ABSTRACT

Title of Dissertation: A SYSTEMATIC AND MINIMALIST APPROACH TO LOWER BARRIERS IN VISUAL DATA EXPLORATION

Mehmet Adil Yalçın, Doctor of Philosophy, 2016

Dissertation directed by: Professor Benjamin B. Bederson
Department of Computer Science
Associate Professor Niklas Elmqvist
College of Information Studies

With the increasing availability and impact of data in our lives, we need to make quicker, more accurate, and intricate data-driven decisions. We can see and interact with data, and identify relevant features, trends, and outliers through visual data representations. In addition, the outcomes of data analysis reflect our cognitive processes, which are strongly influenced by the design of tools. To support visual and interactive data exploration, this thesis presents a systematic and minimalist approach.

First, I present the Cognitive Exploration Framework, which identifies six distinct cognitive stages and provides a high-level structure to design guidelines, and evaluation of analysis tools. Next, in order to reduce decision-making complexities in creating effective interactive data visualizations, I present a minimal, yet expressive, model for tabular data using aggregated data summaries and linked selections. I demonstrate its application to common categorical, numerical, temporal, spatial, and set data types. Based on this model, I developed Keshif as an out-of-the-box, web-based tool to bootstrap the data exploration process. Then, I applied it to 160+

datasets across many domains, aiming to serve journalists, researchers, policy makers, businesses, and those tracking personal data.

Using tools with novel designs and capabilities requires learning and help-seeking for both novices and experts. To provide self-service help for visual data interfaces, I present a data-driven contextual in-situ help system, HelpIn, which contrasts with separated and static videos and manuals. Lastly, I present an evaluation on design and graphical perception for dense visualization of sorted numeric data. I contrast the non-hierarchical treemaps against two multi-column chart designs, wrapped bars and piled bars. The results support that multi-column charts are perceptually more accurate than treemaps, and the unconventional piled bars may require more training to read effectively.

This thesis contributes to our understanding on how to create effective data interfaces by systematically focusing on human-facing challenges through minimalist solutions. Future work to extend the power of data analysis to a broader public should continue to evaluate and improve design approaches to address many remaining cognitive, social, educational, and technical challenges.

A SYSTEMATIC AND MINIMALIST APPROACH TO LOWER BARRIERS IN VISUAL DATA EXPLORATION

by

Mehmet Adil Yalçın

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
2016

Advisory Committee:

Professor Benjamin B. Bederson, Chair
Professor Niklas Elmqvist, Co-chair
Professor Amitabh Varshney
Dr. Catherine Plaisant
Professor Ira Chinoy, Dean's Representative

Dedication

To my mother, and the memory of my father

For their unconditional love and support

Wherever we may be

Acknowledgements

Success stories are rarely the work of a single person working in isolation. I learned that ideas need cross-pollination, nurturing, support, competition, and some resistance to find ways to grow. This thesis is no exception. (I don’t mean to celebrate it as a success story too early, but to reflect the contributions and support of many).

I want to thank my advisors, Ben Bederson and Niklas Elmqvist. I feel lucky and privileged to learn from them. Ben has thought me to be skeptical about almost anything, including the utility of visualizations. Existing tools are still complex (more than potentially necessary), but we made some good progress on the broader, ambitious mission set three years ago. Niklas turned big obstacles into tiny hills. I could not have navigated this path without the map of research-land that he provided. Not only he brilliantly enriched the contributions in this thesis, he also taught me how to be better grounded, a careful listener, and a better speaker. His mentoring and support changed my life, making it possible for me to reach the finish line.

This thesis has an all-star committee (if you haven’t noticed already), with Amitabh Varshney, Catherine Plaisant, and Ira Chinoy. Catherine Plaisant has been a close critique of my work, and made it better at every interaction. Amitabh Varshney has been so gracious, understanding, and always smiling even at tough times, I appreciate his support dearly. Ira has brought his positive attitude and great insights with an eye for hidden stories and the messiness of the real world and the real data. To all my committee members, thank you for your time, feedback and support.

Next, I want to thank my loving wife Sally. She has been my biggest support and my biggest champion since we met. She made me more energized, and stronger. She is also a great first-time user of my prototypes, pointing all that doesn’t feel right, and encouraging me to step up my skills. Sally, you light the way, and you inspire to me to be a better person. I hope I can light your way, all the way, too.

Next, friends step forward! Dear HCIL, it has been great to be here. You have the best green, the best people, the best working space, and a spirit that is hard to find anywhere else. Nilsu, Niloo, Saniye. You gave me strength and great advices when I needed. Turkish friends, on and off campus, made my transition to a new world easier and much more fun. Music, you kept me grooving and feeling. - Thanks, Violetta!

Back to those who helped with the ideas, and were great collaborators and experts on their domains. Thank you, Amy P., Brian W., Michael J., Bill B. at SESYNC, Nick M. at SESYNC, and Andrew W., Elizabeth, Lindsay, Rowie at COMM107. Dear study participants who took time off your day to be stuck in an office looking at some (potentially boring) datasets, you did amazing, and taught me a lot!

Acknowledgements also go to the campus-wide effort to make this thesis possible. Thank you, Jennifer Story at the graduate office, advisors at the international student office, the iSchool staff, and all who coordinated to remove obstacles along the way for graduate students. I also want to thank Linda Macri for organizing the writing retreats. It was the most work done in a week!

Annecim, seni seviyorum. Hayallerimi gerçekleştirmem için ölçülmesi imkansız fedakarlıklar yaptın, bir hayat boyut. Bu tez senin için, ve senin eserin.

Table of Contents

Dedication **ii**

Acknowledgements **iii**

Table of Contents **v**

Chapter 1. Introduction **1**

1.1 Motivation 3
1.1.1 Motivation for a Cognitive Focus on Data Exploration 5
1.1.2 Motivation for a Visual and Interactive Model for Data Exploration 5
1.1.3 Motivation for Set-Typed Data Exploration 6
1.1.4 Motivation for Integrated Contextual Help System Design 7
1.1.5 Motivation for Dense Visualization of Numeric Data 8
1.2 Contributions 9
1.2.1 Cognitive Exploration Framework 11
1.2.2 A Minimal Yet Expressive Model for Data Exploration 11
1.2.3 AggreSet: Set-Typed Data Exploration Technique 13
1.2.4 HelpIn: Data-Driven Contextual In-Situ Help System 15
1.2.5 Evaluation of Multi-Column Bar Charts and Treemaps 16
1.3 Evaluations 18

Chapter 2. Background **20**

1.1 Sensemaking and Data Visualization 20
2.1 Cognition for Sensemaking 21
2.2 Barriers and Costs in Visual Data Exploration 22
2.3 Techniques for Multivariate Data Interfaces 24
2.3.1 Visualization Design Environments (VDEs) 24
2.3.2 Single-Chart Visualizations and Templates 25
2.3.3 Coordinated Multiple Views 26
2.3.4 Domain Specific Systems 27
2.3.5 Foundations 27
2.3.6 Web-based Visualization Tools for Charting and Publishing 27
2.3.7 Set-typed Data Visualization 28
2.4 Help and Documentation Systems 30
2.4.1 Motivating Work 30
2.4.2 Basic Interactive Techniques 31
2.4.3 Video-based Training 32
2.4.4 Context-aware Help Systems 33
2.4.5 Help and Training for Visual Data Interfaces 34
2.5 Visualization Design for Dense Numeric Data 35

Chapter 3. Cognitive Exploration Framework **38**

3.1 The Framework 39

3.1.1 The Six Cognitive Stages 40
3.1.2 The Factor of Decision Making........ 42
3.1.3 The Factor of Existing/New Knowledge........ 43
3.1.4 The Factor of Motivation........ 44
3.2 Design Guides from the Perspective of Cognitive Stages 45
3.2.1 Guides for Planning Data Analysis 46
3.2.2 Guides for Planning Interaction........ 46
3.2.3 Guides for Planning Visualization........ 47
3.2.4 Guides for Assessing Interaction........ 48
3.2.5 Guides for Assessing Visualization........ 49
3.2.6 Guides for Assessing Data Analysis........ 50
3.2.7 Guides across Cognitive Stages........ 50
3.3 An Evaluation Approach to Detect Cognitive Barriers and Activities 51
3.4 Discussion on the Cognitive Exploration Framework 53
3.4.1 Construction of the Framework........ 53
3.4.2 Implications for Design Guides........ 54
3.4.3 Reflections on Cognitive Evaluation of Exploratory Tools 55
3.4.4 Effort Differences across Cognitive Stages........ 57
3.5 Outline of the Thesis 57

Chapter 4. Aggregate Summaries and Linked Selection Model for Visual and Interactive Data Exploration 59
4.1 Aggregate Summaries Model........ 61
4.2 Linked Selection Model 62
4.3 Visual Data Encoding 65

Chapter 5. Keshif – The Implementation of The Exploration Model 67
5.1 Data Browser with Record Display........ 74
5.2 Design Specifics........ 75
5.2.1 Layout Design 75
5.2.2 Attribute and Summaries by Data Type 76
5.2.3 Pointer Based Linked Selection Design 78
5.3 Authoring Data Browsers........ 79
5.3.1 Graphical Authoring........ 80
5.3.2 Programmatic Authoring (API of Browser Configuration)........ 81
5.3.3 Sharing and Collaboration........ 83
5.4 Implementation 84
5.5 Discussion 87
5.6 Limitations 90
5.6.1 Data Model........ 90
5.6.2 Form Factor (Display size and input devices) 91
5.6.3 Collaboration........ 91
5.6.4 Required Skills for Customized Authoring 92
5.6.5 Data Size........ 92
5.6.6 Chart Types 93
5.6.7 Minimalism vs. Expressiveness........ 94

5.6.8 Minimalism vs. Discoverability 95
5.6.9 Minimalism vs. Visual Complexity 96

Chapter 6. AggreSet – Set-Typed Data Exploration Technique 99
6.1 Features of Set-Typed Data 99
6.2 Set Exploration Modeling 102
6.3 Set-typed Data Exploration with AggreSet 108
6.4 Details on Visual Encoding Design 111
6.5 Perceptual Set Ordering for the Set Matrix 112
6.6 Comparison of AggreSet and Other Set Visualization Techniques 116

Chapter 7. User Evaluations of Keshif 122
7.1 Evaluation of Cognitive Barriers with Data Analytics Novices 123
7.1.1 Study Design 124
7.1.2 Barriers in Planning Data Analysis 127
7.1.3 Barriers in Planning Interaction 128
7.1.4 Barriers in Planning Visualization 129
7.1.5 Barriers in Assessing Interaction 130
7.1.6 Barriers in Assessing Visualization 130
7.1.7 Barriers in Assessing Data Analysis 131
7.1.8 The Factor of Existing/New Knowledge 132
7.2 Insight-based Evaluation with Data Analytics Novices 134
7.2.1 Study Design 134
7.2.2 Insight Coding 137
7.2.3 Analysis and Results 143
7.3 Evaluation of AggreSet 147
7.3.1 Expert Review 147
7.3.2 Case Study 151
7.4 Applications over Multiple Domains 156
7.4.1 Sample Use Case Scenario for Data Journalism 156
7.4.2 Public Web-based Data Exploration at www.keshif.me 159
7.4.3 Other Use Cases 160

Chapter 8. Integrated Contextual Help for Data Interfaces 163
8.1 Context in Visual Data Interfaces 164
8.1.1 Data-Driven Context 164
8.1.2 Application-Driven Context 166
8.1.3 History-Driven Context 167
8.1.4 Topic Relevance and Ranking by Context 168
8.2 HelpIn – A Contextual In-Site Help System 169
8.2.1 Seeking Contextual Help 169
8.2.2 Presenting Contextual Help Instruction 170
8.2.3 Instructional Design 170
8.3 Modes of HelpIn 171
8.3.1 Overview 171
8.3.2 Topic Listing 172

8.3.3 Point & Learn ... 173
8.3.4 Guided Tour ... 175
8.3.5 Notifications ... 176
8.3.6 Topic Answers ... 177
8.4 Implementation ... 180
8.5 Evaluation ... 181
8.5.1 Participants ... 181
8.5.2 Study Design ... 182
8.5.3 Collected Data and Metrics ... 183
8.5.4 Training ... 184
8.5.5 Tasks ... 185
8.5.6 Procedure ... 187
8.5.7 Pilot Studies ... 188
8.6 Results ... 188
8.6.1 Task Progress Performance ... 188
8.6.2 Time on Help ... 189
8.6.3 When Help Is Not Needed ... 190
8.6.4 The Characteristics of Help System Use ... 191
8.6.5 Help Topic Listing Search Behavior ... 192
8.6.6 Subjective Preferences ... 193
8.7 Discussion ... 194
8.7.1 Experiment Results ... 194
8.7.2 Generalizing HelpIn ... 195
8.7.3 Help Material and Instructional Design ... 195
8.7.4 The Synergy Between Interface Design and Help Design ... 196
8.7.5 Limitation of Contextual In-Situ Help ... 197

Chapter 9. Dense Visualization Design for Numeric Data 198
9.1 Design Objectives ... 199
9.2 Design Alternatives ... 203
9.2.1 Treemap Technique ... 204
9.2.2 Wrapped Bars ... 205
9.2.3 Piled Bars ... 206
9.2.4 Labels, Use of Color, and Bi-Directional Bars ... 208
9.3 Crowdsourced Perceptual Evaluation ... 213
9.3.1 Tasks ... 213
9.3.2 Experiment Factors and Design ... 216
9.3.3 Chart Parameters ... 217
9.3.4 Participants ... 217
9.3.5 Training and Other Procedures ... 218
9.4 Evaluation for Comparison Task ... 221
9.4.1 Results and Discussions ... 223
9.5 Evaluation for Ranking Task ... 228
9.6 Evaluation for Distribution Overview Task ... 230
9.7 Summary of Results ... 234
9.8 Limitations and Future Work ... 235

Chapter 10. Conclusion 238

10.1 On Cognitive Exploration Framework 239

10.2 On the Data Exploration Model and Keshif, its implementation............. 240

10.3 On AggreSet: Set-Typed Data Exploration.. 240

10.4 On HelpIn: Contextual In-Situ Help 241

10.5 On Dense Visualization of Numeric Data And Piled Bars 242

10.6 Remarks.. 243

Glossary 247

List of Publications 250

Software Packages 251

Bibliography 252

Chapter 1. Introduction

"Engineers solve technical problems that are well behaved. Designers build and innovate solutions to wicked problems, human-messy problems. And on the (computer) mouse, the engineers did a really good job of making the little switches and the things that control the wires and send the information to the computer screen to show the pointer where it should go. (…) But, will people like it? Should you press the button once or twice? Should it make a noise when you do that or not? That's all part of the human experience. So we tried hundreds of different mice, and hundreds of different definitions of how the interaction between the person and the computer would go. Because that's a human experience. You cannot analyze that; you cannot sit down with an equation and figure that out. You have to go into this place called the future we haven't been yet, where computers are friendly, and talk to the user (…) and see what happens. And that's how we build our way forward."

Dave Evans

The Diane Rehm Show, WAMU, October 3rd 2016

The surrounding inspiration of this PhD thesis is that data analysis, as well as our everyday lives, can be profoundly shaped by human-based design. The utility of any tool, or process, that we use depends not only on its internal technology and capabilities, but also on the psychological, educational, social, and aesthetical context within which it operates. I argue that data analysis is a process that is no different. To be able to truly understand data, we face not only computational and algorithmic challenges, but also cognitive, educational, and even cultural ones. Designing the process of bringing data to life, or creating a data dialogue, compliments designing the algorithms. Neither can survive alone. In addition, to use data as a form of communication, or even to draw a conceptual picture implicitly lingering in our minds, data presentations need to appeal to our eyes as well as our mind.

For decades, the information visualization researchers have been developing new techniques, the practitioners have been working to satisfy client needs and develop a culture, and the public has been getting more literate through data-driven journalism and so many everyday interactions with data (information). Yet, we have more work to do to enable rapid, effective, and expressive visual data exploration. Specifically, in a human-centered context, innovation is not merely adding new capabilities and techniques, but also re-shaping the processes to meet the needs of today, and to prepare for a positive vision of the future. The only way to build a timeless sculpture is to remove the excess material, and the imagination of the artist defines what would remain, the core purpose of the material. Similarly, innovation in data and human based design can remove such extraneous material, the complexities, with a new vision for functionality, aesthetics, purpose, and value.

Motivated by the growing importance of making effective data driven decisions, and improving data literacy, for a broad public, and the inspirations of human-centered approaches to problem solving, this thesis presents (i) a framework to understand cognitive aspects of visual data exploration, (ii) a minimal yet expressive model to enable rapid tabular data exploration, (iii) the implementation on this model, Keshif, which has been applied in over 150 settings, (iv) a contextual, in-situ help system design for providing training in visual data interfaces, and (v) an evaluation of alternative visual designs for dense display of numeric datasets.

1.1 Motivation

This thesis is motivated to contribute towards more rapid, effective, and expressive visual data exploration. The need for rapidness reflects the essence of time. The throughput of data-driven knowledge from a dataset depends on making quick observations and a fluid dialogue to support the process. The need for effectiveness reflects the essence of analytical thinking and accuracy in analysis. Given many alternative ways to explore a dataset, there are many potentially misleading paths to inaccurate assessments, or roadblocks to reach specific targeted outcomes. The need for expressiveness reflects the essence of depth and richness of exploration outcomes. The tools and techniques should not only provide a singular view of the data, but a range of views each of which can answer new questions. However, taken together, rapidness, effectiveness, and expressiveness are goals that can oppose each other. Expressiveness can increase the time to make effective decisions, and rapidness can lower the depth and accuracy of assessments of data. The approach of this thesis is finding a balance, and generating new value through by not sacrificing along one dimension while improving among another dimension, recognizing that this is not a zero-sum setting.

Specifically, this thesis decomposes this higher motivation into multiple chapters. These motivations can also be linked together based on targeting various stages and factors of the Cognitive Exploration Framework (See Section 3.5). First, we need to have a clear understanding of the cognitive aspects of data exploration (Chapter 3). Second, we need to develop new, refined models that would create new environments that offer rapid, effective, and expressive exploration (Chapter 4). We need to implement new tools based on these models, and study how people use these them (Chapter 5). Third, we need to enable expressiveness for revealing deeper relations and information within richer data sources (Chapter 6). Fourth, we need to consider how people can be trained, and how they can receive help, so that they are able to quickly learn and effectively apply data analysis and explorations under various conditions and datasets (Chapter 8). Last, but not least, we need to evaluate alternative visualizations by their characteristics in graphical perception and design; to empirically find those that would be more effective under targeted settings (Chapter 9).

1.1.1 Motivation for a Cognitive Focus on Data Exploration

The value of data can be measured by the knowledge we can extract from it. Visual tools support exploration for knowledge discovery by creating an interactive dialogue with data. To evaluate the role of cognition, we focus on the role of a *data explorer* whose primary goal is to understand data by developing and answering questions. This is in contrast to consuming pre-extracted knowledge from a data presentation (such as a news story), communicating results [62], or designing specific interfaces and data exploration spaces for other users [16].

Visualization can amplify people's ability to comprehend data [26]. However, using visual tools for data analysis also requires other cognitive activities, such as forming analysis goals and interaction plans. Barriers to effective cognition can lead us to fruitless paths, inaccurate or false knowledge, lost time, or even the abandonment of exploration because of confusion and frustration. Existing work in modeling visualization or cognitive activities in exploration tend to be frameworks that focus on system components [26], [31], [60], empirical results from specific tools and study setups [54], [80], [84] or surveys [85]. Little work has focused on a comprehensive analysis of the cognitive aspects of visual data exploration.

