194]; (4) *Quality* of the examples [152]; and (5) the *Diversity* of examples [25, 105]. Please revisit chapter 2 (section 2.1.3) for a detailed overview of the literature on each of these factors and related metrics for design outcomes.

Previous research identified key factors that influence design outcomes; our study focuses on three of them: the quantity and diversity of examples, as well as the timing of example introduction. As data informs the design choices a visualization designer makes, we introduced two additional factors relevant to data visualization: the similarities between examples' underlying datasets and the dataset used in the design task in terms of data *topic* and *schema* (i.e., number and types of attributes). These new factors explore whether or not designers curate examples based on closeness in topics or the alignment between the working dataset and an example's data schema, capturing the multidimensionality of similarity in the data context [77, 132]. For instance, if a designer is working on the Boston weather dataset [138], which contains temporal and numerical attributes, is the designer more interested in examples visualizing similar topics

(e.g., average temperatures or melting ice caps), or do they consider a broader set of examples visualizing temporal and numerical data? Additionally, does focusing on topic or schema similarity influence their designs? This expands the scope of example-related factors we analyze in this study to include the *timing* of example introduction, *diversity*, and *quantity* of examples, and the data topic (*topic_sim.*) and schema (*schema_sim.*) similarities.

Regarding design outcomes, we focus on the more obvious quantitative metrics: the *quantity* and *variety* of designs, as well as the conformity of examples, which we refer to as *idea_transfer*. We do not assess subjective factors such as the quality and novelty of examples, nor do we evaluate the effectiveness or creativity of design outcomes. Assessing visualization designs across these axes remains open research for future work.

4.3 Study Design

Our work aims to understand how preexisting and visualization-specific factors influence design outcomes and how designers decide what aspects of examples should be incorporated into their designs. In this section, we discuss our experiment design rationale and procedure.

4.3.1 Experiment Overview

From our survey of prior work, we observed that the study design in most of these works follows the protocol of the foundational study by Smith et al. [196], which examines examples' constraining effects on creativity. In these studies, participants are presented with a design task and various examples (or no examples) and asked to develop designs that fulfill the task.

We considered replicating Smith et al.'s experiment [196] with a data visualization design

context. However, a straightforward replication is not feasible as the design space for data visualization is much broader than previously studied fields. For instance, in Smith et al.’s study, examples are characterized by simple binary features, (e.g., if a tail is present in the example). In contrast, the features of a visualization example can be analyzed along multiple dimensions, including but not limited to mark choices, visual encodings, layout, and graphical styles. Each dimension has numerous values; for instance, a mark choice includes but is not limited to rectangle, circle, polyline, and area. It is unclear which dimensions and values are crucial in selecting and characterizing the features of examples to present to designers. Additionally, there is no clear guidance from prior work on how many examples should be provided in the experiment.

Consequently, we made strategic changes to our study design based on the following considerations. We have five example-related factors to be examined. Controlling for all these factors in a single experiment would be impractical as this would lead to an exponential combination of experiment conditions. Giving participants the freedom to curate examples can naturally give rise to variations in both the example features (e.g., marks, layouts, encodings, etc.) and factors *quantity*, *diversity*, *data schema.sim.*, and *topic.sim.* This approach is more similar to designers’ real-world example curation behavior than providing them with a small set of fixed examples.

Moreover, letting designers curate examples from the web with complete freedom can be impractical. In a pilot study with four participants, we discovered that participants were often distracted by information irrelevant to the design task when exploring examples directly from source pages. Thus, we constructed a corpus of 50 diverse examples for the participants to use in the experiment (section 4.3.3).

We decided on a between-subjects experiment with *two timing conditions*. We included

two design scenarios to capture how our results may generalize across multiple datasets and design goals. This results in **four experimental conditions** (section 4.3.2). We provided an experiment interface for the participants to browse and bookmark relevant examples from the corpus. Participants were asked to explore a given dataset, brainstorm and sketch design ideas for their assigned scenario. These sketches represent the design outcomes of each experiment and were evaluated to identify the *quantity* and *variety* of the designs and instances of *idea-transfer* from examples.

After brainstorming, participants selected two sketches that they perceived satisfied the design goal, which the experimenter implemented to evaluate the feasibility of the designs. In the pilot study, we found that the time needed to assess designs, perform data transformations, and implement sketched visualizations exceeded the allocated time for the experiment. As a result, the experimenter implemented the designs independently after the experiment. The participants then review the implemented version of their designs in a 15-minute follow-up session. Section 4.3.5 describes the experiment procedure in detail.

4.3.2 Experiment Conditions

4.3.2.1 Timing Conditions.

Motivated by a lack of consensus in prior work on when examples should be introduced into the design process, we evaluate two *timing* conditions in our experiment. Particularly, we investigate if a delayed exposure to examples leads to a meaningful difference in the types of examples curated by participants and the number of ideas transferred from examples into designs. Hence, we include a baseline condition (T_{srt}), where participants were provided with the example

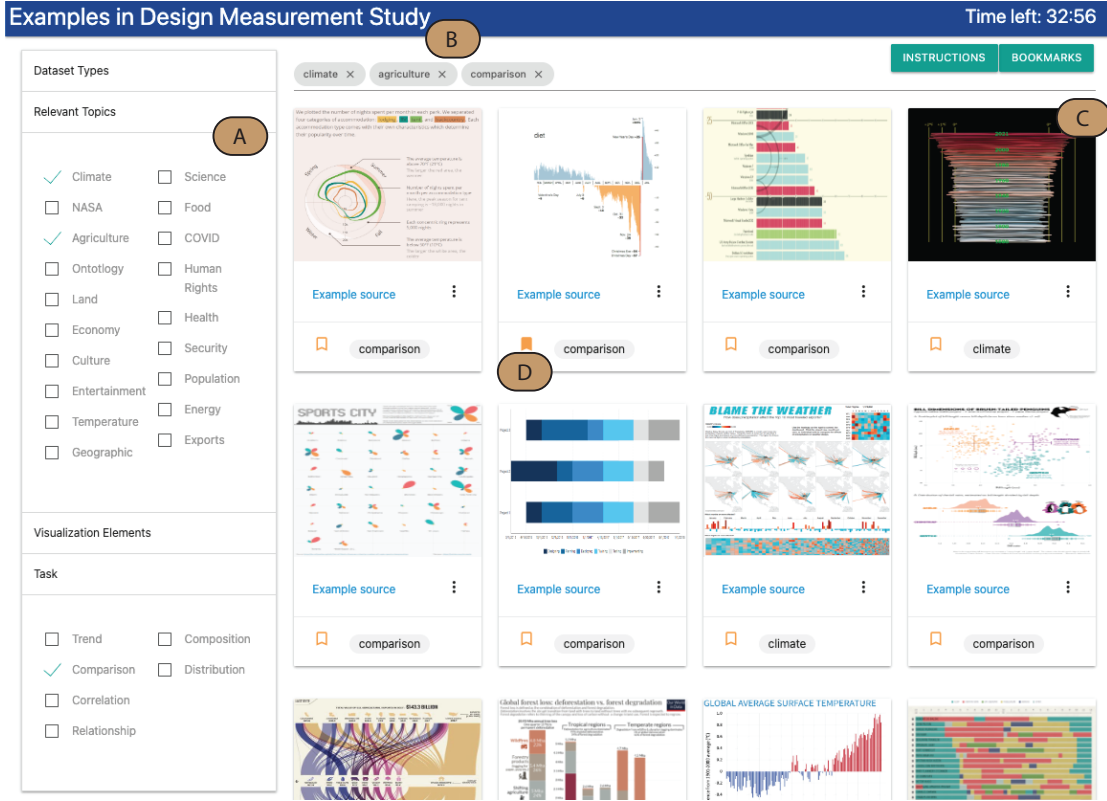


Figure 4.1: A snapshot of the experiment interface provided to participants during the experiment. (A) select tags to filter examples, (B) search for tags with auto-complete suggestions, (C) browse examples in the main panel, and (D) bookmark relevant examples.

corpus at the start of the brainstorming process. In the treatment condition, the introduction of the examples was delayed (T_{del}); participants were first asked to brainstorm design ideas for 15 minutes, then presented with the example corpus, and then asked to continue brainstorming.

4.3.2.2 Design Scenarios.

We included two design scenarios to capture how our results may generalize across design goals and datasets. In selecting datasets, we opt for those not commonly used in data analysis or visualization training to ensure that participants are not inadvertently primed toward certain visualization designs from prior exposure to datasets.

Blog Post Design Scenario (S_1): The first scenario is to design a visualization as the central

figure for a blog post discussing the effects of average temperature warming on the US climate ¹. Participants were provided with a dataset on the yearly average weather recordings from thousands of weather stations around the US from 1921 to 2022 [71]. Since our focus in this study is on the design process and not data analysis, we also included two insights on the rate of warming across the decades. Participants were then asked to come up with designs that communicate these insights using the provided dataset.

Bespoke Design Scenario (S_2): The second scenario focuses on the design of highly customized visualizations, which we describe as bespoke visualization designs. Participants assigned to this condition were given a dataset on vegetable crop production land use, deforestation drivers in Indonesia, and expected yield for vegetable crops [178]. Similar to participants in S_1 , participants were given insights on the activities that are drivers of deforestation in Indonesia and palm oil land use to yield ratio. Participants were then asked to create a visualization highlighting these insights. For this condition, we varied the design task by prompting participants to think creatively instead of focusing on communication as in S_1 . Participants were informed that their designs would be entered into the Iron Viz competition and would be judged on both the novelty and efficacy of their designs.

4.3.3 Examples Corpus and Interface

4.3.3.1 Corpus Construction

We searched for visualization designs shared online on popular blogs and resources such as Our World in Data [2], Storytelling with Data [3], and Information is Beautiful [1]. Informed

¹The prompts, datasets, and stimuli examples are provided on OSF.

Table 4.1: Example selection criteria and definitions

Criteria	Description
Knowledge	How well does the visualization express the basic facts about its underlying dataset (i.e. expressiveness)?
Comprehension	How easy is it for a user to understand the information that is being conveyed in the visualization (i.e. effectiveness)?
Relevance	(A) How valuable is the information in the visualization in relation to the experiment task? (B) How closely related is the visualization to the tasks participants will be asked to perform?
Visual Encoding	How appropriate are the selected data encodings in this visualization?
Aesthetics	How attractive is the visual design for the visualization (e.g., bespoke vs common visualization designs, color schemes used)?
Variety	How does this example improve the variety of designs (i.e. number of similar visualization types) in the entire example set?

by a subset of the evaluation criteria proposed by Burns et al. [42], we evaluated candidate examples across six key criteria: expressed knowledge, comprehension, relevance, encoding choices, aesthetics, and the variety of designs in the entire corpus (see Table 4.1). Two of the authors sampled a total of 50 visualizations. Each selected example was evaluated to ensure it was easy to understand (comprehension), expressive, and had appropriate visual encodings. To allow for serendipity, we optimized our selection process to include standard (e.g., bar charts, bubble plots) and customized (e.g., parallel coordinates plot) visualizations.

Tagging Stimuli Examples. To support easy filtering, two authors manually assigned a set of tags that describe each example’s features based on the four search criteria used by visualization designers identified by Bako et al [15]. These tags capture metadata that describe: 1) the underlying dataset characteristics (e.g., time series, categories), 2) the visual elements in each example (e.g., points, annotation), 3) the visualization task (e.g., comparison, distribution), and 4) the topic (e.g., climate, human rights).

4.3.3.2 Experiment Interface

We developed a web-based tool to instrument our experiment, shown in [Figure 4.1](#). Our tool consists of a faceted search interface that allows users to filter and browse visualization examples. Users can either choose tags from the side panel ([Figure 4.1A](#)) or type in a tag in the search box ([Figure 4.1B](#)). The main view of the interface ([Figure 4.1C](#)) presents users with examples matching the selected tags. Each example is contained in a single card with a thumbnail, a short description, and a link to see an enlarged view of the visualization. Users could bookmark examples by toggling the bookmark icon next to each example ([Figure 4.1D](#)). We logged users' interactions (e.g., clicks, hovers).

4.3.4 Participants

We recruited participants via forums such as Reddit's [r/dataisbeautiful](#), academic and professional mailing lists, and posts on X (previously Twitter). We selected 32 participants (Female=10, Male=22) between the ages of 18 and 44. Participants had between 3 months to 5+ years of experience creating visualizations. Participants reported creating visualizations for a variety of purposes: clients (n=12), blog posts and media graphics (n=7), personal portfolios and projects (n=13), data analysis (n=22), and design studies (n=12). 12 of our participants created visualizations weekly while others created visualizations less than once a week (n=9) or less than once a month (n=11).

To ensure that participants had experience with a wide range of visualization types, we collected information on the designs participants had created in the past, the tools used to implement visualizations, and how often they had created or used them to design visualizations.

We selected participants with experience creating visualizations outside of the three most popular visualization types (i.e., bar charts, line charts, and scatterplots) [23] and had used different types of visualization design tools such as direct manipulation tools (e.g., Tableau, PoweBI), programming tools (e.g., ggplot, D3.js) or design focused tools (e.g., Adobe Illustrator, Canva). Selected participants created a wide array of charts ranging from simple charts like bar charts, line charts, and scatterplots (n=32 for all three) to more complex charts like Chord Diagrams (n=9), Sunbursts (n=13) and Parallel Coordinate charts (n=17). Participants also used a range of tools to create visualizations, such as D3 (n=16), Matplotlib (n=22), Adobe Illustrator (n=17), and Microsoft Excel (n=27). The complete participant demographic information is included on OSF.

4.3.5 Experiment Procedure

Participants were first given a brief introduction to the purpose and objectives of our study, the nature of the tasks they would be asked to complete, and an overview of their rights. Participants were then asked to complete the consent form, after which the experimenter provided participants with a link to the experiment tool and a Google Jamboard for their sketches. Participants were randomly assigned to one of our four conditions ($T_{srt} \times S_1$, $T_{del} \times S_1$, $T_{srt} \times S_2$, and $T_{del} \times S_2$). Each participant was presented with their assigned design scenario and was given 5 minutes to explore the given dataset in a linked Google sheet. Once the 5 minutes were over, the participants were prompted to return to the interface and asked to brainstorm and sketch design ideas for 35 minutes. Participants in the baseline timing (T_{srt}) condition were allowed access to the experiment interface immediately after the dataset exploration session. Participants in the delayed timing condition (T_{del}) were asked to begin the brainstorming activity immediately after

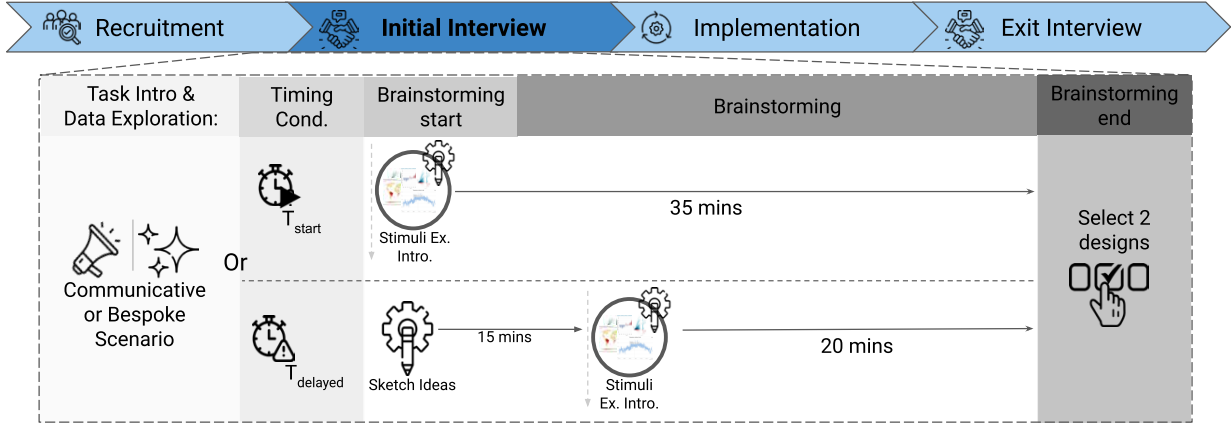


Figure 4.2: A depiction of our experiment protocol focusing on the initial interview. After familiarizing themselves with the scenario description and dataset, participants begin the brainstorming process. Depending on the *timing* condition, participants are either provided with examples at the start of the process (T_{srt}) or asked to brainstorm ideas before introducing examples (T_{del}).

the dataset exploration session. After 15 minutes of brainstorming, participants in the delayed condition were given access to the experiment interface. Once participants were in the experiment interface, they could explore and interact with examples at their discretion. We asked the participants to bookmark relevant examples and to think aloud.

Once the brainstorming session was complete, the experimenter asked the participants to select their top two designs. The experimenter then implemented these two designs asynchronously using Tableau, Data Illustrator [119], D3 [36], Vega-Lite [184], Charticulator [172], or Microsoft Excel at the experimenter’s discretion. A follow-up meeting was scheduled to collect feedback from the participants on the implemented designs and to allow them to complete the exit interview. Both the initial interview and the follow-up interview took place remotely on Zoom, and each interview was audio and video recorded. Participants were compensated with a \$20 gift card.

4.4 Quantifying Curated Examples and Design Outcomes

In this work, our intention is not only to understand how our five factors influence visualization design but also to explore what aspects of the designs are measurable and how to measure the influence of examples on these aspects. Similar to prior work [134], we code the features of the examples in our corpus and the sketches produced by participants (section 4.4.1). Subsequently, we use these codes to derive metrics that describe different *properties of the example sets* curated by participants (section 4.4.2). Below, we discuss our coding process, derived metrics, and results of preliminary data evaluations.

4.4.1 Coding Visualization Design Components

We first performed qualitative coding to identify each visualization’s data types, marks, visual encodings, layouts, compositions, and annotations. One of the authors created an initial codebook based on surveys of existing visualization taxonomies [23,51,88,127,147,167,170]. A subset of 5 random examples was selected from the example corpus, and three authors independently coded the examples and met to discuss the generated codes and refine the code book. Once the codebook was finalized, the three coders independently coded the remaining 45 examples and design sketches produced by participants. The coders achieved Krippendorff’s Alpha inter-rater reliability score of 0.85 [110]. Our codebook is included on OSF.

Data Types. We capture the type of data attributes used in each visualization using the data classifications provided by Mackinlay et al. [127] to code the quantitative-independent (qi), quantitative-dependent (qd), categorical (c), and categorical-date (cd) data attributes encoded in a design. We assign the index for each identified attribute as a concatenation of a suffix and

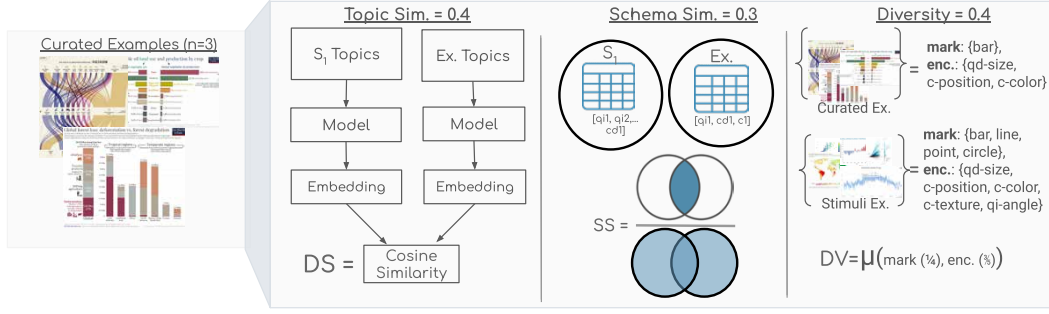


Figure 4.3: An illustration of how example properties are calculated for a set of curated examples. Please read 4.4.2 for more details.

unique identifier (e.g., $qi1, qi2, c1$).

Marks. To understand the basic graphical marks used to represent data items, we coded the individual marks based on the taxonomy of graphical marks described by Heer [88] and Munzner [147]. This includes marks such as points, rects, bars, lines, etc. We treat all graphical items with more than one mark as glyphs.

Visual Encodings. We use Munzner’s taxonomy on visualization channels [147] to capture visual encodings. We capture each encoding as a pair of the data attribute and the visual channel, e.g., if a quantitative-dependent variable is represented as position, we represent it as $[qd1 - position]$.

Layouts. Certain visualizations present multiple marks together as a group and rely on the positions of these individual marks to encode data for that group. For instance, stacked bar charts are composed of individual rect marks of varying lengths stacked together to form a single bar. At the time of performing this analysis, there is no comprehensive taxonomy on visualization layouts. As a result, the primary author surveyed papers [14, 51, 60] and documentation of popular visualization languages [117, 184] and used this to generate a set of codes to represent the layout of marks within an example which includes stacked, layered, packing, treemap,

radial, grouped, network, and branched.

Annotations. Annotations are often used in visualizations to add context [55] or draw readers’ attention to specific parts of a visualization [109,113]. To code the annotations present in each example, we use Ren et al.’s design space exploration of annotated charts [170], which we augment with a newer taxonomy presented by Rahman et al. [167].

Composition. Composite visualizations consist of multiple views. For each composite visualization, we code individual views separately and rely on Javed and Elmqvist’s taxonomy [104] to capture the composition strategy used to combine the views into a single visualization.

Visualization Type. Finally, we rely on Battle et.al.’s taxonomy [23] to capture the types of visualization[s].

4.4.2 Deriving Properties of Curated Example Set

The codes generated from our analysis in section 4.4.1 provide descriptive data on the features present in individual examples. However, we need quantitative metrics to describe the *collective properties* for the entire set of curated examples. We rely on heuristics provided in past literature on the factors that modulate examples influence to derive these metrics (see section 4.2). Specifically, we focus on the *quantity* of curated examples and *diversity* of the entire example set. We also measure the similarity between the examples and the provided dataset captured in the *schema_sim.* and *topic_sim..*

Quantity of Curated Examples: Participants were asked to bookmark examples they felt were relevant or inspiring during the example exploration stage. We also collected data on the examples that participants interacted with (i.e., expanded to inspect additional details closely) to

capture unintentional influences of examples they were exposed to [133, 135]. Together, these bookmarked and expanded examples represent the set of *curated examples* produced by participants.

Topic Similarity: We calculate how close the dataset topic matches the topic[s] of selected examples, which we call the *topic_sim.* of a curated example set. For each curated example, we extracted short descriptions of the visualization from their source website (these descriptions were also presented to participants during the study). We use Open AI’s ChatGPT 3.5 [154] to generate topic summaries for each description, which were manually evaluated for correctness by the lead author. Our scenarios had the following data topics $S_1: \{\text{global warming, temperature trends, data analysis}\}$ and $S_2: \{\text{palm oil, deforestation, land use}\}$. The extracted topics for all examples and the scenario prompt provided to participants were embedded using sentence transformers [169]. We then compute the cosine similarity between the two embeddings for each example. The similarity scores are then normalized. For instance, in Figure 4.3, a participant in the S_1 scenario bookmarked examples *img-003*, *img-010*, and *img-008*, comparing the topic for each of these examples to the scenario topic produces a *topic_sim.* score of ≈ 0.4 . Participants with scores closer to 1 curated examples that are closer to their assigned scenario topic.

Schema Similarity: We were also interested in examining if participants selected examples with similar data schema to the dataset provided to them. For the two datasets used in the scenarios provided to participants, S_1 had a data schema: $\{qi : 4, cd : 1\}$. While S_2 had a data schema: $\{qi : 8, cd : 2, c : 4\}$. We computed a *schema_sim.* score between the participants’ assigned dataset and their curated example set. Related work on schema matching has explored the use of

data types and distributions [27], linguistic similarity within attribute names [115, 130], and meta-data [129] to identify similarities between dataset schema. While we cannot access the underlying data for each example, we borrow techniques from past work and rely on the codes generated for data types in section 4.4.1 to compute this metric. We represent examples and scenario datasets as sets of unique data attributes x and y , respectively. We calculated the Jaccard Index [102] $\{J(x, y) = |x \cap y| / |x \cup y|\}$ between the data attributes in each example and scenario dataset. In Figure 4.3, we can compare the data attributes for our scenario dataset for S_1 to the attributes in our example (ex) = $\{qi_1, cd_1, c_2\}$. The overlapping attributes between these two would be $S_1 \cap \text{ex} = \{qi_1, cd_1\}$, the Jaccard Index of both sets of attributes, gives a *schema_sim.* score of 0.25. We normalized the scores to generate a single score between 0 and 1 per participant. Participants with scores closer to 0 selected examples with less *schema_sim.* to the provided dataset and vice versa.

Diversity: We compute the *diversity* of examples, which captures the explored solution space investigated by designers. Our analysis computes diversity by comparing the proportion of unique visualization components in curated example sets to the unique components in the entire example corpus. Our intuition is that curated example sets with a higher proportion of unique elements per component are more diverse than those with a lower proportion of coverage.

For each set of curated examples, we itemize the unique elements for each visual component as identified in section 4.4.1. This process is repeated for the entire example corpus provided to participants. Figure 4.3 shows a simplified version of this calculation: the curated examples contain *mark* : $\{bar\}$, *encodings* : $\{qd - size, c - position, c - color\}$ and the entire corpus of stimuli examples contains *mark* : $\{bar, line, point, circle\}$, *encodings* : $\{qd -$

size, c - position, c - color, c - texture, qi - angle}. For each component, we calculate the ratio of elements in the curated example set to those in the example corpus. We then compute the *diversity* score by averaging the ratios, which is equivalent to assigning the same weights to each component ratio so that the maximum possible score is 1.

4.4.2.1 Influence of Timing on Curated Examples Properties

Participants curated an average of 7.7 examples ($sd = 4.4$), with average scores of ≈ 0.3 for *topic_sim.* ($sd = 0.04$) and *schema_sim.* ($sd = 0.1$), and ≈ 0.6 for *diversity* ($sd = 0.18$). Figure 4.4 shows the distributions of scores across the three derived metrics by timing and design scenario. One participant did not bookmark or interact with any example during the experiment; data for this participant was excluded from the analysis. During our analysis, we found a high correlation between the *quantity* of curated examples and *diversity* ($r(29) = 0.86, p < .001$). As a result, we exclude the quantity of curated examples for the rest of our analysis as the *diversity* of examples is a more meaningful measure.

Timing influences curated examples' data topic similarity To evaluate how the *timing* of example introduction may have influenced the properties of curated examples, we fit linear mixed effects models with the three calculated metrics (*topic_sim.*, *schema_sim.*, and *diversity*) as dependent variables, *timing* as the fixed effect and scenario as the random effect. Our analysis presents a statistically significant effect of *timing* on the data *topic_sim.* ($\chi^2(1) = 4.35, p = .037$). *Participants who saw examples at the start of the experiment are likely to curate examples whose data topics are 6% ($\beta = .06$) more similar to the dataset they are working on.* We find no statistically significant result for *timing* on *diversity* and *schema_sim.*

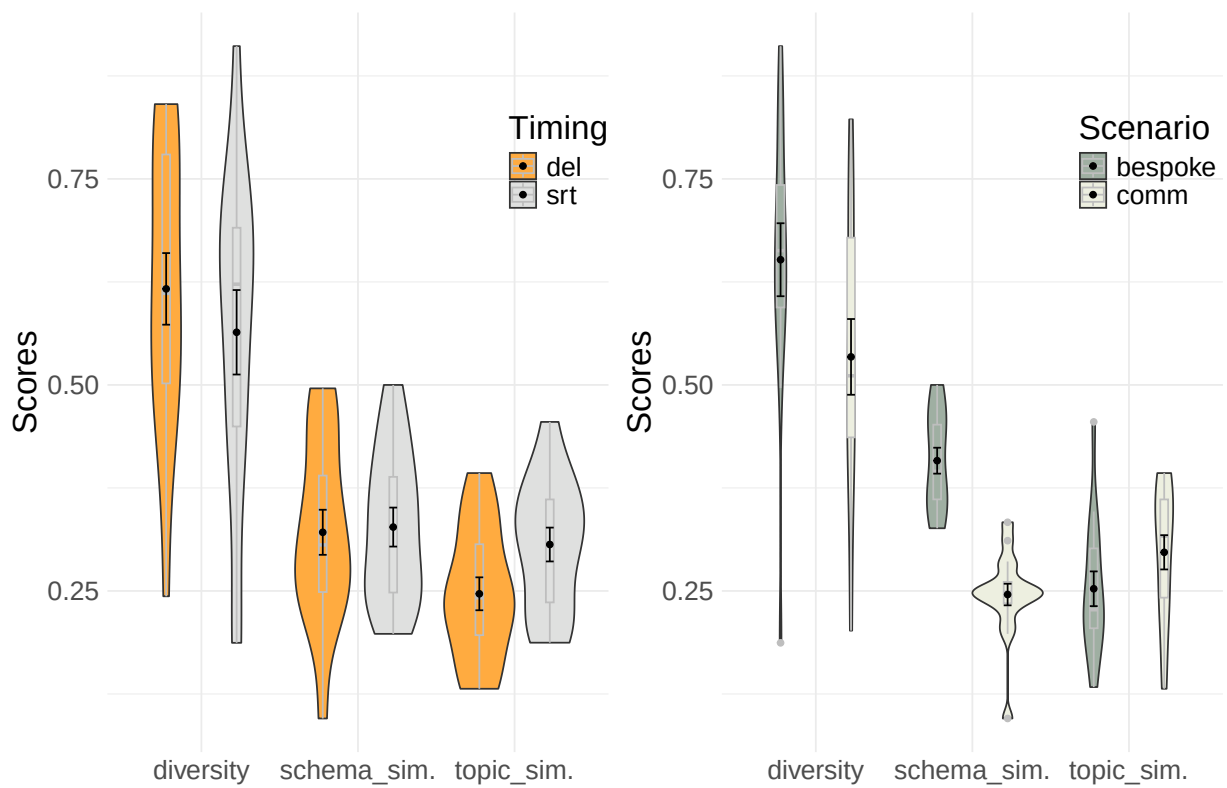


Figure 4.4: Distribution of scores for properties of curated examples based on the different experiment conditions in our study.