1.1.2 Motivation for a Visual and Interactive Model for Data Exploration

Visual data exploration can be performed using visualization design environments (VDEs) (such as Tableau [136], Lyra [123], iVisDesigner [115]) that enable constructing custom visualizations and interactions based on rich visual grammars, interactive features, and data pipelines. VDEs are also designed to support explanatory tasks, such as storytelling and interactive infographics. As a result, VDEs typically define a highly expressive, yet vast and complex query and configuration space that requires users to make many decisions to create effective data views. This process demands high cognitive effort, requires knowledge and experience, and reduces exploratory speed, affecting both novices [54] and experts [16]. Extended discussions of related work in multi-dimensional data analysis, and their limitations in the context of this thesis, are presented in Section 2.3.

1.1.3 Motivation for Set-Typed Data Exploration

Many real-world data collections consist of elements with multiple attributes. Some of these attributes may take multiple categorical values; for example, movies may have multiple genres, recipes have multiple ingredients, students take multiple courses, and publications typically have multiple keywords and authors. These multi-valued categorical attributes are commonly referred as *set-typed* since they implicitly describe set memberships over elements.

Set-typed data has recently received considerable attention in the field of information visualization, with visual representations based on linear lists of set intersections [88], radial node-link diagrams [3], and element matrix compositions [121]. However, common between these and other visual set exploration approaches in the literature is that: (*i*) they scale to a relatively small number of sets; (*ii*) they are optimized for particular set exploration tasks; and (*iii*) they either do not support other element attributes beyond set membership, or the visualization and interaction is designed differently and ad-hoc for other attributes, decreasing consistency.

1.1.4 Motivation for Integrated Contextual Help System Design

Using computer applications effectively can be demanding for both first-time and experienced users. While user interface improvements, better interaction models, and increased familiarity have made applications easier to use, using new interfaces and learning new concepts always pose challenges [131]. In practice, users today expect to use new applications immediately with no or minimal training, and to learn and troubleshoot as they go.

In particular, designing self-instructional interfaces for data science tools faces many challenges because of the overall complexity of data analysis. Even a *visual data interface*, such as Tableau, Spotfire, or Keshif, which are based on interactive visualizations of data to aid sensemaking, must guide experts in translating their analytical knowledge into actual tool features. This step is even more challenging for casual users or novices, who have a limited vocabulary of data analysis, yet are increasingly consuming or searching for data-driven answers in their everyday lives. However, traditional help materials based on static datasets and fixed application settings cannot match the rich context of a live data analysis environment, and thus require the user to translate the abstract information into their task at hand. Integrated help systems have the potential to provide this crucial help and training guidance. Specifically, data interfaces constitute an unprecedented opportunity for *data-driven contextualization* where the features of the underlying dataset, such as variable types or distributions, and analysis settings, such as chart types and data selections, can be used to guide the user to learn the tool and perform data analysis.

1.1.5 Motivation for Dense Visualization of Numeric Data

Lists of numeric measurements for specific items—such as country populations, smartphone prices, or university acceptance rates—are ubiquitous. The sorted bar chart visualizes this data with perceptual effectiveness and simplicity. However, it can only show a few dozen records given standard constrained screen sizes. How can we visualize *more* records—such as 150 countries, 75 tablets, or 300 universities—in a chart, while maintaining perceptual accuracy for data comprehension? Among potential solutions, (i) larger screen spaces for charting may not be available, (ii) interaction, such as scrolling or focus+context, are not supported in ubiquitous print and image medium, and (iii) aggregation of underlying data prevents observing records individually. In addition, there currently exists no detailed evaluation of alternative visualizations and their graphical perception performance targeting this data setting and context.

1.2 Contributions

The contributions of this dissertation are as the following:

- (Chapter 3) We developed the Cognitive Exploration Framework to present a comprehensive structured overview of the cognitive activities and challenges in visual data exploration. The framework can be linked to many design guidelines in data analysis, and can be used for evaluation of data analysis tools as well. This framework is built upon, and extends, existing literature in visual data sensemaking, cognition, and barriers. The results are presented at BELIV 2016 [153].
- (Chapter 4) We developed a visual model for data exploration that reduces decision making in data exploration, and achieves minimalism while maintaining expressibility. We describe the visual, interactive, and analytical components of this framework, and describe its application to multiple common data types.
- (Chapter 5) We implemented a web-based data exploration tool called Keshif based on the proposed framework. The implementation allows creating data browsers using graphical authoring, or using a simple API. The browsers can be embedded, edited and shared easily. Based on our user evaluation with visual data analytics novices, short training and a casual setup, Keshif can lead to rapid data exploration performance as measured by the volume and

depth of data-driven insights. Our study results and performance are comparable to, and on par with, existing studies using advanced tools or novel prototypes with more skilled audiences.

- (Chapter 6) We present a focused design and development for analysis of set-typed (multi-value categorical) data, called AggreSet. This technique has advantages in visual scalability, consistency, and expressiveness compared to the state of the art. The results are published at TVCG as part of InfoVis 2015 proceedings [154].
- (Chapter 8) We present the design of a new contextual in-site (integrated) help system for visual data interfaces (*HelpIn*), with the goal to advance upon rapid help seeking and learning of data interfaces. *HelpIn* takes advantage of active data, and query, visualization states, and includes multiple modes targeting different use cases in help. Our approach clarifies the use of context for both help seeking and help presentation.
- (Chapter 9) We present a detailed evaluation, both in design and in graphical perception performance, for visualization of dense sorted numeric data. We present a novel visualization technique, called Piled Bars, which is an extension of the wrapped bars technique, with advantages in data encoding properties. The evaluation details the perception accuracy in three complimentary tasks (comparison, ranking, overview), as well as various practical use cases and alternatives of the studies alternative techniques.

Next, we describe the contributions of each chapter in more detail.

1.2.1 Cognitive Exploration Framework

The *Cognitive Exploration Framework* for visual data exploration (CEF) is a structured overview of six cognitive stages in data exploration. The factors of decision-making, existing knowledge and motivation are also identified in relation to cognitive activities. By its comprehensive coverage of cognitive activities, the framework can be used to improve and evaluate the design of exploratory tools. We demonstrate the rhetorical power of CEF by using it to categorize a large number of concrete design guides with respect to stages of cognition. In order to use CEF as a lens to evaluate tools, we propose an observational study approach that focuses on identifying failures and challenges in open-ended exploration instead of performance on benchmarked tasks or insights [122].

1.2.2 A Minimal Yet Expressive Model for Data Exploration

To streamline and unify the visualization authoring and data exploration workflow for tabular data, and to reduce complexities and decision-making costs, we propose the aggregate summaries and linked selection model. Data records are aggregated in attribute summaries with visual design based on data type. Our model reduces the search space for choosing visual data encodings by automating visual representations based on data type and semantics using perceptually effective, non-overlapping visual encodings. Thus, the user makes fewer decisions on data representation compared to VDEs, leaving more cognitive resources to reach data-driven insights, and reducing required visual analytics knowledge. The model defines an interactive overview-to-detail flow for visual exploration using three linked selection interactions: (i) *highlighting* (rapidly previewing record groups), (ii) *filtering* (focusing on a record group), and (iii) *comparison* (locking selection of record groups). Despite its minimalism, the model is expressive (enables rich data exploration) by its applicability to common data types (categorical, numerical, temporal, and spatial (Table 1)), and its support for measure functions for aggregates (count, sum, average) and visual scale modes (absolute, part-of). The model achieves scalability in record count by explicit aggregation, and its minimalism enables rapid learning.

Design and Implementation of Keshif: Data Exploration Environment

Based on this model, we designed and implemented Keshif, an open-source, web-based data exploration tool for tabular data, available online at www.keshif.me. Raw data is visualized by authoring a Keshif browser by inserting attribute summaries, the record display (showing records individually), and calculating custom attributes. This authoring can be done using graphical interface, as well as using the minimal JavaScript API of Keshif. Data is then interactively explored through Keshif's unified, consistent linked selections. By enabling authoring within exploration, the two processes are merged in the single environment.

Keshif further specializes summaries based on data semantics for tasks such as categorical sorting, flexible range selections, and navigation (scroll, pan, zoom). To enable exploration of spatial records or self-referencing attributes (networks), the record display can show records on a geographical map or as a node-link diagram, in addition to list views. Keshif browsers are defined with a compact configuration, which can be forked to enable collaboration. Browsers can be publicly shared on the web with a unique URL, or embedded into existing web pages using basic JavaScript and CSS programming, which also can be used to customize the browsers. As a result, Keshif provides an out-of-the-box tabular data exploration solution to enable rapid data exploration.

We present an evaluation of data exploration process from raw data with visual analytics novices in a casual, unguided setting given short training using the insight-based methodology. The results support that Keshif and its model for data exploration enable rapid learning, authoring, and discovery flow, averaging close to two insights shared per minute. We also validate the design through 160+ Keshif browsers on public datasets across many disciplines, enabled by the underlying generic model and its implementation.

1.2.3 AggreSet: Set-Typed Data Exploration Technique

AggreSet is a novel set exploration technique that solves set-exploration challenges noted above through an integrated design of linked visualizations of multiple data dimensions with rapid selection, filtering, and comparison (Figure 12). It addresses the challenges presented in the Section 1.1.3 as the following: (*i*) To improve scalability, AggreSet uses a matrix-based visualization for set relations. Scalability in the number of sets is achieved by the non-overlapping and zoomable nature of the set-matrix. Scalability in the number of elements is achieved by aggregation. (*ii*) Based on an analysis of set-typed data exploration and design guidelines, AggreSet is designed to achieve richness of supported tasks, design efficiency, and consistency. (*iii*) AggreSet embeds the set-matrix in a multi-view layout consisting of histogram-based visualizations that are brushed and linked in a design that does not differentiate between set-typed and multivariate attributes. Specifically, AggreSet achieves improved scalability, richness, consistency, and enables rapid set-typed data exploration through a new matrix-based design for visualizing sets:

Scalability: AggreSet supports concurrent analysis on numerous sets (50+) and many aggregated elements (100,000+) across multiple dimensions. Its scalability comes from non-overlapping visualizations of aggregations over elements, and a scrollable and zoomable matrix view for visualizing relations between sets.

Richness: AggreSet supports a plethora of tasks for exploring relations in set-typed attributes and elements with minimal visual and interaction components. Its multi-view and linked design enables higher-order analysis (e.g. intersection of three or more sets), surpassing the limitations of static 2D set-matrix layouts.

Consistency: The visual and interaction design of AggreSet is consistent across all attribute types; i.e. it does not differentiate between aggregates for sets, set-degrees, set-intersections and other attributes, when applicable.

Rapid exploration: The user can observe many relations on tightly coupled visualizations without performing explicit state changes that slow down interaction. Our visual and interaction design encourages an overview-to-detail exploration.

Matrix design for set relations: AggreSet's set-matrix visualizes set-specific relations: empty, identical and sub-sets. It also presents a new set similarity metric, and a new method for set ordering to perceptually emphasize intersections of set groups.

1.2.4 HelpIn: Data-Driven Contextual In-Situ Help System

To improve help and training for visual data interfaces, we present *HelpIn*, a contextual data-driven in-situ help system. With contextual integration of help instructions using visual callouts, superimposed labels, and dynamic annotation into a live visual data interface, such as in Figure 1, HelpIn responds to active data and application context, and reduces the physical distance of help material to the interface, targeting to weaken the split-attention effect [51]. The features od data, visualizations, and queries, as well as application and task history, is used to help the user to quickly find help material of interest (help seeking) by contextual filtering and ranking, as well as to comprehend dynamic narrative answers. We introduce five modes of help-seeking across the pull/push model (help initiated by the user vs. system) [67]: contextual help on pointed interface elements (Point Learn), topic listing, overview, guided tour, and notifications. In addition, while updating interface design can outdate fixed screenshots or videos, HelpIn allows help material to be adjusted in small pieces during development, enabling iterative maintenance.

We evaluated HelpIn in comparison to its stripped-down version with non-contextual topic index and non-integrated answers using shared instructional material. While our participants showed similar progress on given tasks across the help system conditions, the Point & Learn mode was found the most useful in their feedback, and lead to higher task completion performance while also increasing time spent on help. Given high quality help instructions, the preference across static vs. integrated topic answers were split across on individual level. We also report on help-seeking behaviors for visual analytics, including when, for what, and how.

1.2.5 Evaluation of Multi-Column Bar Charts and Treemaps

Relating to the graphical perception aspects of visual data exploration (and the visualization assessment stage of the Cognitive Exploration Framework, and guidelines thereof), this thesis also focuses on dense data visualizations for sorted numeric data to enable both overviews of all records, and comparisons across records. Figure 31 shows *treemaps* [71], *wrapped bars* [47], and *piled bars* that meet these goals. We considered *treemaps* (TM) because of their common use [17], [143],

[159] for presenting large numbers of records without hierarchical structure, although the technique was originally designed with hierarchical structures in mind [71]. Visualization tools such as Tableau also include treemaps as a suggested plot for a numeric attribute, which leads to its adaptation in various dashboards [35]. We considered *wrapped bars* (WB) and *piled bars* (PB), which are multi-column dense bar charts. Wrapped bars, to our knowledge first introduced by Stephen Few [47], use multiple columns to improve the compactness of the visual representation. Based on this design, we refined wrapped bars into the *piled bars* technique by using a shared baseline for all columns, which visually aligns all record bars, and improves on the data encoding resolution. However, this step introduces overlaps across bars along a row, which we separate visually using a gradient rendering approach. This thesis contributes a detailed analysis of the three designs, and discusses the use of color and bi-directional axis for visualizing negative values and grouped records, as well as showing record labels.

In addition, the thesis reports on the graphical perception performance of the alternative techniques through crowdsourced human experiments, comparing them on three complimentary tasks: comparison, ranking, and overview. In terms of data assessment accuracy, the results suggest that *piled bars* > *wrapped bars* > *treemaps* for comparison task (given a strong outline highlight stimulus); *wrapped bars* > *piled bars* > *treemaps* for ranking task; and *wrapped bars* ≅ *piled bars* > *treemaps* for overview task. The experiments with weaker mark-type stimulus for comparison task show that *piled bars* may not be interpreted correctly with limited training given its significantly lower accuracy performance. We also discuss the effects of column layout and data density on the perception performance.

We developed a JavaScript library to generate the studied chart designs and figures in this paper and the experiments, called chubuk.js, which is available as open source at github.com/adilyalcin/chubuk.js. The experiment data, setup and results are also publicly available at github.com/adilyalcin/chubuk.exp.

1.3 Evaluations

Developing human-centered design and evaluation techniques without actually getting humans to use it and influence the process would be like walking in the dark without a compass. We won't know how much progress we are really making, we won't know when we reach our destination, and we won't even know if our compass is well calibrated! Therefore, while we have our guidelines to act as our compass to our goals, we need to have people to illuminate our path, confirm progress and direction, and better understand our environment.

We evaluated each of our proposed contributions using targeted user studies under various settings. To evaluate the application of Cognitive Exploration Framework to detect cognitive barriers and activities, we used Keshif as the exploration tool and analyzed various challenges across the size stages proposed. To evaluate the exploration model on aggregate data summaries and linked selections, we used insights gained by our participants to measure progress. To evaluate the in-depth exploration of set-typed data, we conducted expert reviews and a short-term case study. These evaluations, all enabled by the extensive implementation of Keshif (Chapter 5), are described in detail in Chapter 7. We present the evaluations of HelpIn, and alternative visualization designs for the dense visualization of numeric data in their respective chapters, Chapter 8 and Chapter 9.

Chapter 2. Background

"A common risk in academic research is getting too caught up in our hammers (powerful solution techniques) and losing track of the nails (the problems that need solving)."

David R. Karger

in "The Semantic Web and End Users: What's Wrong and How to Fix It" [74]

This thesis builds upon a body of decades-long research and practice on data visualization, interaction, interface design, computation, and psychology. In this section, we give an overview of the background and related work that influenced and provided the motivation for the contributions presented thereafter. Additional references are cited in the throughout this thesis as necessary.

1.1 Sensemaking and Data Visualization

Sensemaking is an iterative process of gathering and representing information, developing insights through manipulation, and producing knowledge [139]. The information visualization reference model [26], [31] models visualization pipeline from a system point-of-view as transitions between data, analytical abstraction, visualization abstraction, and view. A nested model [101] can be used to evaluate such systems. Yet, these approaches are not based on cognitive processes in visual exploration. Information foraging [111] describes information search behavior using an analogy with animals hunting and gathering food. However, it does not model the data interfaces, interaction, and the analytical process. The data/frame theory of sensemaking [79] argues that sensemaking is composed of cycles of (i) elaborating a mental frame, (ii) preserving a frame, and (iii) reframing. While it models a reasoning process, it does not model the concrete roles of interaction and visualization, and cannot explicitly guide on supporting these processes.

2.1 Cognition for Sensemaking

Higher mental processes such as attention, language use, memory, perception, problem solving, and thinking, are the focus of cognitive psychology [49]. Cognition is therefore closely related to sensemaking and data visualization. Card et al. [26] define externalized cognition as the use of an external object to reduce mental effort and memory demands when performing a task. David Kirsh [78] extends the role of external representations into rearrangement, persistence, independence, reformulation, and natural encoding, the use of multiple representations, construction, and simplification of control. In a reverse perspective, Liu and Stasko [94] describe mental models as the internal, structural, behavioral, and functional analogues of external visualization systems. They argue that interaction primarily enables external anchoring, information foraging, and cognitive offloading.

Distributed cognition models transitions across cognitive representations, and can be applied to infovis [92]. Walny et al. [145] studied data-sketching as an external representation of data understanding. Their analysis focuses on finalized sketches as the artifacts, and not on the cognitive activities explaining *how* the participants created or iterated upon these sketches. While these studies aim to explain the tools as external representations helping cognition, they are primarily explanatory. We aim to close the gap between theory and practice by building a comprehensive and actionable framework, demonstrating its link to design, and its use for evaluating tools.

Shrinivasan [132] presents an analytical reasoning framework with three components, data/knowledge/navigation, which can be supported by special-purpose views in tools. Van Wijk's model of visualization [149] includes perception, knowledge, and exploration as user-level constructs. Green et al. [55] argues these constructs are cognitive processes informing each other. We focus on data exploration using a holistic model covering a wide range of cognitive activities. We identify six cognitive stages, which encompass perception as an assessment activity, and discuss the cognitive influence of knowledge and motivation factors.

2.2 Barriers and Costs in Visual Data Exploration

Generalizing our everyday interactions with the physical world, Norman's gulfs of execution and evaluation [104] is a simple, effective, and widely adopted model.

However, it does not fully explain visual data exploration, which involves deep analytical thinking and interaction with abstract data interfaces. Lam [85] presents a framework of seven interaction costs, based on a survey of usability problems reported in 484 papers. Our framework builds upon these works by decoupling cognitive and physical activities, and exclusively focuses on the cognition. Amar and Stasko [5] discuss two forms of analytical gaps: worldview gap (what is shown ↔ what needs to be shown to draw a straightforward representational conclusion), and rationale gap (perceiving a relationship ↔ being able to explain the confidence and the usefulness of it). Cognitive stages extend beyond analytical gaps, and aim to clarify the ambiguous definitions across cognitive activities.