4.5 Measuring Design Outcomes and its Modulators

We conducted a series of quantitative analyses to identify what factors modulate the number of design outcomes produced by participants, the variety of design outcomes, and the number of ideas transferred from examples into designs. In this section, we discuss the data collected and the results of our analyses.

4.5.1 Data Description

A total of 128 sketches were produced by participants ($\mu = 4$, $sd = 1.8$). Similar to our coding process described in [subsection 4.4.1](#), we labeled the components of each sketch created by participants. We observe that bar charts account for 25% of all examples bookmarked by participants, making them the most frequently curated examples. Similarly, we see that 33% of the sketches were bar charts as shown in [Figure 4.5b](#). We also find that position was the most commonly used visual channel, making up 45% of all encodings used in the sketches and 40% in the curated examples (see figure [Figure 4.5a](#)). Bar charts are commonly used by designers [17, 23], and their popularity may be a confounding factor for the types of designs participants produced during our experiment. We also considered that the designer’s experience could be a confounding factor in the designs produced across the experiment conditions. We account for this by including the number of years designers have spent creating visualizations in our analyses described in [section 4.5.2](#).

We find significant overlap in components used in curated examples and design sketches; nonetheless, a few sketches deviate from this trend as they contain components not found in curated examples. For instance, we found sketches that use custom glyphs and icons as the primary

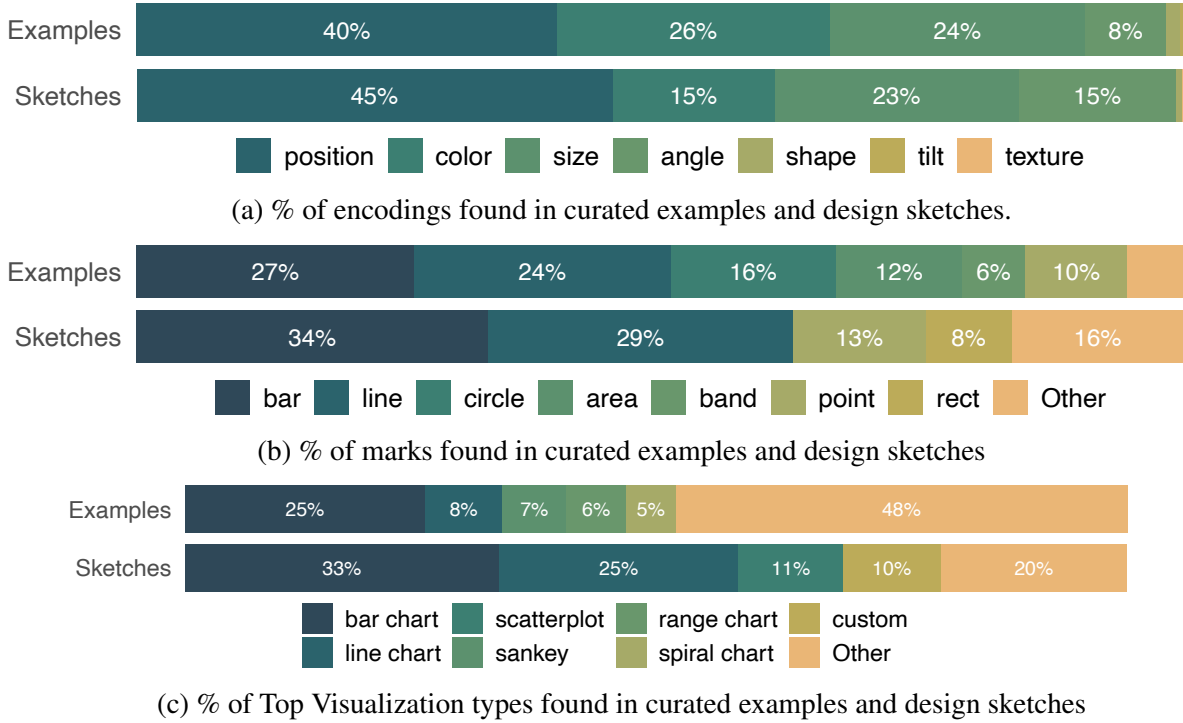


Figure 4.5: Overview of components found in curated examples and design sketches produced by participants. “Other” contains the sum of all components with less than 5% of each group.

marks and texture as an encoding, as seen in Figure 4.5b. The sketches that participants produced also had more annotations and layouts than the curated examples and included compositions and interactions not found in curated examples, as shown in Figure 4.6.

4.5.2 Influence of Timing and Curated Examples On the Quantity and Variety of Designs

We sought to explore the influence of *timing* and curated example properties (*diversity*, *schema_sim.*, and *topic_sim.*) on the *quantity* and *variety* of design outcomes. For example, does curating diverse examples lead to a higher number of designs and more design variety? Or might early exposure to examples lead to a decrease in the number and variety of designs? Here, *quantity* simply refers to the total sketches produced by participants. While

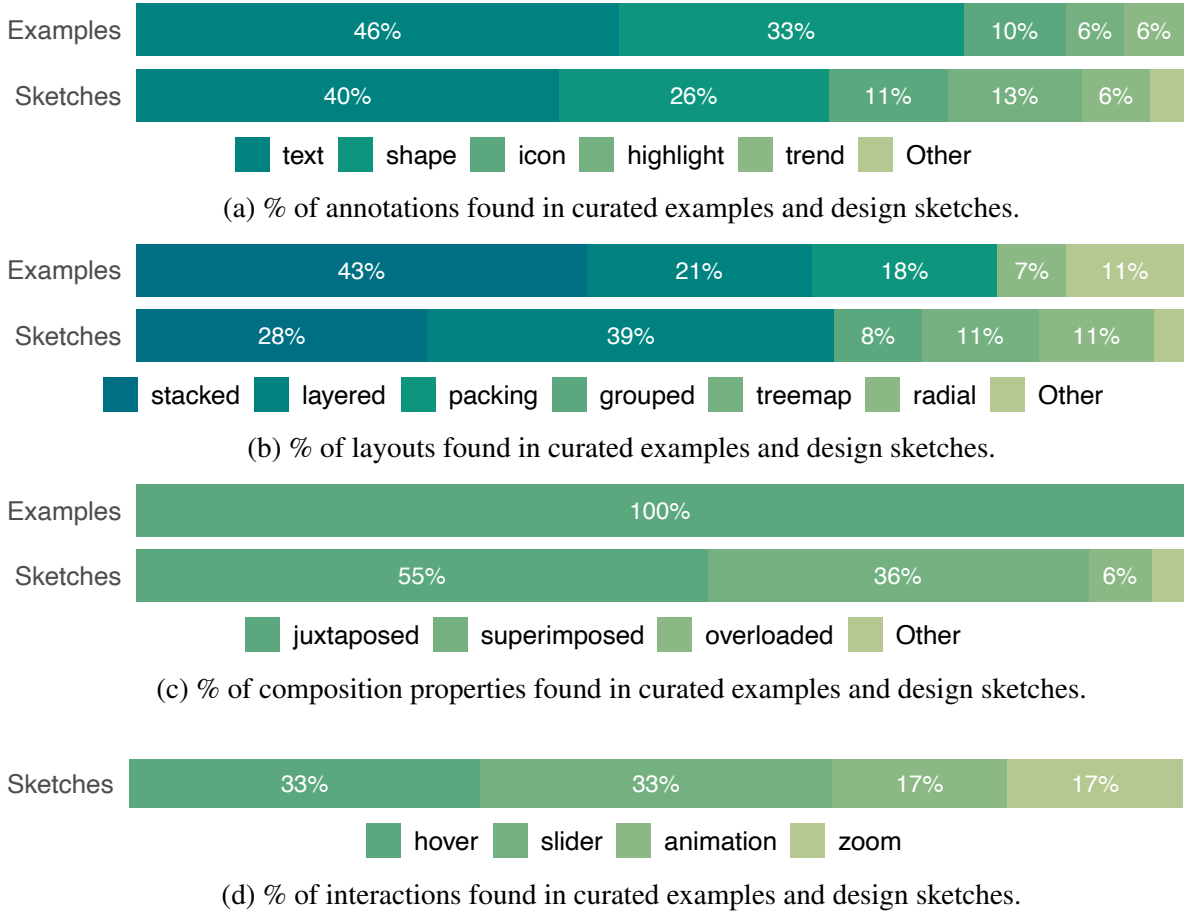


Figure 4.6: An overview of the style-related components found in curated examples and design sketches created by participants.

variety measures the number of unique components (i.e., encodings, marks, etc., identified in [section 4.4](#)) included in design sketches. We fit generalized linear mixed effects (GLME) models using *timing* as the fixed effect with participants' design experience (i.e., years creating visualizations) as the random effect for each dependent variable: *quantity* and *variety*. An analysis of the intercepts of the GLME model which includes the scenario as a random effect, yields very small coefficients (≤ 0) for the scenario. This indicates no systematic effect from including the design scenario in the model. As such, a simpler model without the scenario is preferred. We report the results of these analyses below.

Delay in introducing examples increases the quantity of designs created by participants.

We observe a significant effect of the *timing* on the *quantity* of designs produced ($\chi^2(1) = 5.48, p = .019$). *Participants who were provided with examples at the start of the experiment (T_{srt}) produced (35%) fewer designs on average compared to participants who were provided with examples after brainstorming (T_{del}) ideas ($\beta = -.43$).* These findings are consistent with prior work that finds a delay in introducing examples is more effective at improving the number of ideas produced while problem solving [144, 194].

Delay in introducing examples increases the variety of designs created by participants.

On average, participants used 9 unique components in their designs ($sd = 3.1$). Likelihood ratio tests show a significant effect of *timing* on the *variety* of design ideas used by participants ($\chi^2(1) = 4.54, p = .033$). *Participants who were provided with examples at the start of the experiment (T_{srt}) produced designs with 26% fewer unique components compared to participants who saw examples after brainstorming ideas.*

Example Properties do not influence the number or variety of design outcomes. To assess if curated examples influence the *quantity* and *variety* of design outcomes, we fit generalized linear models for the dependent variables with example properties as independent variables. We observe no significant effect of *diversity*, *topic_sim.*, or *schema_sim.* on either *quantity* or *variety* of design outcomes.

4.5.3 Influence of Curated Examples on Idea Transfer

Part of our investigation includes exploring objective ways to measure the transfer of ideas from examples into final designs. In prior work, idea transfer was measured by simply counting

overlapping features between examples and designs. However, this technique cannot be directly applied here as participants may have underlying biases towards certain designs based on their prior experience, evident in the prevalence of bar charts in the sketches produced by participants. This introduces a confounding factor that needs to be mitigated.

Consequently, we made two accommodations in our analysis. First, there is no way to measure the pure influence of examples on designs for participants exposed to examples at the start of the brainstorming process. Hence, we do not include any of the sketches produced by participants in the T_{srt} condition. This brings our total number of sketches examined to 77, i.e., $\approx 60\%$ of all sketches produced during the experiment. Second, to account for participants' prior knowledge, we assume that *idea-transfer* occurs only if a design component used in a sketch is also present in at least one of the examples bookmarked by a participant but not present in any of the sketches they created before exposure to examples. For instance, if a participant bookmarks example `img-001`, which has a quantitative dependent variable encoded as `size` (`qd-size`) and the same encoding is used in a new sketch(sk_3) but not in any of the sketches they had previously created. We record `qd-size` as a component that has been transferred from `img-001` into sk_3 . We found 41 instances of *idea-transfer* among 13 participants.

Higher schema similarity leads to more transferred ideas. We considered how the properties of examples might influence the number and types of *idea-transfer* by participants. For instance, do higher *diversity*, *schema_sim.* and *topic_sim.* scores lead to a decrease in *idea-transfer*? We fit a generalized linear model to evaluate how the number of ideas transferred (dependent variable) is affected by the curated example set's properties (independent variables).

Our results show a significant effect between *schema_sim.* and the number of transferred ideas ($d = 6.38$, $p = .02$). These results indicate that it is likely that *a unit increase in the schema_sim. of curated examples will result in an increase of 4.08 in the number of transferred ideas*. We find no significant results for *diversity* or *topic_sim.*

Section 5 Summary.

We explored what and how factors such as the timing of example introduction and the properties of curated examples influenced design outcomes. Our findings show that:

- When examples are introduced after initial brainstorming of ideas, designers are likely to produce more designs and use a wider variety of visualization components in their designs.
- Designers who curated more examples resembling their target data schema tend to copy more ideas from examples.

We acknowledge that due to the small population of designers in this exploratory study, further research is needed to validate the generalizability of these results. We discuss the implications of these findings for future work in [section 4.7](#).

4.6 Why and How Designers Incorporate Examples

In this section, we present the results of our qualitative analysis to understand designers' behavior during the experiment and their rationale for using examples. Two coders independently coded video recordings of our interviews, capturing participants' utterances related to the design process. One of the coders consolidated these codes, organizing them into clusters and then refining these clusters to identify emergent themes. Due to space considerations, we focus on

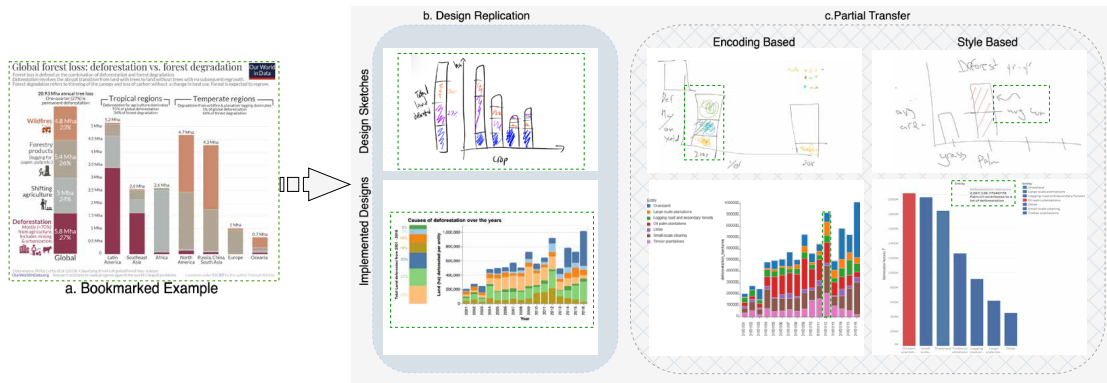


Figure 4.7: A depiction of the forms of idea transfer identified in sketches produced by three participants who bookmarked the same example. In design replication, the entire example is copied, whereas, in partial transfer, participants copy either encodings or styles from the example.

two key themes from our analyses: the features influencing example usage and the forms of idea transfer. We also present a review of infeasible designs and participants’ reflections on implemented designs.

4.6.1 What Example Features Influence Designers?

4.6.1.1 Features that attract designers to examples

Relatedness and uniqueness of the data presented in the example. The primary reason expressed by participants for choosing examples was the relationship between the data they wanted to present and that which is present in the examples. Participants often chose examples that were compatible with their data (n=6) or the examples presented the same data in a unique (n=3) or different way (n=6).

Similarity in visualization tasks. Participants also selected examples that support similar *task[s]* (e.g., comparison, trend) to the task they were trying to present (n=7). P11, when looking at an example of a ridge-line plot (img-041) remarked “...*this is something which we could leverage to try and distinguish the worst 5 years in terms of the temperature...I like this. This is like a*

combination of exactly comparison and annotation.”

Similarity of design to prior ideas. Certain participants were interested in how close the visualization design was to designs they had already created. All but one of the 6 participants who expressed this sentiment had seen examples only *after* brainstorming ideas. These participants were attracted to examples not just for design similarities but also to see if the examples had additional design elements that could improve their original ideas. For instance, when P17 encountered `img-10`, they said, *“Oh, this is like exactly what I did. Did they do anything better? They have annotations, ... instead of trying to put them on the Y-axis, they put a percentage of the total. That’s kind of useful.”*. P17 then proceeded to sketch out a new design that incorporates `img-010`. This revised sketch can be seen in [Figure 4.7b](#).

Visual Composition. Participants also expressed that they were influenced by the *composition* of design elements used in the examples (n=23). These include the overlay of multiple visualizations in a single chart (n=6) or encoding choices used in the design (n=2). Participants were also drawn to the use of annotations (n=6), layouts of marks (n=4), color schemes (n=3), and general design aesthetics (n=2).

4.6.1.2 Features that deter designers from examples

Confusing designs. Participants reject examples if they have difficulty understanding what is being presented in the visualization (n=9). Participants described these examples as being *“too complex”*-P28, *“confusing”*-P10, P9, P28, or *“unreadable”*-P25. Sometimes, the confusion may stem from a lack of clarity on how to apply the example to their data. For instance, P20 remarks on `img-001` *“I’m just worried that since temperature anomaly goes negative. That*

would confuse people a little bit. so I would have to find another way to show that...”

Incompatibility with data. We observed that participants would reject example designs if the example is incompatible with the data they are working on. For instance, participants rejected examples if they had different data types (n=5) or if they felt the design would not represent temporal data properly (n=3). Participants would also reject examples if they could not mentally map the data they had to the data encodings (n=6). This could be because some data transformations must be performed to format the data or introduce new variables.

Lack of inspiration. Finally, designers may reject the examples if they find the designs uninspiring (n=10). Participants often describe these designs as *“not helpful”*. Participants expressed that they found designs to be uninspiring when they were too similar to the ideas they had already brainstormed or were designs they had encountered a lot. One participant said, *“I’ve seen this visualization or similar visualization before. I don’t tend to see as much value in this as the visualization we just did [referring to their sketch]”* -P15

4.6.2 Forms of Idea Transfer

We were interested in understanding *how* designers transferred ideas from examples into their designs. We evaluated the 41 instances of idea transfer we identified in section 4.5.3 and reviewed notes taken during the experiment. *Our observations reveal two forms of idea transfer.*

Partial Idea Transfer. We observed that participants selected individual features from one or more examples to incorporate into their designs. This behavior has been observed in earlier work where visualization designers describe selecting and merging ideas into their designs [15]. Here, we distinguish between features that specify the parameters for binding data to graphical com-

ponents i.e., *encodings* (visual encodings and marks), and those that focus on the presentation of these features within the sketch, i.e., *styles* (layout, composition, and annotations). We found 23 instances of idea transfer, which involved participants copying encodings from examples into their sketches. We call this *encoding level transfer*, and an example of this can be seen in [Figure 4.7c](#) where a bookmarked example (img-010) has encoded the total deforested land as the size of the bars (qd-size) which participant P18 copied to denote the total deforested hectares in their sketch. We found 18 instances of *style-level idea transfer* in our participants' sketches. The most common form of style-level transfer was copying text and shape annotations as shown in [Figure 4.7c](#).

Design Replication. We observed situations where participants copied entire designs from examples. In this form of idea transfer, participants replicated examples, only substituting the data used. An instance of this is shown in [Figure 4.7b](#) where P17 copied example img-010. 13 participants engaged in this behavior, producing 21 sketches where entire designs were replicated. Participants often remarked interest in these designs because the examples presented data in clever, useful, and appealing forms they hadn't considered. For instance, P22, who copied example img-012 remarked: "*It's an interesting way of visualizing color, making it appealing to the eye and interesting to look at... That's pretty visually interesting. It almost looks like mountains and sunsets*". Two participants recalled examples they had seen in the past which were not part of the stimuli corpus and replicated these examples. This behavior of long-term recollection of examples has been reported in prior work [[15](#), [133](#)].

4.6.3 Reflecting on Implemented Designs

As part of our experiment, we implemented a subset of the design sketches created by participants. We present our observations on the feasibility of these designs as well as participants' reflections.

Infeasible Designs. A total of 64 designs were selected by participants, for implementation. However, not all designs selected by participants were successfully implemented; we classify these as *infeasible* designs (n=7) and successfully implemented designs as *feasible* designs (n=57). We find that *a mismatch between the intended visualization form and the data primarily caused infeasible designs*. An example of this type of mismatch is when P2 wanted a Sankey diagram that showed the flow of crop oil production to the total land used for the specific crop and the total deforested land, faceted by a grouping of years (before and after 1996). Given the dataset provided to this participant, the data transformation needed to prepare the data for this design was impractical as the data provided to them for the causes of deforested land did not include association with the crops. We also observed 3 cases of infeasible designs resulting from participants using data not provided in the dataset to create their designs. For example, P21 wanted to contrast temperature change influenced by human activity with expected temperature change without human influence. Overall, these infeasible designs reflect a lack of understanding of data constraints and abstractions, echoing observations made by Bigelow et al. [28]. All infeasible design sketches are included on OSF.

Reflections on implemented Designs. In follow-up interviews, participants were surprised to find that the implemented versions of their designs differed from their original sketches. For some participants, this unforeseen difference was positive (n=6) because the perceived data dis-

tributions or changes were not as drastic as they thought. For others, these differences highlighted design issues they had not considered, such as plotting two attributes with different domains on the same axis. Some participants described their designs as distracting (n=4) or too complex (n=4) or inappropriate for the given task (n=3).

4.7 Discussion

This chapter presents an exploratory study addressing critical questions about the influence of examples on design outcomes. Our investigation reveals that the data schema similarity and timing of example introduction significantly impact how designers curate and extract ideas from examples. We also uncover insights into designers' rationale for selecting examples and how ideas are incorporated into designs. We present discussions on implications for visualization designers, future research directions, and the limitations of our study.

Enhancing design diversity with timed example introduction. A major finding of our work is that delaying the introduction of examples increases both the number and variety of designs. Our observations on the influence of the delayed introduction of examples contribute to ongoing debates in the design community. We demonstrate a positive influence on delayed example introduction, supporting other works that have observed similar positive effects [144, 194, 202]. Building on this insight, designers may benefit more from using examples after initial brainstorming rather than to jumpstart the design process. This approach expands the space of explored ideas, allowing them to merge new concepts and reduce fixation. Furthermore, data visualization educators can adopt this strategy by delaying the use of examples during class exercises, allowing students to brainstorm first before incorporating variations from examples.

Considerations for incorporating example properties into designs. Participants often noted a mismatch between their initial design sketches and final implementations due to data-task incompatibility and overly complex or confusing designs. This issue was especially pronounced for those who fixated on certain examples and tried to force their data to fit those examples. It is important that when using examples, designers first evaluate example property dimensions such as data compatibility, visualization components, and design complexity. Understanding the interactions of these properties and their impact on design feasibility and comprehension could lead to more appropriate idea transference and better design outcomes.

Investigating emergent factors. Based on our analysis of participants' rationale for choosing examples, we find 3 new factors that researchers had not previously evaluated: the similarity of visual tasks, prior design ideas and preferences, and the visual composition of designs. Future work is needed to investigate how to capture these properties and measure their potential influence on design outcomes. This is particularly important for eliciting design preferences, as we observed during our study that designers struggle to verbalize their design intent.

Developing metrics to quantify example properties. As part of our analysis, we needed to develop metrics to quantify the properties of selected examples in terms of their topic and schematic relatedness to datasets, as well as the diversity of designs. We did not find sufficient guidelines in the visualization literature on developing such metrics, especially for the diversity of designs. In this work, we rely on the underlying visual and data components to generate scores and opt for simple methods for metric calculation. However, these techniques are exploratory and based on intuition. More work is needed to evaluate new methods for measuring design diversity.

While developing the code book for identifying the components of designs, we also found

gaps in the literature on taxonomies for describing the layout of marks within a visualization. As a result, our codes for evaluating the layout of visual designs may not be comprehensive as we had to survey a limited set of literature to extract these codes. We encourage the visualization community to look into how we can develop more comprehensive taxonomies for visualization layouts.

Analogical Inspiration for Visualization Design. In this study, we focus on understanding how existing visualization examples influence design outcomes. However, inspiration for designs is not restricted to visualization examples but can also come from artifacts found in nature, art, or physical objects around us. Prior research has shown that examples from fields not closely related to the design task influence the novelty and quality of design outcomes [31, 32, 47]. These works show the potential of augmenting ideation with analogical examples to promote the creation of new and unique concepts. Untangling the influences of analogical inspiration on visualization design outcomes is an interesting direction for future research.

Studying design iterations. During the experiment, we asked participants to sketch the design ideas they generated as they explored the dataset and stimuli examples. These design sketches are the by-product of the brainstorming exercises and represent the first iteration of visualization designs. Prior work has identified that visualization design requires multiple iterations before a final design is produced [209]. Our work only captures the first iteration of the design process; future work could consider a long-term study design where the experimenter and participants could implement and iterate on multiple designs. Nonetheless, our results present novel perspectives on using examples during the design process. Since we have identified that a delay in introducing examples significantly increases design ideas, future work can explore using examples to support

the different design stages. For instance, as a designer creates a visualization design, can we provide suggestions of alternative designs that may be more suited for their selected data attributes and assumed design task?

4.7.1 Limitations of Our Study

We acknowledge that our choice to limit participants to only 50 curated examples introduces limitations to our study design as it does not accurately reflect the real-world example search process for designers. However, we had to make concessions to have our study design be as close to Smith et al.’s protocol as possible. Additionally, both design scenarios in our work are about communicating insights, and we did not study the influence of examples on the design of visualizations for data exploration or analysis. Visualizations often need to be interactive to support analytic tasks. Quantifying and evaluating interaction design is beyond the scope of this work, and future research is needed to investigate this interesting direction. Finally, we considered the participants’ design experience based on their years of experience creating data visualizations. However, this is not a comprehensive measure of design experience. Our future work will examine the development of more robust measures of design experience.

4.8 Summary

In this chapter, we present a controlled experiment to address critical questions about the influence of examples on design outcomes. We find that delaying the introduction of examples increases both the number and variety of design ideas and that the extent of idea transfer—whether partial (encoding or style) or full replication—depends on data-schema similarity.

We also identify key factors guiding example selection, including underlying visual tasks and design compositions, and introduce quantitative metrics to measure these effects, such as data schema similarity and idea transfer. Ultimately, this chapter uncovers the dynamics between examples and design outcomes by establishing the key factors that drive the quantity and variety of design ideas generated and ideas copied during example-aided ideation. Fulfilling the second goal of this dissertation on measuring the example-aided design process.

Chapter 5: Ideation Strategies in Example-Aided Visualization Design

5.1 Introduction

During ideation, the breadth of a designer’s exploration of exemplar artifacts directly impacts the outputs of their ideation process [141, 195, 216]. For instance, prior work has found that a wide exploration leads to increased novelty and design variety [47], whereas a narrow exploration may lead to fixation [159]. Yet, little is known about the behaviors and strategies employed by designers to explore, filter, browse, and manage relevant examples during an ideation task. This gap limits our ability to develop effective support for key tasks that designers engage in during ideation. For example, identifying strategies that lead designers to produce designs with higher variation can inform the development of interactive tools for design retrieval and recommendation, towards fostering more diverse and creative outcomes.

This chapter addresses this gap by presenting an analysis of designers’ behavior data during example-aided ideation from [chapter 4](#), where we investigated the influence of examples on design ideation [15]. In this chapter, we shift our focus to understanding the *process* of example exploration during data visualization design ideation. Specifically, we aim to answer two research questions:

RQ1: What are the common example-aided ideation strategies used by participants, and

how are they influenced by the timing of example introduction?

RQ2: What relationships exist between example ideation strategies and design outcomes, specifically regarding the quantity and variety of designs produced?

During example-aided ideation, a designer may engage in a series of actions such as searching, browsing, and curating examples or sketching ideas [106, 107, 124]. Consequently, an ideation session consists of a sequence of such actions, which can be recorded as time-stamped logs. This work utilizes event sequence analysis techniques on participants' actions to uncover patterns and characterize strategies. We identify five key actions in the ideation processes: (1) **filtering** examples based on the dataset properties, topics of interest, visual elements, and tasks, (2) **browsing** through retrieved examples, (3) **inspecting** specific examples, (4) **curating** interesting examples, and (5) **sketching** design ideas.

To answer **RQ1**, we first focus on the **filtering** actions and the criteria participants utilize to query the examples to understand how the selection criteria evolve over time and how the timing of an example's introduction potentially modulated what criteria they utilized. We then examine all five types of actions, focusing on the frequency and duration of each action as well as differences in action subsequences between different timing conditions. Our findings indicate that participants initially anchor their filtering on either data or topic-related terms. Those who saw examples at the start of brainstorming branched out earlier and engaged more broadly, while those who encountered examples later adopted a deeper, more selective data-focused approach. Additionally, early exposure to examples led to more frequent filtering and longer browsing cycles, whereas later exposure prompted more deliberate filtering and focused inspection of individual examples.

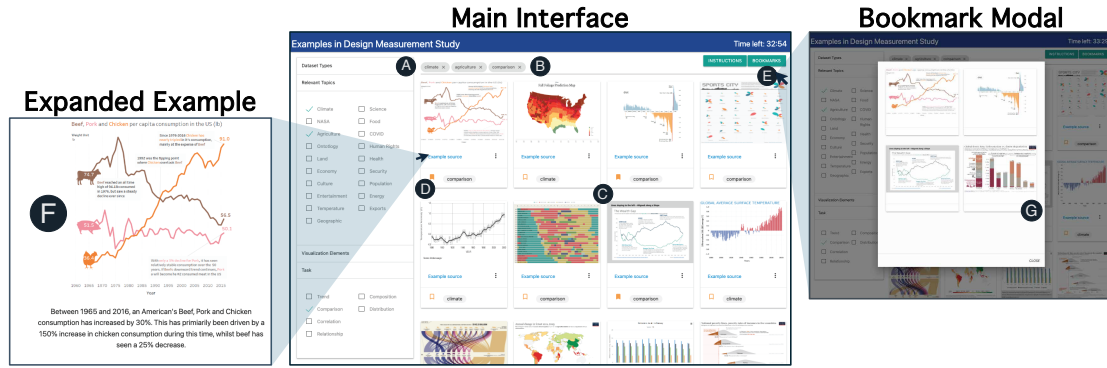


Figure 5.1: A screenshot of the web interface used to instrument the example-aided ideation process. The interface contains a filter panel (A), a search bar (B), and a main view to explore relevant examples (C). Participants can click on “example source” to expand individual examples (F) and toggle (D) to bookmark relevant examples. (E) and (G) are used to review bookmarked examples.