The behavior of novices can reveal barriers that may be reduced or hidden because of existing skills. Grammel et al. [54] performed an observational study on how novices construct information visualizations. While their study suggests barriers in visualization construction, it does not reflect interactive autonomous data exploration since a mediator (Wizard of Oz) created visualizations using verbal descriptions of the participants. Kwon et al. [84] studied behavior of novices to identify visual analytics road-blocks. They gave participants pre-defined tasks and offered guidance, creating a partially explorative process and limiting the extent of reported roadblocks. Lee et al. [86] identified five cognitive activities in the sensemaking of unfamiliar charts: encounter, construct, explore, question, and flounder. However, the *explorer* would avoid creating unfamiliar visualizations [54].

Decision making as a cognitive activity, and its costs and factors, are well-formed within psychology [125]. Yet, decision costs lack a focused discussion in analytics community. Heer et al. [62] discusses “constraining the parameter space that users have to explore”, yet only considering visualizations. Dou et al. [40] studied constrained interactions on solving a math game, with empirical results suggesting that constraints can increase performance.

2.3 *Techniques for Multivariate Data Interfaces*

In this section, we present an overview of existing techniques and practices for multi-variate data interfaces. This thesis focuses on challenges regarding exploration of multi-variate data, and the proposed design and implementation solutions build upon a collection of best practices and with comparison to the state of the art.

2.3.1 Visualization Design Environments (VDEs)

Visualization design environments, such as Tableau [136], enables visualization specification through graphical user interfaces by drag-and-drop on visual encoding shelves. The abstractions in Lyra [123] and iVisDesigner [115] include marks, drop-zones, connectors, handles and data pipelines. However, data ↔ visual encoding task is one of the bottlenecks for infovis novices [54]; they commonly prefer familiar, simple visualizations such as bar and line charts. Kwon et al. [84] also notes that “failure to choose appropriate views” is a roadblock for novices. To improve data exploration process, systems can show *recommended* visualizations based on data types and intended tasks based on a visualization model. Recommendations may be a short list of suggestions, such as Tableau's *Show Me* [95] that uses a rule-based design on selected attribute types on its visual query model VizQL, or a fully automated approach [120]. The context of use can also be considered [53]. Another example is Voyager [151], a faceted browser that generates and recommends alternative data visualizations. However, it does not support querying data, its visualizations are static, and its visual model does not consider semantics, such as spatial view of categorical regions. In contrast to defining a grammar for flexible visualizations, recommending visualizations, and allowing customizations, the proposed exploration model (Chapter 4) and its implementation (Chapter 5) (i) use a set of fixed visual representations and interactions designed to support accurate graphical perception in statistical graphics [32] and to facilitate a rapid data exploration flow, (ii) give the user the control of selections of attributes and data queries, and (iii) provide semantic visual alternatives.

2.3.2 Single-Chart Visualizations and Templates

Chart templates offer a generalized solution for data visualization. They require explicit selection of the chart template (among available options), followed by specification of data ↔ visual encodings on the template slots. ManyEyes [144] was among the first platforms to offer visualization templates as a web service for many chart types, also supporting data upload, hosting, and commenting. Spreadsheet software (e.g. Microsoft Excel, Google Sheets) also offers charting with templates and data specifications. However, templated charts present a bottleneck for novices by requiring visual decisions upfront, and inappropriate decisions may lead to ineffective data views.

2.3.3 Coordinated Multiple Views

Effective data exploration requires multiple perspectives (views) that the user interactively controls. Coordination on interaction (such as by brushing and linking) enables observing data relations across views. Roberts [118] provides a survey on CMVs. Snap-together [105] treats coordination as database join queries. Improvise [146] provides a rich, customizable coordination model on shared objects and dependencies. These systems target high flexibility, expert users, a wide range of use cases and patterns. Their graphical design is based on many menus and configuration options. The targeted audience of such systems is commonly developers rather than the public. As Roberts [118] notes, "Concurrently they (developers) need to decide how the information will be aggregated or abstracted and finally work out how the user interacts with the system." Novice audiences are particularly disadvantaged from these shortcomings. The notion that "theoretically any operation can be coordinated between multiple views" [19], [118], [146] does not consider the increased costs on usability, discoverability, learnability, and decision making for querying relations in the data.

2.3.4 Domain Specific Systems

Domain-specific systems present design solutions, guidelines, and case studies based on a detailed analysis of domain requirements. Examples include energy portfolios [22], online communities [82], funding portfolios [96], temporal transactions [93], and literature surveys [12]. Domain specific systems can assume or emphasize specific properties or relations within their domain, yet potentially limit generalizations, i.e. transfer of solutions across datasets and domains. For example, SurVis [12] focuses on literature datasets including keywords citations. In contrast, we generalize exploration of self-referencing attributes as node-link charts, exploration of categories as sorted histograms, and offer a unified interaction model.

2.3.5 Foundations

Faceted browsing [157], which is based on query previews [56], has become a ubiquitous model of organizing and browsing tabular datasets. Dynamic queries [1] enable querying data using interface elements such as sliders, buttons and maps. Our solution builds on a tight integration of visual representation and interaction, extending the design basis of [1], [142], [157] for rich exploration by including rich visualizations supporting multiple selections, aggregate measures, and scale modes.

2.3.6 Web-based Visualization Tools for Charting and Publishing

Exhibit [68] allows constructing faceted data interfaces using XML specifications. Likewise, Keshif is easy to deploy, while also providing richer exploratory features and a graphical authoring. VisGets [39] provides an exploratory interface for time (histogram), location (bubble map) and tags (word clouds) in document collections. Compared to our system, it does not define a generalized visualization and interaction model, does not support selections to enable side-by-side comparisons, and does not support graphical authoring. Its user evaluation is limited to self-reported usability. In contrast, we present user evaluations using multiple methodologies, including cognitive barriers and data-driven insights with mixed qualitative and quantitative analysis.

2.3.7 Set-typed Data Visualization

This section presents a review of the related work on set visualization based on a categorical approach of visualization types from a recent survey [4]. We refer the reader to this survey for a more thorough analysis. After presenting AggreSet, the proposed technique in Section 6.6, we present a focused comparison and discussion of selected recent techniques.

Euler Diagrams: Sets can be drawn as enclosing boundaries around elements, generating Euler diagrams. Given few set and element counts, Euler diagrams are powerful and can intuitively demonstrate set concepts. However, scalability is an issue. Proposed improvements, such as untangling [116], cannot avoid the inherent visual complexity beyond a few hundred elements and only a few sets, especially when the sets are densely intersecting. An extensive survey of Euler diagrams is presented by Rodgers [119].

Overlays: Sets can also be overlaid on existing visualizations that define element positions (layout) by other attributes [2], [33], [38], [99]. Isocontours are commonly used to enclose elements within sets. Their scale is limited by the element count when elements are not aggregated. Elements appearing in many sets also increase visual overlaps and complexity as in Euler diagrams.

Node-Link and Chord Diagrams: Node-link diagrams visualize set relationships by mapping sets to nodes and set-pair (second degree) intersections to edges. Visual scalability is primarily influenced by the set (node) count and link sparseness (edge count). Circular layouts (chord diagrams) position set nodes along a circle to bring a spatial structure visually. To allow for richer set exploration on such diagrams, RadialSets [3] is based on an interactive circular layout with degree histograms on the set nodes, and uses edges to represent intersections of two or more sets. RadialSets is included in our focused comparison. The design of AggreSet follows previous studies that have shown that when graphs (connected entities) are bigger than twenty nodes, matrix-based visualization performs better than node-link diagrams on many tasks [50].

Matrix-Based Diagrams: A matrix layout is made of rows and columns that list values of a data type. Co-occurrence matrices use the set list on both axes, and cells show set pair intersections. Intersections metrics, such as element count, are commonly visualized using color (*heatmaps*). The resulting visualizations are non-intersecting and easy to read. However, such matrices hide information about higher-order set intersections [87]. AggreSet improves on the set-matrix design with its interactive, multi-dimensional approach. Matrix-based diagrams can also be built using different data dimensions for rows and columns. ConSet [76] uses a matrix with rows from elements and columns from sets. Since elements are not aggregated, its matrix view is not scalable by element count. Among the other approaches, UpSet [88] and OnSet [121] are discussed in a focused comparison (Section 6.6).

2.4 *Help and Documentation Systems*

As using new or rich interfaces can be a demanding task for users with a variety of backgrounds, the design of effective help systems and documentation is an integral part of human-computer interaction research. In this section, we summarize the motivating related work, existing approaches, and the differences of our contributions.

2.4.1 Motivating Work

The principles of minimalist documentation [27] motivates the design and contributions of our work: (i) getting started fast, (ii) training on real tasks (and real data), (iii) reading in any order, (iv) coordinating system and training, (v) using the situation (context), (vi) exploiting prior knowledge, and (vii) supporting reasoning and improvisation. While our implementation also aims to (viii) support error recognition and recovery, as well as (ix) develop optimal training designs, we do not claim contributions on these principles. Our design and contributions also reflect Caldwell and White's help-system design goals [25] of navigability, consistency, relevance, coherence, conciseness, reuse, and fidelity, while we do not aim to guarantee completeness. Earlier studies have empirically shown that physical and temporal separation of information sources undermine learning, i.e. the split-attention effect [51]. Our design of HelpIn enables rapid switching between consulting help and using the interface (analyzing data) [8], while avoiding interference with the main interface use, and remaining unobtrusive while the user focuses on the original task [127]. We aim to guide the user through complex operations by demonstration in the context of the user's own interface [57]. Our integration of help into the data interface also reflects the guideline of *showing instead of telling* [10], and advances the state of the art in visual data representations to support contextual and integrated help.

2.4.2 Basic Interactive Techniques

Help topic indices are commonly used to offer alphabetical, hierarchical, and search-based access to help. However, empirical studies suggest that users often avoid using both paper *and* online help manuals, and are frustrated by navigation, terms of indexing, and level of explanations [106]. As a common UI pattern, *tooltips* (callouts) are simple snippets that offer brief information next to a UI component on demand (such as on mouse hover). However, they generally present static (non-contextual) descriptive information, and they are not indexed for navigation. *Guided tours* use a sequence of tooltips as a fixed, step-by-step introduction to various interface components and tasks; however, they cannot provide help on-demand and on targeted questions. *Overlays* with multiple tooltips can describe multiple components at once (for example [160]). A multi-layered approach [73] can structure help material from simple (on first-use) to complex (on continued use). In a similar fashion, training wheel strategy [29] blocks complex actions and error states on introductory use. *Automated wizards* aim to complete specific tasks on behalf of the user with minimal interruption. This contrasts with teaching how to carry out data analysis under different datasets and a rich range of configurations.

2.4.3 Video-based Training

Videos can introduce multiple interface features in a recorded sequence, often using voice-over explanations. The research on video-based training commonly aims to allow navigation by video-content. In order to provide a content-annotated timeline, Tools cape [77] uses crowdsourcing to extract annotations, and Waken [11] identifies events and interface components by image processing. Nguyen and Lie [103] propose controlling the video playback by making the videos partially interactive *within the captured video frame*, while Pongnumkull et al. [113] propose synchronizing a tutorial video to a live interface when the user aims to achieve the same task on video. However, videos fundamentally present a *fixed linear flow* using *static material* that cannot adjust to active application. Users can become disengaged in a video training for reasons including long segments, abstract conceptual information, inconsistencies within and compared to other documentation, or extensive zooming [112]. Future changes in interface design can outdate existing videos. Therefore, producing and maintaining high-quality videos remains demanding, and video materials are limited in supporting integrated and contextual help.

2.4.4 Context-aware Help Systems

AmbientHelp [98] uses a secondary monitor to continuously and ambiently present help material (videos and manuals) outside of the primary work monitor, with relevance detected using most recent user actions. Targeting web-search applications, Ekstrand et al. [41] propose context-profiles including recently used tools, actions, and open interface components. HelpIn, on the other hand, provides descriptions of data elements with an interpretation of actual live data. Myers et al. [102] focus on answering why and why-not questions in user interfaces. Their query model can extract topics from pointed elements or recent actions, and present answers with textual description and relevant interface components highlighted. Yeh et al. [158] use screenshots to overlay the help on the interface directly. However, image-targeting rules can result in false positives/negatives, and is not robust to changes in interface design. Also, this system cannot be aware of the full application state or underlying data, or control the application. A key distinguishing element of HelpIn is that it provides descriptions of live data with explanations of how to interpret and act on that data in context of the data interface itself.

2.4.5 Help and Training for Visual Data Interfaces

Existing studies on visualization help commonly focus on providing training for a single visualization design or technique. Recently, Kwon and Lee [83] studied the effectiveness of different learning approaches for scatterplots (static, video, and interactive). Other recent approaches include converting visualizations to natural language descriptions of data features and potential insights, such as recent *Wordsmith* [161] and *Narratives* [162] tools developed for dashboards created with Tableau software. While HelpIn also features customized narrations, these come in response to help seeking rather than detecting and presenting potential insights. Our method also enables finding relevant help topics rather than insights. To our knowledge, there is no comprehensive, integrated, and responsive help system developed for rich visual data interfaces as in the scope of our work.

Closely tied to help and training, literacy and knowledge have received attention in visual data analytics community. For assessing visualization literacy, Boy et al. [21] propose a principled approach based on Item Response Theory. In the Cognitive Exploration Framework [153], knowledge is modeled to influence cognitive activities in visual data exploration, as a dynamic construct that can be extended with new knowledge of data and of the application over use. These discussions on visual literacy and sensemaking further motivate our work towards improving help for data interfaces.

2.5 *Visualization Design for Dense Numeric Data*

Increasing data density is among Tufte's visualization guidelines [141]. Another goal of effective visualization design is graphical perception accuracy, requiring a careful design process, and evaluation of alternative designs.

Fekete et al. [45] demonstrated the use of treemaps to visualize up to a million records on large screens. Under such settings, many records occupy a few pixels, and the visualization primarily supports perceiving overviews of record groups, and comparison across records with larger size. In this thesis, the aim is high legibility of each value in the chart, thus avoiding large data scales in a limited chart area. Kong et al. [81] compared the perceptual performance of treemaps to *single-column* bar charts *in a hierarchical setting* with up to 8,000 records at the leaf-level in a 600x400 pixel chart size. They reported, "As data density increases, treemaps become faster than bar charts while exhibiting equivalent accuracy." This effect may be due to the tiny size of *single-column* bars at dense displays that makes them harder to observe, which could be mitigated by using *multiple columns*. Their study did not consider the use of treemaps in a non-hierarchical setting, and tasks for data overview. Therefore, the study in this thesis contrasts to existing studies with its motivation, data types, and inclusion of visual overview and ranking tasks.

Among the techniques for dense information visualization, *horizon charts* [46] display time-series data in a compact chart height using a refined filled line chart. They divide the numeric data axis into equal sized bands, and collapse the bands while adjusting the color darkness per band. The chart height is reduced in the order of the number of bands while trends can still be observed. Heer et al. [63] studied perception of horizon charts and identified the effect of banding and chart height on estimation accuracy and speed. Javed et al. [70] discussed design alternatives to visualize multiple time series in a limited area, including braided charts , and assessed perceptual performance with lab experiments. Fuchs et al. [48] evaluated alternative glyph designs for time series data in small multiple settings, where each glyphs represents dense temporal data.

Evaluating the graphical perception of visualization design has a long history in the field of statistical graphics. The comparison task used by Cleveland and McGill in 1984 [32] has become an established method to assess graphical perception. Talbot et al. [138] extended their results on bar chart perception to better understand the reasons for performance differences across aligned and nonaligned bars, and the effects of separation and distracting bars. Perceptual studies have been extended to a crowdsourced methodology by Heer et al. [61]. Their results were aligned with results in lab settings, albeit with more variance. The uncontrollable display size and viewing distance across crowdsourced participants can be balanced by recruiting more participants from a wide online population than traditional lab settings with few participants. Following other recent studies [21], [134] and targeting casual use of the studied charts, we used online crowdsourcing for our graphical perception experiments.

Chapter 3. Cognitive Exploration Framework

"The best design gets out of the way between the viewer's brain and the content."

Edward Tufte

We begin the contributions of this thesis by presenting the *Cognitive Exploration Framework* for visual data exploration, which offers a structured basis for understanding cognitive activities, design guidelines, and evaluating data analysis and exploration tools. The design of the visual data exploration model (Chapter 4) and its implementation *Keshif* (Chapter 5), the contextual help system *HelpIn* (Chapter 8), as well as our perceptual evaluation on dense visualizations of sorted numeric data (Chapter 9) builds upon the foundations of the cognitive activities and design guidelines proposed in this chapter.

In this chapter, we first describe the six orthogonal cognitive stages of visual data exploration, as well as three factors into these activities. We then present how the framework can be used to categorize a large number of design guidelines. Lastly, we propose an evaluation approach that focuses on cognitive barriers and activities, revealing challenges on all six stages of the framework, as well as opportunities for improving the design of tools. The results from our user study using this approach is presented in Section 7.1. This chapter concludes with discussions and remarks on how the framework was constructed, implications of the framework for future design guidelines, our reflections on the process and the results on our evaluation, and effort levels across cognitive stages.

3.1 The Framework

We present the *Cognitive Exploration Framework* (CEF in short) (Figure 1), which identifies six cognitive stages in visual data exploration as a combination of two activities—planning and assessing—across data analysis, visualization, and interaction. Cognitive barriers are impediments that can be observed, categorized and studied across these orthogonal cognitive stages. In addition, the framework identifies the factors of decision costs, existing knowledge and motivation, which interact with cognitive stages to influence the exploratory process and outcomes.

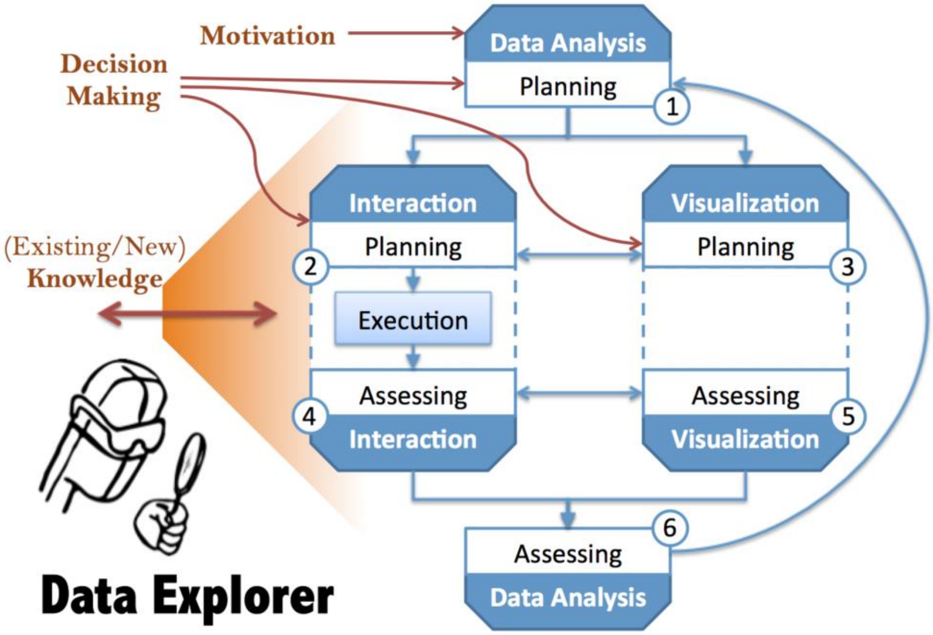


Figure 1- The *cognitive exploration framework* with six stages (shown within blue boxes) and three factors: decision-making, motivation, and existing/new knowledge (shown in red text).