In our analysis for **RQ2**, we grouped participants based on their design outcomes—the quantity and variety of sketches produced—and then re-examined the emergent strategies for their ideation process. We find that consistent anchors and flexibility in filtering strategies are associated with higher design quantity and variety, respectively. Furthermore, *inspecting* and *filtering* actions are key to design outcomes: designers with high sketch quantity rely on intensive *inspecting* mixed with thoughtful *filtering*, whereas those with greater design variety engage in more diverse, *inspecting*-focused sessions.

Our work contributes a novel application of event sequencing techniques to understanding example-aided ideation. Furthermore, we present a characterization of the ideation strategies that lead to improved outcomes for data visualization design.

In the experiment detailed in [chapter 4](#), participants were provided with 50 stimuli examples that they could browse to identify relevant or inspiring designs. These 50 examples were manually curated from online blogs and resources such as Our World in Data ¹ and Information

¹Our World in Data

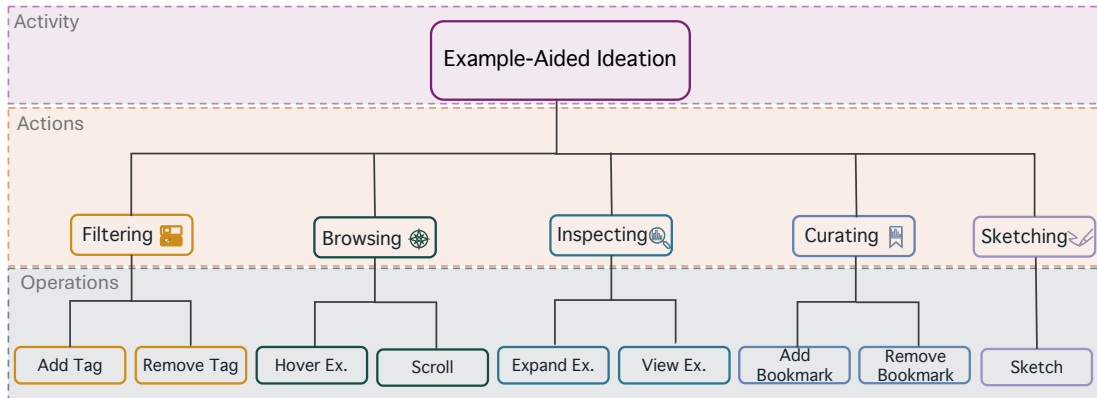


Figure 5.2: An overview of our operationalization of the operations and actions involved in the example-aided ideation activity.

is Beautiful ². To instrument the example browsing process, we provided participants with a web interface to filter and browse through the stimuli examples (see Figure 5.1). Using this interface, participants could either select or type provided tags to filter examples (Figure 5.1 A & B). Once tags are selected, the interface updates the main view (Figure 5.1 C) to present examples that match the tags selected by the participant in a grid view. Participants can then scroll through the presented designs, hover over each example, view a short description of the example, or view an enlarged version of the example (Figure 5.1 F). Once participants find an inspiring example, they can toggle the bookmark icon (Figure 5.1 D) to bookmark that example. We also provided a bookmark panel that participants could open to review bookmarked examples (Figure 5.1 E & G).

5.2 Operationalization of Example-Aided Ideation

In the experiment detailed in chapter 4, participants were provided with 50 stimuli examples that they could browse to identify relevant or inspiring designs. These 50 examples were

²Information Is Beautiful

manually curated from online blogs and resources such as Our World in Data ³ and Information is Beautiful ⁴. To instrument the example browsing process, we provided participants with a web interface to filter and browse through the stimuli examples (see Figure 5.1). Using this interface, participants could either select or type provided tags to filter examples (Figure 5.1 A & B). Once tags are selected, the interface updates the main view (Figure 5.1 C) to present examples that match the tags selected by the participant in a grid view. Participants can then scroll through the presented designs, hover over each example, view a short description of the example, or view an enlarged version of the example (Figure 5.1 F). Once participants find an inspiring example, they can toggle the bookmark icon (Figure 5.1 D) to bookmark that example. We also provided a bookmark panel that participants could open to review bookmarked examples (Figure 5.1 E & G).

During the experiment, we collected timestamped logs of participants' interactions within the interface. This includes interactions such as hovers over examples, button clicks, toggles, and chip selections. Motivated by Activity Theory [148] and Gotz and Zhou's characterization of visual analytic activities [82], we operationalize a hierarchical structure of the data for our analysis to capture the *activity*, *actions*, and *operations* participants engage in during the experiment.

Individual event logs collected during the experiment represent the *operations* that participants were engaged in. These low-level *operations* can be classified into five semantically meaningful *actions*: **filtering**, **browsing**, **inspecting**, **curating**, and **sketching**. The *actions* in turn constitute the example-aided ideation *activity* participants were engaged in during the experiment, which is the focus of our analysis in this work (see Figure 5.2). Below, we contextual-

³Our World in Data

⁴Information Is Beautiful

ize the connections between the five actions and underlying operations participants engaged in during the example-aided ideation activity. Additionally, we discuss how each action reflects a particular mental state that guides participants' interactions and design process. An overview of participants' ideation actions can be found on OSF.

Filtering. Within the interface, participants could `add` and `remove` tags to query a subset of the 50 stimuli examples. These operations of adding and removing tags constitute a filtering action, which corresponds to the mental state of *querying* for examples. In this state, participants are actively refining their search criteria to surface relevant examples that align with their evolving design goals. To facilitate this filtering action, each stimuli example was assigned a set of tags that describe metadata about the constituent visualization design based on the four search criteria designers utilize during example search [15], which we refer to as “*tag groups*”:

- **Data:** These tags describe the characteristics of the underlying dataset used to construct the visualization example, such as the data type (e.g., “*categorical*”, “*numeric*”) or the number of encoded variables (e.g., “*three or more variables*”).
- **Topic:** Tags were assigned to capture the main topic being presented in the example. For instance, a visualization presenting data on the net change in forest area measures received the tags “*climate*” and “*geographic*”.
- **Visual:** We also assigned tags to capture the visual elements utilized in the visualization, such as marks (“*bar*”, “*line*”, “*point*”), layouts (e.g., “*stacked*”, “*grouped*”), and annotations.
- **Task:** We assigned tags to capture the inferred intent of the visualization based on Amar et al.’s taxonomy of low-level analytical tasks [9]. For instance, a visualization showing 200

years of immigration to the United States from different parts of the world was annotated with the tags “*comparison*” and “*trend*”.

Browsing. Participants can casually `scroll` through and `hover` over each example shown in the main view. Each `hover` operation was logged, with the sequence of hovers captured as a browsing action. This browsing action reflects a mental state of *exploration*, where participants are actively scanning and gathering inspiration from examples. During this exploratory state, participants may refine their understanding of the design space before committing to an idea or strategy.

Inspecting. During the course of browsing through examples, participants can choose to target a specific example by `clicking` on the ellipse to view the example description or on the example source to see an expanded view of the example. These two operations delineate an inspecting action, which indicates a deeper engagement with a specific design to *internalize* an example. By carefully examining a specific example, participants may evaluate its components and decide to integrate aspects of the design into their own design ideas.

Curating. Once the participant identifies an inspiring example, they can choose to toggle the bookmark icon to either `add` or `remove` a bookmark. Additionally, during the course of the browsing session, participants can `open` the bookmark modal to review examples that have been bookmarked. All operations involving adding, removing, and reviewing bookmarks are referred to as curating actions. This action reflects the mental state of *managing* inspiring or relevant examples, where participants actively organize and refine their pool of reference materials. By curating examples, they create a personalized collection that aligns with their evolving design ideas.

Sketching. Participants were asked to sketch out design ideas as they explored examples. As such, participants could spontaneously break from their interactions with examples to sketch out design ideas. We manually annotated the start and end times for each of these recorded `sketch` operations, which constitute a sketching action. This action represents the mental state of *externalizing* ideas formed during the ideation process, allowing participants to translate their design concepts into tangible representations. By sketching, they refine emerging ideas and iterate on potential designs that solve their design task.

5.3 Analysis Methods

Our two research questions examine strategies in participants' example-aided ideation actions across two axes: the timing conditions participants were assigned to (**RQ1**) and the properties of their design outcomes (**RQ2**). For our analysis, we examine similarities and differences in ideation strategies for different cohorts of participants along two key dimensions: (1) the prevalence and evolution of example `filtering` criteria and (2) the emergent patterns associated with all five types of actions. We place a special emphasis on the analysis of `filtering` actions because they determine the space of examples participants engaged with during the ideation process. In the following subsections, we provide a brief review of event sequence analysis for browsing behavior and introduce our approaches to measure filtering and ideation strategies.

5.3.1 Analyzing Temporal Patterns in Browsing Behavior

Behavior data is inherently temporal. As such, event sequence analytics techniques and systems have been successfully applied to study user behavior across various domains. For example,

TipoVis [85] analyzes social and communicative behaviors among children, helping developmental psychology researchers to identify overlapping behavioral events. Similar approaches have also been applied to different contexts, such as online learning [50, 83, 145], collaborative writing behaviors [52], web session logs [112], login and security session analysis [151], clickstream data [118, 120, 230], and interactive visualization exploration [70].

These works illustrate the versatility of event sequence analysis in understanding complex user behaviors, from consumer analytics to education and collaborative work. We extend these techniques to examine design gallery browsing behavior, focusing on two key analytical objectives: pattern exploration and cohort comparison. Pattern exploration identifies common browsing strategies, while cohort comparison reveals behavioral differences across groups with varying conditions or outcomes. Since existing visualization systems already support these tasks [232], we adapt and repurpose these tools for our study.

Our approach integrates multiple complementary techniques with statistical analysis to develop a comprehensive understanding of example-aided ideation behavior. COCO [131] integrates visual analytics and statistical testing for comparing event sequences across different user groups. We apply COCO to identify statistically significant differences in the example-aided ideation patterns between different groups of participants (e.g., participants exposed to examples at different times). SentenTree [100]—originally designed for analyzing social media text—creates a graph-based representation while preserving linguistic context. Zinat et al. [233] adapted SentenTree for event sequence data, using a summarization algorithm to identify frequent subsequences and represent them as a connected graph. SentenTree reveals how behavior patterns branch and evolve over time. We utilize Sententree to study the evolution of user filtering actions during example-aided ideation.

5.3.2 Example Filtering Strategies

All participants performed 690 filtering actions in total. One participant did not engage in any filtering actions, and data for this participant was dropped from the analysis. However, the selection of tag groups is not mutually exclusive—tags from multiple groups are often used in conjunction for filtering. As such, we were interested in understanding how participants combined and navigated through tag groups over the course of their filtering actions. We chose to focus on higher-level tag groups rather than low-level tag values as this approach offers a clearer representation of participants’ filtering strategies.

Tag-Group States: Since selected tag groups are not mutually exclusive, there are 16 possible states (shown in [Figure 5.3](#)) that capture the set of selected tag groups at any time. We analyze the frequency distribution across all 16 states, including repeated visits to the same state. We perform Chi-Square tests for homogeneity on the frequency distributions to identify states with statistically significant frequencies.

Tag-Group Evolution: To identify emerging patterns in participants’ filtering actions over the course of the experiment, we filtered out non-filtering actions to isolate each participant’s sequential filtering actions. We analyzed how their filtering actions evolved over time and conducted an assessment of the selected tag groups using a re-implementation of SentenTree [[100](#)] for event sequence data developed by Zinat et al. [[233](#)]. For our comparison of transition evolution across the timing conditions, we adjusted the support value to ensure a minimum of two sequences are included in each reported pattern (overall=0.05, start=0.1, delayed=0.1). Our analysis of tag-group evolution is qualitative and intended to complement our quantitative findings. In the following sections, we offer a descriptive overview of tag evolution, and detailed snapshots can be found

on OSF.

5.3.3 Example-Aided Ideation Strategies

We identified five high-level actions in [section 5.2](#). To identify emergent strategies in participants' actions during example-aided ideation, we aggregate consecutive events of the same action into “bursts”, defined as a continuous period spent on a single action before transitioning to another. For instance, if a participant explores several examples through repetitive `hover` operations, we merge them into a single `browsing` burst. For the remainder of this chapter, we refer to these “bursts” simply as actions. We perform the four key analyses detailed below.

Action Frequency: To measure the prevalence of different action types, we count the frequency of each action among participants. We perform Chi-Square test for homogeneity of variance on the action frequencies to identify statistically significant differences across cohorts.

Duration of Action: The duration of each action reveals how participants allocate their time during the example-aided ideation session. We analyze action durations to compare typical engagement time for each action burst while controlling for the 15-minute difference in total example-aided ideation time between participants in *start* and *delayed* timing conditions. Note that this is different from the total time spent per action across the entire session.

Action Transition Frequency: We analyze transitions between consecutive actions to capture how participants switch between different tasks during ideation. Statistical comparisons of action transitions across cohorts posed methodological challenges. With five browsing actions and 20 possible transitions (excluding self-transitions), many transitions were rare or absent, violating chi-square test assumptions. For example, the `filtering`→`inspecting` transition was entirely absent

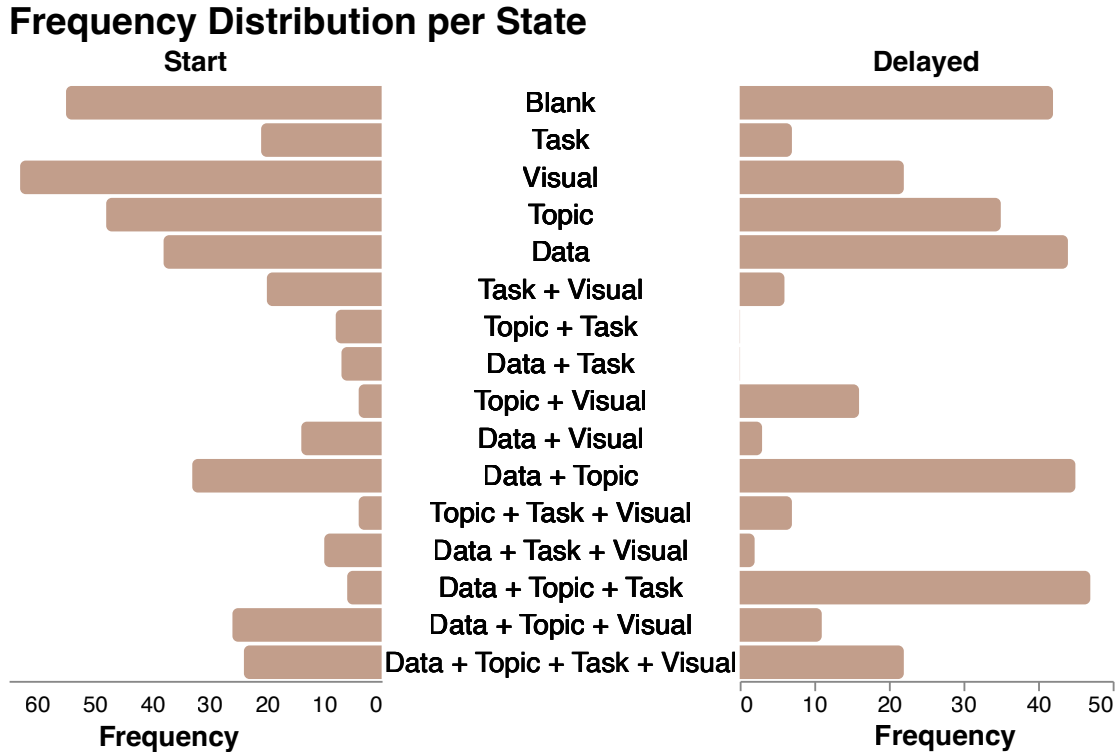


Figure 5.3: Frequency distribution across all 16 states by timing condition. The “Blank” state is a state where no tags are selected.

in the *start* condition. Given these constraints, we adopt a descriptive approach, calculating transition frequencies and percentages within each condition.

Action Subsequences Comparison: Using COCO [131], we analyze n-gram action subsequences (length 2–10) to identify recurring ideation patterns, considering significant subsequences at $p < 0.05$.

5.4 RQ1: Effect of Timing on Ideation Strategies

To investigate whether the timing of the example introduction influenced ideation strategies (RQ1), we conducted an analysis of participants’ actions across the *start* and *delayed* timing conditions. We summarize our key findings below. We refer readers to our OSF repository for a full report of all the analysis results.

5.4.1 Filtering Strategies

We found that participants primarily interacted with tags in the **topic** (n=336) and **data** (n=332) tag group categories. This was followed by the **visual** (n=254) and **task** (n=191) tag groups. The tag group associated with the first filtering action is overwhelmingly the **data** tag group. 25 participants started with **data**-related tag groups, while 5 began with **topic**, and 1 with **task**. As each participant progressed through their example ideation session, we observed an “*anchoring strategy*”, where participants anchor on the initial tag group they selected (i.e., **data** or **topic**) and repeatedly cycle through the tags within the same group. Later in the session, participants begin adding or removing tag groups while keeping the anchor group selected. Many participants incorporate **visual** tag groups only near the end of their example ideation.

Wider vs. deeper exploration patterns There were a total of 690 filtering actions; 381 of these actions were by participants in the *start* condition, while the remaining 309 were by the participants in the *delayed* condition. While participants in the delayed condition had less time to interact with examples, they generated only 18.6% less filtering actions as their counterparts in the start condition who had access to examples 15 minutes earlier.

A ChiSq test for homogeneity on the *frequency distribution* across all 16 states (Figure 5.3) revealed a statistically significant difference with a medium effect size between the *start* and *delayed* conditions ($\chi^2 = 107.39, p < .01, df = 15, V = 0.4$). Specifically, we find that participants in the start condition had higher frequencies for the **visual** state (adjusted standard residuals (z) = 2.37, $p < .01$). On the other hand, participants in the delayed group had higher frequencies for **topic**+ **visual** ($z = 2.38, p < .01$) and **data** + **topic** + **task** ($z = 4.83, p < .01$).

These results indicate that the timing of the example introduction potentially impacted which tag groups participants focused on. Furthermore, we observed that participants in the start group had a much broader exploration of the filtering tag space, with four core branches emerging after a max of 3 filtering events. On the other hand, participants in the delayed condition had a much narrower exploration of the filtering tag space, with branching happening much later in their filtering events. Participants in the delayed condition were also particular about when they engaged with certain tags. They had very little engagement with the **task** tag group when at most two tag groups were selected, avoiding the **topic** + **task** and **data** + **task** tag groups combinations altogether (see [Figure 5.3](#)).

Dominant anchors emerge across timing conditions. In the delayed condition, participants primarily utilized a **data**-anchored filtering strategy (n=15). These participants expanded their selected tag groups at different points to include **topic** (i.e., **data** + **topic**) and later on include **task** (i.e., to **data** + **topic** + **task**) after a minimum of 2 filtering actions. In contrast, participants in the start condition had 2 distinct anchors on **data** (n=10) and **topic** (n=5) in their exploration tree (see [Figure 5.4](#)). One participant utilized a **task**-anchoring strategy. Participants who began with the **data**-anchored filtering had a similar expansion pattern to those in the delayed condition. However, their tag group expansion advances beyond **data** + **topic** + **task** to include **visual**, leading them to a state where all tag groups are selected. Participants in the start condition who anchored on **topic** had less deviation in their exploration and also had shorter filtering operations than those who focused on **data**. Furthermore, we observed a recurring pattern among the **topic**-anchored filtering, where participants frequently contracted to the blank state before adding another term within the **topic** tag group.

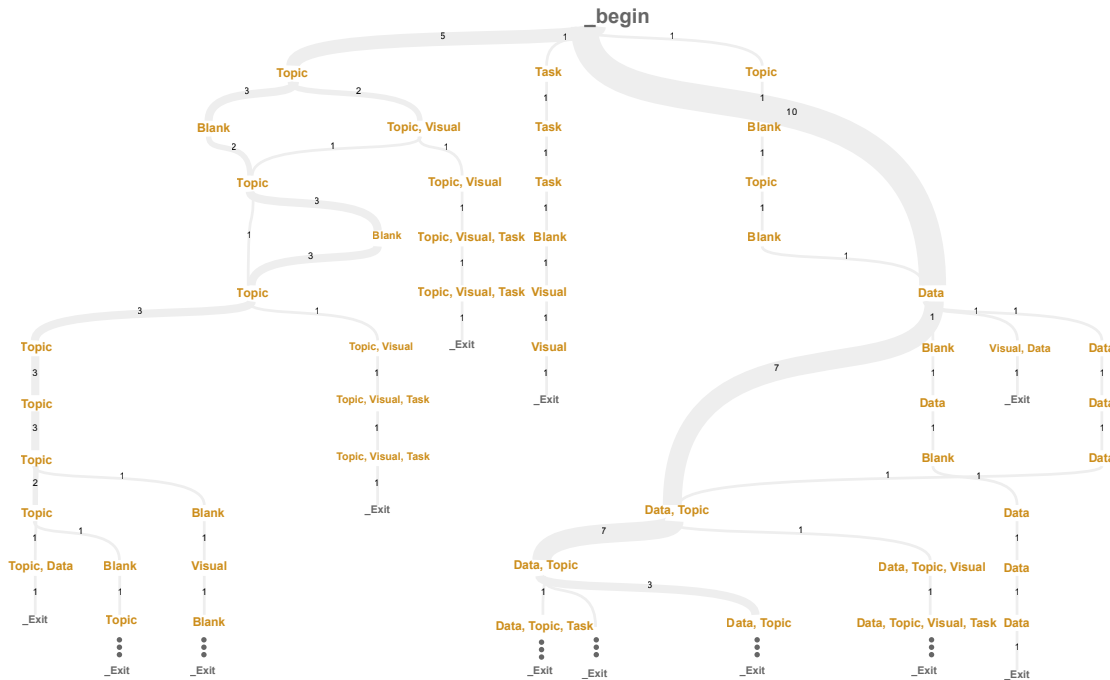


Figure 5.4: An example of filtering evolution over time. Presented in this figure are the filtering activities of participants in the start condition, demonstrating dominant anchors on **data** and **topic** tag groups.

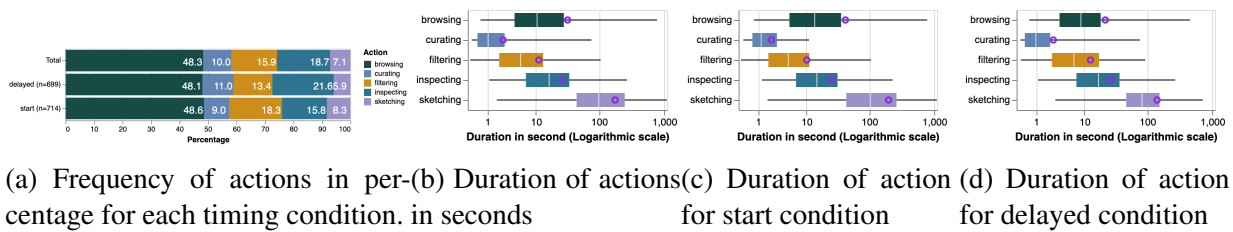
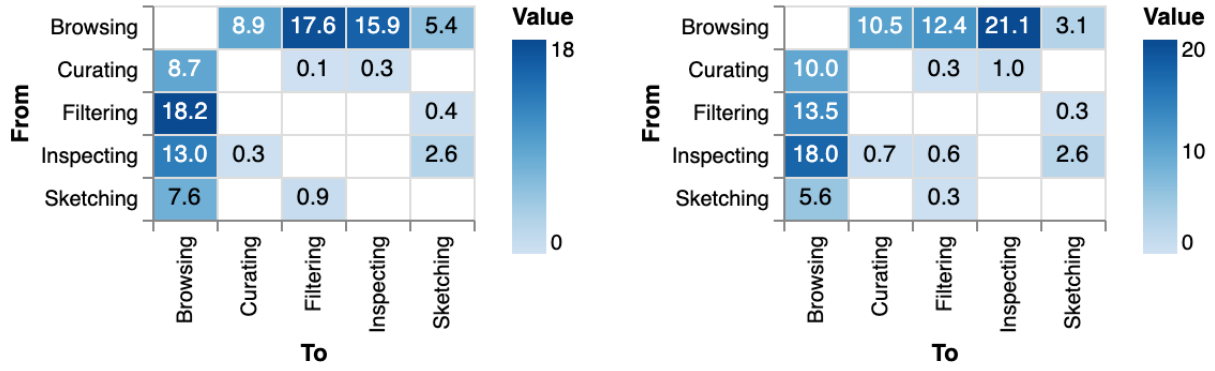


Figure 5.5: Action frequency and duration across different timing conditions

5.4.2 Ideation Strategies

In total, 1,413 actions of all five types were recorded during the ideation sessions. Of these actions, 714 occurred in the *start* condition and 699 in the *delayed* condition, resulting in 1391 (1413 – 32) pairwise transitions across 32 participants (start = 698, delayed = 683).

Browsing is the most frequent action, accounting for nearly half of all actions, whereas **sketching** is the least common (Figure 5.5a). As shown in Figure 5.5b, action durations are



(a) Start timing condition. (b) Delayed timing condition.
Figure 5.6: Transition distribution for start and delayed timing conditions.

heavily skewed, with mean values often more than twice the median, indicating a long tail of outlier bursts. The overall duration of all action categories showed significant deviations from normality (Shapiro-Wilk test). Among the actions, *sketching* is the longest, while *curating* is the shortest.

Timing condition affects action frequency and duration A Chi-Square test ($\chi^2 = 16.01, p - value < 0.01, df = 4, V = 0.11$) revealed significant differences in *action frequency distributions* across conditions, though with a small effect size. Adjusted standardized residuals (z) identified significant differences for *inspecting* ($z = 2.78, p - value < 0.01$), and *filtering* ($z = 2.52, p < 0.05$). The frequency distributions across timing conditions are shown in Figure 5.5a.

A Mann-Whitney U test on *action durations* showed statistically significant differences in *browsing* ($U = 63609.5, p < 0.001$), *filtering* ($U = 5077.5, p < 0.05$), and *curating* ($U = 3013, p < 0.05$). Duration distributions for both timing conditions are shown in Figure 5.5c and Figure 5.5d.

Start condition participants primarily focused on looking for examples Participants who saw examples at the start of their ideation session perform *filtering* more frequently than participants

in the *delayed* condition (*start* = 18.3%, *delayed* = 13.5%) as shown in Figure 5.5a. Their dominant transition features frequent **browsing** $\Leftrightarrow$ **filtering** alternations, making up 35.8% of their transitions (Figure 5.6a).

Subsequence analysis shows they repeatedly refine search criteria and then browse through resulting examples. They are also six times more likely to iterate between **filtering** and **browsing** before **curating**. The subsequence “**browsing**→ **filtering**→ **browsing**→ **filtering**→ **browsing**→ **filtering**→ **browsing**→ **curating**” appears in 37.5% of *start* condition participants vs. 6.25% of *delayed*. They also have longer **browsing** actions, with a 57.5% longer median duration compared to those in the *delayed* condition (13.7s vs 8.7s). These findings suggest that early example availability encourages more frequent filter refinement and iterative browsing.

Delayed condition participants primarily focused on examining examples Participants in the delayed condition participants spend less time per **browsing** action and have a higher frequency of **inspecting** actions (*start* = 21.6% vs *delayed* = 15.8%). This emphasis is evident in their transition patterns, where **browsing** $\Leftrightarrow$ **inspecting** transitions make up 39.1% of all transitions (Figure 5.6b). These participants also demonstrated 25% faster **curating**.

Interestingly, while participants in the *delayed* condition performed fewer **filtering** actions overall, they spent significantly more time per **filtering** operation (median = 6.6s vs. 4.8s; a 37.5% increase). These findings suggest they were more deliberate and thoughtful when adjusting **filtering** criteria, likely searching for specific examples that aligned with their already-formed design concepts.

5.5 RQ2: Quantity of Designs

In our initial experiment, we examined three metrics to assess participants' design outcomes: the quantity and variety of sketches all participants produced, as well as the ideas transferred from examples into sketches [16]. Since idea transfer can only be measured for participants in the delayed condition, we choose not to include it in our RQ2 analysis. As such, for **RQ2**, we focus on how the *quantity* and *variety* of design sketches produced by participants differ based on the filtering and ideation strategies they use. Below, we present the results of our analysis to understand how example filtering and ideation strategies differ based on the quantity of sketches generated.

5.5.1 Cohort Overview

Overall, participants produced an average of 4.0 ($sd = 1.8$) sketches. We first group participants into 3 cohorts based on the quantity of sketches they produced: *Low*, *Medium*, and *High*. These cohorts were created by calculating the average number of sketches and their standard deviation, then using half a standard deviation as the threshold. Participants with sketch counts below ($mean - 0.5 \times sd$) were categorized as Low, those above ($mean + 0.5 \times sd$) as High, and those in between were classified as Medium. Details on participant cohorts for the quantity of sketches can be found in [Figure 5.1](#). In this section, we use L_q , M_q , and H_q to refer to the low, medium, and high cohorts in sketch quantity. Similar to our analysis in [Section 5.4](#), we explored the example filtering and ideation strategies used by different participant cohorts.

Table 5.1: Participants split for quantity of sketches.

Cohort	Sketches	Participants
Low	≤ 3	<i>Start</i> = 10 <i>Delayed</i> = 5
Medium	4 – 5	<i>Start</i> = 5 <i>Delayed</i> = 6
High	≥ 6	<i>Start</i> = 1 <i>Delayed</i> = 5

Table 5.2: Participants split for variety of sketches.

Cohort	Variety	Participants
Low	< 7.6	<i>Start</i> = 7 <i>Delayed</i> = 3
Medium	7.6 – 10.6	<i>Start</i> = 7 <i>Delayed</i> = 5
High	> 10.6	<i>Start</i> = 2 <i>Delayed</i> = 8

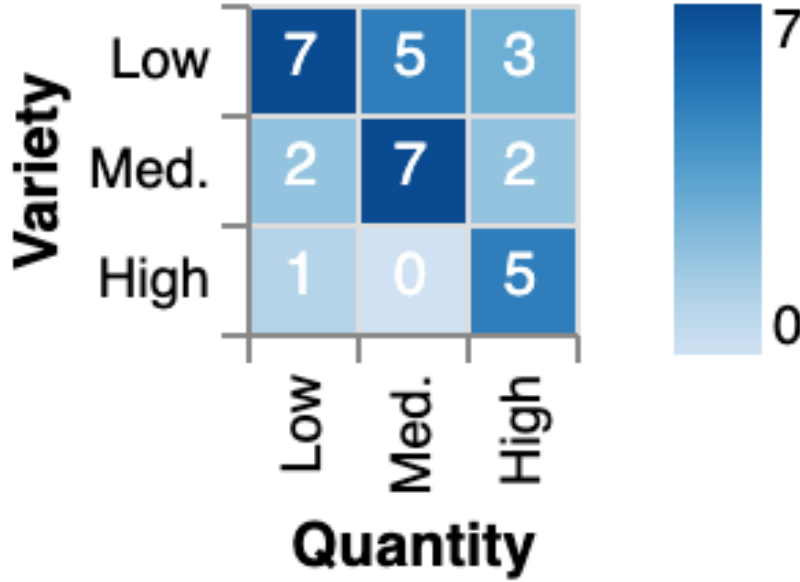


Figure 5.7: Participant distribution across quantity and variety of sketches.

5.5.2 Filtering Strategies by Sketch Quantity

A consistent **filtering** anchor yields more sketches. The Chi-Square test for homogeneity on the frequency distribution for all tag group states across the low (L_q), medium (M_q), and high (H_q) cohorts produced a statistically significant difference with a moderate effect size ($\chi^2 = 172.32, p < 0.01, df = 30, V = 0.35$). Results for states with statistically significant differences across all cohorts are shown in Table 5.3, while frequency distributions are shown in Figure 5.8.

Our qualitative evaluation of tag-group evolution reveals distinct filtering patterns across

Frequency Distribution per State by Sketch Cohort

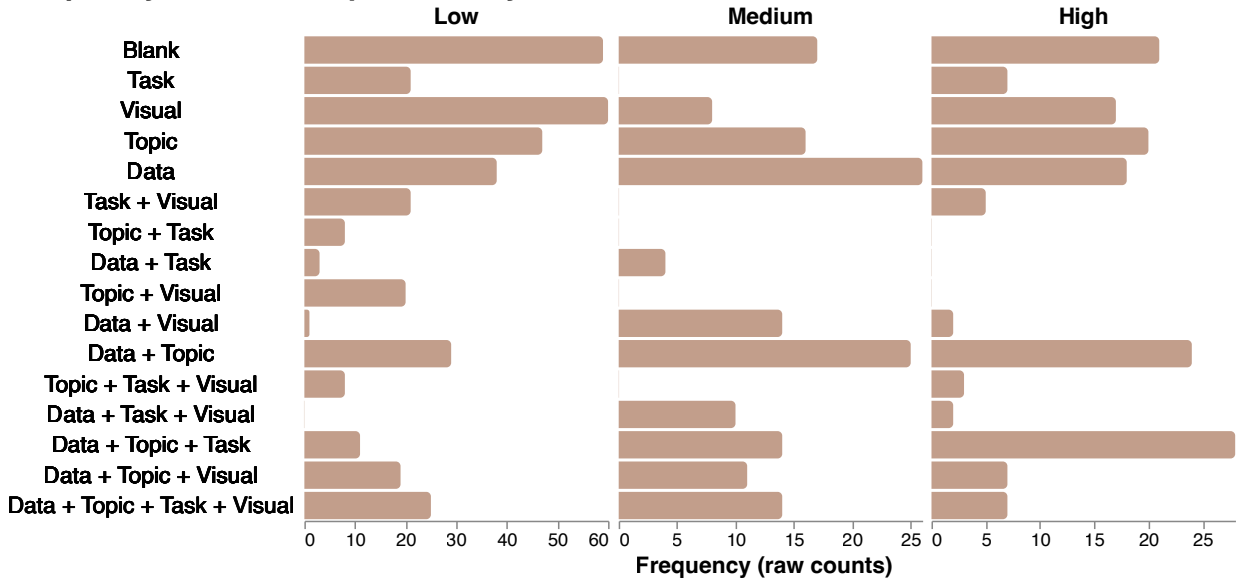


Figure 5.8: Frequency distribution for the quantity of sketches cohorts.

cohorts. In the L_q cohort, 73% (n=11) participants started their exploration with a **data**-anchored strategy. We observed some participants in the **data** anchored strategy switched their anchor to **visual** or **topic** anchors, oscillating between the blank state and these anchors for a minimum of 18 filtering actions. A similar pattern was also observed for the four participants in the L_q cohort who started with a **topic**-anchored strategy. In contrast, in the M_q and H_q cohorts showed less oscillation in their anchors. Notably, the H_q cohort contained two additional branches where

Table 5.3: Significant differences across quantity cohorts for tag states. z-scores are shown only for significant results (*).

State	Ordering	Z-scores
visual	$L_q^* > H_q > M_q^*$	$L_q(z = 2.2, p < .01)$ $M_q(z = -2.7, p < .01)$
data + visual	$M_q^* > H_q > L_q^*$	$M_q(z = 5.6, p < .01)$ $L_q(z = -2.6, p < .01)$
data + topic + task	$H_q^* > M_q > L_q^*$	$H_q(z = 4.5, p < .01)$ $M_q(z = -3.3, p < .01)$

participants remained in their initial anchored strategy and gradually expanded their filtering criteria to include all tag groups added. The complete overview of tag-group evolution for each cohort is included on OSF.

Overall, these results highlight a pattern of repeated switching between anchors for participants in the L_q cohort and demonstrate that *a consistent anchor with progressive expansion of the filtering space is potentially associated with higher sketch production.*

5.5.3 Ideation Strategies by Sketch Quantity

Table 5.4: Comparison of ideation strategies by quantity of sketches.

Characteristic	Low Quantity of Sketches	Medium Quantity of Sketches	High Quantity of Sketches
Primary Strategy	Filtering -dominated	Curating -focused	Inspecting -intensive
Filtering Actions	Significantly more frequent (19.2%; $z = 3.2, p < .01$) with the shortest duration (4.9s median)	Significantly less frequent (12.9%; $z = -2.1, p < .01$)	Moderate frequency (13.2%) but $1.7\times$ longer duration than Low (8.6s median)
Inspecting Actions	Significantly less frequent (16.1%; $z = -2.4, p < .01$)	Moderate frequency (18.3%) with $1.6\times$ longer duration than other cohorts (23.7s median vs. (L_q : 13.6s and H_q : 14.8s))	Significantly more frequent (25.3%, $z = 3.3, p < .01$)
Browsing Duration	Moderate (11.4s median)	Longest (12.2s median)	Shortest (8.2s median), at least 27% shorter than others
Dominant Transitions	Browsing $\leftrightarrow$ Filtering (37.7%)	More balanced between Browsing $\leftrightarrow$ Filtering (25.6%) and Browsing $\leftrightarrow$ Inspecting (34.1%)	Browsing $\leftrightarrow$ Inspecting (44.7%)
Secondary Transitions	Browsing $\leftrightarrow$ Inspecting (29.1%)	Browsing $\leftrightarrow$ Curating (22.9%), 30% more than other cohorts	Lowest Browsing $\leftrightarrow$ Filtering (24.1%)
Characteristic Sub-sequences	Extended filtering -browsing chains with occasional inspecting	Methodical curating cycles with thorough inspecting	Extensive browsing- inspecting chains; distinctive curating-inspecting transitions (4.3% vs 0 – 0.6% in other cohorts)
Sample Significant Subsequence	“ Filtering → Browsing→ Inspecting → Browsing→ Filtering → Browsing” (53.33% vs 9.09% in Medium)	“Browsing→ Curating → Browsing→ Inspecting → Browsing→ Inspecting → Browsing→ Curating → Browsing” (45.45% vs 6.67% in Low)	“Browsing→ Inspecting → Browsing→ Inspecting → Browsing→ Inspecting ” (83.33% vs 33.33% in Low)

Differences in **filtering**, **inspecting**, and browsing across cohorts The frequency distribution of actions across quantity cohorts (see Figure 5.9a) is as follows: $L_q(n = 660)$, $M_q(n =$

465), $H_q(n = 288)$. A Chi-Square test ($\chi^2 = 22.88, p < 0.01, df = 8, V = 0.09$) revealed significant differences in action frequency, with a small effect size. Adjusted standardized residuals highlight the strongest contrasts in **inspecting** and **filtering** behaviors (see Table 5.4).

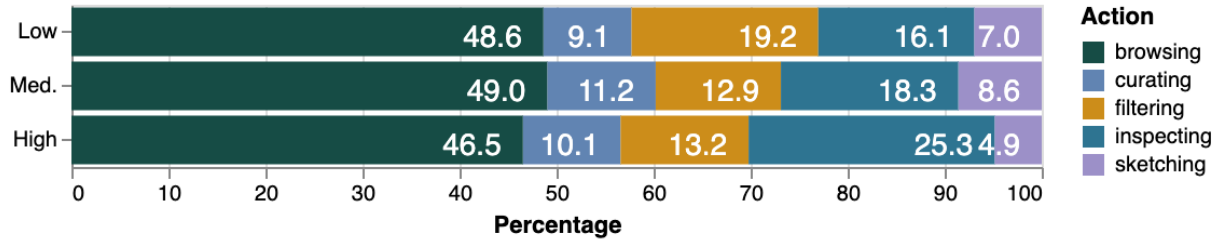
Kruskal-Wallis H test to compare ranked action durations across quantity cohorts, reveal significant differences in **inspecting** ($H = 11.7, p < 0.01$), **filtering** ($H = 7.2, p < 0.05$), and **browsing** ($H = 9.0, p < 0.05$). Post-hoc Dunn's tests with Holm correction identify specific cohort differences (see Table 5.5). Median action durations across cohorts are shown in Figure 5.9b.

Ideation strategies shifts from **filtering** to **curating** to **inspecting** as sketch production increases. As sketch production increases from low to high, participants demonstrate a clear evolution in their example-aided ideation strategies, as shown in Table 5.4. Participants who generate a low quantity of sketches predominantly employ a **filtering**-heavy approach characterized by frequent (Figure 5.9a) but brief **filtering** actions (Figure 5.9b), with **filtering** $\leftrightarrow$ **browsing** as their dominant transition (Figure 5.10a). Their significant subsequences reveal extended **filtering**-**browsing** chains interspersed with occasional **inspecting**.

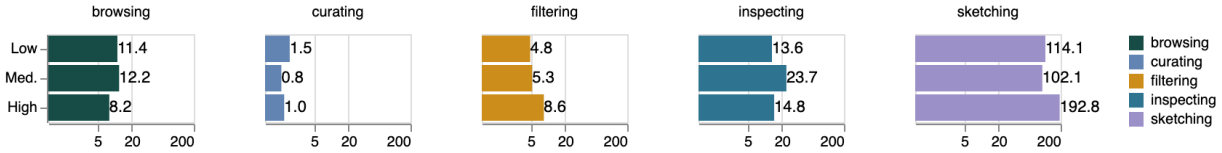
Participants who generate a medium quantity of sketches shift to a more **curating**-focused

Table 5.5: Statistically significant pairwise posthoc analyses results across quantity cohorts for durations.

Action	Ordering	Significant Pairs
browsing	$M_q > L_q > H_q$	$H_q - L_q (p < .05)$ $H_q - M_q (p < .05)$
filtering	$H_q > M_q > L_q$	$L_q - H_q (p < .05)$
inspecting	$M_q > H_q > L_q$	$M_q - L_q (p < .01)$ $M_q - H_q (p < .05)$



(a) Frequency of actions in percentage.



(b) Median duration of actions in seconds.

Figure 5.9: Frequency in percentage and duration of actions in seconds (logarithmic scale) for each sketch quantity cohort

methodology with longer *inspecting* times ($1.6\times$ longer than other cohorts), and significantly more *browsing-curating* transitions, as shown in Figure 5.9b. On the other hand, participants in the high quantity cohort adopt an *inspecting*-intensive strategy with the highest proportion of *inspecting* actions (Figure 5.9a), shorter *browsing* durations (Figure 5.9b), and distinctive *inspecting-curating* patterns that suggest more deliberate assessment of interesting examples.

These results suggest that participants who generated a lower number of sketches invest in recurrent refining of search criteria rather than examining examples in depth. On the other hand, more productive designers may employ more systematic approaches to example-aided ideation: less but deliberate query refinement and more thorough analysis of relevant examples. For detailed comparisons of action frequencies, durations, transition patterns, and characteristic subsequences across quantity cohorts, please refer to Table 5.4.

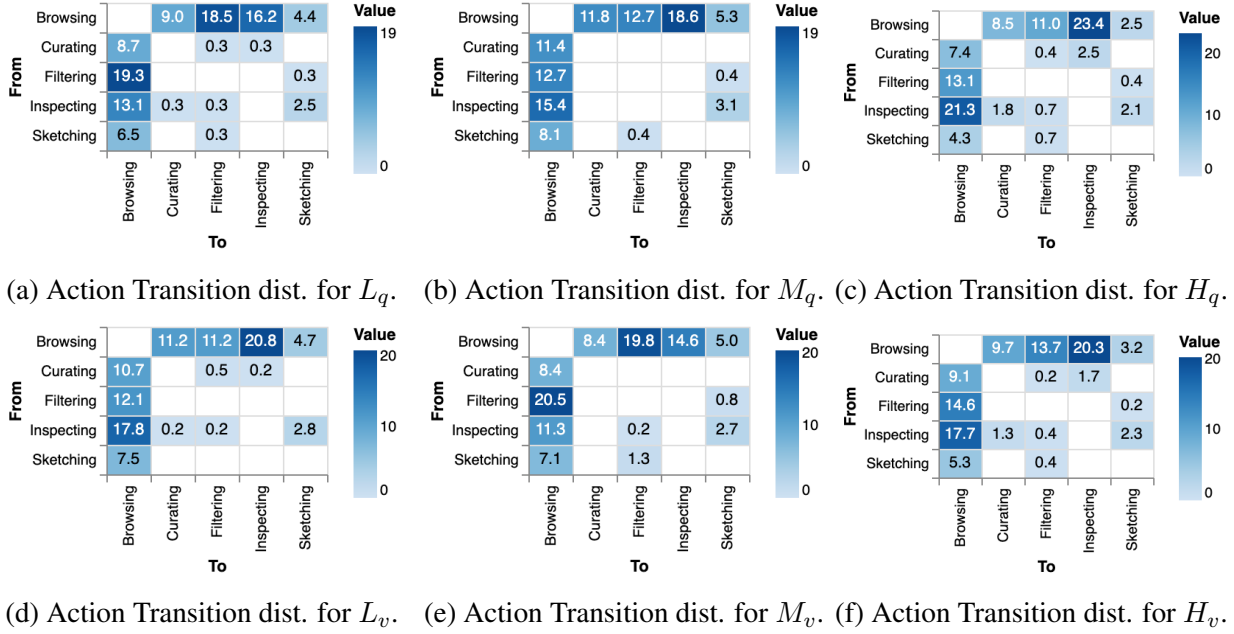


Figure 5.10: Action Transition distributions (dist.) for the quantity (row 1) and variety (row 2) cohorts.

5.6 RQ2: Variety of Sketches

The second part of our analysis for **RQ2** is to understand how the example filtering and ideation strategies relate to the variety of designs. The variety of design sketches provides a measure of the range of ideas produced by a participant. To assess variety, we calculated the number of unique elements (e.g., visual encodings, marks, compositions) included in the sketches produced by participants [16]. Overall, participants had an average design variety score of 9.1 ($sd = 3.1$).

We use the same grouping strategy described in subsection 5.5.1 to create low (L_v), medium (M_v), and high (H_v) cohorts for design variety. Figure 5.2 presents details for the variety cohort population, and Figure 5.7 shows the overlap in participants between the quantity and variety cohorts.

Table 5.6: Significant differences across all variety cohorts (z-scores shown only for significant results, marked with an asterisk).

State	Ordering	Z-scores
visual	$M_v^* > H_v > L_v^*$	$M_v(z = 2.97, p < .01)$ $L_v(z = -2.7, p < .01)$
data + topic + task	$H_v^* > L_v > M_v^*$	$H_v(z = 3.5, p < .01)$ $M_v(z = -4.1, p < .01)$

5.6.1 Filtering Strategies by Variety of Sketches

We also performed a Chi-Square test for homogeneity on the frequency of visits to each state to evaluate if the tag groups that participants utilized to filter examples differed based on the variety of produced designs. Our results show a statistically significant difference in the transition frequencies across participants with low, medium, and high variety ($\chi^2 = 223.36, p < 0.01, df = 30, V = 0.41$). We observed statistically significant differences in frequencies across all cohorts for the only two states shown in Table 5.6. The frequency distribution across all states is presented in Figure 5.11.

Our qualitative analysis of the tag-group evolution between the two cohorts reveals that participants in the L_v cohort typically remain in their initial chosen anchor (data, topic, or task) throughout the ideation process. Participants in the M_v cohort exhibit similar filtering strategies as participants in the low sketch quantity cohort who have a very narrow filtering pattern. On the other hand, participants in the high variety (H_v) cohort began with a data-anchored filtering strategy. Still, they switched anchors later in the ideation process, frequently oscillating between topic and a blank state before switching to a cyclical visual or topic anchored filtering strategy. These results suggest that a dynamic and evolving strategy that allows for *flexibility in filtering*

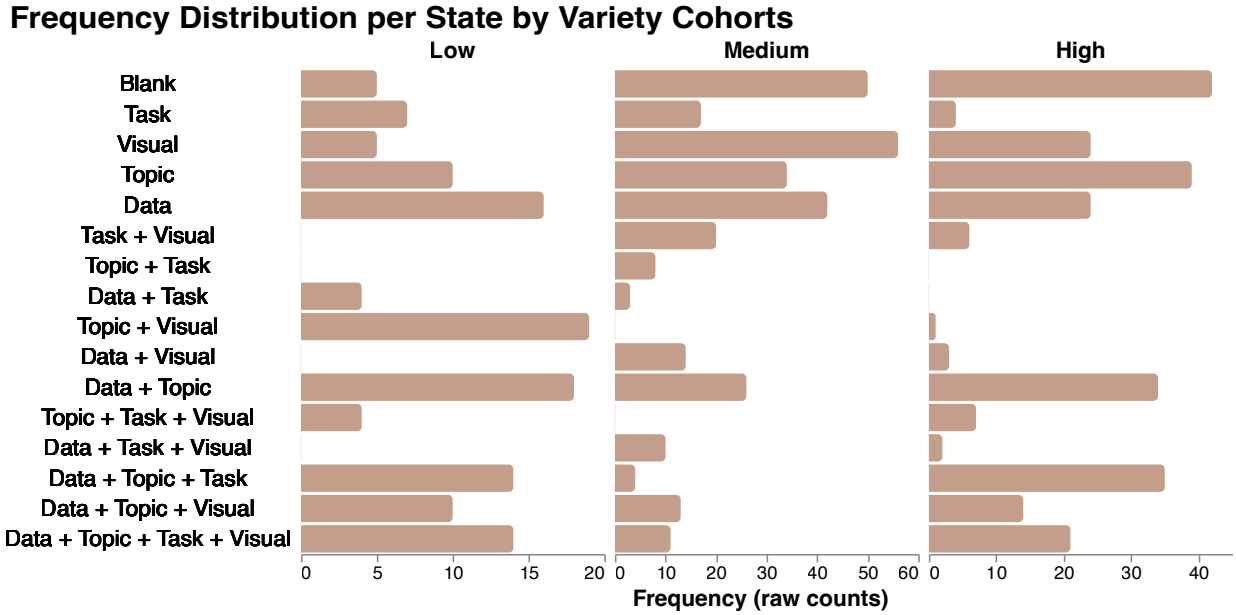


Figure 5.11: Frequency distribution for all 16 states across the low, medium, and high sketch variety cohorts.

strategies potentially leads to higher variety in design outcomes.

5.6.2 Ideation Strategies by Variety of Sketches

Similar to our analysis in section 5.4, we explored the example filtering and ideation strategies used by different participant cohorts based on the variety of sketches they produced.

Filtering, **inspecting**, **curating** and **browsing** drive sketch variety cohort differences. The action distribution across variety cohorts is as follows: $L_v(n = 438)$, $M_v(n = 491)$, $H_v(n = 484)$. A Chi-Square test ($\chi^2 = 25.64, p < 0.01, df = 8, V = 0.10$) revealed significant differences in action frequency, though with a small effect size. Adjusted standardized residuals show the strongest contrasts in **inspecting** and **filtering** behaviors, similar to quantity cohorts (Table 5.8).

The Kruskal-Wallis H tests reveal significant differences in two actions: **browsing** ($H = 7.18, p < 0.05$) and **curating** ($H = 7.39, p < 0.05$). Post-hoc Dunn's tests with Holm correction

identify specific cohort differences (Table 5.7). Figure 5.12b and Figure 5.12a visualizes median action durations and frequencies across cohorts, respectively.

Low and high design variety cohorts: shared inspecting-heavy ideation, different curating, filtering strategies Both L_v and H_v cohorts demonstrate similar inspecting frequencies (Figure 5.12a) and dominant browsing $\Leftrightarrow$ inspecting transition (Figure 5.10d, Figure 5.10f). However, they differ significantly in several key aspects. First, while both cohorts prioritize curating transitions over sketching, L_v participants have 66% longer median curating duration (Figure 5.12b). Second, L_v participants engage in filtering significantly less often, whereas H_v participants filter slightly more often. Third, H_v participants have 25% shorter median browsing duration than other cohorts (Figure 5.12b), and the most non-browsing transitions (Figure 5.10f), suggesting a more varied ideation strategy. Finally, L_v participants focus more on curating-browsing chains. On the other hand, H_v participants demonstrate extensive browsing-inspecting chains. Table 5.8 shows detailed comparisons of action frequencies, durations, transition patterns, and characteristic subsequences across variety cohorts.

These findings suggest that L_v participants demonstrate a curating-focused approach to ideation. But they were they were less effective at translating these curated examples into design elements. In contrast, H_v participants' indicated maintaining design variety through inspecting-heavy approach with rapid browsing.

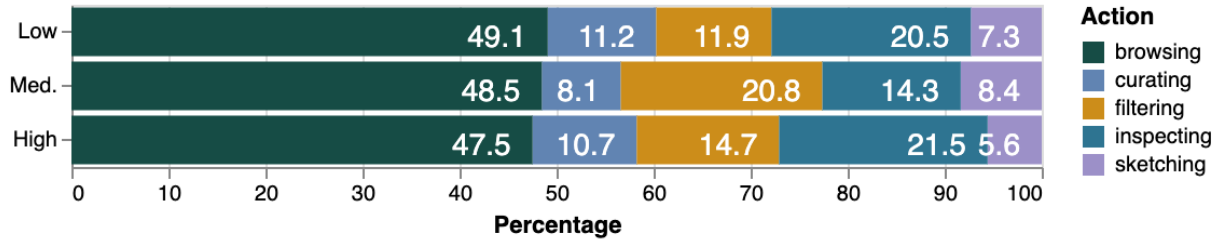
Table 5.7: Statistically significant pairwise posthoc analyses results across variety cohorts for action durations

Action	Ordering	significant pairs
browsing	$M_v > L_v > H_v$	$M_v - H_v (p < .05)$
curating	$L_v > H_v > M_v$	$L_v - H_v (p < .05)$

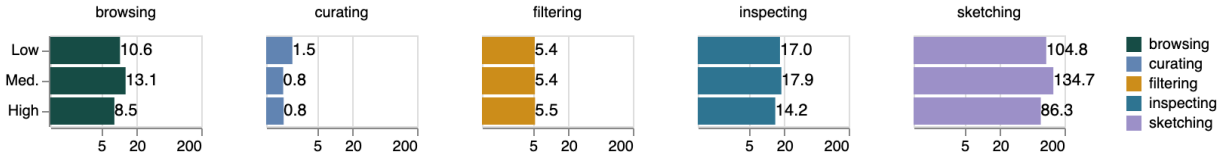
Table 5.8: Comparison of ideation strategies by variety of sketches

Characteristic	Low Variety of Sketches	Medium Variety of Sketches	High Variety of Sketches
Primary Strategy	Inspecting and Curating -dominated	Filtering -dominant	Inspecting -heavy
Filtering Actions	Significantly less frequent (11.9%; $z = -2.8, p < .01$)	Significantly more frequent (20.8%; $z = 3.6, p < .01$)	Moderate frequency (14.7%)
Inspecting Actions	Higher frequency (20.6%)	Significantly less frequent (14.3%; $z = -3.1, p < .01$)	Significantly high frequency (21.5%; $z = 2.0, p < .05$)
Browsing Duration	Moderate (median = 10.6s)	Longest (median = 13.1s)	Significantly shorter (median = 8.5s), 20% shorter than other cohorts
Curating Duration	Substantially longer (median = 1.5s), 66% longer than other cohorts	Shortest (median = 0.8s)	Moderate (median = 0.9s)
Dominant Transitions	Browsing ↔ Inspecting (38.6% of transitions)	Browsing ↔ Filtering (40.3% of transitions)	Browsing ↔ Inspecting (38.0% of transitions)
Secondary Transitions	Curating transitions exceed sketching by 1.5×	Balanced proportion between curating (16.7%) and sketching (16.9%)	Curating transitions nearly 2× sketching transitions
Non-browsing Transitions	Lowest (4.0% of transitions)	Medium (5.0% of transitions)	Highest (6.5% of transitions)
Characteristic Sub-sequences	Extensive curating -browsing chains	Extended filtering -browsing chains	Extensive browsing- inspecting chains
Sample Significant Sub-sequence	“Browsing → Curating → Browsing → Curating → Browsing → Inspecting ” (60% of Low vs. 10% of High and 8.33% of Medium)	“Browsing → Filtering → Browsing → Filtering → Browsing → Filtering ” (83.3% of Medium vs. 30% of Low)	“Browsing → Inspecting → Browsing → Inspecting → Browsing → Inspecting ” (80% of High vs. 33.3% of Medium)

Medium sketch variety cohort: **filtering**-focused ideation The M_v cohort demonstrates a **filtering**-dominant approach to example-aided ideation, setting it apart from both H_v and L_v cohorts. This cohort engages in **filtering** frequently while simultaneously showing the lowest frequency of **inspecting** actions among all groups (Figure 5.12a). Consequently, their interaction patterns are heavily dominated by browsing $\leftrightarrow$ **filtering** transitions (Figure 5.10e). M_v participants also exhibited extended **filtering**-browsing chains. These repetitive **filtering** chains often originated early in ideation sessions. This suggests that while active **filtering** helps participants explore more examples, the most varied design outcomes may require a deeper inspection of relevant examples.



(a) Frequency of actions in percentage.



(b) Median duration of actions in seconds.

Figure 5.12: Frequency and duration of actions in seconds (logarithmic scale) for each variety cohort.

5.7 Discussion

This study examines the strategies data visualization designers engage in during example-aided ideation and how these strategies potentially differ based on the properties of designs participants produced. The results of this work offer key insights into designers' ideation process. We discuss the implications of our findings for future work below.

Adapting timed introduction of examples to promote wider exploration and example internalization. The choice of when to introduce examples can alter how designers approach the exploration process. In the *start* condition, where examples are available from the beginning of the ideation process, designers tend to query a wide range of examples, engaging in more open-ended **filtering** and longer **browsing** actions. This indicates that the space of examples explored by these participants was much broader, but their interactions with the examples were more casual. Conversely, in the *delayed* condition, where designers have already begun **sketching** before seeing examples, their example **filtering** was more focused and anchored on dataset properties.

Their ideation process was heavily interspersed by *inspecting* actions. This highlights that a delayed introduction of examples leads designers to engage in a more structured ideation process with more emphasis on internalizing examples before deciding to use them. Visualization educators can consider timing the introduction of examples strategically in classroom activities to promote broader or focused engagement with examples.

Filtering strategies and their impact on design outcomes. Our analysis reveals that *filtering* strategies potentially influence both the quantity and variety of sketches. A higher number of sketches is linked to a stable, progressively expanding strategy, where participants begin with a consistent *data* anchor and gradually incorporate additional tag groups such as *topic*, *visual*, and *task*. In contrast, high design variety is associated with a dynamic *filtering* process characterized by frequent switching between different *filtering* anchors, *mirroring the deliberate jumps in example search space observed by Chan et. al. [45]*.

These findings suggest a potential tradeoff in *filtering* strategies that promote high design quantity and those that enhance high design variety. Notably, there are strong similarities between the *filtering* patterns across some cohorts. For instance, participants with low sketch quantity have similar *filtering* patterns as those with medium variety. Future work is needed to investigate if a tradeoff truly exists between strategies that promote high sketch quantity and variety, as well as to identify what *filtering* strategies balance design quantity and variety.

Developing adaptive design tools and galleries. Our results indicate that there are ideation strategies that are linked to both the quantity and variety of designs produced by participants. For instance, participants who produced a high (H_v) and low (L_v) variety of designs both had *inspecting*-heavy ideation strategies with dominant *browsing* $\leftrightarrow$ *inspecting* transitions. We also

see that as sketch production increases, participants' ideation strategies shift from being **filtering**-dominant (L_q) to more **curating**-focused (M_q) and **inspecting**-heavy (H_q). Furthermore, when comparing M_q and H_q , we found fewer statistically significant differences in their action patterns than in other pairwise comparisons.

These insights on the associations between ideation strategies and design outcomes can be utilized for developing adaptive design tools that are capable of real-time monitoring of designers' strategies as they explore examples or create designs. By detecting suboptimal patterns—such as the repetitive **browsing-curating** chain observed in low variety cohort—and providing corrective feedback or tailored recommendations, such tools could help steer designers toward more balanced approaches that promote both high sketch quantity and variety. Future work can investigate how to facilitate such real-time monitoring and detection of ideation strategies and what interventions may be suitable to steer designers off sub-optimal strategies.