3.1.1 The Six Cognitive Stages

We describe the cognitive stages using arguments in existing literature below, and show them in exploratory flow in Figure 1.

1. Planning Data Analysis: Form goals [29], determine domain parameters [1], characterize task and data [36].

2. Planning Interaction: Form system operations [29], translate queries to attributes [14], execute appropriate interactions [28].

3. Planning Visualization: Design visual mappings / encodings [36] [14], choose appropriate views [28].

4. Assessing Interaction: Evaluate state-change [29], adapt mental model to views [28], the gulf of evaluation [37].

5. Assessing Visualization: Perceive / interpret visualizations [28], visual-cluttering and view-change costs [29].

6. Assessing Data Analysis: Reason about outcomes, observe trends, generate hypotheses, make predictions, assess uncertainty [1], and build confidence.

The framework defines *visualization* as the purposefully organized representation of data in an abstract visual language. *Interaction* is the communication between the data and the explorer through the data interface. It encompasses all elements beyond the visual data encoding, such as control panels, buttons, and multiple views. Therefore, in the framework, the notion of visualization strictly relates to the visual representation of data, and does not cover any interactivity.

In terms of activities in data exploration, CEF identifies two activity groups—*planning* and *assessing*—that apply across data analysis, interaction, and visualization. Planning activities involve consciously setting goals, making decisions, and identifying courses of individual actions to be taken to reach goals. Assessment activities evaluate the courses of actions taken, data visualizations (through perception), the changes in the interface, and also include reasoning on whether the analytical goals have been answered based on available data, or not. The *Cognitive Exploration Framework* models execution, such as by mouse or touch, as a physical, non-cognitive stage that follows planning interaction, and leads to cognitive assessment stages. It is therefore left out of the scope of cognitive analysis.

In Cognitive Exploration Framework, exploration flows from data analysis planning to analytical data assessment to generate knowledge (insights). This is a cyclic and dynamic flow, i.e. exploration can continue with new paths influenced by insights obtained. If a path does not lead to knowledge, or if the explorer is stuck, s/he may retreat to produce new plans or change goals, although time would be lost and motivation may be reduced. The explorer may also act without a purposeful plan, such as selecting a data subset out of curiosity, and reach insights by observing relations revealed by these actions. Therefore, while the path ideally starts with a well-defined data analysis plan, we recognize it can also be driven by serendipitous interactions.

Next, we discuss three factors that influence the presented model of cognition.

3.1.2 The Factor of Decision Making

Increasing options in the exploratory process needs to be assessed not only by what they may enable (richer insights), but also by their cognitive costs. Given many options to choose from, making a decision is harder, and a decision is less likely to be optimal [125]. For example, finding the most effective visualization can be overwhelming given the combination of chart types, glyph types, color, and other visual encodings, especially for novices [54] but also for experienced designers [16].

Avoiding a decision also can be costly. Kobsa reported that Spotfire users tended to use scatter-plot, its default visualization, (therefore avoiding chart decision) when another chart would better fit [80].

Cognitive Exploration Framework generalizes decision costs in data exploration across all planning activities in visualization, interaction, and data analysis. We argue that the options faced in the process of exploration directly influence the decision costs and therefore the cognitive activities. While the examples given above relate to decision factors in visualization, decision-making also applies to data analysis (such as identifying which questions to follow, and which selections to make), and interaction (such as selecting across two alternative actions that may produce the same high-level outcome, or the sequence of actions). Every decision is likely to have a positive, or negative, outcome in the exploratory process. CEF recognizes and emphasizes the factor of decision making as a potential cost of the cognitive activities in the process of data exploration.

3.1.3 The Factor of Existing/New Knowledge

The explorer does not only process the data and its interface; s/he also has existing knowledge about the data domain, interface, and visualizations. This knowledge can help across all cognitive stages. For example, recalling personal experiences can help forming new queries, and assessing results in a broader context [86]. As the explorer gains more skills, the plans and assessments can improve. However, existing knowledge is limited, non-universal, and varying across people. In addition, knowledge is dynamic, i.e. there is learning during exploration and use of the tool. The explorer iteratively uses, builds, and evaluates knowledge constructs [79]. S/he does not only learn about the explored data, but also about the interface, interactions, and visualizations, which can lead to more effective use of the tool over time.

3.1.4 The Factor of Motivation

What are the driving forces of the explorer to engage in data exploration? *Cognitive Exploration Framework* identifies potential answers as the motivation factor. Motivation can follow the curiosity, such as to understand the data content and features. Being in the flow is another motivational construct. The flow—the balance between the challenge of a task and user skills—can apply within the context of interface use [13] and visual analysis [55]. Creativity is also motivating, and is applicable to data analysis (finding goals), interaction (combining features of the interface), and visualization (finding new forms to see new data perspectives). Emotions can also be motivating. Harrison et al. [59] found that emotion (affect) priming can influence perception of visualization. We propose that this result can apply to a wider range of activities in data exploration. Positive mood can increase motivation, and therefore exploration success.

3.2 *Design Guides from the Perspective of Cognitive Stages*

In visual data exploration, the data interface becomes the communicative channel between the cognition (mind) and the data. Supporting cognition (and reducing barriers) is therefore most related to the design of the tool interfaces rather than their computational models. In turn, what is the relation between design and the cognitive stages? How can the cognitive barriers be reduced by principled design? To answer these questions, we contribute a new categorization of 29 concrete and common design guides by linking them across six orthogonal stages of the *Cognitive Exploration Framework*. This section can be used to guide and improve the design of data exploration tools. The wide range of principles covered supports the rhetorical power of the CEF, which creates an orthogonal space for analyzing cognitive activities.

The selection of design guides is based on the existing practices and literature. Although we aimed to present a wide coverage and effective exemplars for each stage, offering a complete list of guides is impossible, and an extensive list is out of our scope. These guides should not be taken as rules of design, but rather directions to consider in designing tools to better support cognitive activities.

3.2.1 Guides for Planning Data Analysis

- Promote overview-to-detail exploration [129]. Starting with the data overview helps the explorer build a high-level mental model. Reveal detailed relations by interaction progressively.
- Show only relevant exploratory paths. Promote never-ending exploration [43]. Prevent queries leading to zero results [56]. *Systematic yet flexible discovery* [110] enumerates exploratory paths to suggest unexplored areas and communicate progress.
- Make exploration steps easily reversible [43]. This motivates action and reduces decision costs.
- Provide traces of exploration paths. To form new goals, the explorer may use action histories [64].

3.2.2 Guides for Planning Interaction

- Use direct manipulation [43], [130]. This reduces the cognitive distance between planning and execution through a continuous representation or metaphors of objects in the interface.
- Integrate interface with visualizations [43], [55]. This promotes visual coherence in a single immersive environment. Scented widgets [150] suggest designs on merging visualizations with interface elements such as dynamic

query widgets [128]. Legends can also be designed as interactive widgets [117].

- Show only relevant interaction options. Design to provide context; reveal interactions relevant to the selected object. Design based on the context; reveal contextual interfaces only when the explorer interacts with relevant object (e.g. show action icons on mouse-over).
- Indicate affordances of visual objects clearly [43]. Use visual cues to suggest interactivity [20].
- Design to fit the cognitive and conceptual model of the explorer. Allow searching for concrete data values, expose con-text of data attributes and their semantic relations, and support partial specification of exploration paths [54].
- Make every step useful and pleasing [43]. An action should not lead to a confusing, ineffective interface.

3.2.3 Guides for Planning Visualization

The primary means to support cognition in planning visualization is reducing the visualization parameters and options, starting with showing sensible defaults [64].

- Show only appropriate visualization options for the underlying data types and intended tasks. Recommendations may be a short list of suggestions; such as Tableau’s “show me” feature [95], which uses a rule-based design on selected attribute types, or a fully automated approach [120]. The context of use can also be considered [53].

- Support alternative visualizations to reveal relations that cannot be explored with existing views. Alternatives should be functional and add minimum decision costs. For example, given cities and their populations, an ordered list would reveal the cities with most/least populations, a histogram would reveal the population distribution, a map would reveal the spatial context, and a line chart would reveal temporal changes.

A common practice in visualization design is *templating*, in which the explorer selects a chart type first, and then decides which attributes to map to template parameters: axes, color, size, and so on. However, using visualization templates can impede cognitive activities because they require the explorer to understand the template parameters to make effective mappings [54]. Thinking is restructured to the terms of the template parameters from the terms of exploratory goals, potentially creating a mismatch of mental representations. Templates can be richer than fixed chart types such as flexible shelf-based systems [136] that construct a parameterized visualization space. We argue that revealing systematic parameters of a visualization design space should not be the basis of constructing visualizations for exploration.

3.2.4 Guides for Assessing Interaction

- Make system status clearly visible [104].

- Link multiple views on interaction [118]. Having multiple views increases the cognitive load with more visual information to digest. Linking views reveals relations between data representations, and can improve mental models. Linking should be consistent and intuitive.
- Provide real-time feedback after interaction [43]. A visual feedback delay, as short as 500ms, can decrease exploration activity and data coverage [90].
- Animate transitions between interface states [43]. Avoid abrupt changes and provide a sense of direction.

3.2.5 Guides for Assessing Visualization

- Use effective visual encodings. Graphical perception studies [32] report how accurately and rapidly we perceive data graphics across different encodings.
- Use appropriate scales, grids, labels, legends [62].
- Aim to reduce visual complexity. Avoid overlapping glyphs since they are a basic form of visual complexity.
- Avoid duplicate representations. Duplication of the same data point may increase cognitive efforts, as it requires understanding relations across multiple glyphs of the same data. Each additional glyph also takes screen space, which is a limited re-source that should be carefully used.
- Aggregate data, when it cannot effectively fit in limited screen space, and to provide overviews.

- Show conceptual data domain. For example, use matching icons (as glyphs or isographs) and matching colors for categories [126] where appropriate. Show uncertainty [5] when data has an uncertainty measure.
- Animate transitions of data glyphs [65].
- Use available screen space effectively. Adapt the visualizations based on display size.

3.2.6 Guides for Assessing Data Analysis

- Provide multiple views (perspectives) of data [55], [118]. One visual representation cannot show all aspects of rich data. Simultaneously observing multiple views can reveal relations across individual views.
- Provide analytical models for statistical analysis. Without tool support, the explorer may not be able to accurately evaluate their findings using statistical methods such as hypothesis testing with significance [5].
- Show the semantic context of data [54], such as description of data attributes, categories, and data values.

3.2.7 Guides across Cognitive Stages

- Aim for consistency. Inconsistencies in visualization, interaction, or interface design make it harder to form goals and action sequences, make decisions, perceive data, and the interface state. Therefore, consistency can influence both planning and assessing stages across multiple artifacts.

- Aim for minimalism. Make design as little as possible [97], [141], [163]. Showing only relevant paths and options in context of active state is a form of minimalism, which can support cognition for planning. Minimalism can also present complex systems as having fewer components that are easier to evaluate, thus supporting cognition for assessment.

3.3 *An Evaluation Approach to Detect Cognitive Barriers and Activities*

The success of data exploration depends on cognitive activities, and the cognitive barriers faced within these activities. The goal of the proposed evaluation is to better understand the behavior of the analyst/explorer, and to use this understanding to reduce barriers by improvements in design. In this section, we discuss how cognitive activities can be observed per each stage in evaluating an exploration tool, and how the framework provides a high-level structure to this evaluation. The goal is not to describe evaluation of a specific design guide, or a single stage of cognition, such as visualization perception, which require different setups. We don’t aim to present new guidelines, or a comprehensive analysis of an existing tool. Rather, we present a new evaluation approach, which can be considered as a specialization of usability testing, as a lens that focuses on and reveals barriers to cognitive activities.

We argue that detecting cognitive barriers requires focusing on failures, such as lack of goals, not being-in-flow, ineffective plans, and invalid insights. This is in contrast to the common practice of searching for success stories of our tools. Using benchmark tasks on fixed datasets does not facilitate autonomous, self-driven exploration. Furthermore, it may fail to motivate participants with a wide range of interests and background, or alienate them. We suggest that, the participants should express their interests in selecting data domain and their exploration goals, in order to improve their motivation and success. Furthermore, to expose all cognitive activities clearly, participants should be encouraged to interact with the tool directly without guidance by the facilitator. In contrast, usability studies commonly focus on physical execution problems and surface-level software use activities with pre-defined, benchmark tasks. Their goal does not include revealing the cognitive processes of the user. To summarize, the proposed study protocol aims to position participants as explorers, aiming to discover meaningful data-driven knowledge in an open-ended setting to answer their own questions based on their interests.

Revealing cognitive activities in depth requires moving beyond basic observations. For example, the explorer may want to sort a list alphabetically, interact with various interface components to find this feature, and then give up and change her goal. Detecting such a process as a negative outcome is instrumental to under-standing cognitive activities, especially when such tasks may not be explicitly enumerated. How can such failed actions and goals be observed by the analyst or some algorithm? Software logs [58], eye tracking [135], and brain scans [6] have some, yet limited, power in describing reasoning and exploration processes. Alternatively, encouraging verbal communication and analyzing the discourse can allow observing parts of the cognitive processes [44].

As the basis of the proposed protocol, we suggest that cognitive activities can be revealed with the facilitator observing the exploration process for potential challenges, asking for clarifications, prompting for more communication based on exploratory stages and reasoning behind actions of the participant. These interventions should be minimal and focused on cognitive activities, not a test of knowledge or a measure of success. Surveys and others forms of external cognition can also facilitate communication of cognitive processes. Our position is that, taken together, observations, interventions, surveys and external cognitive methods can lead to identification of a rich set of cognitive activities in data exploration.

We applied this suggested protocol to the evaluation of Keshif, the tool presented within this thesis. The results are presented and discussed in Section 7.1.

3.4 Discussion on the Cognitive Exploration Framework

3.4.1 Construction of the Framework

We presented the *Cognitive Exploration Framework* to provide a comprehensive overview of cognitive activities, the role of design in cognition, and how barriers to cognition can be the focus of evaluation of tools. To construct the framework, we iteratively identified and refined various arguments about cognition and barriers in related literature (see Section 2) as well as my own experiences in evaluation and design. For example, the gulf of execution and evaluation [104] models physical or lower-level cognitive activities, while Lam [85] focuses on interaction-related usability problems, which are integrated to our framework after separating physical execution stages. The framework is further enriched and supported by other arguments such as positioning of analytical gaps and activities [5], and results from empirical user studies [16], [54], [84]. Overall, we had noticed similar themes across taxonomies and empirical studies stated in different perspectives. We hope that the six-stage orthogonal overview of cognitive activities of CEF and its relation to design and evaluation will provide a concrete, lean basis to understand and improve how we cognitively explore and analyze data.

3.4.2 Implications for Design Guides

The overview of design guides suggests that existing literature provides many guidelines and discussions for interaction and visualization design. However, high-level data analysis and planning are cognitive activities with further opportunities for more results and guidelines with new focused studies. One of the challenges is identifying *how* people reason about data, and plan for data analysis steps. Another challenge is evaluating high-level outcomes of exploration and cognitive planning activities. Equipped with better models for cognition and evaluation methods that expose new metrics and processes, new improvements and guides may be made achieved. The results and examples from our user evaluation support that high-level cognitive activities can be analyzed qualitatively by observing failures in user behavior and verbal feedback. The framework can be used to target and analyze specific cognitive stages to propose new guidelines or experimental studies.

3.4.3 Reflections on Cognitive Evaluation of Exploratory Tools

To detect cognitive barriers, we designed and ran a user experiment (Section 7.1) with an open-ended exploratory setting, allowed the participants to choose a dataset and exploratory goals of their interest to increase motivation, and applied brief interruptions to encourage the participants to communicate their exploratory process and their negative emotions/experiences in a safe environment. While insight-based methodologies [122] focus on the success stories to quantify the observed value of a tool, a principled way to understand *failures* reveal opportunities for improvement. Our evaluation is a reflection of the open-ended data exploration approach, aiming for the unknown and intangible in the process of exploratory cognition and generating qualitative, rather than quantitative, value. We have shown that the *Cognitive Exploration Framework* can be applied in practice to detect and categorize observed barriers on cognition effectively, although we did not create CEF on empirical results from the particular study reported in this dissertation.

The proposed study design can be replicated or modified to study cognition in more depth. While we used think-aloud protocol and discourse analysis along with actions observed in video and notes taken by the facilitator, this approach has its own limitations, especially for achieving comprehensiveness. This qualitative analysis can be coupled with other forms of behavior tracking, such as software logs and eye movements, to add quantitative support for detecting cognitive activities. Using pair analytics protocol [9], the cognitive stages can be distributed across subject matter expert (high-level cognition in data analysis) and visual analytics expert (low-level cognition in interaction and visualization).

In retrospect, we observed that the participants rarely used the cards, one of the strategies employed to encourage communication of cognitive stages, to express their emotional and exploration state. While external anchoring may be beneficial to reveal more activities, the participants were either immersed in their data exploration, or not paying attention to the cards that were displayed on the table next to the study laptop. Embedding these feedback mechanisms on the interface of the tool may make them more prominent. The benefit of such external mechanisms can be studied further to detect if they lead to more communication. As the study included a small number of participants, we used the survey as a way to collect more feedback from the participants rather than to build a semi-quantitative analysis. Selected quotes reported in Section 7.1 include feedback during the completion of post-exploration survey. We suggest the use of post-exploration surveys to create opportunities to gather more feedback about the experience of the participants.

Since our goal was to find exemplar barriers in this preliminary study, we did not fully transcribe the sessions, which require higher effort and resources. Having more participants, full transcriptions, and multiple passes over the recorded material may reveal more cognitive activities in the use of a studied tool.

3.4.4 Effort Differences across Cognitive Stages

Do all cognitive stages require the same mental effort? Daniel Kahneman [72] argues that our cognitive activities are two-folded: system-1 (thinking fast) and system-2 (thinking slow). System-1 is how we make quick decisions, take short-cuts, apply our cognitive biases, etc. It is less deliberate and more spontaneous. System-2 is how we engage in a more effortful thinking, be more analytical, evaluate facts, and even actions of system-1. We argue that the stages of planning and assessing data analysis requires higher cognitive efforts as a slow thinking activity, and that fast thinking activities include perception of visualizations, evaluation of interface and planning for low-level actions respectively. Future research may investigate the differences of effort in cognitive activities under various settings.

3.5 *Outline of the Thesis*

The structure and motivation for the rest of this thesis is also supported by the Cognitive Exploration Framework (Figure 2). The aggregate summaries and linked selections model (Chapter 4) aims to support rapid tabular data exploration by reducing decision-making costs by presenting a minimal visualization and interaction basis. Its implementation and extension Keshif (Chapter 5) is also designed to follow many design guidelines mentioned in Section 3.2 to lower barriers in assessing interaction and visualization. We present interaction and visualization strategies on set-typed data in AggreSet (Chapter 6). Next, We focus on the knowledge component, as related to interaction planning and assessment, and propose a contextual help system for visual data interfaces (Chapter 8). Last, we focus on perception (assessing visualization) for dense numeric data, and present a new chart design, Piled Bars, and detailed evaluation across alternative designs (Chapter 9).