These results also have implications for the design gallery creators. Design galleries are repositories that present examples of designs for libraries, most commonly in the form of tutorials [223]. Design galleries may also be configured as dynamic repositories that could empower users by adapting to their exploration strategies. For instance, future research can investigate how design galleries can adjust displayed examples based on inferred strategies from a designer's exploration.

Supporting diverse visualization retrieval. Our findings on designers' **filtering** strategies may offer valuable insights for design retrieval tools. We demonstrate that a dynamic, evolving **filtering** approach—characterized by a stable initial anchor that gradually expands to include additional tag groups—supports both high sketch quantity and variety. However, the current tools for

visualization retrieval focus on supporting the retrieval of similar visualizations [114, 186, 224], which are in conflict with the needs of visualization designers who may prefer diverse output. There is a need for future work to examine how we can develop tools that facilitate better exploration of visualization designs, especially those that emphasize semantically diverse designs [221] and those that are tool or language-agnostic.

5.7.1 Limitations

Our work focuses on understanding participants' example-aided ideation strategies during a controlled experiment using a faceted browsing interface for querying examples. While this setup facilitates our analysis goals for this chapter and our previous study [16], we acknowledge that it does not fully capture the real-world search process, which relies on search engines and includes additional modalities such as text queries. Furthermore, our study only focuses on the first ideation phase. As a result, the designs evaluated are not final polished solutions. Future work can examine how the results of these studies scale to other search modalities and multiple ideation sessions.

5.8 Summary

In this chapter, we deepen our understanding of the process of example-aided design by uncovering the mechanics of example-aided ideation strategies and how they support or constrain design outcomes. Through an analysis of the actions of 32 visualization designers during a data visualization ideation task, we assess how the timing of example exposure shapes example-aided design ideation strategies and how these strategies relate to the quantity and variety of sketches

designers produce during an ideation session. We find that early access to examples encourages broad, exploratory filtering and browsing, while delayed exposure fosters a more deliberate, data-focused filtering approach. Our analysis further reveals that consistent anchoring combined with flexible filtering and inspection practices is associated with both higher sketch production and increased design variety. The insights from our work collectively capture the dynamic nature of example-aided ideation, providing a methodical overview of how designers filter and navigate through examples, and completing our investigation into the process of example-aided design ideation.

Chapter 6: Mirny: An Example-Driven Prototyping Tool for Interactive D3 Visualizations

6.1 Introduction

Creating thoughtful and captivating data visualizations necessitates careful consideration of multiple design alternatives as designers program their visualizations, i.e., *visualization prototyping* [15, 28, 136, 209]. This chapter explores the interplay between *visualization programming*, i.e., creating interactive visualizations via languages, and *visualization design ideation*, in supporting the *visualization prototyping* process. With visualization programming, a user can create dozens of interactive visualizations by permuting a few core variables within a single program. We see these benefits in well-designed languages like D3 with thousands of visualization programs observed online [23, 24]. However, users must be knowledgeable of the underlying language, which can be a huge hurdle for individuals with limited or no programming experience [24, 183, 209], constraining them to only visualizations that they can practically implement.

Previous chapters have demonstrated the benefits of exploring and iterating over multiple design alternatives (in the form of examples) for visualization ideation outcomes. Yet, the process of evaluating multiple alternative designs can still be an arduous task as visualization designers struggle to anticipate how real data may impact abstract designs [209]. As such, visualization

designers often need to manually replicate and modify design examples themselves in order to properly evaluate design alternatives. Visualization programming and design ideation thus offer complementary yet distinct benefits to the visualization prototyping process. However, it is difficult to balance out the programming effort needed to explore and evaluate visualization design alternatives *and* accompanying interactions. Thus pointing to a critical question: How can we make it easier for users to prototype their interactive design ideas through complex visualization languages like D3?

In [chapter 4](#), we observed the replicate & modify strategy that designers use to adapt examples to their data and analytical contexts, suggesting a path to abstract visualization code examples into general-purpose programming templates [87], and generate code recommendations based on these pre-defined structures (e.g., Falx [211]). In prior work, programming templates are often treated as pre-defined *static* structures, where users can only modify fixed design parameters to create visualizations. However, users may be interested in changing multiple parts of a template such as aesthetics (e.g., color schemes), data encodings, interaction styles (e.g., slider versus brush filters), and mark types. Moreover, these parameters are chosen by the tool developers, not their users. Additionally, prior work finds that designers often dissect existing examples to extract new functionality they want to build on [15, 24]; for example, users could start working with a bar chart example but eventually transform it into a completely different visualization. By limiting how users can modify a template, we hamper these creative design activities.

In this chapter, we propose a *dynamic* approach to template-based design for interactive visualizations. Rather than restricting how templates can be modified, our goal is to identify the *design patterns* encapsulated within these templates and develop automated features to identify and extend these patterns as D3 users program in real time. These features must be *resilient*,

meaning they should work regardless of whether the user is populating template code or writing code from scratch. To achieve this, we leverage a suite of D3 templates derived in our prior work [17] to build two automated support features for D3: (1) a user-driven recommendation model that suggests suitable interaction[s] to add to a user’s visualization code and (2) a code augmentation feature that can integrate code snippets directly into live user code. In this way, we can help users *preserve their creative design thinking* as they code, and work towards closing the gap between visualization design and visualization programming. To demonstrate the utility of these features, we implemented them in Mirny, a design-focused environment for rapid prototyping of interactive visualizations in D3 (shown in [Figure 6.1](#)).

We evaluated these features through a study with 20 D3 users. We find that using the automated features implemented in Mirny, participants were able to prototype design ideas in less time and with fewer programming iterations compared to their typical D3 implementation workflows. We observed participants modifying templates in a variety of interesting ways, which we synthesize into three high-level modification strategies: changing the visual styles (e.g. mark colors or mark types), introducing data transformations (e.g., to aggregate or sort data), and removing or adding new data encodings.

In summary, this work makes the following contributions:

- We present an example-driven *model for recommending interactions* users can implement within a D3 prototype
- We develop a code augmentation feature to *automatically integrate interaction templates* derived from 2500 code examples into live D3 code.
- We implement these features in Mirny and evaluate Mirny through *a user study with 20 D3 users*. Our results show Mirny enabled participants to implement interactive D3 visualiza-

tions in less time and with fewer iterations.

- We synthesize *three high-level modification strategies* used by designers to customize visualization templates.

Using our findings, we suggest research opportunities to enhance template-based design approaches within visualization languages. All our data and code are shared on OSF ¹

6.2 Motivating Usage Scenario for Mirny

Sandra is a student who wants to create some interesting visualizations for the final project in her data science course. Although this course was broad in scope, it lacked depth in terms of learning interactive visualization techniques. Sandra has heard that D3 can support a wide variety of interactive visualizations. Although she has limited experience using D3, Sandra wants to create a D3 visualization, preferably with a simple dataset to start, such as the palmerpenguins dataset [80]. Sandra decides to use Mirny, shown in Figure 6.1, to program her visualizations. Sandra loads Mirny and imports her dataset, she chooses a Scatterplot from Mirny's Template Panel (Figure 6.1a). Mirny automatically generates the D3 code for a Scatterplot from the data as shown in (Figure 6.1b), and the resulting visualization is rendered in the output panel (Figure 6.1c). Mirny recommends a ranked list of common interactions for Scatterplots (Figure 6.1d).

Sandra sees nothing wrong with the scatterplot, but she wonders what the distribution of penguins is across the three islands recorded in the dataset. For her second prototype, she selects a bar chart from Mirny's Template Panel. Mirny automatically generates the D3 code for a bar chart a populates the Editor (Figure 6.1a), saving Sandra the effort of figuring out how to change her design to a different visualization type. Sandra edits the code to use the island attribute for

¹<https://osf.io/97mru/>

the x-axis and the count of penguins for the Y-axis. The overall visualization looks okay, but Sandra still isn't satisfied. She navigates back her Scatterplot prototype using the Template Panel and changes the color encoding from the species attribute to the island attribute.

Sandra likes the current visualization design but wants to add an interaction. Sandra clicks on the first item from the recommendation panel (*hover*) and Mirny automatically extends her current D3 program with the corresponding code, highlighting the new code block and providing a summary of its behavior in the Editor ([Figure 6.1e](#)). The new code displays additional information about a data point when a hover event is triggered. However, Sandra notices that some of the data points are really close to each other. She also adds a zoom interaction from the recommendation panel, so her viewers can zoom into tightly packed sections of the visualization. Once Sandra completes her visualization prototypes, she exports the code from the editor as a JS file and saves the visualization output in SVG format.

Using Mirny, a D3 user like Sandra can quickly prototype multiple interactive visualizations. Mirny eases the cumbersome process of programming D3 code, enabling users to explore visualization design alternatives using a single interactive interface. Although this example focuses on a new D3 user (Sandra), it can also easily apply to a more advanced user. For example, a professional analyst who has used D3 in several projects may find it tedious to edit an old example from a previous project, and may still prefer to fill in one of Mirny's templates instead. We discuss the experiences of both new and more advanced D3 users in [section 6.8](#).

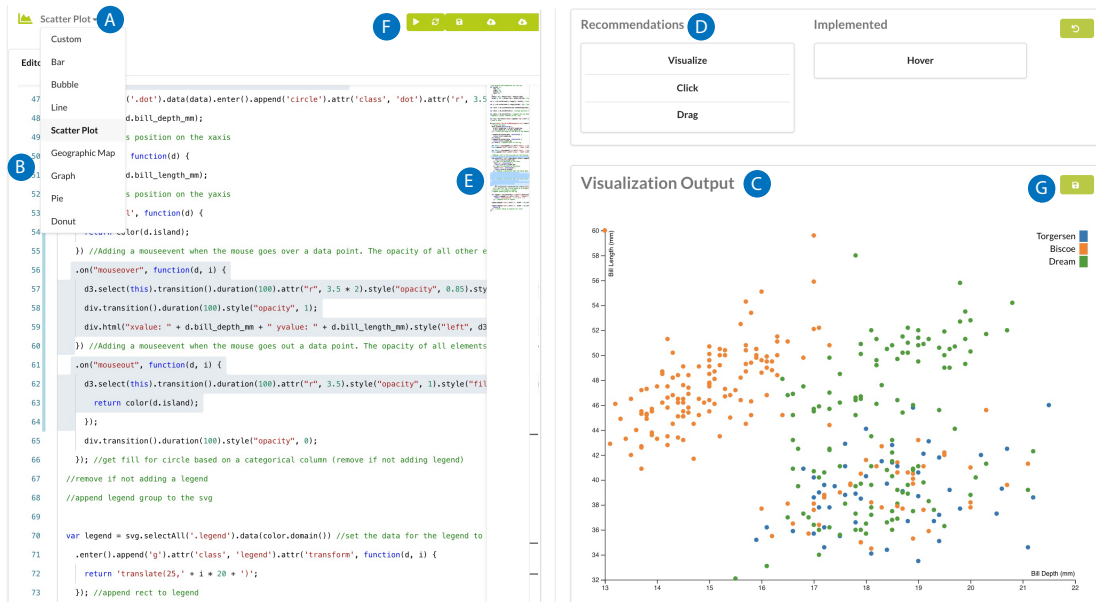
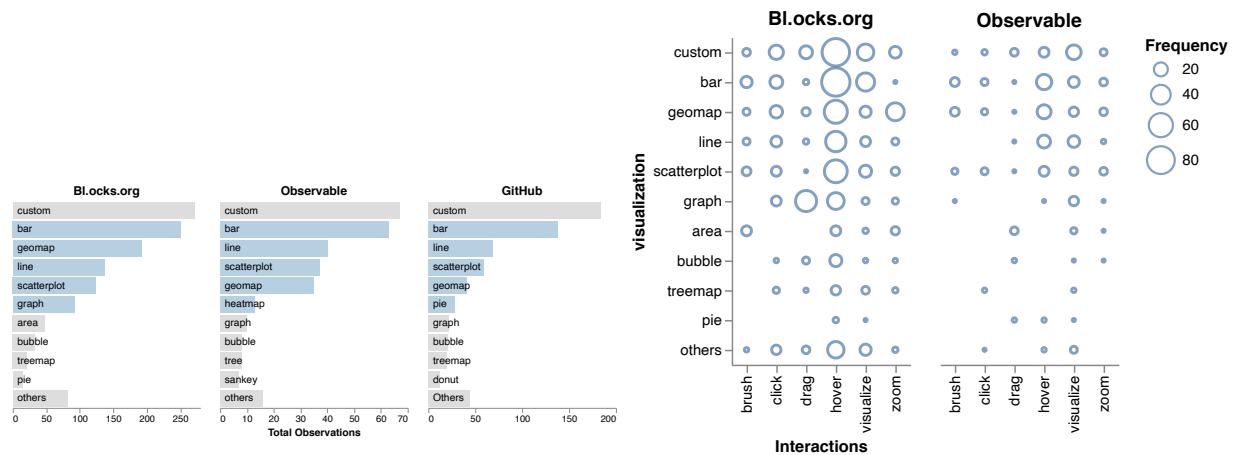


Figure 6.1: An example of an interactive visualization designed using Mirny. A user can select a visualization such as a *scatterplot* to create from the **Template Panel (A)**. The code responsible for creating the *scatterplot* is displayed within the **Editor (B)** and the corresponding visualization is rendered in the **Visualization Panel (C)**. A list of recommended interactions is displayed to the user in the **Recommendation Panel (D)**. A user can then choose a recommended interaction such as a *Hover* and the code responsible for implementing the interaction is added to the interface and highlighted for the user to view and modify (**E**). Once a user is satisfied with their visualization, they can export the code and visualization using the **Controls (F)** and (**G**).

6.3 Preliminary Study: Lessons from Analyzing D3 Examples Online

The goal of our work is to streamline the process of prototyping visualization designs with complex toolkits like D3. To do this, we first need to understand *what* kinds of visualizations D3 users create and *how* these users implement these visualizations in D3. In our prior work [17], we qualitatively analyzed 2500 D3 examples shared on GitHub [163], Bl.ocks.org [34] and Observable [35]. Through this analysis, we investigated common implementation patterns D3 users adopt when creating interactive visualizations. In this section, we summarize the key results from this analysis and describe how these results influence design considerations for the Mirny system.



(a) Rankings of observed visualizations in our analysis.

(b) Observed visualization-interaction pairings.

Figure 6.2: Results from the analysis carried out on the three example corpus [17] (a) presents the frequency of the top visualization types and their corresponding interactions observed in our datasets. Visualizations with 1% and less frequency are captured in the "others" category. (b) shows the number of times different interaction types were implemented for each visualization type in our corpus.

Takeaway 1: A Small Subset of Visualizations are Commonly Implemented in D3. To better understand which visualizations tend to be implemented in D3, we counted the incidence of visualization types across our corpus using Battle et al.'s D3 visualization taxonomy [17, 23]. Our observations reveal that 80% of the D3 visualizations shared online can be accounted for by just five visualization types (see Figure 6.2a): Bar charts, Geomaps, Line charts, Scatterplots, and Force Directed Graphs. Our findings suggest that *a small subset of visualizations cover the majority of D3 users' needs.*

Takeaway 2: D3 Implementation Patterns are Consistent Within Visualization Types. In a subsequent round of analysis, we divided our corpus by visualization type and analyzed the code for similarities and differences across examples, such as differences in the rendering strategy or differences in which interactions were implemented [17]. We found that in most cases, *the underlying structure of the D3 code examples did not vary.* In other words, the underlying structure and API calls remained the same within each visualization type. As a result, we observe clear patterns

in how different D3 users implement the same types of visualizations, with variations made to suit users' design preferences. We also found that D3 users copy code from existing examples, and even attribute the original sources in their visualization programs, corroborating observances of code reuse in the D3 community by prior work [24]. To make our findings actionable for the visualization community, we curated and shared templates for 8 popular D3 visualization types.

Takeaway 3: Interactions are Commonly Implemented in D3 and Vary by Visualization Type. Interactions are known to be challenging to implement in D3 [24, 184], and thus are a promising focus for design templates. To better understand how D3 users currently implement interactions, we classified observed interactions in our corpus using a subset of Brehmer and Munzer's multi-level typology of visualization tasks [41]. We observed 6 common interaction widget types: *Brush*, *Click*, *Drag*, *Hover*, *Visualize*, and *Zoom*. 52% of the examples we classified contained at least one or more combinations of interaction widgets (e.g *Hover*, *Click*, *Zoom*).

First, our findings show that interactions involving the selection of data points such as *Hover* are used in the vast majority of interactive examples (62%, see Figure 6.2b). Second, some interactions seem to be coupled with specific visualization types; for example, D3 network graphs often support the drag interactions. Finally, the total interactions implemented varied by visualization type. For instance, network graphs tended to have at least one interaction whereas Line charts tended to have no interactions implemented. This suggests that in D3, *interaction programming is dependent on the visualization design process, requiring support features that are both design-aware and code-aware.*

6.3.1 Design Considerations

Based on these results and our personal experiences using D3, we provide three design considerations for Mirny.

C1: Prioritize familiar interfaces/workflows. Past research highlights a need for tools to be integrated into users’ workflows [24,66] and familiar interfaces [75]. Furthermore, compatibility with other development platforms was one of the original design goals of D3 [36]. For these reasons, we prioritize compatibility with users’ existing workflows and preferred interfaces, reducing the burden of introducing yet another new tool to do their design prototyping process. Specifically, Mirny is designed to use either visualization templates *or* existing code as input and makes it easy to export code so it can be used in other tools such as Observable [35]. We summarize the Mirny interface design and explain how it supports visualization prototyping in [section 6.4](#).

C2: Design adaptive recommendations to match current practice. Our prior work reveals that with the widespread implementation of D3 visualizations comes myriad conventions for when and how to implement interactions for specific D3 visualization types [17]. To design effective interactions, D3 users have to navigate a complex space of existing design examples to identify compatible implementations. By providing automated recommendations for compatible interactions (inspired by existing recommenders, e.g., [81, 187, 205]), we can help users quickly prototype their envisioned designs, rather than forcing them to scour the internet for relevant examples. We present our recommendation engine to support interaction prototyping in [section 6.5](#).

C3: Support automated code augmentation. Prior work [24, 97] has established code reuse as a prominent implementation strategy employed by D3 users. In our past work, we developed

code templates for common data visualizations and interactions that D3 users implement [17]. However, the templates alone are not enough as users may have to modify or add to the templates to realize their target designs. To facilitate template modification and customization, users need automated integration of template[s] into existing user code, which we refer to as *code augmentation*. This way, a user can prototype multiple interactive visualization designs with a simple mouse click. We explain how we support automated code augmentation in [section 6.6](#).

6.4 Mirny System Overview

We implement two automated features in Mirny (recommendations and code augmentation) to facilitate interactive visualization prototyping in D3. These features are situated in a design-focused interface structured for integration with existing user workflows (addressing design considerations **C1** and **C2**). In this section, we describe Mirny’s overall design (see [Figure 6.3](#)).

The Mirny web interface provides a user with an interactive graphical user interface through which they can interact with the Mirny system. The web interface consists of four core views: **Templates Panel** ([Figure 6.1A](#)), **Editor** ([Figure 6.1B](#)), **Visualization Panel** ([Figure 6.1C](#)) and the **Recommendation Panel** ([Figure 6.1D](#)).

Templates Panel. The Templates Panel is designed to help D3 users quickly generate an initial visualization prototype for a given dataset. Revisiting our example in [section 6.2](#): after Sandra uploads her dataset, she selects a Scatterplot from Mirny’s list of templates. Mirny uses this template to identify attributes with compatible data types and generate an initial prototype visualization, which Sandra can view in the Editor (code) and Visualization Panel (output). Mirny

supports templates for the 8 most popular visualization types identified in our prior work [17]: Bar charts, Line charts, Scatterplots, Geographic Maps, Force Directed Graphs, Pie Charts, Donut charts, and Bubble plots. Mirny also leverages templates for 6 interaction types (Hover, Click, Brush, Drag, Zoom, and Visualize), which are displayed in the Recommendation Panel.

Editor. The Mirny Editor is a live JavaScript editor tailored for prototyping interactive D3 visualizations. Given our focus on integration with existing user workflows, the Editor is designed in a similar style to complementary editors, such as those in Vega-Lite [184] and Observable [35]. For example, when Sandra modifies the code for her Scatterplot prototype in Mirny, it is as if she is making the change with similar editors, smoothing Mirny’s learning curve.

Visualization Panel. Mirny displays the visual output for the current prototype in the Visualization Panel. This panel updates live and is fully interactive, so when a user modifies the code, they can quickly test the results in the Visualization Panel. For example, when Sandra adds a Hover interaction to her Scatterplot, she can immediately test the output using this panel.

Recommendation Panel. Parallel to the user’s coding process, Mirny also recommends compatible interaction designs that could be added to the current prototype. These recommendations are displayed in the Recommendation Panel, ranked by their relevance to the target visualization type. Mirny also shows which interactions have already been implemented, such as Sandra’s selected Hover interaction. The recommendations are powered by Mirny’s recommendation engine, described in [section 6.5](#). Users can click on any recommendation and it will automatically be integrated into their current prototype using Mirny’s code augmentation feature, described in [section 6.6](#).

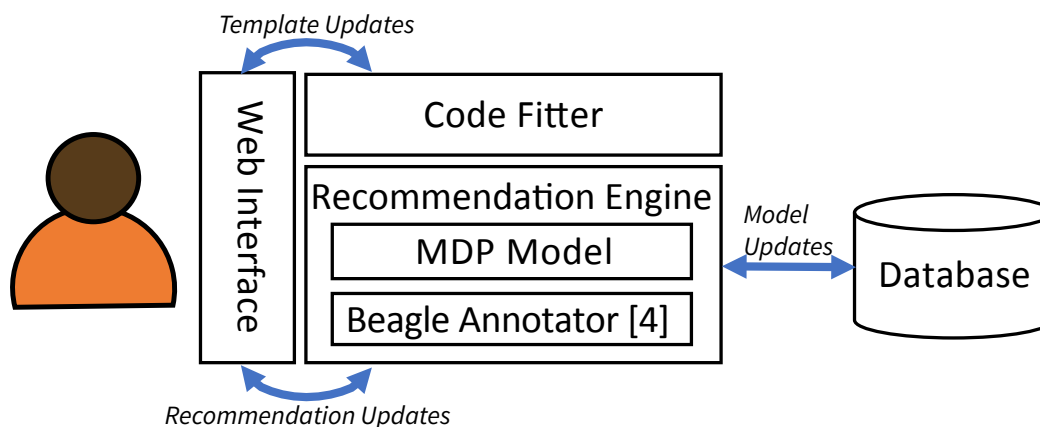


Figure 6.3: Mirny’s architecture. The Mirny interface captures user requests for interactions and recommendations. Using the Code Fitter, it generates the appropriate D3 specification and renders the results for the user via the Interface. Requests are also sent to the prediction model to receive recommended interactions for the user.

6.5 Recommending Code Snippets to Make D3 Visualizations Interactive

Given the prevalent combinations of visualization and interaction types observed in the wild (see [section 6.3](#)), D3 users are clearly interested in *interactive* visualizations. However, it may not be obvious to D3 users which interactions complement a given visualization. Revisiting our motivating example, Sandra may not know which interactions to implement to enhance her Scatterplot, such as Hover, Zoom, or Brush.

One approach to solving this problem is to *automatically recommend compatible interactions* anchored on the implementation practices of past D3 users, for example, by training machine learning models to predict complementary interactions for a given visualization (addressing design consideration **C2**). However, users’ programming behaviors and design preferences evolve as their interests shift over time [81, 205]. Thus, a one-shot approach to model training risks generating stale results as users advance their visualization projects. To keep up with the evolving nature of user preferences, our recommendations must also *adapt* to shifts in user be-

havior over time. Hence, we leverage Markov Decision Process (MDP) [26] to model users' interaction preferences and recommend complementary interactions to users as they program D3 visualizations. Here, we provide an overview of the prediction models that drive Mirny's recommendations.

6.5.1 Predicting Which Interactions to Implement

Modeling Interaction States. To make recommendations, we first need an understanding of the current state of a user's visualization code and use that information to predict what interactions to recommend to the user. We define the MDP for our recommendation engine as a set of *states* S , where each state $s \in S$ is a set of the interactions that have already been implemented in the user's code $\{i_1, i_2 \dots i_n\}$. Given that the user could pick any number of interactions to implement, we create a separate state for every possible combination of interactions that we observed from Blocks.org. Furthermore, this interaction set often starts out empty as D3 users seem to implement functional visualizations before incorporating interactions. For this reason, we make the empty state the initial state for our model.

For any given state $s \in S$, our MDP model produces a ranked list of valid interaction recommendations, ordered by how likely they are to be implemented based on observed frequencies of interaction types among our D3 examples (see [section 6.3](#)).

Adjusting the Model Based on User Feedback. Given our interaction states, we model potential interaction implementation steps through a set of *user actions* A , where each action $a \in A$ represents a user's reaction to a recommended interaction i_r . The user can react in one of four ways, given the user's current state s , and the corresponding future state s' :

- **Accept** the recommendation by clicking on the corresponding button in the Mirny interface (see [Figure 6.1D](#)), thereby moving to state s' , given by the equation:

$$P[s, i_r, s'] = \frac{observations(s')}{observations(s)}, \quad (6.1)$$

where *observations* computes the total observations of s (i.e., the combination of interactions) in our corpus.

- **Export** the visualization code files (see [Figure 6.1E](#)) and exiting Mirny. This is a cue that the user may be satisfied with her visualization and wants to save it for future use, calculated as:

$$P[s, i_r, export] = \frac{observations(s')}{\sum_{s \in S} observations(export|s)} \quad (6.2)$$

- **Undo** an added interaction, suggesting the user was dissatisfied with the interaction and wants to remove it:

$$P[s, i_r, undo] = \frac{observations(s')}{\sum_{s \in S} observations(undo|s)} \quad (6.3)$$

- **Ignore** all of Mirny's recommendations I_r by not clicking on any of them in the interface, remaining in the same state s :

$$P[s, i_r, s] = 1 - P[s, i_r, export] - P[s, I_r, s'] - P[s, i_r, undo] \quad (6.4)$$

Each time the user transitions to a different state, the probability distributions for outgoing transitions from s will be recalculated to account for this additional information. We normalize

all corresponding probabilities to ensure that they sum to one. Note that since the total interactions that a user will implement are relatively small, these distributions can be computed easily.

Predicting Transitions Between Interaction States. We model how likely the user is to transition from one state s to another s' as a transition probability between states. This is represented as a *transition function* T , which computes the probability that an interaction i recommended in state $s_i \in S$ leads to state $s_j \in S$, or $T(s_i|i_r, s_j)$. Given the user's current state $s \in S$, a recommended interaction i_r , and a corresponding future state $s' \in S$, we define our transition function probability as follows:

$$T_r(s, i_r, s') = P[s, i_r, s'] \quad (6.5)$$

Note that an interaction will only be recommended if it has been observed before. However, we derive our states and transition probabilities using our observations from [section 6.3](#), ensuring that our MDP model covers D3 interactions that users often implement.

When a user does or does not choose one of Mirny's recommendations, the MDP is rewarded or penalized accordingly with a positive or negative *reward* R . We assume that the reward R for a user's reaction to recommendation i_r to be: a small positive number if the user **accepts** i_r , a large positive value if the user **exports**, a large negative value if the user **undoes**, and 0 if the user **ignores** i_r . In this way, the MDP can adapt to match users' behavior.

6.5.2 MDP Example.

We demonstrate the behavior of the recommendation engine using a simplified example based on our motivating scenario with Sandra from [section 6.2](#). A simplified MDP model is depicted in [Figure 6.4](#). In this example, we assume that there are only two possible interac-

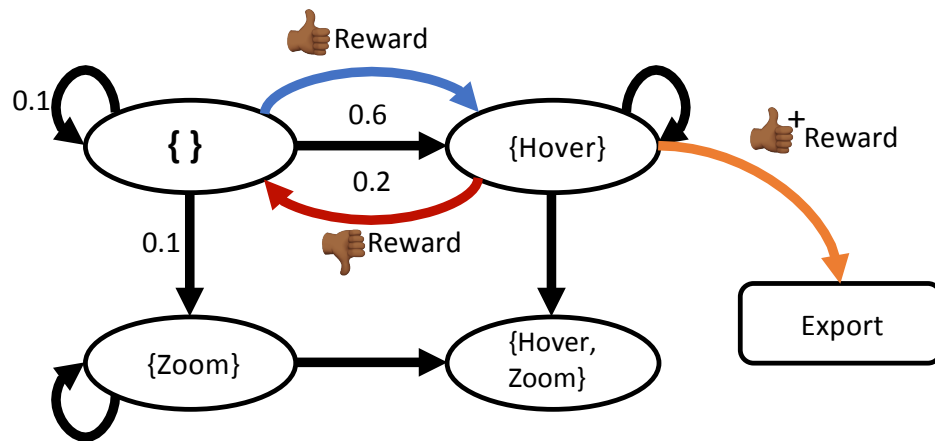


Figure 6.4: The Mirny recommendation engine models user interaction state using a Markov decision process (or MDP) model. Here is a simplified example of how the Mirny MDP model is structured with two interactions (zoom and hover), explained in [subsection 6.5.2](#).

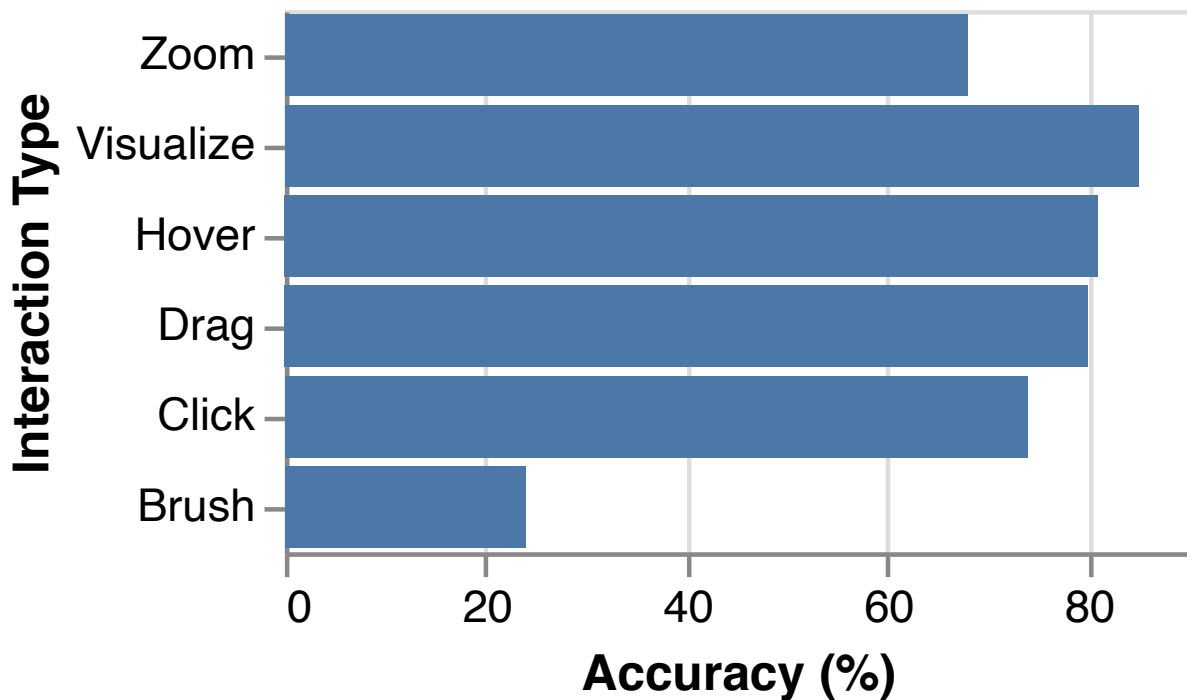


Figure 6.5: Results of the LOO cross-validation of the recommendation model per interaction type.

tions that Sandra will want to implement: zoom and hover. With two interactions, there will be four possible states for our model, shown in [Figure 6.4](#): no interactions implemented (the initial state), zoom has been implemented, hover has been implemented, and all interactions have been implemented (the terminal state).

Assuming Sandra starts in the initial empty state with her Scatterplot, we can calculate initial transition probabilities, where four (fictitious) transition probabilities are provided in [Figure 6.4](#). The intuition behind these probabilities is as follows: Sandra may iterate on her Scatterplot before incorporating any interactions (represented with probability 0.1), and Sandra is much more likely to implement hover than zoom interactions (probabilities 0.6 and 0.1, respectively). There is also a possibility that Sandra could implement a hover interaction but then change her mind and click on the undo button (represented with probability 0.2). For the empty state, our model provides a ranked list with two items, where the first item is hover, and the second item is zoom.

In our motivating example, Sandra clicks on the first recommendation of hover, representing the **accept** reaction. In this case, the MDP model receives a small positive reward for providing a useful recommendation, represented by the blue line in [Figure 6.4](#). However, suppose that Sandra **undos** her last added interaction, this results in a negative reward to the MDP model, represented by the red line in [Figure 6.4](#). If Sandra **exports** her visualization after adding the interaction, the MDP model receives a large positive reward for providing helpful recommendation[s] to Sandra.

6.5.3 Preliminary Model Accuracy Results

To ensure that our recommendations align with our observations from [section 6.3](#), we performed leave-one-out cross-validation [72] of the MDP model using our coded dataset. For each visualization example from our dataset, assuming the initial empty state, we classify a recommendation as correct only if they include all interactions implemented within that example. Overall, our model has an average accuracy of 76% (see [Figure 6.5](#)). For certain visualization types, this accuracy can be improved, e.g., for zooming interactions (68% accuracy). We find that our model performs poorly for brush interactions due to the relatively small number of data points for this interaction type in our training data (see [Figure 6.2](#)). Such cases can be improved by adding more input observations for the model to train on. However, this evaluation does not take into account how users' reactions to recommendations may improve the model's predictions over time. We discuss our evaluation with actual D3 users in [section 6.7](#).

6.5.4 Predicting Visualization types

We described how the recommendation engine supports interaction predictions when the user selects a visualization template first. However, our implementation in the Mirny system can also support recommendations even when the user does not select a template. To do this, Mirny utilizes the Beagle Annotator developed by Battle et al. [23] to enable real-time prediction of the visualization a user is currently implementing. With this annotator, Mirny can predict if a user is implementing a Scatterplot, or some other visualization type by extracting the SVG elements of the visualization and running it through the Beagle Annotator's decision tree classifiers.

6.6 Augmenting User Code

Recommending relevant interactions could help D3 users find and prototype interesting interactions for their visualization designs. However, these recommendations alone would not be very useful to visualization designers. Users would still have to search for example implementations and figure out how to integrate the code into their prototypes. A major goal of this work is *to reduce the challenge of integrating code from different sources when prototyping visualization designs*. We envision a user clicking just a single button to automatically add an interaction to her design specification (addressing design consideration **C3**).

The D3 API supports six interactions (i.e. *Hover, Click, Brush, Drag, Visualize and Zoom*) which have been identified and analyzed in [section 6.3](#). The underlying code base for these interactions has been manually extracted into pre-generated templates that Mirny uses to automatically generate code for selected interactions. Once a user selects a recommended interaction, we apply a series of code rewriting steps to merge the template into the user’s current code file. Specifically, we translate the user’s code file into its corresponding abstract syntax trees (or ASTs), analyze the trees for potential insertion points, and merge the interaction code template with the user’s code file at a selected insertion point. This process happens in four steps which we describe below.

Step 1: Identify Input Variables for the Interaction Template. Interactions often result in changes to the state of the visualization. This may include displaying additional information about the selected data point through a tooltip or making modifications to the state of the selected mark (or non-selected marks) such as changing its position on a drag event. To create meaningful code from interaction templates, specific inputs from the user’s code need to be extracted. How-

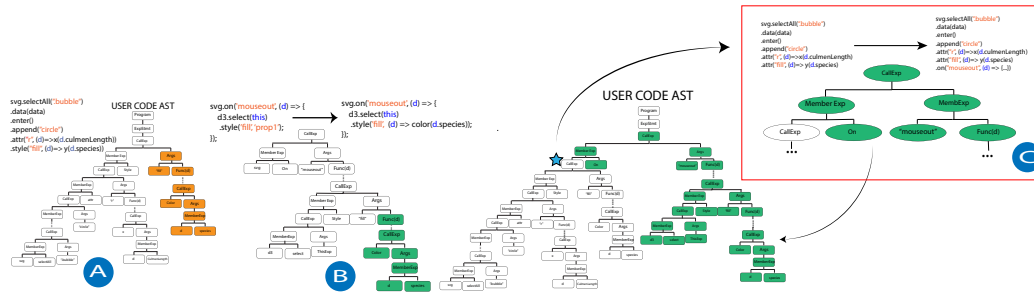


Figure 6.6: Given a recommended code snippet, e.g., a template for “Hover” interactions in D3, the Mirny system supports *automatic code augmentation*, where the user’s D3 code is mapped to an Abstract Syntax Tree (AST) that Mirny uses to populate and insert the code snippet. In (A), Mirny detects the “fill” variable in the user’s current code (highlighted in orange). In (B), Mirny uses the “fill” variable to automatically populate the “Hover” template (highlighted in green). Finally, in (C), Mirny inserts the populated “Hover” template into the user’s code at a detected anchor point in the AST (marked with a star).

ever, given the abstract nature of code, it is initially unclear which variables in the user’s code file map to the required inputs within a given interaction code template. Considering our running example from [section 6.2](#), if the hover interaction needs to update certain properties of selected circle marks in Sandra’s Scatterplot, then the code that sets properties for the circle marks, and the data attributes passed as inputs to the mark properties, need to be incorporated into the hover template.

Hence, the first step of the code augmentation process is to identify analogous variables between the user’s code and the interaction template. Since the properties needed for the interaction templates are known beforehand, we can identify the appropriate data attributes in the user’s code by parsing through its AST.

Using our motivating example, when Sandra selects the *Hover* interaction from the recommendation panel, our templates assume that at the end of the *Hover* event, properties like the color and radius of the circle mark would need to be restored to its original value. The Code Fitter parses through Sandra’s code to identify the input variables that are needed to make a viable

template. For instance, suppose Sandra’s code creates and sets the fill of the circle marks in a scatter plot using the following code:

```
svg.selectAll('.bubble')  
  .data(data)  
  .enter().append('circle')  
  ...  
  .attr('r', (d) => radius(d.culmenLength));  
  .style('fill', (d) => color(d.species));
```

The code fitter parses through the AST of her code to find where the circle marks’ `fill` attribute is assigned and extract the assigned value (or in this case function) (see [Figure 6.6a](#)).

Step 2: Populate the Interaction Template. Once all the required input variables have been identified, the interaction template is modified to include these input variables. The generic interaction template contains `prop` variables that are replaced with the appropriate input variables from the user’s code. For instance, our default `Hover` template changes all marks to a template-specified color. At the end of the hover interaction, this color will need to be reset. Thus all API calls are updated with the right variable names and parameters as needed. In our running example, the `mouseout` event of the hover interaction will be populated with the original function to determine the circle’s `fill` as seen in [Figure 6.6b](#). If an attribute is not found within the code (see description in step 1), a default value is assigned. However, designers have the discretion to change the default values our templates provide after the code augmentation process is completed.

Generic Template	Populated Template
<pre> .on('mouseout', (d) => { d3.select(this) .style("fill", "prop1"); }); </pre>	<pre> .on('mouseout', (d) => { d3.select(this) .style("fill", (d) => color(d.species)); }); </pre>

Step 3: Identify Anchor Points in the Users’ code for Insertion. In D3, the specification of interactions is *imperative* [184], hence, the interaction code cannot be placed just anywhere within an existing D3 specification. Before modifying the user’s code, we must first identify appropriate locations within the code to include new interactions. We refer to these locations as *anchor points*.

Anchor points signify nodes within the AST that need to be changed to support a new interaction. For example, to add the Hover interaction to Sandra’s code, a new node that maps to the interaction code has to be created and added to her code at the point where the visual marks are created. Because we know she is creating a Scatterplot, this would translate to the code block that contains the `.append(circle)` API call. ASTs are traversed using a depth-first approach, hence, identifying the anchor point simply involves iterating over the AST, with the objective of finding the node which has the appropriate data binding API call in its child branch. For our hover example, the anchor point would translate to the first node we identify that has the `.append(circle)` expression as a member of its child nodes as seen in [Figure 6.6c](#).

Anchor points for most visualizations generally can be found where the data binding to visual marks occurs. However, for complex interactions such as *Zoom* or *Brush* interactions helper functions and new visual elements need to be added to the user’s code, as such multiple anchor points are needed. Helper functions need to be accessed from anywhere in the D3 script

(i.e. have a global scope), hence their anchor point is by default the body of the entire script. Elements, such as `rects`, to enable zooming or brushing over sections of the visualization need to be added over the entire SVG, as such they use the node that binds the `<SVG>` element to the web page, as their anchor point.

Step 4: Insert the Populated Template into the User’s Code. Once a template has been populated and the anchor point has been found, the template code is then added as a new node in the user code at the selected anchor point. As a result, the populated template code becomes a new sub-tree within the AST of the user’s code. The template’s root node can either be *appended* or *pre-pended* to and in some cases *replace* the anchor point.

The nature of the template node and the semantic correctness of the code are taken into consideration when determining which of the insertion actions to perform. Code that introduces a new code block after the anchor point typically will use an *append* operation, whereas new code that goes before the anchor point will use a *pre-pend* operation. For instance, adding new code that defines a tooltip for the Hover interaction will be included before we call any hover event. As such, the code for the tooltip will be added before the anchor point using a *pre-pend* operation. The *replace* operation is usually used when adding new code as part of an already existing code sequence in the user’s code. This is due to the fact that ASTs are traditionally traversed depth-first, so the new code will have to become a parent node to the anchor point. In our motivating example for the *Hover* interaction, the code that handles the `mouseover` and `mouseout` events will need to be added to the entire anchor point’s branch. This will require a *replace* operation to make the template code’s AST a parent node of the anchor point as seen in [Figure 6.6c](#). Finally, Mirny converts the new AST back into code and updates the Editor to display the updated code file.

```
svg.selectAll('.bubble')  
  .data(data)  
  .enter().append('circle')  
  ...  
  .attr('r', (d) => radius(d.culmenLength));  
  .style('fill', (d) => color(d.species));  
  .on('mouseover', (d) => { ...});  
  .on('mouseout', (d) => { ...});
```

Limitations with Babel. The code augmentation feature helps users add interactions to their live D3 code in a single click. However, the code fitter uses the Babel compiler [198] to transform D3 code into its corresponding AST, as a result, there is a limitation to its operation. The Babel compiler requires that the code to be transformed is both semantically and syntactically correct. Hence, the code augmentation feature will not work if a user selects an interaction on incorrect D3 code. In these cases, Mirny will alert the user about the issues and not complete the code augmentation process.

6.7 Evaluation: User Study Design

In previous sections, we introduced two automated features with the aim of supporting template-based visualization prototyping. We implemented these features in a design-focused environment called Mirny. In this section, we focus on evaluating if these automated features lead to measurable improvements in how D3 users prototype interactive visualizations [173, 183]. Our study was preregistered on aspredicted.org².

Participants. We recruited 20 participants from mailing lists and social networking sites (4

²Pre-Registration available at <https://aspredicted.org/t2hb3.pdf>

female, 15 male, and 1 non-binary) between the ages of 18 to 44. To ensure that all participants had a sufficient understanding of D3 to complete the study tasks under time constraints, we required participants to have at least 3 months of experience working with D3.js. 11 participants had 3 months to 1 year of experience, 3 had 2 - 3 years of experience, 4 had 4 - 5 years of experience and 2 had 5+ years of experience. Participants self-report being proficient in creating visualizations. Each session took approximately 1 hour and 30 minutes and participants were compensated with a \$20 Amazon gift card.

Tasks. To observe how participants use these automated features in various visualization design contexts, participants were asked to perform two tasks for the study [173,217]. Participants were allowed to create whatever visualization or interaction type they chose as long as they met the requirements for each task.

- *Task One: Targeted visualization design:* For this task, participants were asked to use the Cherry Blossoms dataset to create a visualization meeting specific design requirements. The output visualization[s] had to meet two requirements: target users of the output visualization should be able to (a) identify correlations between two specific variables, and (b) interact with the data points in the visualization.
- *Task Two: Exploratory visualization design:* Participants were asked to choose between the popular Cars³ and Iris [11] datasets, and then brainstorm what type of visualization they would like to create. Then, participants were asked to program this visualization and include at least one interaction in their visualization.

Experiment Conditions. Participants experienced two different Mirny setups across our tasks,

³<https://www.kaggle.com/abineshkumark/carsdata>

resulting in four total experiment conditions ($2 \text{ tasks} \times 2 \text{ interface setups}$). In one setup (baseline), participants only had access to Mirny’s visualization templates, but not Mirny’s interaction recommendations. In the second setup, participants had access to all of Mirny’s features. Condition order was counterbalanced across the visualization tasks. We tuned our experiment conditions through a pilot study, described below.

Pilot. We conducted a pilot study to fine-tune our initial protocol. One important parameter we sought to tune was task difficulty, i.e., should participants be asked to create D3 visualizations from scratch as a task condition (the common case), or should participants be provided with Mirny’s templates? In the pilot study, all participants (2 female, 1 male) were familiarized with Mirny and were asked to perform the tasks described above. For task one, we provided access to Mirny’s visualization templates for two of the three pilot participants, while one participant created visualizations from scratch. Participants were able to use both the predefined templates and recommendations for task 2.

We observed that the participants who were provided with templates only used Mirny’s templates while the third participant copied code from a D3 example they found online. The participant who had to write their D3 code from scratch spent 45 minutes programming a single visualization and was unable to implement any interactions, thus unable to complete the task. The participants who were provided with templates took less than 15 minutes to create a single interactive visualization for the same task. These results show that it would have been difficult to conduct a one-hour experiment if participants had to create visualizations from scratch, and the experiment time could be drastically reduced if templates were always provided. Hence, we decided to provide templates for all tasks and experiment conditions. However, users could always opt to write their own code from scratch if desired.

Protocol. Participants were required to complete a demographic survey to determine eligibility for the study. Each session started with a Pre-study session where participants were familiarized with the study and signed the consent form. They then proceeded to a training session where the experimenter walked them through the interface functionality via a prerecorded demo of the interface. Participants were then given 5 minutes to explore the full Mirny interface themselves and ask any questions they had about the interface. Once participants were comfortable with the interface, they proceeded to complete our two study tasks. The experimenter limited guidance to only situations where the user reached a complete standstill and was unable to figure out the next step. Participants were allowed to use any external resource they wanted such as Stack Overflow or Blocks.org. No time limits were placed for any of the tasks. Once all the tasks were completed, the participants completed a short exit survey and interview.

Data Collection For each session, an experimenter observed the participants while they performed their tasks and took notes. Logs of participants' interactions with Mirny were also recorded through the Mirny back-end, which captured all click and keyboard events. We collected data from the post-task survey on participants' ratings of Mirny's features to capture participant feedback. Finally, video and audio were recorded to capture participants' interactions with the interface and responses to the exit interview questions.

6.8 Evaluation: Analysis & Results

Here we present three analyses of our study data: measuring session duration and design prototypes[s] ([subsection 6.8.1](#)), classifying template usage and modification ([subsection 6.8.2](#)), and summarizing participant feedback ([subsection 6.8.3](#)).

Number of Interactive Visualizations Produced

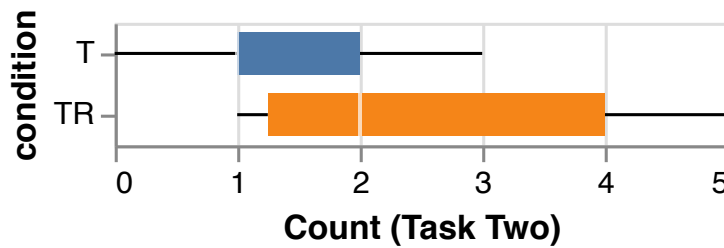
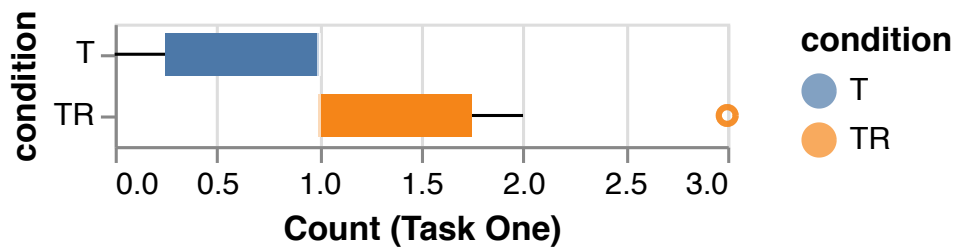


Figure 6.7: Count of interactive visualizations produced by participants.

Time taken to implement Interactions (Seconds)

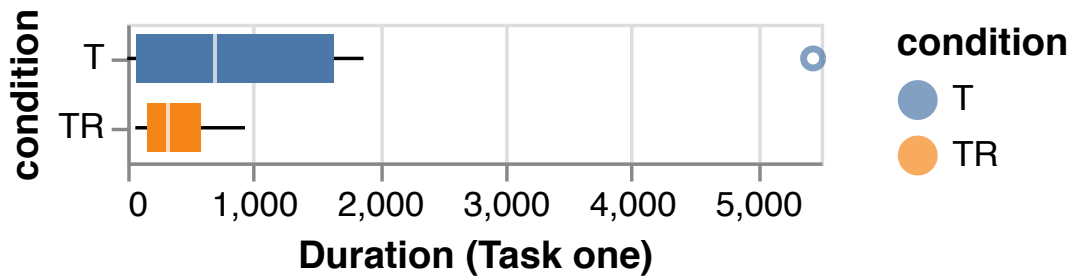


Figure 6.8: Distribution of time taken to program a single interaction.

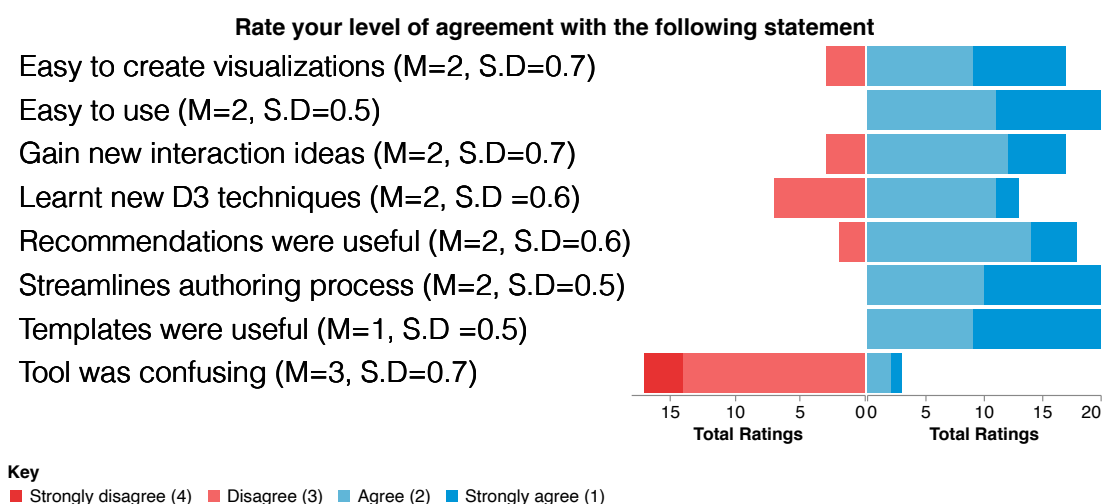


Figure 6.9: Distribution of post-study questionnaire responses.

6.8.1 Measuring Session Duration and Design Prototypes

We formed two hypotheses to test how Mirny’s code recommendation and augmentation support the visualization prototyping process:

- **H1:** Users produce a higher number of interactive visualizations provided with visualization templates and interaction recommendations, versus only visualization templates.
- **H2:** Users spend less time implementing interactive visualizations when provided with visualization templates and interaction recommendations, compared to when users only have access to visualization templates.

To test our hypotheses, we collected data on the number of visualizations created, the number of interactions participants implemented for each visualization (**H1**), and the time taken to implement a fully interactive visualization, i.e., time taken to render a single visualization + time taken to implement a single interaction (**H2**). We fit linear mixed effects models [21] with the condition and condition order as fixed effects and participant and dataset as random effects.

Based on the results of our pilots, templates were provided for all participants. As a result, we did not statistically test the number of visualizations explored by each participant since all participants used the templates. On average participants created 2 visualizations for each design task. We find a significant effect for interface layout on the number of interactions [**H1**: ($\chi^2(3, N = 20) = 13.406, p = 0.011$)] and time to implement an interaction [**H2**: ($\chi^2(1, N = 20) = 3.85, p = 0.0499$)]. Recommendations + code augmentation led to *1.9 more interactions implemented per visualization* compared to only 0.7 interactions when using only templates. Participants also *implemented interactions 3.2x faster* when recommendations and code augmentation were provided (5 minutes versus 16 minutes for templates only). Our survey results show that participants found the recommendations to be useful (Median(M)=2/Agree, $\sigma = 0.6$) and instrumental in helping them gain new ideas of interactions to implement (Median(M)=2/Agree, $\sigma = 0.6$).

6.8.2 Classifying Template Usage and Modification

To assess the efficacy of our automated features in helping users program visualizations, we formed two hypotheses:

- **H3**: Users produce a higher number of executable visualizations when provided with visualization templates compared to when users program visualizations from scratch.
- **H4**: Users find D3 more interesting and more useful when provided with both visualization templates and interaction recommendations, versus only visualization templates.

However, based on the results of our pilot study we needed to provide templates for all participants, affecting our analysis methods. While we do not quantitatively test **H3**, we have

informal support for **H4** based on our observations of user behavior and feedback from participants.

Participants default to templates to complete tasks. We expected that some participants would want to write their own code and provided an option to support this. Contrary to our expectations, *every participant in the study* defaulted to using Mirny’s templates, regardless of D3 expertise. Most participants settled on Scatterplots as their final design. We suspect that some participants may have been influenced by the word ‘correlation’ in task one which primed participants to choose scatterplots as their final design. Unfortunately, this behavior was unintentional and the fixation on scatterplots could have been mitigated by a more careful examination of the task description. Nevertheless, participants often explored other chart types such as line charts, bar charts, and pie charts before deciding what visualization to use.

Template Modification Strategies. Our findings amplify prior observations that templates are a great starting point for users but require modifications to fit users’ needs. We observed a total of *33 distinct code modifications* to Mirny’s visualization templates (n=22) and interaction templates (n=11). The majority of these modifications involved changes to accommodate *stylistic preferences* of participants (n=22), such as changes to the fills for marks and their associated color schemes, changes to the displayed layout of visual elements such as text, or changes to add new behavior to interactions. For example, P16 modified the code for a hover interaction to remove the default tooltip and instead change the opacity and size of other circles outside of the `species` class of the selected data point(s). P16 also modified the code to increase the size of all circles in the selected species class to make them more prominent.

We observed 6 occurrences of participants modifying Mirny’s visualization templates to *transform data* such as aggregating, sorting, or grouping data attributes. For instance, P15 modi-

fied the bar chart template to aggregate the number of cars produced each year and group them by country of origin. We also observed 5 modifications to *introduce new data encodings* or visual marks to the templates, which resulted in particularly interesting visualization redesigns. For example, P19 introduced an encoding mapping the `LastFrost` attribute to circle size in their Scatterplot implementation; more examples can be seen in [Figure 6.10](#).

Unexpected uses of templates. We observed an unexpected use case for templates by some participants while completing the design tasks. A small subset of our participants (n=4) used an initial template to *perform exploratory data analysis*, then chose a different template later on to present the results. For instance, when completing task two, P7 used the bar chart template to analyze the distributions of various data attributes. Then, they used the resulting insights to select attributes to present in their final Scatterplot.

6.8.3 Participant Feedback

Participants had varied reactions to the automated features provided in the Mirny system. In particular, participants often explained that the templates helped them integrate their design thinking into their programming process because Mirny provided starter code which helped save time finding suitable code. Participants mentioned that on their own, they would have to spend hours finding example code, which is known to be a major design challenge for D3 users [24] and even visualization designers in general [15]. For instance, P11 comments “*those [templates] are pretty valuable because there’s always that boilerplate of code you would have to write if you don’t have those templates.*” Similarly, P10 said “*I think it’s good that it’s smart enough to do this. And normally, doing these interactions is not easy. For example, I wouldn’t know how to*

implement brushing. I know, it's a very common thing. But I believe every time I want to do it, I have to search for it online." Participants mentioned that having access to the code for both the visualizations and interactions helped them to improve their understanding of the corresponding D3 designs.

The ability to quickly iterate over different interaction prototypes also allowed them to weigh their design options to understand and identify which designs were suitable for the task. This feature also helped participants to avoid the sunken cost fallacy that would have otherwise made them stick to a specific design only because of the time they had invested in creating their design. P20 states *"...instead of having to look for separate things myself, all these recommendations are in one place so it's very easy to try out which interaction is suitable for which kind of visualization. So I just click and then I can undo and then be like, okay, I probably don't want this. Whereas if I were to implement them myself, I would want to use something that I made, because I already spent time on it. So I want to use it, even though I probably shouldn't."*

More experienced participants explained that access to code may not be useful for them due to their existing knowledge of D3, but also agreed that an environment similar to what Mirny provides would be useful for novices who are just starting to learn D3. For example, P9 instructs a visualization class and mentions that teaching D3 is a difficult task as students typically need to learn HTML, CSS, JavaScript, and D3 all at the same time, hindering students' ability to prototype different visualization designs to understand their perceptual pros and cons. *"I also teach people d3. And I think this would be great...Having something like this, that creates the chart and the code and comments the code, so you can read the code. They [students] would like cry tears of joy to have this."*

Participants welcomed the automated features demonstrated in Mirny, however, our ap-

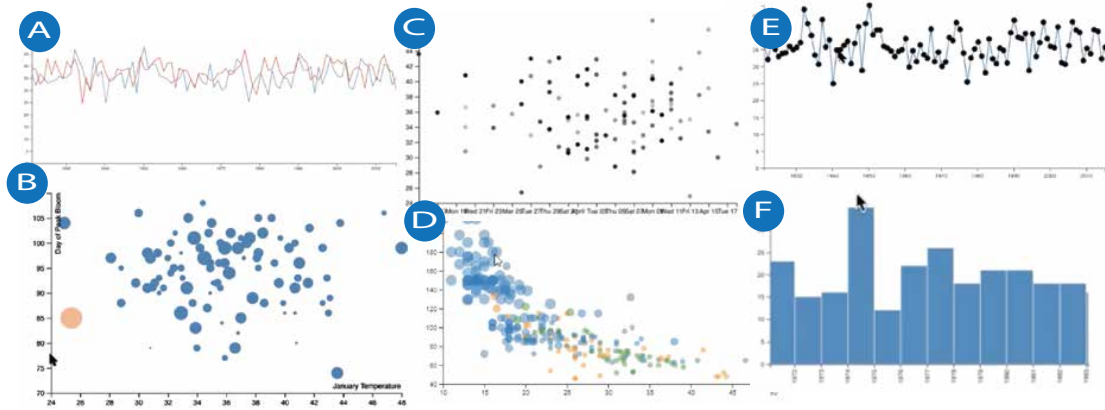


Figure 6.10: Examples of visualization designs created by participants during the experiment. (A), (B), (C), and (E) are visualizations of the Cherry Blossoms dataset depicting different attributes in the dataset such as yearly records of January Snow and February Snow (A), day of peak bloom vs January Temperature (B), etc. (D) and (F) are visualizations of attributes from the cars dataset.

proach is not perfect, and participants made suggestions for further refinement. Participants expressed the need for insight on why certain recommendations are made, e.g., P10 said *“instead of having the recommendation shift, just give more insight as to why this is being recommended, basically”*. Participants also suggested including more widgets to support further customization of D3 code, e.g., P15 said *“...maybe adding something like the components, to support selecting an axis or legend. If those are also given as part of templates to completely customize code, that might be more helpful”*. Participants also found the live updates feature provided in Mirny to be a hindrance when debugging their code. Participants suggested adding a run button to allow them to update the visualization output when desired.