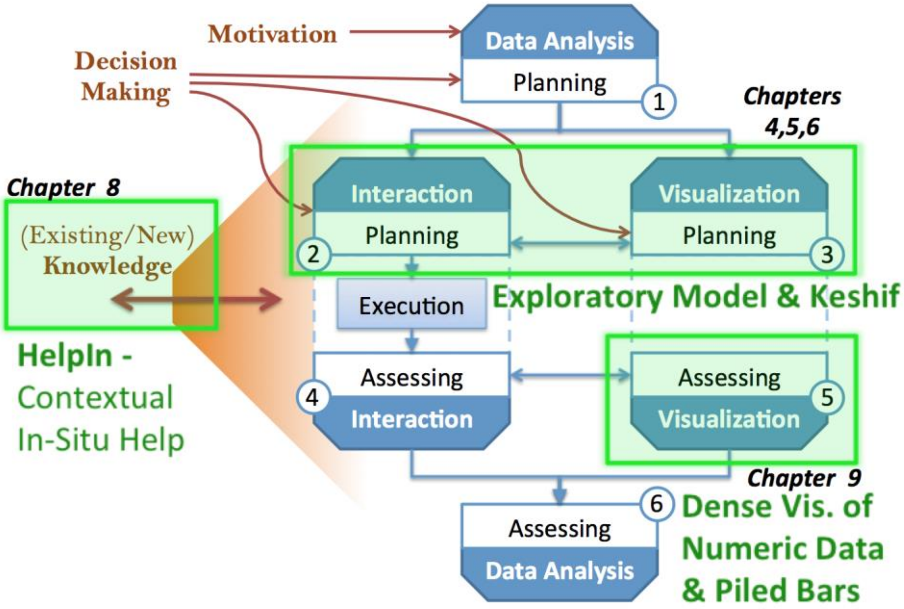


Figure 2- The outline of the thesis based on the *Cognitive Exploration Framework*.

Chapter 4. Aggregate Summaries and Linked Selection Model for Visual and Interactive Data Exploration

"Design is the conscious effort to impose a meaningful order."

Victor Papanek

in "Design for the Real World: Human Ecology and Social Change" [108]

To streamline and unify the visualization authoring and data exploration workflow for tabular data, we propose the aggregate summaries and linked selection model (Figure 2). This model provides a minimal yet expressive design basis to enable rapid visual and interactive data exploration. Data record attributes are summarized by aggregating records and measuring group characteristics. The visualization design of aggregates is based on the attribute data type (Table 1), and support *absolute* and *part-of-active* scale encoding of measured aggregate characteristics. This model reduces the search space for choosing visual data encodings by automating visual representations based on data type and semantics using perceptually effective, non-overlapping visual encodings. Thus, the user makes fewer decisions on data representation compared to visualization design environments, leaving more cognitive resources to reach data-driven insights, and reducing required visual analytics knowledge. The model defines an interactive overview-to-detail flow for visual exploration using three linked selection interactions: (i) *highlighting* (rapidly previewing record groups), (ii) *filtering* (focusing on a record group), and (iii) *comparison* (locking selection of record groups). Despite its minimalism, the model is expressive (enables rich data exploration) by its applicability to multiple common data types (categorical, numerical, temporal, and spatial (Table 1)), and its support for measure functions for aggregates (count, sum, average) and visual scale modes (absolute, part-of). The model achieves scalability in record count by explicit aggregation, and its minimalism enables rapid learning.

The data model is designed for common tabular data: records with attributes (categorical or interval). Categorical data may be single or multi valued (set-typed) [33], and may describe spatial regions. Interval data may be numeric or timestamp. New attributes can be calculated per record using existing attributes, such as to split/parse text values and to compute weighted averages from a list of numeric attributes.

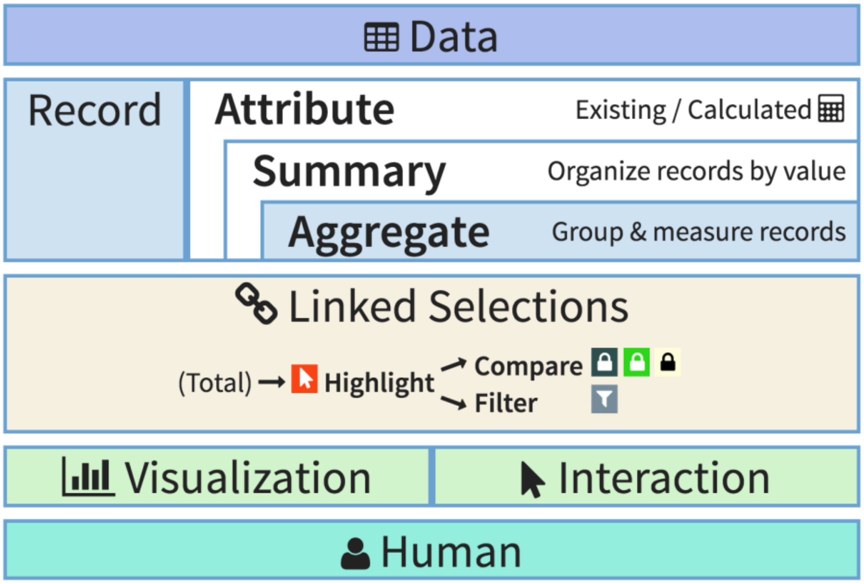


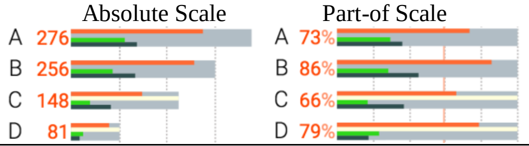
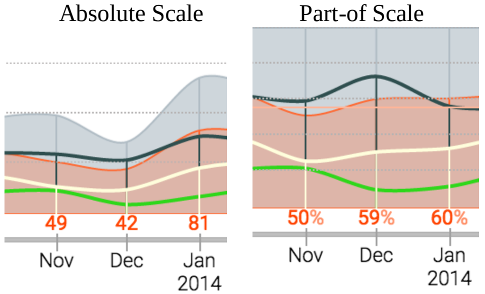
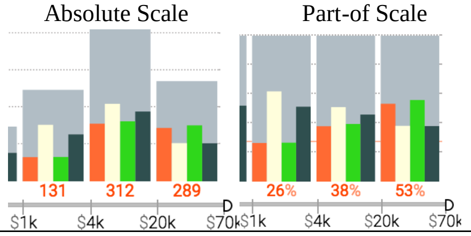
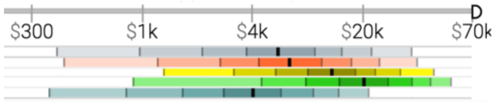
Figure 3- The aggregate summaries and linked selection model creates a data↔human interface. Data consists of records with attributes. Attributes are summarized to aggregates, which measure group characteristics. Three linked selection modes provide the exploratory dialogue with records and aggregates. Highlighting precedes filtering and comparison.

4.1 Aggregate Summaries Model

Given an attribute of a dataset, a ***summary*** extracts attribute values of all records, and ***aggregates*** records by their value, either as discrete categories or as interval (range) bins (Table 1). The ***aggregate measure metric*** computes a numeric characteristic of the aggregated record group, either (i) *count* (e.g. count of car accidents), (ii) *sum* (e.g. total injured people in accidents), or (iii) *average* (e.g. average car speed in accidents). *Count*, the default metric, provides a familiar faceted data overview [34]. *Sum* and *average* metrics use the record values of a chosen numeric attribute (e.g. the number of injured people, or the car speed). Median and percentile characteristics of a record group can be shown by the percentile aggregations on numeric summaries upon selection of the group. Thus, our model achieves expressiveness by revealing a wide range of group statistics on multiple attribute types.

4.2 Linked Selection Model

The model defines three selection interactions for three complementary tasks: highlighting, filtering, and comparison. **Highlighting** allows rapidly previewing characteristics of the records in the selected aggregate. **Filtering** focuses on records within the selected aggregate by removing the records outside of the selection. It is an explicit, permanent selection compared to the highlighting selection for preview. Filtering criteria can be refined incrementally using multiple summaries and selections. Group **comparison** allows comparing characteristics of multiple record groups side-by-side by locking a highlight selection. Without compare-selection, comparing distributions across multiple selections would require memorization over time and higher mental effort. Thus, compare-selection allows capturing and storing a selection state to facilitate group-wise and side-by-side comparison of records. In practice, we limit the number of compared selections to three in order to accommodate capabilities of human perception. To model the exploration process, highlight selection precedes (previews) all filtering and comparison selections. With the total selection representing all the records, our model allows exploration of distributions of *six* record groups concurrently.

Data Type	Glyph	Visualization
Category	Bar (Category)	
	Encoding	→ Length (Width)
	Position	Category order, next to category label
Time	Line (Interval range bin)	
	Encoding	↑ Length for measure value. ↔ Line connects bins. Area-fill for non-compare selections.
	Position	Interval Range
Number	Bar (Interval range bin)	
	Encoding	↑ Length (Height)
	Position	Interval Range
Percentile (Distribution)	Block (Percentile range)	 Distribution of a numerical attribute. Simple alternative to box-plots without visualization of outliers. Percentiles are independent of scale mode.
	Encoding	Color: Four fixed percentile ranges with 10% steps. Darker color towards the median (50%).
	Position	The percentile ranges of the selected records

(Table continues on the next page)

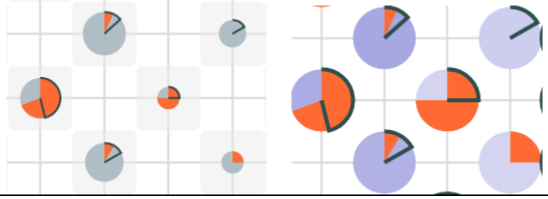
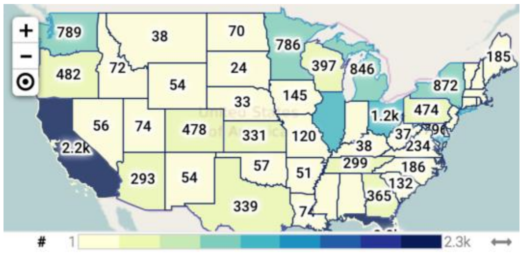
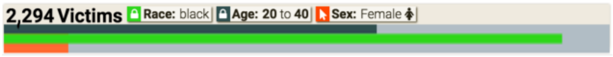
Data Type	Glyph	Visualization
Set Pair (Multi-Value Category)	Disc	Absolute Scale Part-of Scale 
	Encoding	Filtered: ◎Circular area. Highlighted: Arc area (0°-360°) Compared: Arc border (0°-360°) Total: None. Exists: Cell background color. Strength: Circle color (part-of scale). For details, see AggreSet [33].
	Position	Set-pair location on grid. Small glyph size.
Spatial Area	Region (Map)	 789 38 70 786 397 846 185 482 72 54 24 872 33 145 474 1.2k 56 74 478 331 120 37 96 2.2k 38 234 299 186 293 54 57 51 132 365 339 74 # 1 2.3k In part-of scale, color is scaled from 0% to the maximum % value of all (filtered) regions.
	Encoding	Color: [0 - max(distribution)]. Visualizes one distribution by color mapping. Default is filtered selection. Highlight-selection takes precedence when enabled.
	Position	Geographically defined. Fixed shape and size.
No-Value (Missing)	Icon	⊘ ⊘ ⊘ Aggregates records with no-value in summary.
	Encoding	Color (0-max(filtered))
	Position	Fixed (Lower-left corner of summary).
All Records (Global)	Bar (Full width)	 2,294 Victims Race: black Age: 20 to 40 Sex: Female
	Encoding	→ Length (Width)
	Position	Fixed (Top of the browser)

Table 1- Visual encodings for aggregations across multiple data types and selections. The visual encodings are designed to minimize overlaps, support accurate graphical perception, enable fluid interaction, and achieve scalability and consistency.

4.3 Visual Data Encoding

Measure Function	Measure Summary	Relation across selection values (distributions)	Measure Scale	
			Absolute (Shared scale in summary)	Part-of (Scale per aggregate)
Count	NA	Total ≥ Filtered Filtered ≥ Highlighted Filtered ≥ Compared	0->max(filtered value of aggregates in summary)	0->filtered value of aggregate. Presented in percentage (0-100%)
Sum (Total)	Numeric Attribute			
Average		NA	0->max(filtered/compared value of aggregates)	Not applicable, not well defined

Table 2- Properties of three measure functions and two measure scale modes.

Aggregates visualize measured values by color-coding the selected record distributions (Total ⬜, Filtered ▩, Highlighted 🟧, Compared ⬛ 🟩 ⬜) (Table 1). Measured values are visually encoded based on the aggregate glyph type, such as by length, color, or area on a quantitative scale with two alternatives: *absolute scale* and *part-of scale* (Table 2). Absolute scale constructs a scale that is shared across all aggregates in the summary. Part-of scale constructs a scale per-aggregate that encodes highlighted/compared measure value as percentage of filtered value. Comparisons are side-by-side along a shared axis. Filtered selection distribution is emphasized by setting the maximum range of the axis on filtered selection. Highlighted and compared values are within the scale limits when count and sum measure functions are used, as the subset of records measure less than the filtered set of records. However, this relation does not hold under average measure. In our design for this case, the measure scale is updated to cover values of compare selection, but not of highlight selection since frequent scale updates on rapid highlighting can be distracting.

Chapter 5. Keshif – The Implementation of The Exploration Model

*"Real artists ship." - **Steve Jobs***

Based on the proposed data exploration model, I implemented Keshif [1], an open-source, web-based data exploration tool for tabular data, available online at www.keshif.me. Raw data is visualized by authoring a Keshif browser (examples shown in Figure 3, Figure 4, Figure 5, Figure 6) by inserting attribute summaries, the record display (showing records individually), and calculating custom attributes. Data is then interactively explored through Keshif's unified, consistent linked selection model. To enable exploration of spatial records or self-referencing attributes (networks), the record display can show records on a geographical map or as a node-link diagram, in addition to list views. Summaries are further specialized on data semantics for tasks such as categorical sorting, flexible range selections, and navigation (scroll, pan, zoom), as summarized in Table 4. Keshif browsers are defined with a compact JSON-like configuration, which can be forked to enable collaboration. Browsers can be publicly shared on the web with a unique URL, or

[1] Keşif (keshif) means discovery and exploration in Turkish.

embedded into existing web pages using basic JavaScript and CSS programming, which also can be used to customize the browsers. As a result, Keshif provides an out-of-the-box tabular data exploration solution to enable rapid data exploration.

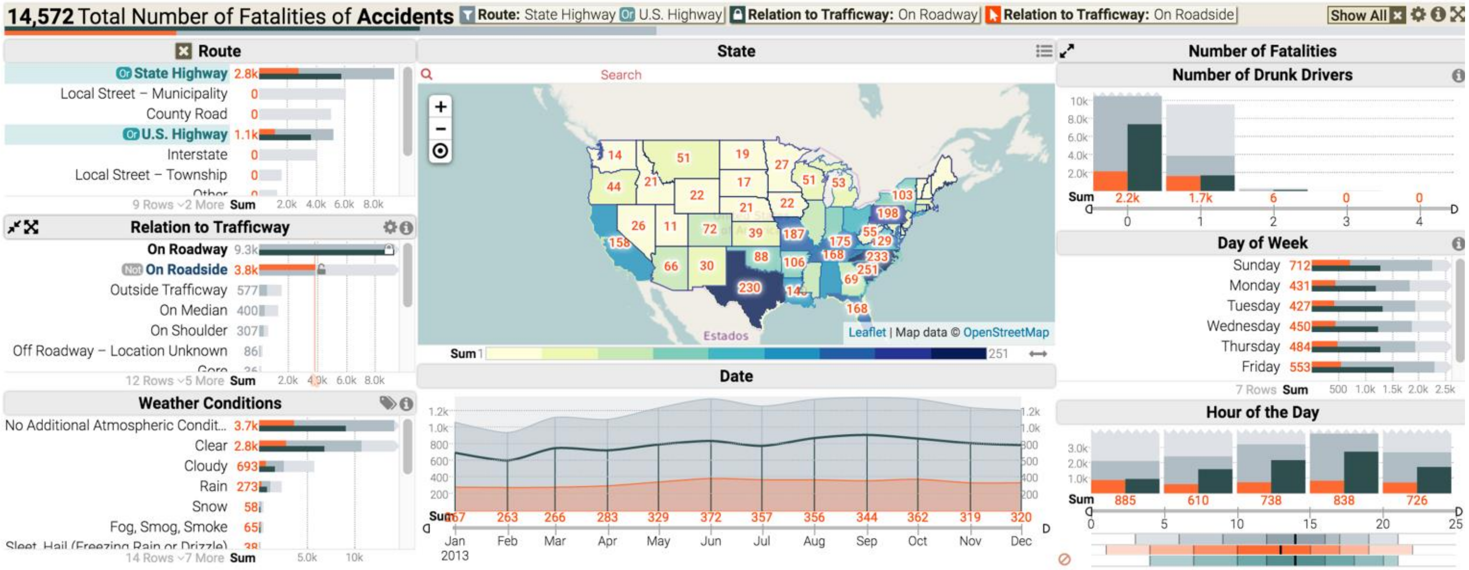
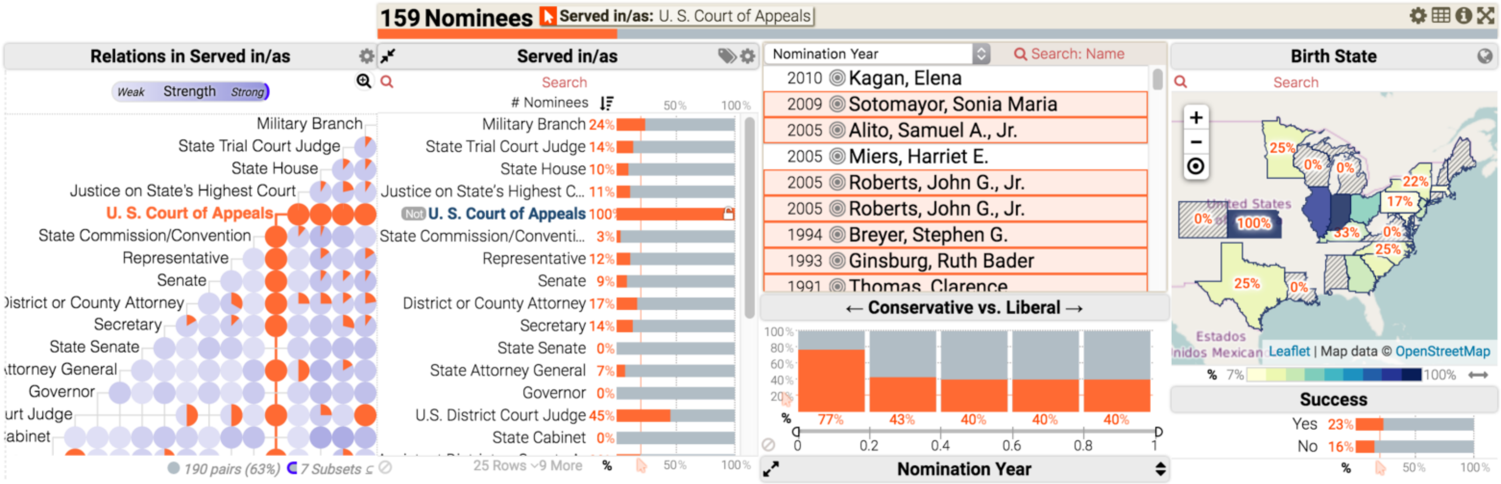


Figure 4- This Keshif browser enables exploration of fatal traffic accidents in 2013 in the United States. Selected attributes are summarized using data aggregations, measuring the total number of fatalities. Visualizations show data distributions of three linked selections (▼,🔒,↖), and are minimal and effective per attribute data type. This view shows bar, line, map, and percentile charts. Accidents on State Highway or U.S. Highway (route categories) are selected by filtering ▼. Roadway accidents are selected by locking 🔒, and roadside accidents are selected by highlighting ↖ on mouse-over. This browser and exploratory view can be easily and rapidly authored from raw data using the graphical interface; saved, shared, and embedded into existing web pages.