6.9 Discussion

In this work, we present an approach to visualization prototyping that relies on template-based recommendation and augmentation. Our results demonstrate that our approach allows users to generate multiple visualization and interaction designs with significantly less effort and

time. Our evaluation highlights strategies that participants engage in to modify templates to fit their design goals. Participants’ feedback also shows that our approach saves users valuable time that they would have otherwise spent seeking design implementations; allowing them to invest their time exploring and evaluating design alternatives. We reflect on the implications of our findings and future directions for research on understanding visualization design, supporting user creativity and autonomy as well as advancing template-driven design.

Understanding visualization design reasoning. Visualization designers often start building their designs from an example or template [15], however, these templates are just a starting point for further design exploration. We find that when provided with visualization templates, participants would often make changes to implement their desired design ideas. We find that the key reasons why participants *augment our templates* are to: (1) accommodate their stylistic preferences, (2) perform data transformations, and (3) introduce new encoding[s] to the visualization. Modifying templates or examples is a common phenomenon amongst designers, yet the visualization community has hardly explored how designers build on and source ideas for their visual designs. In the future, we need to deeply investigate and understand the designer’s reasoning during the design process to understand how the context of use and design tasks influence the changes a user makes to design a template. Furthermore, more research is needed to understand when and how code augmentation fits into the visualization prototyping process. Finally, research also needs to explore the role datasets and stylistic preferences play in the visualization design process.

Balancing automation and user agency in visualization generation. The recent rise in program synthesis tools like GitHub’s Copilot has made it possible to automatically generate code [49]. However, as with all automated systems, users lose control over the code specifica-

tions that are generated as these tools often generate buggy or outdated code that a user neither understands nor knows how to fix [13, 61]; a limitation acknowledged in Copilot’s documentation [78]. While our approach relies on the same principles of automated code generation, its code generation process is built on manually curated code templates that incorporate domain-relevant expertise, safeguarding users from some limitations of automatic code generators. However, we acknowledge that the manual nature of Mirny’s template curation limits its scalability to more examples. Future work could explore ensemble strategies that incorporate the strengths of both approaches. For instance, one could integrate Mirny’s domain-specific data on D3 visualizations and its recommendation engine with Copilot to learn from a wider array of visualization and general programming examples to generate recommendations.

Visualization design is still a very subjective process that requires human input and creativity, evidenced by our participants’ desire to modify the generated visualization designs. Current visualization recommenders (e.g., Show Me [127], Voyager [217, 218], etc.), often treat system-generated designs as final output with little leeway (if any) for modifying the recommended designs, limiting user creativity. In the future, research needs to explore how automated visualization authoring tools can simultaneously preserve user agency and individual design preferences in the designs they generate or recommend [20]. A potential approach is to explore how users can be incorporated into the design generation process providing an opportunity for the user and recommenders to co-design data visualizations.

Modular visualization construction. Inspiration for visualization designs often comes from multiple sources and designers often piece together components from these sources into their designs (i.e., select & merge) [15, 228]. Participants like P15 in our study also expressed interest in templates that break visualization elements into modular code components to support

further customization. This request echoes the sentiment of other D3 users shared online [24]. Future research on the use of design templates should allow for dialog between users, templates, and authoring tools to improve the visualization programming experience. For instance, modularized templates for each part of a visualization (e.g axes, marks) could allow users to mix and match designs.

Furthermore, we can explore how AI can be used to advance support for no-code modular visualization construction for visualization designers. For instance, recent work has demonstrated how AI can be used to support the generation of UI widgets for visualization modification [204] or the blending of multiple visual references into a single design [53]. Similar affordances can be extended to support merging design components from multiple examples.

Supporting complex template transformations. We acknowledge that our choice to focus on only the common D3 visualization types and code integration for interactions is modest. However, this work is only a starting point in supporting dynamic template transformations for visualization design. We observed during our user study that users may be interested in changing a template from one visualization form to another. For instance changing a line chart to a connected scatter plot (see Figure 6.10b). An important direction for future work is to explore extending our automated features to support more complex template transformations such as changing the structure and arrangement of visual marks in response to data transformations. For instance, how can we dynamically transform a template that implements a bar chart to a stacked bar chart when a user performs data aggregation?

6.10 Summary

In this chapter, we present a new approach to supporting visualization prototyping by using code examples to power two template-driven automated features to help users program interactive D3 visualizations with less time and effort: (1) a *recommendation engine* for suggesting complementary interactions to add to a D3 visualization, and (2) *automatic code augmentation* to incorporate suggested code snippets to live user code, even when users are *replicating* code copied from external sources. We demonstrate these features in Mirny and in a user study with 20 D3 users, we find that they enable users to create interactive visualizations faster and in fewer iterations compared to the typical D3 development workflow (i.e., when automated features are not available). We also observe three key strategies users employ to customize visualization templates, as well as unexpected patterns of template reuse, underscoring how examples can scaffold rapid, creative authoring.

By embedding example-driven recommendations and live code augmentation into an authoring environment, this work directly addresses a component of this dissertation’s central question of how to *support* example-aided design. Mirny validates that tailored tooling can materially enhance the creative visualization process. In doing so, it closes the loop between qualitative insights into example usage, controlled measurements of their effects, and concrete tool support—laying a practical foundation for future co-design systems and guidelines.

Chapter 7: Conclusion

Examples play a critical role in the visualization design process, offering designers a way to understand problems, evaluate their design choices, and find inspiration for new ideas. Yet we have no formal evaluations of how visualization designers use examples and how factors unique to data visualization modulate the example-aided design process. As a result, we have a proliferation of tools with the wrong assumptions about designers’ workflows and inefficient support for adapting ideas from examples.

This dissertation addresses these issues through a systematic investigation into the process of example-aided design, answering the question: “*How can we understand, measure, and support example-aided design in data visualization authoring?*” To this end, this dissertation presents: (1) a qualitative interview study to characterize designers’ motivations and strategies ([chapter 3](#)) [15]; (2) a controlled experiment to quantify how example attributes and timing influence design outcomes ([chapter 4](#)) [16]; (3) an analysis of interaction sequences to reveal how ideation behaviors map to creative success ([chapter 5](#)) [19]; and (4) Mirny, a tool that operationalizes example adaptation strategies for interactive D3 visualization authoring ([chapter 6](#)) [17, 18]. Collectively, these contributions provide a holistic view of the example-aided design process. The interview study presented in [chapter 3](#) uncovered that designers rely on examples to both bridge gaps in understanding and to spark creativity. It also itemized the strategies designers use

to search for, curate, and adapt ideas gleaned from examples which complement existing findings in design literature [62, 101, 179, 188]. The controlled experiment in [chapter 4](#) demonstrated that data-schema similarity, visual complexity, and early exposure significantly impact the diversity, novelty, and conformity of designs. Furthermore, the analysis of participants’ behavior during example-aided ideation in [chapter 5](#) showed that flexible filtering and intensive example inspection correlate with richer ideation outcomes. The quantitative results in [chapter 4](#) and [chapter 5](#) contribute to ongoing debates in design by demonstrating the positive influence of delayed example introduction [144, 194] and uncover previously unrecognized factors and strategies such as data-schema similarity, visual composition, and filtering anchors, that shape example-aided visualization outcomes. Finally, Mirny validated that embedding these insights into an authoring environment can expand designers’ expressive capabilities.

7.1 Future Research Directions

This dissertation provides key insights into the process of example-aided design; however, it is only an initial exploration into this space. There is still a vast research agenda on developing computational support tools to enhance example-aided visualization design authoring. In this section, we highlight some of these interesting research directions.

7.1.1 Exploring the Dynamics of Example-Aided Creative Ideation

Creative ideation comprises iterative cycles of problem understanding, idea generation, and idea evaluation. Prior work has shown that examples can spark inspiration and scaffold design decisions [15, 16, 68, 93, 159, 195, 196], yet we still lack a nuanced understanding of when and

how different ideation phases benefit from example reuse versus when they suffer from fixation or cognitive overload. Building on the insights from Chapters 3–5, future research should probe the full cycles of creative ideation. For instance, future studies can be conducted to adopt fine-grained logging and think-aloud protocols to capture transitions between the different stages of creative ideation, mapping these behaviors to specific tool affordances (e.g., GUI controls, auto-suggestions) to reveal patterns of constructive versus inhibitive influence.

A second thread in this direction is to investigate the trade-offs that emerge during iterative cycles of creative ideation. As designers cycle between problem definition and solution exploration, they must balance divergent thinking (broad idea generation) against convergent thinking (narrowing toward a viable design) [108]. Computational interventions such as adaptive example filters, progressive complexity scaling, or context-aware prompts could be designed to encourage shifts between these modes at opportune moments. For instance, we can evaluate “ideation nudges” that detect when a designer is stuck in convergent thinking and then temporarily broaden the example corpus, or conversely, focus suggestions when the solution space is too broad. Controlled lab studies and field deployments will be crucial to calibrate these interventions, measuring their impact not only on creative output but also on designers’ subjective experience and flow.

Finally, to move beyond purely subjective judgments of creativity, we aim to develop and embed objective metrics for evaluating different dimensions of visualization designs into our tooling ecosystem. Inspired by computational creativity research, potential metrics include measures of structural novelty (e.g., deviation from prior examples in layout or encoding), semantic diversity (e.g., diversity in dataset–visual mapping), and exploration breadth (e.g., coverage of the design space over time). By integrating these metrics into live tools, we can give designers real-time feedback on their ideation trajectories, flagging when they are retracing familiar paths

or encouraging them to explore underutilized design regions. Embedding such evaluative signals directly into example-aided interfaces will enable a new class of “creativity-aware” visualization tools that not only scaffold idea generation but also help designers self-monitor and optimize their creative workflows.

7.1.2 Advancing Human–AI Co-Design for Visualization Authoring

In [chapter 6](#), this dissertation introduced the first iteration of design support tools for example-aided design that operationalizes one of the three example adaptation strategies identified in [chapter 3](#). Mirny demonstrates the feasibility of embedding example-aided design strategies into visualization design tools, opening up avenues of research on developing co-design environments that leverage artificial intelligence (AI) and the example-aided strategies we have uncovered to support both technical and non-technical users.

First, while tools like Mirny are suited for users with technical expertise who can program their own visualization designs, most domain experts have little to no technical skills. We therefore need interfaces that embed example reuse into higher-level workflows, allowing users to scaffold creative exploration without writing code. Secondly, there are still two additional strategies—*select & merge*, *trial & error*—that we need to accommodate in visualization authoring tools.

Conversely, we observe a phenomenon of fluid ideation, where designers oscillate between well-formed goals and open-ended sketching. This phenomenon highlights a critical challenge for the new AI design tools: interpreting shifting intent in real time. At one moment, a user may have a clear design intent and is focused on implementing that target idea. At the next, they

may be forming the idea as they iterate over their design. To address this, future research must investigate multimodal intent elicitation (e.g., conversational prompts and gestural commands, dynamic sketch recognition, and interaction-trace analysis) with the goal of determining how each modality can signal the user’s current ideation phase. By aligning AI suggestions (e.g., example recommendations, code snippets, or layout templates) with these subtle cues, co-design tools can stay in sync with evolving creative workflows.

Finally, we envision adaptive, personalized co-design systems that learn from individual designers’ behaviors and preferences over time. Building on our sequence-analysis findings in [chapter 5](#), such systems could model a user’s favored example-adaptation strategies (e.g., selective filtering vs. deep inspection) and proactively surface examples or interface controls that nudge them toward more divergent or convergent thinking as needed. Research questions include: 1) which interaction patterns best predict creative breakthroughs? 2) how can we balance exploration and focus through real-time interventions? and 3) can embedded creativity metrics guide AI to calibrate its support dynamically? Addressing these questions will pave the way for next-generation visualization tools that democratize expressive, example-driven authoring across skill levels.

7.1.3 Integrating Example, Domain, and Perceptual Knowledge in Visualization Design

During design, a designer potentially draws on three distinct knowledge sources: specialized domain conventions, design guidelines and heuristics, and implicit design cues gleaned from examples. Examples teach designers which visual encodings pair well with particular data at-

tributes; domain knowledge embeds conventions (for instance, spiral layouts for protein structures in biomedical engineering); and perceptual studies offer guidelines on element discriminability and compositional effectiveness. Yet these three axes of knowledge sources remain siloed both in visualization theory and tooling. A unified framework that weaves together these strands would enable tools to recommend not only “what works” in the abstract but “what works here” for a given dataset, domain context, and perceptual objective.

To construct such a framework, future work should formalize mappings among the three axes. On the example axis, we can mine large visualization corpora to learn associations between data schemas and successful encodings. On the domain axis, we must codify discipline-specific conventions, drawing on domain literature and expert interviews to capture conventional designs like glyph-based gene expression maps or geographic choropleths in epidemiology. On the perceptual axis, we need to formalize the results from graphical perception studies into empirical thresholds or weights that can be objectively measured and evaluated. Bringing these three axes into a common representational ontology will allow us to query for visualizations that: 1) match the data, 2) respect domain conventions, and 3) conform to perceptual best practices.

Once these knowledge sources are linked, we can translate the framework into co-design tools. For example, a recommendation engine could surface analogical examples from adjacent domains—showing a spiral layout from genomics to suggest an alternative pattern for time-series weather data—while simultaneously filtering for perceptual clarity and domain appropriateness. By embedding this tri-axial intelligence in authoring environments, we empower designers to discover novel visualization designs and support analogical inspiration systematically. Ultimately, this integrated framework will not only bridge theoretical gaps but also inform more creative, context-aware workflows for both technical and non-technical visualization designers.

Bibliography

- [1] Information is beautiful. <https://informationisbeautiful.net/>. Accessed: March 28, 2024.
- [2] Our world in data. <https://ourworldindata.org/>. Accessed: March 28, 2024.
- [3] Storytelling with data. <https://www.storytellingwithdata.com/>. Accessed: March 28, 2024.
- [4] Tableau public: Your data has a story. share it with the world. <https://public.tableau.com/en-us/s/>. Accessed: 2022-03-21.
- [5] Adobe. Behance. <https://www.behance.net/>, 2005. Accessed: 2022-03-22.
- [6] M. Agogu  , A. Kazak  i, A. Hatchuel, P. Le Masson, B. Weil, N. Poirel, and M. Cassotti. The impact of type of examples on originality: Explaining fixation and stimulation effects. *The Journal of Creative Behavior*, 48(1):1–12, 2014.
- [7] S. Alspaugh, N. Zokaei, A. Liu, C. Jin, and M. A. Hearst. Futzing and moseying: Interviews with professional data analysts on exploration practices. *IEEE transactions on visualization and computer graphics*, 25(1):22–31, 2018.
- [8] T. M. Amabile. The social psychology of creativity: A componential conceptualization. *Journal of personality and social psychology*, 45(2):357, 1983.
- [9] R. Amar, J. Eagan, and J. Stasko. Low-level components of analytic activity in information visualization. In *Proceedings of the Proceedings of the 2005 IEEE Symposium on Information Visualization*, INFOVIS ’05, page 15, USA, 2005. IEEE Computer Society.
- [10] S. Amershi and M. R. Morris. Cosearch: a system for co-located collaborative web search. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 1647–1656, 2008.
- [11] E. Andersen. The irises of the gasp e peninsula. *Bulletin of the American Iris Society*, 59:2–5, 1935.
- [12] S. G. Archambault, J. Helouvry, B. Strohl, and G. Williams. Data visualization as a communication tool. *Library Hi Tech News*, 2015.

- [13] O. Asare, M. Nagappan, and N. Asokan. Is github’s copilot as bad as humans at introducing vulnerabilities in code?, 2022.
- [14] B. Bach, Z. Wang, M. Farinella, D. Murray-Rust, and N. Henry Riche. Design patterns for data comics. In *Proceedings of the 2018 chi conference on human factors in computing systems*, pages 1–12, 2018.
- [15] H. K. Bako, X. Liu, L. Battle, and Z. Liu. Understanding how designers find and use data visualization examples. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):1048–1058, 2023.
- [16] H. K. Bako, X. Liu, G. Ko, H. Song, L. Battle, and Z. Liu. Unveiling how examples shape visualization design outcomes. *IEEE Transactions on Visualization and Computer Graphics*, 31(1):1137–1147, 2025.
- [17] H. K. Bako, A. Varma, A. Faboro, M. Haider, F. Nerrise, B. Kenah, and L. Battle. Streamlining visualization authoring in d3 through user-driven templates. In *2022 IEEE Visualization and Visual Analytics (VIS)*, pages 16–20, 2022.
- [18] H. K. Bako, A. Varma, A. Faboro, M. Haider, F. Nerrise, B. Kenah, J. P. Dickerson, and L. Battle. User-driven support for visualization prototyping in d3. In *Proceedings of the 28th International Conference on Intelligent User Interfaces, IUI ’23*, page 958–972, New York, NY, USA, 2023. Association for Computing Machinery.
- [19] H. K. Bako, K. T. Zinat, S. N. Sakhamuri, and Z. Liu. Patterns of inspiration: Analyzing ideation strategies in example-aided visualization design. In *Under Review*, 2025.
- [20] C. S. Bao, S. Li, S. G. Flores, M. Correll, and L. Battle. Recommendations for visualization recommendations: Exploring preferences and priorities in public health. *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. ACM, 2022.
- [21] D. J. Barr, R. Levy, C. Scheepers, and H. J. Tily. Random effects structure for confirmatory hypothesis testing: Keep it maximal. *Journal of memory and language*, 68(3):255–278, 2013.
- [22] J. Baruah and P. B. Paulus. Category assignment and relatedness in the group ideation process. *Journal of Experimental Social Psychology*, 47(6):1070–1077, 2011.
- [23] L. Battle, P. Duan, Z. Miranda, D. Mukusheva, R. Chang, and M. Stonebraker. Beagle: Automated extraction and interpretation of visualizations from the web. In *Proceedings of the 2018 CHI conference on human factors in computing systems*, pages 1–8, 2018.
- [24] L. Battle, D. Feng, and K. Webber. Exploring d3 implementation challenges on stack overflow. In *2022 IEEE Visualization and Visual Analytics (VIS)*, pages 1–5. IEEE, 2022.
- [25] W. A. Baughman and M. D. Mumford. Process-analytic models of creative capacities: Operations influencing the combination-and-reorganization process. *Creativity Research Journal*, 8(1):37–62, 1995.

- [26] R. Bellman. A markovian decision process. *Indiana Univ. Math. J.*, 6:679–684, 1957.
- [27] J. Berlin and A. Motro. Database schema matching using machine learning with feature selection. In *Advanced Information Systems Engineering: 14th International Conference, CAiSE 2002 Toronto, Canada, May 27–31, 2002 Proceedings 14*. Springer, 2002.
- [28] A. Bigelow, S. Drucker, D. Fisher, and M. Meyer. Reflections on how designers design with data. In *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces, AVI ’14*, page 17–24, New York, NY, USA, 2014. Association for Computing Machinery.
- [29] M. Bilalić, P. McLeod, and F. Gobet. Inflexibility of experts—reality or myth? quantifying the einstellung effect in chess masters. *Cognitive psychology*, 56(2):73–102, 2008.
- [30] M. Bilalić, P. McLeod, and F. Gobet. Why good thoughts block better ones: The mechanism of the pernicious einstellung (set) effect. *Cognition*, 108(3):652–661, 2008.
- [31] N. Bonnardel. Creativity in design activities: The role of analogies in a constrained cognitive environment. In *Proceedings of the 3rd conference on Creativity & cognition*, pages 158–165, 1999.
- [32] N. Bonnardel and E. Marmèche. Evocation processes by novice and expert designers: Towards stimulating analogical thinking. *Creativity and Innovation Management*, 13(3):176–186, 2004.
- [33] N. Bonnardel and E. Marmèche. Favoursing creativity in design projects. *Studying Designers*, 5:23–36, 2005.
- [34] M. Bostock. Popular blocks, 2016 (accessed November, 2019).
- [35] M. Bostock and M. Meckfessel. Observable, 2016 (accessed July, 2021).
- [36] M. Bostock, V. Ogievetsky, and J. Heer. D³ data-driven documents. *IEEE transactions on visualization and computer graphics*, 17(12):2301–2309, 2011.
- [37] J. Brandt, M. Dontcheva, M. Weskamp, and S. R. Klemmer. Example-centric programming: integrating web search into the development environment. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI ’10*, page 513–522, New York, NY, USA, 2010. Association for Computing Machinery.
- [38] J. Brandt, P. J. Guo, J. Lewenstein, M. Dontcheva, and S. R. Klemmer. Two studies of opportunistic programming: interleaving web foraging, learning, and writing code. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI ’09*, page 1589–1598, New York, NY, USA, 2009. Association for Computing Machinery.
- [39] M. Brehmer and R. Kosara. From jam session to recital: Synchronous communication and collaboration around data in organizations. *arXiv preprint arXiv:2107.09042*, 2021.
- [40] M. Brehmer, R. Kosara, and C. Hull. Generative design inspiration for glyphs with diatoms. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):389–399, 2022.

- [41] M. Brehmer and T. Munzner. A multi-level typology of abstract visualization tasks. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2376–2385, 2013.
- [42] A. Burns, C. Xiong, S. Franconeri, A. Cairo, and N. Mahyar. How to evaluate data visualizations across different levels of understanding. In *2020 IEEE Workshop on Evaluation and Beyond - Methodological Approaches to Visualization (BELIV)*, pages 19–28, 2020.
- [43] C. Cardoso and P. Badke-Schaub. The influence of different pictorial representations during idea generation. *The Journal of Creative Behavior*, 45(2):130–146, 2011.
- [44] S. Carpendale. Evaluating information visualizations. In *Information visualization: Human-centered issues and perspectives*, pages 19–45. Springer, 2008.
- [45] J. Chan, Z. Ding, E. Kamrah, and M. Fuge. Formulating or fixating: Effects of examples on problem solving vary as a function of example presentation interface design. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, CHI ’24, New York, NY, USA, 2024. Association for Computing Machinery.
- [46] J. Chan, S. P. Dow, and C. D. Schunn. Do the best design ideas (really) come from conceptually distant sources of inspiration? *Engineering a Better Future: Interplay between Engineering, Social Sciences, and Innovation*, pages 111–139, 2018.
- [47] J. Chan, K. Fu, C. Schunn, J. Cagan, K. Wood, and K. Kotovsky. On the benefits and pitfalls of analogies for innovative design: Ideation performance based on analogical distance, commonness, and modality of examples. 2011.
- [48] J. Chan, P. Siangliulue, D. Qori McDonald, R. Liu, R. Moradinezhad, S. Aman, E. T. Solovey, K. Z. Gajos, and S. P. Dow. Semantically far inspirations considered harmful? accounting for cognitive states in collaborative ideation. In *Proceedings of the 2017 ACM SIGCHI Conference on Creativity and Cognition*, C&C ’17, page 93–105, New York, NY, USA, 2017. Association for Computing Machinery.
- [49] M. Chen, J. Tworek, H. Jun, Q. Yuan, H. P. d. O. Pinto, J. Kaplan, H. Edwards, Y. Burda, N. Joseph, G. Brockman, A. Ray, R. Puri, G. Krueger, M. Petrov, H. Khlaaf, G. Sastri, P. Mishkin, B. Chan, S. Gray, N. Ryder, M. Pavlov, A. Power, L. Kaiser, M. Bavarian, C. Winter, P. Tillet, F. P. Such, D. Cummings, M. Plappert, F. Chantzis, E. Barnes, A. Herbert-Voss, W. H. Guss, A. Nichol, A. Paino, N. Tezak, J. Tang, I. Babuschkin, S. Balaji, S. Jain, W. Saunders, C. Hesse, A. N. Carr, J. Leike, J. Achiam, V. Misra, E. Morikawa, A. Radford, M. Knight, M. Brundage, M. Murati, K. Mayer, P. Welinder, B. McGrew, D. Amodei, S. McCandlish, I. Sutskever, and W. Zaremba. Evaluating large language models trained on code, 2021.
- [50] Q. Chen, X. Yue, X. Plantaz, Y. Chen, C. Shi, T.-C. Pong, and H. Qu. Viseq: Visual analytics of learning sequence in massive open online courses. *IEEE Transactions on Visualization and Computer Graphics*, 26(3):1622–1636, 2020.
- [51] X. Chen, W. Zeng, Y. Lin, H. M. Al-manee, J. Roberts, and R. Chang. Composition and configuration patterns in multiple-view visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):1514–1524, 2021.

- [52] Y. Chen, Y. Xiao, K. T. Zinat, N. Yamashita, G. Gao, and Z. Liu. Comparing native and non-native english speakers' behaviors in collaborative writing through visual analytics. *arXiv preprint arXiv:2502.18681*, 2025.
- [53] L. B. Chilton, S. Petridis, and M. Agrawala. Visiblends: A flexible workflow for visual blends. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, page 1–14, New York, NY, USA, 2019. Association for Computing Machinery.
- [54] I. Chiu and L. H. Shu. Investigating effects of oppositely related semantic stimuli on design concept creativity. *Journal of Engineering Design*, 23(4):271–296, 2012.
- [55] E. K. Choe, B. Lee, and m. schraefel. Characterizing visualization insights from quantified selfers' personal data presentations. *IEEE Computer Graphics and Applications*, 35(4):28–37, 2015.
- [56] E. G. Chrysikou and R. W. Weisberg. Following the wrong footsteps: fixation effects of pictorial examples in a design problem-solving task. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 31(5):1134, 2005.
- [57] N. Crilly. Fixation and creativity in concept development: The attitudes and practices of expert designers. *Design studies*, 38:54–91, 2015.
- [58] N. Crilly and C. Cardoso. Where next for research on fixation, inspiration and creativity in design? *Design Studies*, 50:1–38, 2017.
- [59] W. Cui, J. Wang, H. Huang, Y. Wang, C.-Y. Lin, H. Zhang, and D. Zhang. A Mixed-Initiative Approach to Reusing Infographic Charts. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):173–183, Jan. 2022. Conference Name: IEEE Transactions on Visualization and Computer Graphics.
- [60] W. Cui, X. Zhang, Y. Wang, H. Huang, B. Chen, L. Fang, H. Zhang, J.-G. Lou, and D. Zhang. Text-to-viz: Automatic generation of infographics from proportion-related natural language statements. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):906–916, 2020.
- [61] A. M. Dakhel, V. Majdinasab, A. Nikanjam, F. Khomh, M. C. Desmarais, Z. Ming, and Jiang. Github copilot ai pair programmer: Asset or liability?, 2022.
- [62] P. Dalsgaard, M. M. Biskjaer, and J. Frich. Capturing and revisiting ideas in the design process: A longitudinal technology probe study. *Design Studies*, 88:101200, 2023.
- [63] S. Dasgupta, W. Hale, A. Monroy-Hernández, and B. M. Hill. Remixing as a pathway to computational thinking. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*, CSCW '16, page 1438–1449, New York, NY, USA, 2016. Association for Computing Machinery.

- [64] V. Dibia and Ç. Demiralp. Data2vis: Automatic generation of data visualizations using sequence-to-sequence recurrent neural networks. *IEEE computer graphics and applications*, 39(5):33–46, 2019.
- [65] S. P. Dow, A. Glassco, J. Kass, M. Schwarz, D. L. Schwartz, and S. R. Klemmer. Parallel prototyping leads to better design results, more divergence, and increased self-efficacy. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 17(4):1–24, 2010.
- [66] 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*, page 1–12. Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems. ACM, 2020.
- [67] C. Eckert and M. Stacey. Fortune favours only the prepared mind: Why sources of inspiration are essential for continuing creativity. *Creativity and innovation management*, 7(1):9–16, 1998.
- [68] C. Eckert and M. Stacey. Sources of inspiration: a language of design. *Design Studies*, 21(5), 2000.
- [69] J.-D. Fekete. The infovis toolkit. In *IEEE Symposium on Information Visualization*, pages 167–174, 2004.
- [70] 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*, 25(1):501–511, 2019.
- [71] N. C. for Environmental Information. National time series: Climate at a glance. <https://www.ncei.noaa.gov/access/monitoring/climate-at-a-glance/national/time-series>, 2022. Accessed: Dec 12, 2022.
- [72] J. Friedman, T. Hastie, and R. Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.
- [73] K. Fu, J. Cagan, and K. Kotovsky. Design Team Convergence: The Influence of Example Solution Quality. *Journal of Mechanical Design*, 132(11):111005, 11 2010.
- [74] K. Fu, J. Chan, J. Cagan, K. Kotovsky, C. Schunn, and K. Wood. The meaning of “near” and “far”: the impact of structuring design databases and the effect of distance of analogy on design output. *Journal of Mechanical Design*, 135(2):021007, 2013.
- [75] J. Gale, M. Seiden, and Ç. Demiralp. Sigma worksheet: Interactive construction of olap queries. *arXiv preprint arXiv:2012.00697*, 2020.
- [76] L. Gao, X. Ma, J. Lin, and J. Callan. Precise zero-shot dense retrieval without relevance labels, 2022.
- [77] D. Gentner and A. B. Markman. Structural alignment in comparison: No difference without similarity. *Psychological science*, 5(3):152–158, 1994.