159 Nominees Served in/as: U. S. Court of Appeals

Relations in Served in/as

Served in/as

Search

Nominees

Military Branch 24%
State Trial Court Judge 14%
State House 10%
Justice on State's Highest C... 11%
Not U. S. Court of Appeals 100%
State Commission/Conventi... 3%
Representative 12%
Senate 9%
District or County Attorney 17%
Secretary 14%
State Senate 0%
State Attorney General 7%
Governor 0%
U.S. District Court Judge 45%
State Cabinet 0%

25 Rows ~9 More

190 pairs (63%) 7 Subsets

Nomination Year Search: Name

2010 Kagan, Elena
2009 Sotomayor, Sonia Maria
2005 Alito, Samuel A., Jr.
2005 Miers, Harriet E.
2005 Roberts, John G., Jr.
2005 Roberts, John G., Jr.
1994 Breyer, Stephen G.
1993 Ginsburg, Ruth Bader
1991 Thomas, Clarence

← Conservative vs. Liberal →

Nomination Year

Birth State

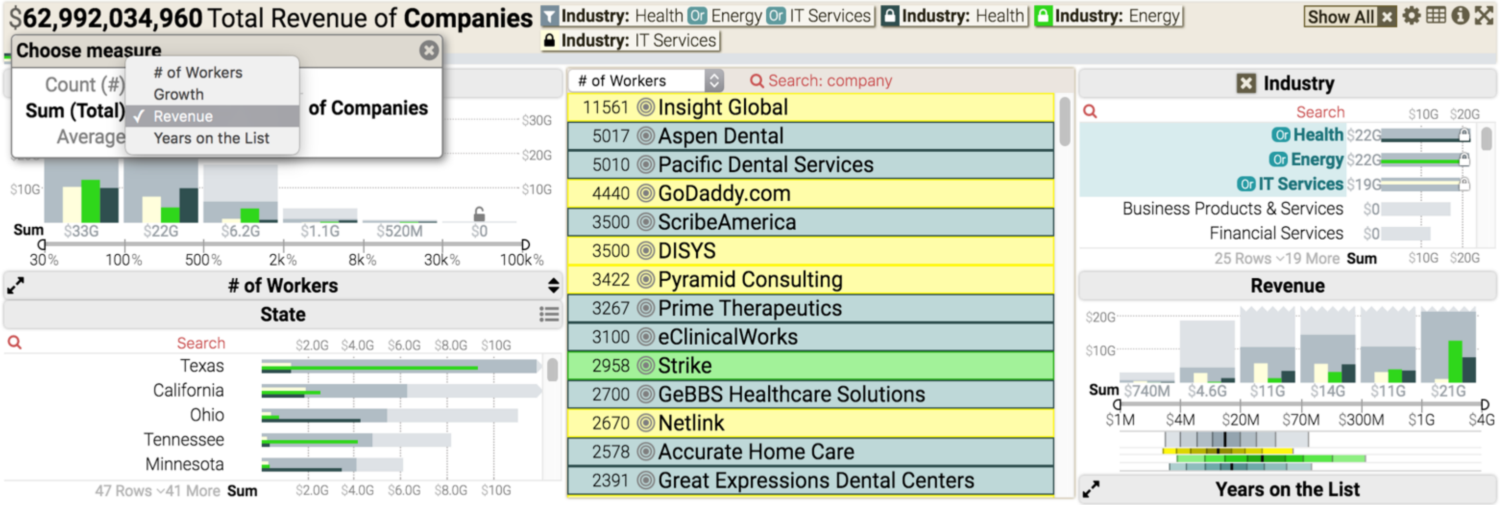
Search

Estados Unidos Mexicanos Leaflet | Map data © OpenStreetMap

Success

Yes 23%
No 16%

Figure 5- Keshif data browser showing the U.S. Supreme Court nominees. Scales show part-of (%) relations. The set matrix [33] shows relations across positions that the nominee served in before the nomination. Nominees that served in U.S. Court of Appeals are highlighted 🟥 by mouse-over. The map view uses color-coding to show the percentage of nominees served in the selected position among all candidates from that state. Some states do not have a nominee (gaps). Some states have none (0%) that served in selected position (dashed regions).



# of Workers	Search: company
11561	Insight Global
5017	Aspen Dental
5010	Pacific Dental Services
4440	GoDaddy.com
3500	ScribeAmerica
3500	DISYS
3422	Pyramid Consulting
3267	Prime Therapeutics
3100	eClinicalWorks
2958	Strike
2700	GeBBS Healthcare Solutions
2670	Netlink
2578	Accurate Home Care
2391	Great Expressions Dental Centers

Figure 6- Keshif data browser showing rapidly growing companies of 2014 by www.inc.com using three active compare-selections. The total revenue is the aggregate measure, selected using the popup panel on the top left corner. This panel is revealed upon clicking the global summary text area, a natural interaction. The companies are filtered ▼ on three industries: health, energy, IT services. Each industry is then selected for comparison. In the record display, companies that fall into respective industries (per compare selections) are automatically color-coded ■ 🟩 □.

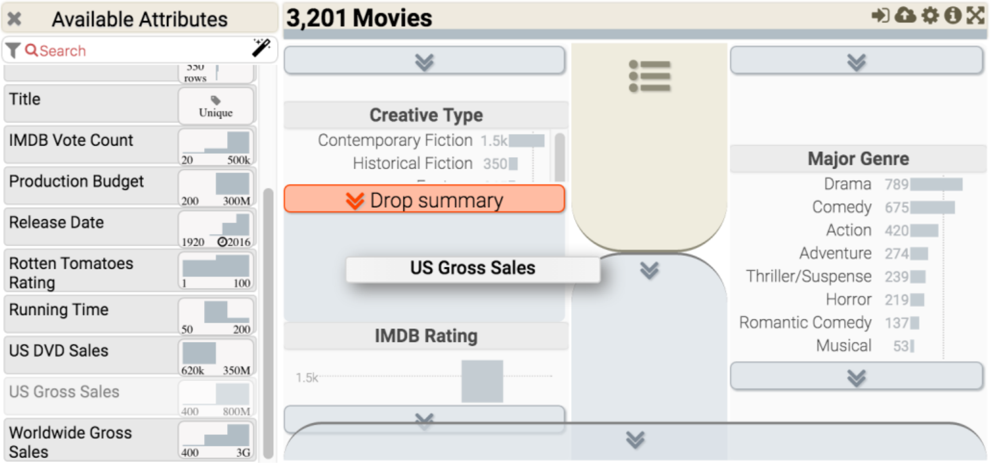


Figure 7- Browsers are authored using drag-and-drop from *available attributes* panel to create summaries in four browser panels (left, right, middle, bottom), or to list records individually. In this view, *US Gross Sales* is dragged and placed between *Creative Type* and *IMDB Rating* summaries. The browser layout is adjusted to reveal drop zones across all panels and between summaries.

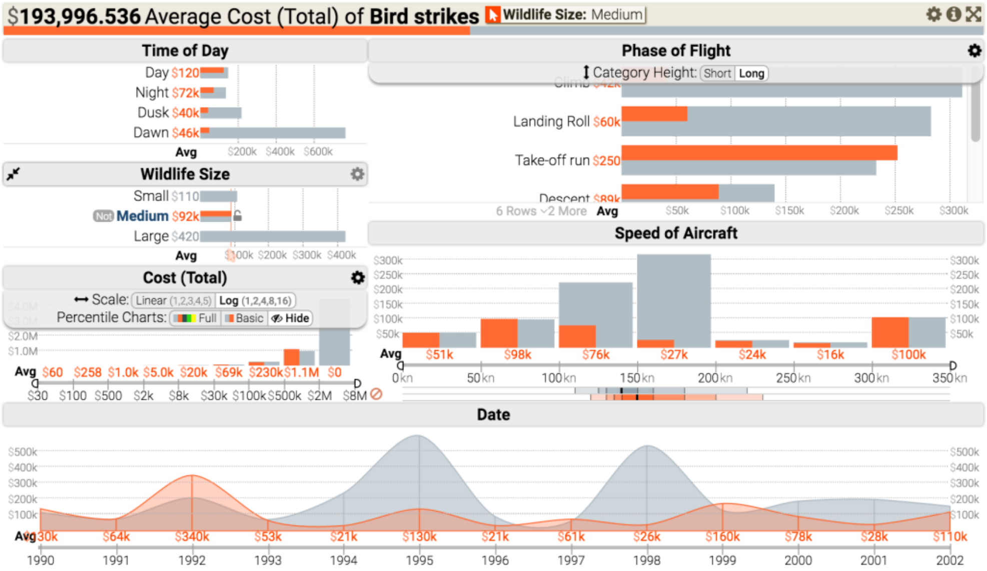
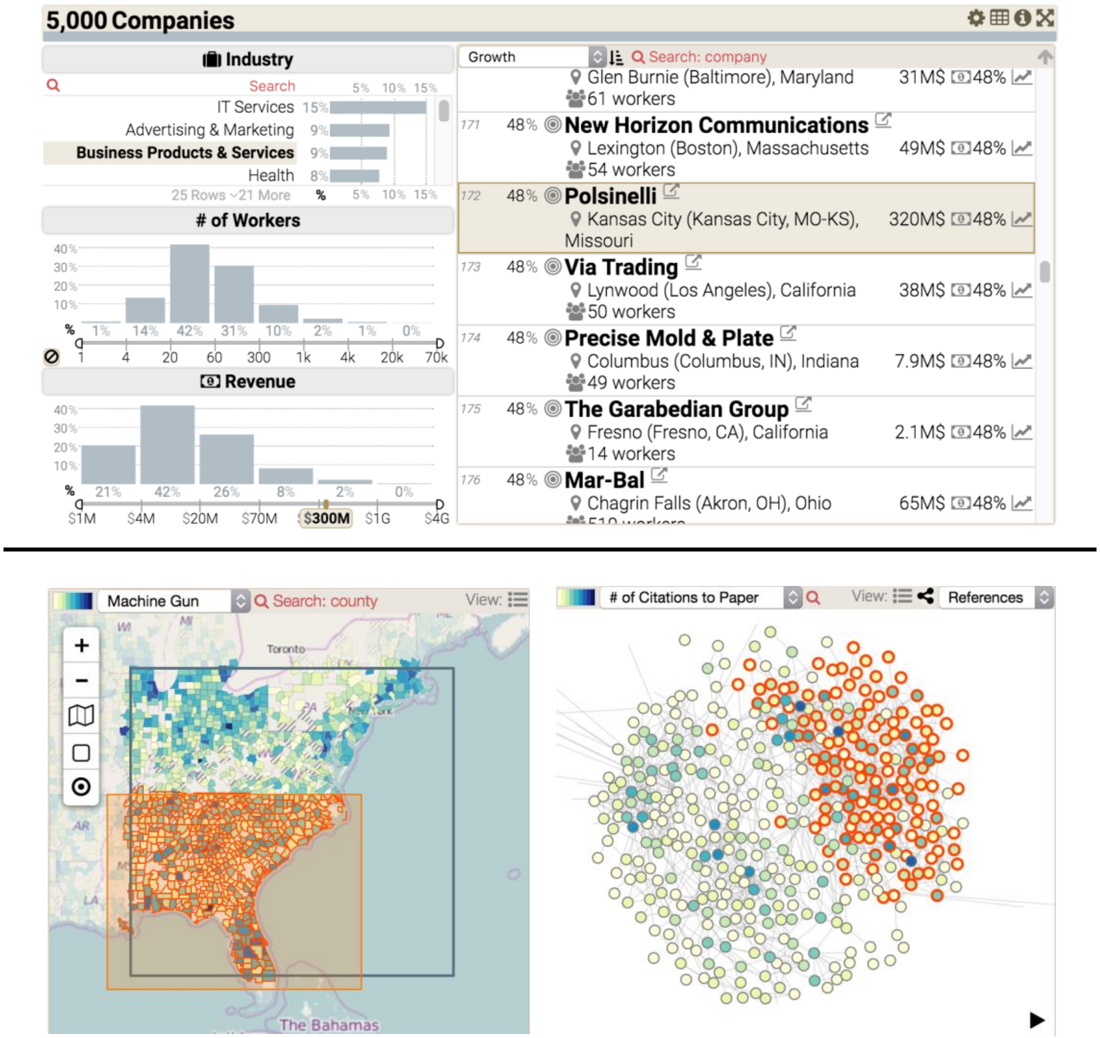

$193,996.536 Average Cost (Total) of Bird strikes
Wildlife Size: Medium
Time of Day
Day $120
Night $72k
Dusk $40k
Dawn $46k
Avg $200k $400k $600k
Wildlife Size
Small $110
Not Medium $92k
Large $420
Avg $100k $200k $300k $400k
Cost (Total)
Scale: Linear (1,2,3,4,5) Log (1,2,4,8,16)
Percentile Charts: Full Basic Hide
$2.0M
$1.0M
Avg $60 $258 $1.0k $5.0k $20k $69k $230k $1.1M $0
$30 $100 $500 $2k $8k $30k $100k $500k $2M $8M
Phase of Flight
Category Height: Short Long
Landing Roll $60k
Take-off run $250
Descent $89k
6 Rows 2 More Avg $50k $100k $150k $200k $250k $300k
Speed of Aircraft
$300k $250k $200k $150k $100k $50k
Avg $51k $98k $76k $27k $24k $16k $100k
0kn 50kn 100kn 150kn 200kn 250kn 300kn 350kn
Date
$500k $400k $300k $200k $100k
Avg $30k $64k $340k $53k $21k $130k $21k $61k $26k $160k $78k $28k $110k
1990 1991 1992 1993 1994 1995 1996 1997 1998 1999 2000 2001 2002

Figure 8- Exploring BirdStrikes dataset. The aggregate measure function is the *average* of *cost*. Medium size birds are highlighted 🟥. The highlight selection shows the average damage per each aggregate related to medium-size birds. The average cost is not steady over time, and chart reveals no correlation between all birds and medium size birds.



Glen Burnie (Baltimore), Maryland 31M$ 48%
61 workers

171 48% New Horizon Communications
Lexington (Boston), Massachusetts 49M$ 48%
54 workers

172 48% Polsinelli
Kansas City (Kansas City, MO-KS), Missouri 320M$ 48%

173 48% Via Trading
Lynwood (Los Angeles), California 38M$ 48%
50 workers

174 48% Precise Mold & Plate
Columbus (Columbus, IN), Indiana 7.9M$ 48%
49 workers

175 48% The Garabedian Group
Fresno (Fresno, CA), California 2.1M$ 48%
14 workers

176 48% Mar-Bal
Chagrin Falls (Akron, OH), Ohio 65M$ 48%

Figure 9-Alternative record display views. **Top**) List view with custom content and styling. Polsinelli (a record) is highlighted. Summaries on the left reveal its characteristics with consistent color use: Business Prod. & Services, unknown (∅) number of workers, $300M revenue. **Bottom Left**) Map view shows US counties and the number of machine guns they received from military. In the map view, records can be selected spatially into flexible aggregates. Counties within the black rectangle are selected by filtering (click+drag), and orange counties within orange box are selected by highlighting (shift+drag). **Bottom Right**) Node-link view based on citations between papers from the InfoVis conferences [13]. Node color shows the number of citations to the paper. Papers of InfoVis conference are highlighted with orange border.

5.1 *Data Browser with Record Display*

Keshif data browser builds the data exploration space around attribute summaries and a record display that shows records individually either as a list (row / grid), on a map if the records define spatial boundaries, or as a node-link diagram if the records are explicitly inter-connected (such as references across publications) (Figure 8). The visual encodings are summarized in Table 3. All attributes of a record can be viewed in a pop-up window by clicking ◎ in the list view, or the region/node in map/node-link views. The header panel summarizes the complete dataset and visualizes the selection characteristics using the global aggregate. All active selections are shown as breadcrumbs in the header panel, encoded by color and icons for a quickly accessible overview of the data selection (exploration) state.

Record View	**Organize by (numeric attribute)**	**Filtered-Out Records**	**Highlight / Compare Encoding**
List	Sort	Removed	Fill color
Map	Fill Color	Transparent	Border color
Node-Link	Fill Color	Removed	Border color

Table 3- The form and visual encoding used by the record display for visualizing individual records. List view is the default. Map view is supported if the records have a spatial component. Node-link view is supported if the records have an attribute that refers to other records. The form of record display can be switched during exploration.

The active aggregate measure function and the scale mode are shared across all summaries to provide a consistent interface. The controls are mapped to conceptual visual elements, aimed to minimize control-specific UI components. The aggregate measure is set by clicking to global aggregate (Figure 5); the scale mode is set by clicking on the measure scale axis. In contrast to Tableau [136] and Voyager [151] where *record count* is shown along record attributes, Keshif clearly distinguishes record count as an aggregate measure function. Measure labels can be shown in absolute or percentage values under *count* and *sum* measure functions. For example, an aggregate of 343 female employees among 870 (filtered) employees can be labeled as 39%, providing a quick percentage-overview of the record groups (Figure 8, Left). Clicking **# - %** icons on the chart corner changes this mode.

5.2 Design Specifics

This section presents specialization details of the layout (browser), visualization (summaries), and the interaction (linked selection) design of Keshif.

5.2.1 Layout Design

The Keshif browser layout is designed to avoid overlaps across summaries and the record display, and to simplify layout configuration. The browser defines four panels (left, right, middle, bottom) that can include multiple stacked summaries. The summary height is automatically distributed across all summaries in a panel. The individual summaries can be collapsed to their header (Figure 5, # of Workers), which opens more space for other summaries in the panel. The record display is positioned in the middle, perceptually binding selections across all summaries positioned around it. The browser header panel holds the global summary and selection breadcrumbs. This constrained design minimizes decisions on the layout and positioning to speed up data exploration.

Summary	Form	Navigation
Categorical	List	↕ Scroll (1D)
	Map	✣ Pan & Zoom (2D)
Interval	Histogram	🔍Zoom to filtered range 🔍Zoom to total range (Fine vs. coarse bins)
	Line	
Set-Pair	Matrix	✣ Pan & Zoom (2D)

Summary	Form	Specialization	
Categorical	List	**Sorting**: Automated re-sorting after filtering to emphasize most relevant first. Multiple sorting options, custom category ordering, and inverse sorting are supported.	♦ Label text search under many categories. ♦ Multiple logics for selection (**And**/ Or / Not). **And** is only applicable to multi-valued categorical attributes. See AggreSet [33] for details.
	Map	**Select** records by spatial query (rectangle)	
Interval	Histogram	♦ Linear/log scale binning, based on data distribution, can be changed in UI. ♦ Supports percentile chart. ♦ Supports unit names (10 mg, $100, etc.)	♦ Bin range is based on value range (min/max) and summary width. ♦ Flexible range queries to select records beyond fixed ranges. ♦ Filtered range is always visible.
	Line	♦ Only linear-scale binning (horizontal axis).	
Set-Pair	Matrix	♦ Visualizes set-pair strength & subset relations (design on data-semantic) ♦ Connected (next) to categorical list summary with synchronized scrolling navigation.	

Table 4- Specializations on summary and form types reflect data types, semantics, and tasks.

5.2.2 Attribute and Summaries by Data Type

The summary design is further specialized on data type and semantics as summarized. Table 4 presents an overview of the specializations. An attribute summary can support alternative data semantics by adjusting its visual form with the setting controlled by a button that demonstrates the context. In our implementation, categorical attributes that define spatial boundary definitions (such as countries) can be shown as a list ≡ (to emphasize sorted ranks), or on a map 🌐 (to emphasize spatial distributions). The icons in summary header allow controlling the mode. The relations in multi-value categorical attributes are revealed in set-pair matrix [154] using 🏷. By default, the percentile chart is not visible in numeric summaries to keep the interface minimal. It can be shown using the summary configuration pop-up panel (Figure 7 – Cost summary), which also allows adjusting the binning to linear or log scale if applicable.

Existing attributes of a raw dataset may need to be transformed or reformatted for effective representation and analysis. Keshif allows specifying calculated attributes as functions that return a new value given a record and its attributes. This provides a highly flexible customization pipeline to describe units of analysis, and can support pre-processing stages such as converting values (e.g. “10k” to 10,000 and “20M” to 20,000,000, i.e. strings to numbers). In Figure 3, the *Day of Week* summary is extracted from the *Date* attribute. In Figure 4, the services held by the nominees are combined to a simple list merged from multiple attributes, each of which define the location of a service if the service had been held. This allows summarizing the service types in a compact form instead of summarizing them individually. Calculated attributes can also be used to lookup/merge external tables. For example, in a publication browser, a calculated attribute for *Countries of Authors* can return the list of countries of all authors of a paper by a lookup on the author table that stores the author country. Calculated attributes also enable defining rich HTML markup for individual records in the record display (Figure 8, Left). Keshif interface also supports common data transformation shortcuts in its graphical interface, such as (i) extracting month, hour or week-of-day from a time attribute, (ii) extracting the set-degree from a set-typed attribute, and (iii) splitting categories into multiple values by tokenization, such as splitting “A;B;C” on “;” to generate the categorical list [“A”, “B”, “C”].

5.2.3 Pointer Based Linked Selection Design

Keshif implements a pointer (mouse) based interaction design for selecting records and record groups. Mouse-over on an aggregate ↖ sets the highlight-selection. Clicking on an aggregate ▼ sets the filter-selection, an explicit action compared to mouse-over. Compare-selection 🔒 can be set by clicking on 🔒 that appears on a highlighted aggregate. Alternatively, shift+click on a highlighted selection sets comparison as well, and enables comparison of aggregate designs that may not reveal a lock icon by design, such as no-value or map region aggregates. To enable flexible interval selections (beyond fixed bins), shift+mousemove, click+mousemove, and shift+mousemove+click along horizontal axis set highlight, filter, and compare selections respectively. The aggregate measure text label color also reflects which distribution it displays. Activating highlight selection sets the text label to orange. Mouse-over on a compare selection, on the breadcrumb or charts, updates the labels to show the values. All visual encoding transitions (such as length, color, and size changes) are animated. Categories are resorted with staged animations after filtering to show most frequent/relevant on top.

Selecting a record by mouse-over reveals its attribute values in all summaries (Figure 8, Left), a doubly linked selection that highlights the record on demand within the context of distributions of all records. In the node-link view, mouse-over selection of a record also highlights its neighboring records. To avoid unintentional triggering of highlight-selection (and visual flickering) on mouse-move across the screen, we added a delay that is linear to mouse speed, activated above a threshold. Slower, deliberate mouse moves immediately enable highlighting, while fast moves respond with a minor delay.

5.3 Authoring Data Browsers

Enabling out-of-the-box data exploration requires easily importing new datasets into the exploration environment. In Keshif, data browsers can be authored / created after importing a dataset using two approaches: using JavaScript API (which also serves as a storage/exchange format), or the graphical interface, which supports drag drop interaction. Authoring is designed as a mode that can be enabled during exploration, as well as after data import, so that exploration process can be enriched with modifying the data summaries within the browser.

Keshif, including its API, is primarily designed to let the user define *what* is being visualized and explored, not *how*. This is in contrast to grammars of visualization such as Vega Lite and ggplot, which have a compositional approach to create a range of chart designs. It also contrasts with chart templating approaches such Excel, Raw, Datamatic, and Quadrigram, since Keshif automates the visualizations and interaction, and the data dialogue is driven by the user based on key exploratory tasks rather than selecting charts and mapping data to template parameters.

Lastly, customizations of Keshif browsers are most commonly aimed to express metadata, such as *ordinal* categories and unit names of numeric attributes (such as km, or $), as well as basic data transformations such as parsing time components from a text field, and splitting a text field into multiple categories by a delimiter. The API currently does not aim to store exploration state, such as specific selections. We created a descriptive, concise API for Keshif browsers that support the common needs we identified on 160+ public datasets.

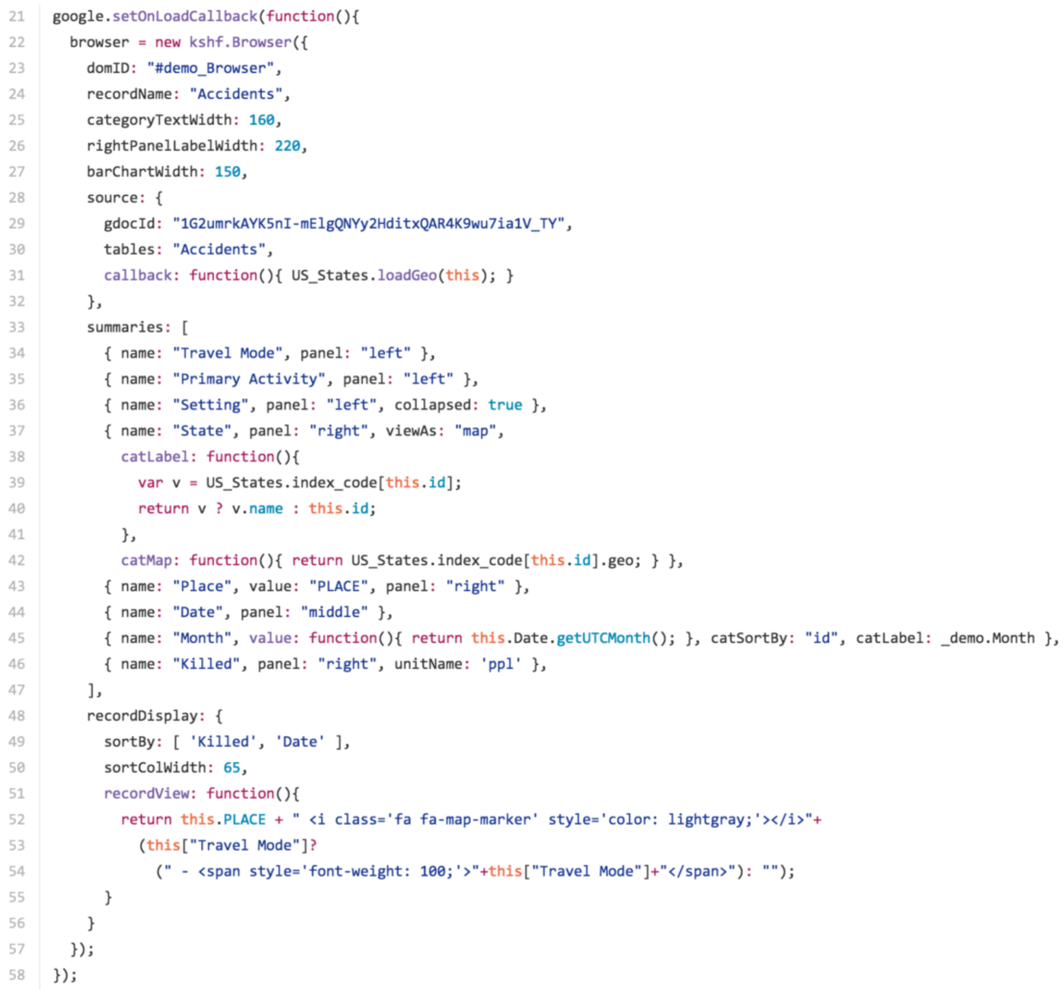
5.3.1 Graphical Authoring

Authoring enables converting raw data to an explorable form in data browsers, as well as modifying existing browsers to explore different perspectives of data. In graphical authoring mode, the available attributes panel (Figure 6) shows the attributes that do not appear in the data browser. Each attribute includes a small visualization thumbnail showing its distribution overviews with category count, or interval range. To organize the attributes, they are sorted by data type first (categorical, numeric, and time), and then by distribution characteristics. Attributes can be added to, removed from, and moved across four panels in the browser by drag-and-drop. To simplify the arrangement (a non-exploratory task) for rapid exploration, double-clicking on an available attribute adds its summary on a panel chosen based on the data type (such as categorical: left, interval: right, time: bottom), and remaining panel space. Calculated attributes can be defined in a popup panel with title and function body written in JavaScript, and evaluated live.

5.3.2 Programmatic Authoring (API of Browser Configuration)

The JavaScript API of Keshif (Figure 9) enables flexible, customizable, and persistent configuration of data browsers. The format of this configuration is minimalistic, and can be easily learned and used by web programmers. The API has a single entry-point: instantiation of a kshf.Browser object with a browser configuration, which describes the data source, the list of summaries (position, name, function, and other configurations such as sorting of categorical data or unit name for integer values), and the record display (including sorting options, record view, etc.). Multiple browsers can be added to a single web-page by instantiating multiple kshf.Browser objects. Figure 9 demonstrates functional customizations for key objectives including loading custom data (such as GeoJSON of a country, an XML file, or even BibTeX entries for literature surveys), describing a data feature to summarize (such as extracting months from a Date attribute), and describing HTML components of how a component should be rendered (such as merging multiple attributes, with custom styling). While the visual and interaction design is tightly controlled and not aimed to be end-user configurable, these callbacks provide key flexibility so that Keshif can fit many data sources, domains, and settings of analysis. In addition, Keshif browser configuration can be serialized to/from JSON objects. To handle custom callback functions in a configuration, we convert these functions to strings on export, and evaluate functions as string definitions using JavaScript `eval` function on configuration load.

The end-user API documentation is available at github.com/adilyalcin/Keshif/wiki.



```
google.setOnLoadCallback(function(){
  browser = new kshf.Browser({
    domID: "#demo_Browser",
    recordName: "Accidents",
    categoryTextWidth: 160,
    rightPanelLabelWidth: 220,
    barChartWidth: 150,
    source: {
      gdocId: "1G2umrkAYK5nI-mElgQNYy2HditxQAR4K9wu7ia1V_TY",
      tables: "Accidents",
      callback: function(){ US_States.loadGeo(this); }
    },
    summaries: [
      { name: "Travel Mode", panel: "left" },
      { name: "Primary Activity", panel: "left" },
      { name: "Setting", panel: "left", collapsed: true },
      { name: "State", panel: "right", viewAs: "map",
        catLabel: function(){
          var v = US_States.index_code[this.id];
          return v ? v.name : this.id;
        },
        catMap: function(){ return US_States.index_code[this.id].geo; } },
      { name: "Place", value: "PLACE", panel: "right" },
      { name: "Date", panel: "middle" },
      { name: "Month", value: function(){ return this.Date.getUTCMonth(); }, catSortBy: "id", catLabel: _demo.Month },
      { name: "Killed", panel: "right", unitName: 'ppl' },
    ],
    recordDisplay: {
      sortBy: [ 'Killed', 'Date' ],
      sortColWidth: 65,
      recordView: function(){
        return this.PLACE + " <i class='fa fa-map-marker' style='color: lightgray;'></i>"+
          (this["Travel Mode"]?
            (" - <span style='font-weight: 100;'>"+this["Travel Mode"]+"</span>"): "");
      }
    }
  });
});
```

Figure 10- Keshif configuration for an avalanche accidents dataset. This browser can be accessed at keshif.me/demo/AvalancheAccidents. The full source of the web-page is available at

github.com/adilyalcin/Keshif/blob/master/demo/AvalancheAccidents.html

5.3.3 Sharing and Collaboration

To enable saving, hosting, loading, and editing browser configurations easily as JSON objects, we implemented a GitHub Gist-based storage and authentication, similar to the blockbuilder.org and bl.ocks.org services. Gist configuration are stored and loaded using unique IDs, such as keshif.me/gist/?82d0d3caed8e93ea5ff8, with code hosted at gist.github.com/82d0d3caed8e93ea5ff8. This allows easy version-control and forking of browser configurations. Our Gist integration also can manage custom CSS style files along with browser configuration.

5.4 *Implementation*

Keshif is implemented as a cross-platform tool based on modern web standards of JavaScript, HTML and CSS. As a strictly client-side tool, Keshif is a lightweight system that does not require a server installation or maintenance. Datasets can be loaded from cloud services that host spreadsheets (such as Google Sheets) or documents (such as CSV or JSON files on Google Drive or Dropbox), in addition to files hosted at a local server, or uploaded from local computer (non-persistent). Essentially, a Keshif browser can be built on any data resource that a web browser can access, and Keshif does not control data authentication and security protocols of the data sources, which can be set up using the cloud services.

Keshif's client-side basis puts a practical limit on the data volume that can be loaded into browser's memory, while a demonstration with 220k+ records is available as a NYC bike-trip data browser (See Figure 11). This dataset, with 8 active summaries, can be interactively browsed (queried without significant delay, about 500ms to 1 seconds in filtering performance, faster for highlight selection) in Macbook Pro (Retina, Mid 2012) with 2.3Ghz Intel Core i7 processor and 8 GB

1600 Mhz DDR3 memory with NVIDIA GeForce GT 650M GPU running on MacOS Sierra operating system. The performance significantly depends on the browser (type and version, including JavaScript runtime), operating system, and hardware. In addition, the query execution speed is related linearly to the selected number of records, and the number of aggregates they appear in, as Keshif currently implements a linear pass over each element, and checks if they meet the query condition for each potential filter, and propogate selection changes to each aggregate that the record appears in. Therefore, making selections of an aggregate that has 50k records responds slower compared to an aggregate with 10k records. In this dataset, filtering queries can take to complete and start refreshing the charts.

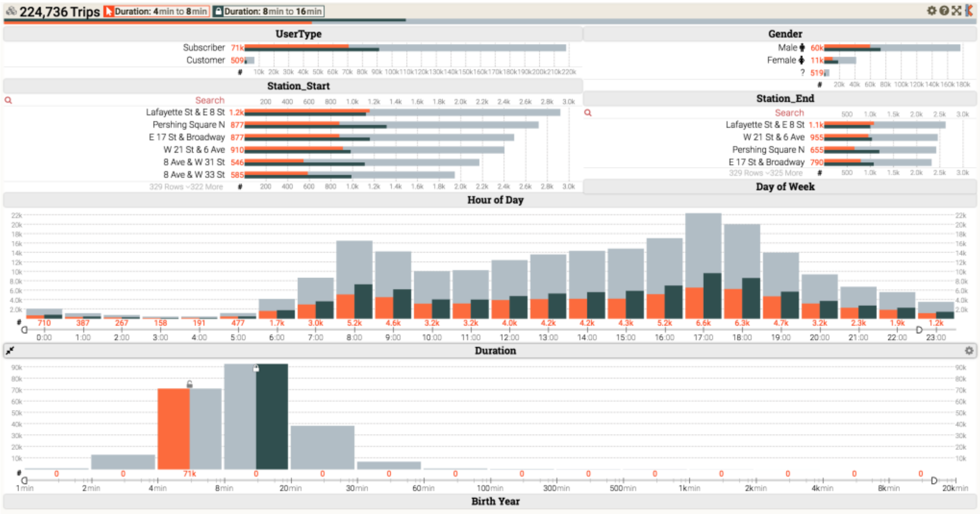

224,736 Trips
Duration: 4min to 8min
Duration: 8min to 16min
UserType
Subscriber 71k
Customer 509
10k 20k 30k 40k 50k 60k 70k 80k 90k 100k 110k 120k 130k 140k 150k 160k 170k 180k 190k 200k 210k 220k
Gender
Male 60k
Female 11k
? 519
20k 40k 60k 80k 100k 120k 140k 160k 180k
Station_Start
Search
200 400 600 800 1.0k 1.2k 1.4k 1.6k 1.8k 2.0k 2.2k 2.4k 2.6k 2.8k 3.0k
Lafayette St & E 8 St 1.2k
Pershing Square N 877
E 17 St & Broadway 877
W 21 St & 6 Ave 910
8 Ave & W 31 St 546
8 Ave & W 33 St 585
329 Rows ~322 More
Station_End
Search
500 1.0k 1.5k 2.0k 2.5k 3.0k
Lafayette St & E 8 St 1.1k
W 21 St & 6 Ave 955
Pershing Square N 655
E 17 St & Broadway 790
329 Rows ~325 More
Day of Week
Hour of Day
22k 20k 18k 16k 14k 12k 10k 8.0k 6.0k 4.0k 2.0k
710 387 267 158 191 477 1.7k 3.0k 5.2k 4.6k 3.2k 3.2k 4.0k 4.2k 4.2k 4.3k 5.2k 6.6k 6.3k 4.7k 3.2k 2.3k 1.9k 1.2k
0:00 1:00 2:00 3:00 4:00 5:00 6:00 7:00 8:00 9:00 10:00 11:00 12:00 13:00 14:00 15:00 16:00 17:00 18:00 19:00 20:00 21:00 22:00 23:00
Duration
90k 80k 70k 60k 50k 40k 30k 20k 10k
0 0 71k 0 0 0 0 0 0 0 0 0 0 0
1min 2min 4min 8min 20min 30min 60min 100min 300min 500min 1kmin 2kmin 4kmin 8kmin 20kmin
Birth Year

Figure 11- NYC Bike-Trips Data on Keshif (available at http://keshif.me/demo/nycbiketrips)

Our implementation emphasizes a lean, minimalist approach as well. To keep our development stack minimal and have full control over the implementation and user interface design, we opted not to use frameworks such as React and Angular, or even jQuery. The *only* core dependency of the current Keshif implementation is D3, which is used to bind custom data structures to page components, create visualizations, and update these components interactively. We implemented our own internal aggregation and cached computations, since Keshif support query models not supported by off-the-shelf tools like Crossfilter. The JavaScript code is developed and maintained under a single file, keshif.js. In addition, Keshif uses Leaflet to render interactive maps, and PapaParse to load and parse CSV files when necessary. Keshif browser styling is implemented using less, a CSS preprocessor, which simplifies hierarchical styling and cross-browser compatibility. Our current unminified JavaScript implementation is over 11kLOC (460KB), and less stylesheet is over 4k LOC (138KB). Keshif also uses FontAwesome, which provides a clean, consistent, and familiar icon design for many actions and objects in Keshif interface. Furthermore, we implemented most animations using CSS3 transitions instead of using d3.transition(), making it more concise, simpler to develop and maintain. We used CSS flexbox display model to implement flexible and responsive layout components. Since rendering records individually (in record display) can hurt rendering performance given large datasets, we implemented an infinite scrolling strategy, creating page DOM elements conservatively and dynamically on scroll and filter.

5.5 Discussion

In the design and development of Keshif, our end goal is to lower barriers in generic tabular data exploration in order to reach more insights (knowledge) from raw data in a short time. The barriers are higher for novices in data analytics with lesser existing knowledge to make good decisions in visualization and interaction. Our user study with novices in short-term data exploration (Section 7.2) suggests high performance using Keshif, with the volume, range and characteristics of insights comparable to skilled users on advanced tools as reported in other studies.