- [78] GitHub. About github copilot.
- [79] M. Gonçalves, C. Cardoso, and P. Badke-Schaub. What inspires designers? preferences on inspirational approaches during idea generation. *Design studies*, 35(1):29–53, 2014.
- [80] K. B. Gorman, T. D. Williams, and W. R. Fraser. Ecological sexual dimorphism and environmental variability within a community of antarctic penguins (genus *pygoscelis*). *PloS one*, 9(3):e90081, 2014.
- [81] D. Gotz and Z. Wen. Behavior-driven visualization recommendation. In *Proceedings of the 14th international conference on Intelligent user interfaces*, pages 315–324, 2009.
- [82] D. Gotz and M. X. Zhou. Characterizing users’ visual analytic activity for insight provenance. In *2008 IEEE Symposium on Visual Analytics Science and Technology*, pages 123–130, 2008.
- [83] M. C. Goulden, E. Gronda, Y. Yang, Z. Zhang, J. Tao, C. Wang, X. Duan, G. A. Ambrose, K. Abbott, and P. Miller. Ccvis: Visual analytics of student online learning behaviors using course clickstream data. *Electronic Imaging*, 31:1–12, 2019.
- [84] S. Gulwani. Programming by examples: applications, algorithms, and ambiguity resolution. In *Proceedings of the 19th International Symposium on Principles and Practice of Declarative Programming, PPDP ’17*, page 2, New York, NY, USA, 2017. Association for Computing Machinery.
- [85] Y. Han, A. Rozga, N. Dimitrova, G. D. Abowd, and J. Stasko. Visual analysis of proximal temporal relationships of social and communicative behaviors. *Computer Graphics Forum*, 34(3):51–60, 2015.
- [86] A. Hargadon and R. I. Sutton. Technology brokering and innovation in a product development firm. *Administrative science quarterly*, pages 716–749, 1997.
- [87] J. Harper and M. Agrawala. Converting basic d3 charts into reusable style templates. *IEEE Transactions on Visualization and Computer Graphics*, 24(3):1274–1286, 2018.
- [88] J. Heer. Data types, graphical marks, and visual encoding channels. <https://observablehq.com/@uwdata/data-types-graphical-marks-and-visual-encoding-channels>, Sep 2021. Accessed: Sept 28, 2023.
- [89] J. Heer, S. K. Card, and J. A. Landay. prefuse: a toolkit for interactive information visualization. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI ’05*, page 421–430, New York, NY, USA, 2005. Association for Computing Machinery.
- [90] M. Helms, S. S. Vattam, and A. K. Goel. Biologically inspired design: process and products. *Design studies*, 30(5):606–622, 2009.

- [91] M. E. Helms, S. Vattam, and A. Goel. Compound analogical design, or how to make a surfboard disappear. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 30, 2008.
- [92] S. R. Herring, C.-C. Chang, J. Krantzler, and B. P. Bailey. Getting inspired! understanding how and why examples are used in creative design practice. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 87–96, 2009.
- [93] S. R. Herring, B. R. Jones, and B. P. Bailey. Idea generation techniques among creative professionals. In *2009 42nd Hawaii International Conference on System Sciences*, pages 1–10. IEEE, 2009.
- [94] T. Hill, V. Charles, J. Stiller, and A. Isaac. “searching for inspiration”: User needs and search architecture in europeana collections. *Proceedings of the Association for Information Science and Technology*, 53(1):1–7, 2016.
- [95] R. Holmes and G. C. Murphy. Using structural context to recommend source code examples. In *Proceedings of the 27th International Conference on Software Engineering, ICSE ’05*, page 117–125, New York, NY, USA, 2005. Association for Computing Machinery.
- [96] T. Hope, R. Tamari, H. Kang, D. Hershcovich, J. Chan, A. Kittur, and D. Shahaf. Scaling creative inspiration with fine-grained functional facets of product ideas. *arXiv preprint arXiv:2102.09761*, 2021.
- [97] E. Hoque and M. Agrawala. Searching the visual style and structure of d3 visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):1236–1245, 2020.
- [98] P. A. Howard-Jones, S.-J. Blakemore, E. A. Samuel, I. R. Summers, and G. Claxton. Semantic divergence and creative story generation: An fmri investigation. *Cognitive Brain Research*, 25(1):240–250, 2005.
- [99] K. Hu, M. A. Bakker, S. Li, T. Kraska, and C. Hidalgo. Vizml: A machine learning approach to visualization recommendation. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–12, 2019.
- [100] M. Hu, K. Wongsuphasawat, and J. Stasko. Visualizing social media content with senten-tree. *IEEE transactions on visualization and computer graphics*, 23(1):621–630, 2016.
- [101] N. Inie and P. Dalsgaard. How interaction designers use tools to manage ideas. *ACM Trans. Comput.-Hum. Interact.*, 27(2), Mar. 2020.
- [102] P. Jaccard. The distribution of the flora in the alpine zone. 1. *New phytologist*, 11(2):37–50, 1912.
- [103] D. G. Jansson and S. M. Smith. Design fixation. *Design Studies*, 12(1):3–11, 1991.
- [104] W. Javed and N. Elmqvist. Exploring the design space of composite visualization. In *2012 IEEE Pacific Visualization Symposium*, pages 1–8. IEEE, 2012.

- [105] E. Kamrah, S. F. Ghoreishi, Z. Ding, J. Chan, and M. Fuge. How diverse initial samples help and hurt bayesian optimizers. *Journal of Mechanical Design*, 145(11):111703, 2023.
- [106] A. Kerne, N. Lupfer, R. Linder, Y. Qu, A. Valdez, A. Jain, K. Keith, M. Carrasco, J. Vane-gas, and A. Billingsley. Strategies of free-form web curation: Processes of creative en-gagement with prior work. In *Proceedings of the 2017 ACM SIGCHI Conference on Cre-ativity and Cognition, C&C '17*, page 380–392, New York, NY, USA, 2017. Association for Computing Machinery.
- [107] A. Kerne, A. M. Webb, S. M. Smith, R. Linder, N. Lupfer, Y. Qu, J. Moeller, and S. Dama-raj. Using metrics of curation to evaluate information-based ideation. *ACM Transactions on Computer-Human Interaction (ToCHI)*, 21(3):1–48, 2014.
- [108] K. H. Kim and R. A. Pierce. Convergent versus divergent thinking. *Encyclopedia of creativity, invention, innovation and entrepreneurship*, pages 245–250, 2013.
- [109] R. Kosara and J. Mackinlay. Storytelling: The next step for visualization. *Computer*, 46(5):44–50, 2013.
- [110] K. Krippendorff. Estimating the reliability, systematic error and random error of interval data. *Educational and psychological measurement*, 30(1):61–70, 1970.
- [111] C. Kulkarni, S. P. Dow, and S. R. Klemmer. Early and repeated exposure to examples improves creative work. In *Design thinking research: Building innovation eco-systems*, pages 49–62. Springer, 2013.
- [112] H. Lam, D. Russell, D. Tang, and T. Munzner. Session viewer: Visual exploratory analysis of web session logs. In *2007 IEEE Symposium on Visual Analytics Science and Technol-ogy*, pages 147–154, 2007.
- [113] B. Lee, R. H. Kazi, and G. Smith. Sketchstory: Telling more engaging stories with data through freeform sketching. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2416–2425, 2013.
- [114] H. Li, Y. Wang, A. Wu, H. Wei, and H. Qu. Structure-aware visualization retrieval. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, New York, NY, USA, 2022. Association for Computing Machinery.
- [115] W.-S. Li and C. Clifton. Semantic integration in heterogeneous databases using neural networks. In *Proceedings of the 20th International Conference on Very Large Data Bases, VLDB '94*, page 1–12, San Francisco, CA, USA, 1994. Morgan Kaufmann Publishers Inc.
- [116] J. S. Linsey, I. Tseng, K. Fu, J. Cagan, K. L. Wood, and C. Schunn. A Study of Design Fix-ation, Its Mitigation and Perception in Engineering Design Faculty. *Journal of Mechanical Design*, 132(4):041003, 04 2010.
- [117] Z. Liu, C. Chen, F. Morales, and Y. Zhao. Atlas: Grammar-based procedural generation of data visualizations. In *2021 IEEE Visualization Conference (VIS)*. IEEE, 2021.

- [118] Z. Liu, B. Kerr, M. Dontcheva, J. Grover, M. Hoffman, and A. Wilson. Coreflow: Extracting and visualizing branching patterns from event sequences. *Computer Graphics Forum*, 36(3):527–538, 2017.
- [119] Z. Liu, J. Thompson, A. Wilson, M. Dontcheva, J. Delorey, S. Grigg, B. Kerr, and J. Stasko. Data illustrator: Augmenting vector design tools with lazy data binding for expressive visualization authoring. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI ’18, page 1–13, New York, NY, USA, 2018. Association for Computing Machinery.
- [120] Z. Liu, Y. Wang, M. Dontcheva, M. Hoffman, S. Walker, and A. Wilson. Patterns and sequences: Interactive exploration of clickstreams to understand common visitor paths. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):321–330, 2017.
- [121] A. Lucero. Framing, aligning, paradoxing, abstracting, and directing: how design mood boards work. In *Proceedings of the Designing Interactive Systems Conference*, DIS ’12, page 438–447, New York, NY, USA, 2012. Association for Computing Machinery.
- [122] A. S. Luchins. Mechanization in problem solving: The effect of einstellung. *Psychological monographs*, 54(6):i, 1942.
- [123] Y. Luo, X. Qin, N. Tang, and G. Li. Deepeye: Towards automatic data visualization. In *2018 IEEE 34th International Conference on Data Engineering (ICDE)*, pages 101–112, 2018.
- [124] N. Lupfer, A. Kerne, A. M. Webb, and R. Linder. Patterns of free-form curation: Visual thinking with web content. In *Proceedings of the 24th ACM International Conference on Multimedia*, MM ’16, page 12–21, New York, NY, USA, 2016. Association for Computing Machinery.
- [125] R. Ma, H. Mei, H. Guan, W. Huang, F. Zhang, C. Xin, W. Dai, X. Wen, and W. Chen. Ladv: Deep learning assisted authoring of dashboard visualizations from images and sketches. *IEEE Transactions on Visualization and Computer Graphics*, 27(9):3717–3732, 2020.
- [126] J. Mackinlay. Automating the design of graphical presentations of relational information. *ACM Trans. Graph.*, 5(2):110–141, apr 1986.
- [127] J. Mackinlay, P. Hanrahan, and C. Stolte. Show me: Automatic presentation for visual analysis. *IEEE transactions on visualization and computer graphics*, 13(6):1137–1144, 2007.
- [128] S. MacNeil, Z. Huang, K. Chen, Z. Ding, A. Yu, K. Nakai, and S. P. Dow. Freeform templates: Combining freeform curation with structured templates. In *Proceedings of the 15th Conference on Creativity and Cognition*, C&C ’23, page 478–488, New York, NY, USA, 2023. Association for Computing Machinery.
- [129] J. Madhavan, P. A. Bernstein, A. Doan, and A. Halevy. Corpus-based schema matching. In *21st International Conference on Data Engineering (ICDE’05)*, pages 57–68, 2005.

- [130] J. Madhavan, P. A. Bernstein, and E. Rahm. Generic schema matching with cupid. In *Proceedings of the 27th International Conference on Very Large Data Bases, VLDB '01*, page 49–58, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.
- [131] S. Malik, F. Du, M. Monroe, E. Onukwugha, C. Plaisant, and B. Shneiderman. Cohort comparison of event sequences with balanced integration of visual analytics and statistics. In *Proceedings of the 20th International Conference on Intelligent User Interfaces, IUI '15*, page 38–49, New York, 2015. ACM.
- [132] A. B. Markman and D. Gentner. Commonalities and differences in similarity comparisons. *Memory & cognition*, 24(2):235–249, 1996.
- [133] R. L. Marsh and J. D. Landau. Availability in cryptomnesia: Assessing its role in two paradigms of unconscious plagiarism. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 21(6):1568, 1995.
- [134] R. L. Marsh, J. D. Landau, and J. L. Hicks. How examples may (and may not) constrain creativity. *Memory & cognition*, 24(5):669–680, 1996.
- [135] R. L. Marsh, T. B. Ward, and J. D. Landau. The inadvertent use of prior knowledge in a generative cognitive task. *Memory & cognition*, 27(1):94–105, 1999.
- [136] S. McKenna, D. Mazur, J. Agutter, and M. Meyer. Design activity framework for visualization design. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2191–2200, 2014.
- [137] A. M. McNutt and R. Chugh. Integrated visualization editing via parameterized declarative templates. 2021.
- [138] S. MEHER. Boston weather 2013-2023. https://www.kaggle.com/datasets/swaroopmeher/boston-weather-2013-2023?select=boston_weather_data.csv, March 2023. Accessed: June 24, 2024.
- [139] J. Meyer, B. Bederson, and J.-D. Fekete. Agile2d. <https://www.cs.umd.edu/hcil/agile2d/index.shtml>, 2002. Accessed: April 11, 2024.
- [140] Microsoft Corporation. Microsoft excel.
- [141] S. R. Miller and B. P. Bailey. Searching for inspiration: An in-depth look at designers example finding practices. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 46407, page V007T07A035. American Society of Mechanical Engineers, 2014.
- [142] A. Monroy-Hernández. *Designing for remixing: Supporting an online community of amateur creators*. PhD thesis, Massachusetts Institute of Technology, 2012.
- [143] D. Moritz, C. Wang, G. L. Nelson, H. Lin, A. M. Smith, B. Howe, and J. Heer. Formalizing visualization design knowledge as constraints: Actionable and extensible models in draco. *IEEE transactions on visualization and computer graphics*, 25(1):438–448, 2018.

- [144] J. Moss, K. Kotovsky, and J. Cagan. The effect of incidental hints when problems are suspended before, during, or after an impasse. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 37(1):140, 2011.
- [145] X. Mu, K. Xu, Q. Chen, F. Du, Y. Wang, and H. Qu. MOOCad: Visual Analysis of Anomalous Learning Activities in Massive Open Online Courses. In J. Johansson, F. Sadlo, and G. E. Marai, editors, *EuroVis 2019 - Short Papers*. The Eurographics Association, 2019.
- [146] M. D. Mumford, M. I. Mobley, R. Reiter-Palmon, C. E. Uhlman, and L. M. Doares. Process analytic models of creative capacities. *Creativity research journal*, 4(2):91–122, 1991.
- [147] T. Munzner. *Visualization analysis and design*. CRC press, 2014.
- [148] B. A. Nardi. *Context and consciousness: Activity theory and human-computer interaction*. mit Press, 1996.
- [149] 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*, 27(2):369–379, 2021.
- [150] S. M. Nasehi, J. Sillito, F. Maurer, and C. Burns. What makes a good code example?: A study of programming q&a in stackoverflow. In *2012 28th IEEE International Conference on Software Maintenance (ICSM)*, pages 25–34, 2012.
- [151] P. H. Nguyen, R. Henkin, S. Chen, N. Andrienko, G. Andrienko, O. Thonnard, and C. Turkay. Vasabi: Hierarchical user profiles for interactive visual user behaviour analytics. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):77–86, 2020.
- [152] B. A. Nijstad, W. Stroebe, and H. F. Lodewijkx. Cognitive stimulation and interference in groups: Exposure effects in an idea generation task. *Journal of experimental social psychology*, 38(6):535–544, 2002.
- [153] H. Niu, I. Keivanloo, and Y. Zou. Learning to rank code examples for code search engines. *Empirical Software Engineering*, 22:259–291, 2017.
- [154] OpenAI. Gpt-3.5. <https://openai.com/gpt-3>, 2021. Accessed: March 4, 2024.
- [155] A. F. Osborn. *Applied imagination*. 1953.
- [156] S. Palani, Z. Ding, S. MacNeil, and S. P. Dow. The ”active search” hypothesis: How search strategies relate to creative learning. In *Proceedings of the 2021 Conference on Human Information Interaction and Retrieval, CHIIR ’21*, page 325–329, New York, NY, USA, 2021. Association for Computing Machinery.
- [157] S. Palani, Z. Ding, A. Nguyen, A. Chuang, S. MacNeil, and S. P. Dow. Conotate: Suggesting queries based on notes promotes knowledge discovery. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, CHI ’21*, New York, NY, USA, 2021. Association for Computing Machinery.

- [158] P. Parsons, P. Shukla, and C. Park. Fixation and creativity in data visualization design: Experiences and perspectives of practitioners. In *2021 IEEE Visualization Conference (VIS)*, pages 76–80, 2021.
- [159] M. Perttula and P. Sipilä. The idea exposure paradigm in design idea generation. *Journal of Engineering Design*, 18(1):93–102, 2007.
- [160] Pinterest. Pinterest. <https://www.pinterest.com/>, 2009. Accessed: 2022-03-22.
- [161] C. Plaisant. The challenge of information visualization evaluation. In *Proceedings of the working conference on Advanced visual interfaces*, pages 109–116, 2004.
- [162] Plotly. Collaborative data science, 2015.
- [163] T. Preston-Werner, C. Wanstrath, P. Hyett, and S. Chaconl. Github, 2008 (accessed July, 2019).
- [164] A. Purcell and J. S. Gero. Design and other types of fixation. *Design Studies*, 17(4):363–383, 1996. Special Issue: Design Cognition and Computation.
- [165] A. T. Purcell, P. F. Williams, J. S. Gero, and B. Colbron. Fixation effects: Do they exist in design problem solving? *Environment and Planning B: Planning and Design*, 20(3):333–345, 1993.
- [166] C. Qian, S. Sun, W. Cui, J.-G. Lou, H. Zhang, and D. Zhang. Retrieve-then-adapt: Example-based automatic generation for proportion-related infographics. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):443–452, 2020.
- [167] M. D. Rahman, G. J. Quadri, B. Doppalapudi, D. A. Szafrir, and P. Rosen. A qualitative analysis of common practices in annotations: A taxonomy and design space, 2023.
- [168] Reddit. R/dataisbeautiful. <https://www.reddit.com/r/dataisbeautiful/>, 2005. Accessed: 2022-03-22.
- [169] N. Reimers and I. Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019.
- [170] D. Ren, M. Brehmer, B. Lee, T. Höllerer, and E. K. Choe. Chartaccent: Annotation for data-driven storytelling. In *2017 IEEE Pacific Visualization Symposium (PacificVis)*, pages 230–239, 2017.
- [171] D. Ren, T. Höllerer, and X. Yuan. ivisdesigner: Expressive interactive design of information visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2092–2101, 2014.
- [172] D. Ren, B. Lee, and M. Brehmer. Charticulator: Interactive construction of bespoke chart layouts. *IEEE transactions on visualization and computer graphics*, 25(1):789–799, 2018.

- [173] D. Ren, B. Lee, M. Brehmer, and N. H. Riche. Reflecting on the evaluation of visualization authoring systems: Position paper. In *2018 IEEE Evaluation and Beyond-Methodological Approaches for Visualization (BELIV)*, pages 86–92. IEEE, 2018.
- [174] A. Renkl. Learning from worked examples: How to prepare students for meaningful problem solving. 2014.
- [175] A. Renkl. Toward an instructionally oriented theory of example-based learning. *Cognitive science*, 38(1):1–37, 2014.
- [176] E. F. Rietzschel, B. A. Nijstad, and W. Stroebe. Relative accessibility of domain knowledge and creativity: The effects of knowledge activation on the quantity and originality of generated ideas. *Journal of experimental social psychology*, 43(6), 2007.
- [177] E. L. Rissland. Example-based reasoning. In *Informal reasoning and education*, pages 187–208. Routledge, 2012.
- [178] H. Ritchie and M. Roser. Forests and deforestation. <https://ourworldindata.org/forests-and-deforestation>, 2021.
- [179] E. Rosselli Del Turco and P. Dalsgaard. “my ideas come little by little”: how graphic professionals manage ideas. In *Proceedings of the 16th Conference on Creativity & Cognition, C&C ’24*, page 452–463, New York, NY, USA, 2024. Association for Computing Machinery.
- [180] T. Russell-Rose and T. Tate. *Designing the search experience: The information architecture of discovery*. Newnes, 2012.
- [181] B. Saleh, M. Dontcheva, A. Hertzmann, and Z. Liu. Learning style similarity for searching infographics. *arXiv preprint arXiv:1505.01214*, 2015.
- [182] A. Satyanarayan and J. Heer. Lyra: An interactive visualization design environment. *Computer Graphics Forum*, 33, 06 2014.
- [183] A. Satyanarayan, B. Lee, D. Ren, J. Heer, J. Stasko, J. Thompson, M. Brehmer, and Z. Liu. Critical reflections on visualization authoring systems. *IEEE transactions on visualization and computer graphics*, 26(1):461–471, 2019.
- [184] A. Satyanarayan, D. Moritz, K. Wongsuphasawat, and J. Heer. Vega-lite: A grammar of interactive graphics. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):341–350, 2017.
- [185] E. Segel and J. Heer. Narrative visualization: Telling stories with data. *IEEE transactions on visualization and computer graphics*, 16(6):1139–1148, 2010.
- [186] V. Setlur, A. Kanyuka, and A. Srinivasan. Olio: A semantic search interface for data repositories. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology, UIST ’23*, New York, NY, USA, 2023. Association for Computing Machinery.

- [187] G. Shani, R. I. Brafman, and D. Heckerman. An mdp-based recommender system. In *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence*, UAI'02, page 453–460, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc.
- [188] M. Sharmin, B. P. Bailey, C. Coats, and K. Hamilton. Understanding knowledge management practices for early design activity and its implications for reuse. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, page 2367–2376, New York, NY, USA, 2009. Association for Computing Machinery.
- [189] R. N. Shepard. Recognition memory for words, sentences, and pictures. *Journal of verbal Learning and verbal Behavior*, 6(1):156–163, 1967.
- [190] Y. Shi, P. Liu, S. Chen, M. Sun, and N. Cao. Supporting expressive and faithful pictorial visualization design with visual style transfer. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):236–246, 2022.
- [191] B. Shneiderman. Creating creativity: user interfaces for supporting innovation. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 7(1):114–138, 2000.
- [192] P. Siangliulue, K. C. Arnold, K. Z. Gajos, and S. P. Dow. Toward collaborative ideation at scale: Leveraging ideas from others to generate more creative and diverse ideas. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, CSCW '15, page 937–945, New York, NY, USA, 2015. Association for Computing Machinery.
- [193] P. Siangliulue, J. Chan, K. Z. Gajos, and S. P. Dow. Providing timely examples improves the quantity and quality of generated ideas. In *Proceedings of the 2015 ACM SIGCHI Conference on Creativity and Cognition*, pages 83–92, 2015.
- [194] U. N. Sio, K. Kotovsky, and J. Cagan. Fixation or inspiration? a meta-analytic review of the role of examples on design processes. *Design Studies*, 39:70–99, 2015.
- [195] G. F. Smith. Idea-generation techniques: A formulary of active ingredients. *The Journal of Creative Behavior*, 32(2):107–134, 1998.
- [196] S. M. Smith, T. B. Ward, and J. S. Schumacher. Constraining effects of examples in a creative generation task. *Memory & cognition*, 21(6):837–845, 1993.
- [197] K. Son, D. Choi, T. S. Kim, Y.-H. Kim, and J. Kim. Genquery: Supporting expressive visual search with generative models. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, CHI '24, New York, NY, USA, 2024. Association for Computing Machinery.
- [198] O. Source. Babel, 2020-07.
- [199] A. Taylor and H. R. Greve. Superman or the fantastic four? knowledge combination and experience in innovative teams. *Academy of management Journal*, 49(4):723–740, 2006.

- [200] J. Teevan, C. Alvarado, M. S. Ackerman, and D. R. Karger. The perfect search engine is not enough: a study of orienteering behavior in directed search. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '04, page 415–422, New York, NY, USA, 2004. Association for Computing Machinery.
- [201] T. Tsandilas. Structgraphics: Flexible visualization design through data-agnostic and reusable graphical structures. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):315–325, 2020.
- [202] I. Tseng, J. Moss, J. Cagan, and K. Kotovsky. The role of timing and analogical similarity in the stimulation of idea generation in design. *Design studies*, 29(3):203–221, 2008.
- [203] E. R. Tufte and D. Robins. *Visual explanations*. Graphics Cheshire, CT, 1997.
- [204] P. Vaithilingam, E. L. Glassman, J. P. Inala, and C. Wang. Dynavis: Dynamically synthesized ui widgets for visualization editing. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, CHI '24, New York, NY, USA, 2024. Association for Computing Machinery.
- [205] M. Vartak, S. Huang, T. Siddiqui, S. Madden, and A. Parameswaran. Towards visualization recommendation systems. *ACM SIGMOD Record*, 45(4):34–39, 2017.
- [206] F. B. Viegas, M. Wattenberg, F. van Ham, J. Kriss, and M. McKeon. Manyeyes: a site for visualization at internet scale. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1121–1128, 2007.
- [207] R. Vuillemot and J. Boy. Structuring visualization mock-ups at the graphical level by dividing the display space. *IEEE transactions on visualization and computer graphics*, 24(1):424–434, 2017.
- [208] G. Wallas. *The art of thought*, volume 10. Harcourt, Brace, 1926.
- [209] J. Walny, C. Frisson, M. West, D. Kosminsky, S. Knudsen, S. Carpendale, and W. Willett. Data changes everything: Challenges and opportunities in data visualization design handoff. *IEEE Transactions on Visualization & Computer Graphics*, 26(01):12–22, jan 2020.
- [210] C. Wang, Y. Feng, R. Bodik, A. Cheung, and I. Dillig. Visualization by example. *Proceedings of the ACM on Programming Languages*, 4(POPL):1–28, 2019.
- [211] C. Wang, Y. Feng, R. Bodik, I. Dillig, A. Cheung, and A. J. Ko. Falx: Synthesis-powered visualization authoring. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–15, 2021.
- [212] T. B. Ward. Structured imagination: The role of category structure in exemplar generation. *Cognitive psychology*, 27(1):1–40, 1994.
- [213] A. Warr and E. O'Neill. Understanding design as a social creative process. In *Proceedings of the 5th Conference on Creativity & Cognition*, pages 118–127, 2005.

- [214] T. R. Weirich, N. Tschakert, and S. Kozlowski. Teaching data analytics skills in auditing classes using tableau. *Journal of Emerging Technologies in Accounting*, 15(2):137–150, 2018.
- [215] M. Whitelaw et al. Generous interfaces for digital cultural collections. 2015.
- [216] J. O. Wilson, D. Rosen, B. A. Nelson, and J. Yen. The effects of biological examples in idea generation. *Design Studies*, 31(2):169–186, 2010.
- [217] K. Wongsuphasawat, D. Moritz, A. Anand, J. Mackinlay, B. Howe, and J. Heer. Voyager: Exploratory analysis via faceted browsing of visualization recommendations. *IEEE transactions on visualization and computer graphics*, 22(1):649–658, 2015.
- [218] K. Wongsuphasawat, Z. Qu, D. Moritz, R. Chang, F. Ouk, A. Anand, J. Mackinlay, B. Howe, and J. Heer. Voyager 2: Augmenting visual analysis with partial view specifications. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 2648–2659, 2017.
- [219] Z. Wu, V. Le, A. Tiwari, S. Gulwani, A. Radhakrishna, I. Radiček, G. Soares, X. Wang, Z. Li, and T. Xie. NL2viz: natural language to visualization via constrained syntax-guided synthesis. In *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering, ESEC/FSE 2022*, page 972–983, New York, NY, USA, 2022. Association for Computing Machinery.
- [220] Z. Wu, Q. Xu, Y. Liu, Z. Peng, Y. Xu, and X. Ma. Exploring designers’ practice of online example management for supporting mobile ui design. In *Proceedings of the 23rd International Conference on Mobile Human-Computer Interaction*, pages 1–12, 2021.
- [221] X. Xu, R. Xiong, B. Wang, D. Min, and S. P. Dow. Ideaterelate: An examples gallery that helps creators explore ideas in relation to their own. *Proceedings of the ACM on Human-Computer Interaction*, 5(CSCW2):1–18, 2021.
- [222] C. Yan and Y. He. Auto-suggest: Learning-to-recommend data preparation steps using data science notebooks. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data, SIGMOD ’20*, page 1539–1554, New York, NY, USA, 2020. Association for Computing Machinery.
- [223] J. Yang, A. M. McNutt, and L. Battle. Considering visualization example galleries. In *2024 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pages 329–343, 2024.
- [224] Y. Ye, R. Huang, and W. Zeng. Visatlas: An image-based exploration and query system for large visualization collections via neural image embedding. *IEEE Transactions on Visualization and Computer Graphics*, 30(7):3224–3240, 2024.
- [225] R. Zehrung, A. Singhal, M. Correll, and L. Battle. Vis ex machina: An analysis of trust in human versus algorithmically generated visualization recommendations. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, CHI ’21*, New York, NY, USA, 2021. Association for Computing Machinery.

- [226] L. Zeng, R. W. Proctor, and G. Salvendy. Fostering creativity in product and service development: validation in the domain of information technology. *Human Factors*, 53(3):245–270, 2011.
- [227] Z. Zeng, P. Moh, F. Du, J. Hoffswell, T. Y. Lee, S. Malik, E. Koh, and L. Battle. An evaluation-focused framework for visualization recommendation algorithms. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):346–356, 2022.
- [228] J. E. Zhang, N. Sultanum, A. Bezerianos, and F. Chevalier. Dataquilt: Extracting visual elements from images to craft pictorial visualizations. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, page 1–13, New York, NY, USA, 2020. Association for Computing Machinery.
- [229] Y. Zhang, R. Capra, and Y. Li. An in-situ study of information needs in design-related creative projects. In *Proceedings of the 2020 Conference on Human Information Interaction and Retrieval*, CHIIR ’20, page 113–123, New York, NY, USA, 2020. Association for Computing Machinery.
- [230] J. Zhao, Z. Liu, M. Dontcheva, A. Hertzmann, and A. Wilson. Matrixwave: Visual comparison of event sequence data. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI ’15, page 259–268, New York, 2015. ACM.
- [231] C. Zhu-Tian, Y. Wang, Q. Wang, Y. Wang, and H. Qu. Towards automated infographic design: Deep learning-based auto-extraction of extensible timeline. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):917–926, Jan. 2020.
- [232] K. T. Zinat, S. N. Sakhamuri, A. S. Chen, and Z. Liu. A multi-level task framework for event sequence analysis. *IEEE Transactions on Visualization and Computer Graphics*, 31(1):842–852, 2025.
- [233] K. T. Zinat, J. Yang, A. Gandhi, N. Mitra, and Z. Liu. A comparative evaluation of visual summarization techniques for event sequences. *Computer Graphics Forum*, 42(3):173–185, 2023.