In contrast to existing visualization and charting environments that emphasize flexibility on design and support for non-exploratory tasks, we instead focused on building an immersive data exploration environment with extended best practices and refined design. Our implementation automates the aggregated visualization and linked selection interaction model that addresses the reported limitations, preferences and cognitive processes of the users [54]. Our integrated design extends upon effective and common techniques such as the overview-to-detail flow for information seeking [1], faceted browsing [157], coordinated multiple views [118] with brushing and linking. Specifically, we introduced a linked selection model composed of three complimentary selections (highlighting, filtering, and comparisons), aggregated visualizations including global overviews, no-value aggregations and semantic alternatives, and scale and measurement modes on aggregations for alternative high-utility views into data.

We argue that the effectiveness of the resulting data exploration space follows our design motivations in data exploration, and our basis in effective principles and techniques for visualization and interaction. By focusing on a core set of features seamlessly integrated to provide an expressive and consistent exploration space, the end-system is both greater and different than the sum of its individual components, following gestalt principles. Therefore, our contribution also lies in the definition and demonstration of the combination of our systematic components. Furthermore, our implementation advances the state-of-the-art in web-based visualization engineering as an open source tool used by thousands of visitors and hundreds of developers as of the time of submission.

While this dissertation reflects the refined design of our solution, we had considered and iterated on alternatives some of which were found to be limited or inferior. All selection and visualization states (such as measure function and scale modes) are shared across all summaries to create a consistent and easy-to-control interface, which is in contrast to flexible coordination models which require more training and decision making. For visualizing compared selections, we chose side-by-side rather than stacked designs since stacking only works when selected record groups are exclusive, therefore not applicable for multiple selections across summaries or in a multi-valued categorical summary. We avoided categorical wordclouds because of their limitations in perceptual accuracy, well-defined ordering, and use of size encoding, compared to bar charts. Scatterplots and ||-coords present challenges in scalable overviews. Our focus on univariate visualizations implies that Keshif achieves multivariate exploration with synchronized interactive views rather than multiple variables visualized in a single chart.

We presented the aggregate glyph designs for visualization on selected common data types rather than a design basis applicable to a wide range of chart types. We did not aim to provide generalizations for exploratory visualizations, although we present components and design features that can be applied to new data types and semantics. For example, spatial points (lat-long) can be aggregated on a map using the circular glyphs of set-pair matrix. Our design can be extended to support aggregate hierarchies to represent categorical hierarchies, and merge aggregates for higher-level overviews.

Lastly, we modeled exploration process to start with raw data, and have not proposed models to *capture* the process of exploration directly. The raw data is converted to a dashboard through its metadata, including attribute descriptions (which are shown on mouse-over on ℹ) and codebooks (converting integer codes to string labels, as commonly used in some datasets). The source of the data can be linked on the browser using ▦ icon, which can be manually adjusted to link to a page including data dictionary or detailed source information. However, beyond these supporting features to provide links and descriptive information on data, Keshif does not aim to provide a data dictionary or reflect the process of how data was collected. It also does not aim to support *editing* data. While data quality or coverage issues can be exposed visually through creating dashboards quickly, it does not offer views specifically focusing on detecting potential data issues. Supporting automated data quality checking with integrated visual reporting in the system may be a part of the future extensions of Keshif.

5.6 *Limitations*

In this section, we identify some of the limitations of the proposed exploration model and its implementation, Keshif, through multiple perspectives: Limitations of data model (what kind of data types are supported, not supported, and cannot be supported), form factor (what kind of devices can be used), collaboration (what kind of collaborative tasks are (not) supported), skills (how user skills influence the outcomes), data size (the limits of data size in our implementation), and chart types (what kind of charts may (not) be supported). We also contrast the goal of minimalism, to achieving expressiveness, discoverability, and visual complexity, which can be opposing goals when considered together.

5.6.1 Data Model

Our data model is strictly tabular, and a Keshif browser presents a single record type (table), where each record ideally presents a single observation (an event, entity, person, etc.). Calculated attributes enable linking to additional tables to merge multiple datasets. This data model design is consistent and minimal, yet places limitations on the supported data structures. For example, raw data in aggregated forms cannot be explored with full flexibility in selections, and spatio-temporal datasets that describes observations across multiple dimensions do not lead to effective attribute summarizations in our design.

5.6.2 Form Factor (Display size and input devices)

Keshif is designed for desktop/laptop form factors with pointer-based (mouse/touchpad) interaction. It does not aim to scale to small (mobile) displays or large displays effectively. Showing multiple charts in a small screen with linked selections and brushing may not be an optimum design approach for small screens. Likewise, large spaces would present different interaction requirements and opportunities, as well as the need to scale charts into larger form factors which can be observed from both a short and far distance. Keshif is also not designed for rich touch interaction. Some buttons and selection targets are smaller than recommended sizes for touch interaction, and we did not discuss alternative inputs with multi-touch, such as zooming or more advanced dragging capabilities. Future work can focus on design extensions for a wider range of display and input characteristics.

5.6.3 Collaboration

Our problem space models the user as an individual with a motivation to understand tabular datasets. While browsers can be forked, refined, and shared, we do not propose a model for synch or asynch collaboration in data exploration. Our model does not present solutions for provenance of insights or interface use. Based on our focus on exploratory process of data understanding rather than data presentation, Keshif is not designed to support custom annotations, or exporting charts.

5.6.4 Required Skills for Customized Authoring

While Keshif offers a graphical interface for authoring and exploration, features such as calculated attributes, API customizations and custom data loading callbacks target a more skilled audience (such as with some web development experience). While informal feedback from some external users with novice coding skills noted that Keshif API can be learned and used through example browser configurations, we are looking forward to extending graphical features for authoring and calculations, while maintaining Keshif's lean and clean design.

5.6.5 Data Size

Keshif is currently implemented as a client-side tool that runs on a web browser locally. While the lack of a server query backend limits scalability in practice because of computational limitations, it also makes Keshif easy to deploy, maintain, and integrate with existing data sources and web pages. The aggregated visualizations of Keshif can support larger datasets by design given appropriate data backends that support aggregated and flexible queries. The future work to offload computation from client to server side includes development of remote and scalable data backends, incremental data transmission, and rapid query models.

5.6.6 Chart Types

Previous section includes discussions regarding the chart types selected and the visualization designs. To clarify the limitations, we do not represent the multiple selections and aggregate glyphs approach as a full grammar that would automatically support data types and use cases beyond those presented or discussed. For example, summarizing multiple measurements of a single variable is not supported, such as, given a list of cities with various indicators, Keshif currently cannot summarize *population over time* as a single, interactive, integrated chart. Current model would only be able to summarize population of cities at a *single* time-point using a histogram. Showing multiple selections on time-series data while supporting different aggregation modes (count/sum/average), data types and visualization settings is a challenge not addressed in this dissertation. However, extending the model to lat-long data types with dynamic spatial aggregations is possible, and the model can also be extended to support bi-variate analysis with additional effort, as an extension of the set-matrix design already presented as a scalable basis for scatterplot-like relations across two variables.

Bi-variate analysis in a single chart is also supported only for multi-valued categories in Keshif. Generalized charting solutions include scatterplots or heatmaps with two axis using different attributes. Data summaries in Keshif are designed to *aggregate* data, and be scalable. For example, scatterplots would not scale to different input sizes or conform to the idea of ‘summarizing’ the data. However, the record display can be extended to support a scatterplot view (where each record is a point), or alternative charts where each record is presented only once (such as parallel coordinates, or bump charts). Adding such “features” would require considering how it would be enabled and used in the exploratory process without violating the minimalist and systematic design basis of the work presented in this thesis.

5.6.7 Minimalism vs. Expressiveness

As noted in the motivations (Section 1.1), minimalism and expressiveness can be opposing goals. When one wants to make a system more expressive, it is generally achieved through adding new features, which may not be aligned cohesively with existing features, and reduce its minimalism, usability, and learnability. The proposed model, and its implementation Keshif, aims to achieve minimalism through connected components, consistency, and minimal UI. It targets core, common data types, and core data analysis tasks, such as comparison, ranking, filtering, and observing trends, using alternative measurements within record groups (aggregations). The features are designed to work together seamlessly, rather than as isolated parts of an amalgamation of various charts and analysis options.

The limitation of expressiveness includes not only data model and collaboration, but also other tasks such as data presentation, and the possible data queries. For example, while SQL might be used to query a database in very flexible ways by chaining and merging different selections, the proposed model only presents a single, fully synchronized query model. Other visualization or data preparation tools such as Tableau can include more flexible ways to formulate new data properties using data not only from each row, but by using metrics from all the dataset and visual structure to enrich data visualizations, such as generating Perato charts. These example functions include ranking, running count/sum/average, window count/sum/average and combinations thereof. We did not propose a fully flexible, all-generalized model to transform and re-purpose data into new formats. However, by using full JavaScript specifications, we enabled various transformations for data attributes per record. The selections and linking strategy of Keshif is also single-purposed, as such cannot be as flexible as Snap-together [105] and Improvise [146].

5.6.8 Minimalism vs. Discoverability

Another point of friction across different goals is between minimalism and discoverability. Reducing icons, and revealing options only on certain interactions (such as revealing locking icon after highlight selection (mouse-over), or changing aggregate metrics through a single, shared icon) may lead to a design for which the features are harder to discover. We have observed these limitations in our user studies with novice users of Keshif (See Sections 7.1 and 7.2 for examples). These limitations in discoverability was among the factors that lead us to design an integrated help system, most relevantly its Guided Tour mode (See Section 8.3.4) and Topic Listing mode (See Section 8.3.2). However, the capabilities of Keshif still require some learning investment, and using it effectively requires analytical thinking. Having a menu-less approach where data *becomes* the interface is a passionate goal. Yet, with increasing expressiveness, discoverability can become a new profound barrier to in-depth data analysis. Making the current design easier to *discover* is one of the future design challenges.

5.6.9 Minimalism vs. Visual Complexity

While Keshif aims to achieve a systematic minimalism, we have observed that the visualizations and interactions it enables may be visually complex or confusing for some audiences and some settings. One source of complexity is the multi-selection visualization glyph design of Keshif. Having up six colors on a single aggregate glyph representing different selections (Table 1) can be confusing to first-time users. To limit the impact of this complexity factor, Keshif starts the exploration process from the overview (total selection), and any future selections are enabled explicitly by mouse-over or clicking, giving full control to the user.

Another contributor to complexity is the frequent animated updates on mouse-over. While we implemented a thresholded delay to selection while the mouse is moving to prevent highly frequent updates, the highly interactive nature of making selections, where every action might lead to a change in the interface, can be confusing to some users, as we have observed on various occasions through feedback. One way to counter this observed complexity effect is to offer limiting highlight selection, or increasing its delay threshold, for novice audiences. This would decrease the rapidness of data exploration through quickly observing multiple sub-groups by moving the mouse, however with benefits to readability.

The perception of visual complexity also depends on the viewer, their domain knowledge, and motivation. For example, a data browser with ten charts describing various aspects of the data may have high utility for a domain expert who would like to explore relations across multiple attributes simultaneously through linked selections. However, such an interface may be too *busy* or *distracting* for a casual person who may not wish to see all these trends, and they may gradually increase complexity as they prefer. The ideal situation would be to bootstrap their exploration with few selected basic attributes (summaries), and encourage exploration of other attributes afterwards. This example also points to the complexity introduced by having multiple simultaneous and highly connected charts on a data dashboard. One way to reduce complexity would be to enable expanding one chart to a full-size to cover the browser, and limit exploration across multiple summaries. This may simplify (limit) the data presented on the screen, and can also allow seeing more details in a single chart (such as a larger map, or an extended multi-column list).

Lastly, we have developed HelpIn (Chapter 8) to counter the complexity of the interface by offering live, contextual, integrated descriptions to help readability of the data interface, charts, and various interactions. One limitation is that this help-based approach is based on existing design of the tool, and does not make it inherently simpler or more effective, but aims to close the gaps with additional features. While we argue that getting help, and training, for a data interface/tool is crucial for effective use, we also recognize that the first goal should be to create a better designed interface rather than providing help when discoverability, usability, and confusion issues arise.

Chapter 6. AggreSet – Set-Typed Data Exploration Technique

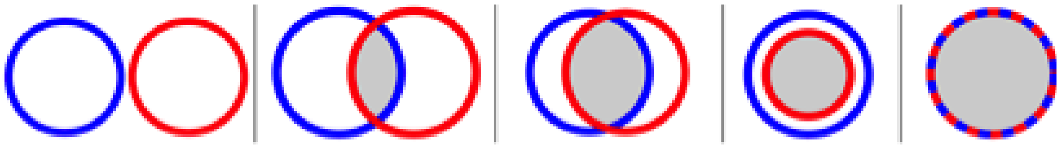
"Every doorway, every intersection has a story."

Katherine Dunn

AggreSet specializes the proposed data exploration model to meet the challenges of set-typed data exploration. In this section, we present features of set-typed data, the detailed design of AggreSet, and how it makes set-typed data explorable.

6.1 Features of Set-Typed Data

Set-typed data implicitly define relations between sets (A, B) based on their intersection (Q=A∩B). Figure 10 orders intersection in increasing strength: disjoint sets, partial subsets, proper subsets and identical sets. Revealing these relations are among set visualization goals. *Disjoint* relation (Q=∅) represents empty intersection. It is very common in sparsely connected sets. *Identity* relation (A=B=Q) represents the strongest connection. It requires both sets to contain the same elements. *Proper subset* relation is the strongest relation when sets have different number of elements. One set subsumes the other, i.e. all elements that appear in the smaller set are also in the larger set (A⊂B, Q=A or B⊂A, Q=B). In datasets, many set-pairs are in *partial* relation. The sets have some shared items, and each set has some unique element compared to the other ($Q\neq\varnothing$, $A\setminus B\neq\varnothing$, $B\setminus A\neq\varnothing$).



a) Disjoint b) Partial (weak) c) Partial (strong) d) Proper Subset e) Identity

Figure 12- Relations between two sets based on shared elements.

To model relations between sets, we define the *strength* of a set pair {A,B} on a continuous scale from disjoint (0) to subset (1), computed as $|A\cap B|/\min(|A|,|B|)$. The set-pair intersection gets stronger as the sets share more elements, and the strength reaches one when the sets share all the elements they can share. This metric presents a normalized context to set-pair relations, a form of similarity, and is an alternative to characterisation by element count, an absolute value on an unbounded scale.

In contrast, the Jaccard Index, a common set-relation metric, normalizes the intersection size of two sets with their union size ($|A\cap B|/|A\cup B|$), also ranging from 0 (disjoint) to 1 (identical). However, this metric produces an unbalanced distribution since high values (toward equity) are much less likely to occur than strength metric (toward subset-ness) given varying set sizes. There are also other similarity metrics representing deviation from expected values using statistical inference assuming a marginal independence between sets [3], [88]. Such metrics return positive or negative values depending on whether the observed element count is higher or lower than expected. Deviation results can be compared relatively across sets and their intersections, while the *strength* metric is meaningful in absolute form (*subset*-ness) as well as for comparison.

6.2 Set Exploration Modeling

Set exploration is conceptually non-trivial; there are many tasks that involve intersections and relations between multiple sets and other element attributes [4]. To support a rich and comprehensive ability to explore set-typed data, we present a new modeling for data representations, low-level actions, and high-level tasks. This data and low-level action model is shown in Figure 11 below.

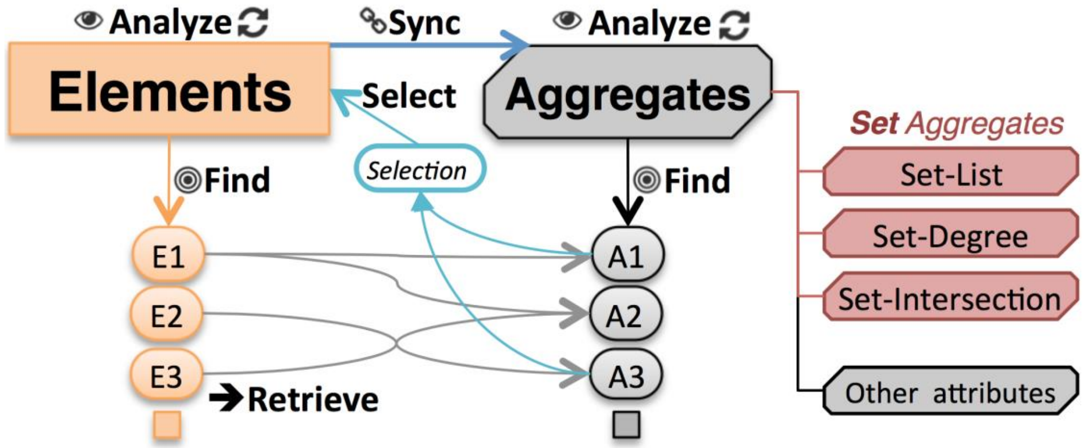
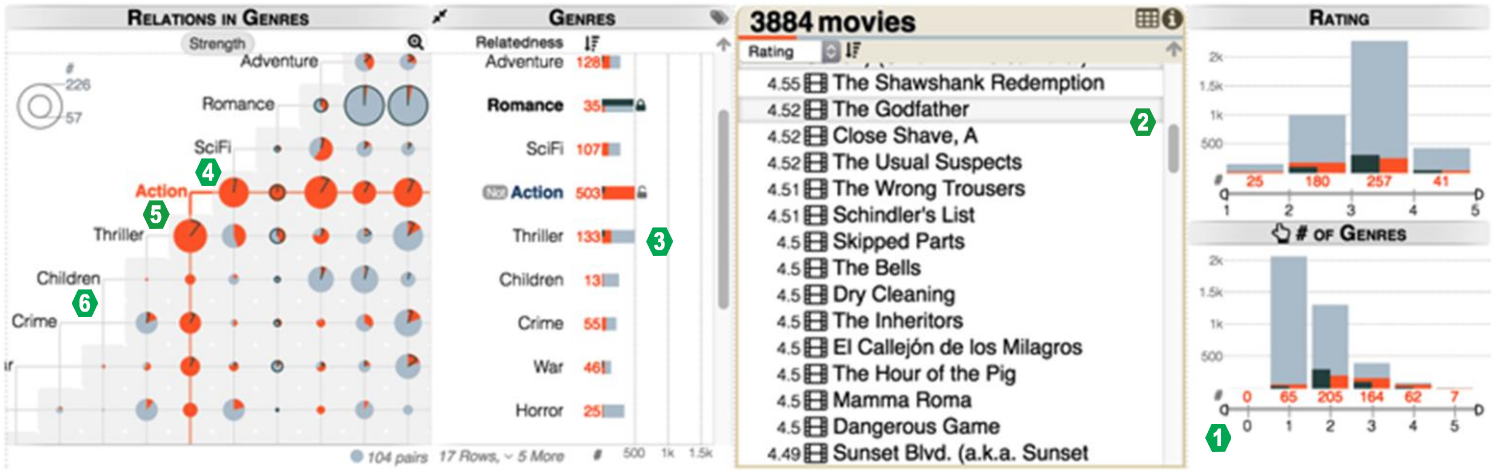


Figure 13- Our set exploration model for data and low-level actions. Elements are mapped to aggregates, and actions are defined across data types. A set-typed attribute is decomposed into three forms of element aggregates: set-list, set-degree, and set-intersection. This model distinguishes the explicit set-list from set-intersections, and allows for exploration using set-degrees directly. Given a group of elements/ aggregates, you can **Find** an element/ set with some characteristic, or **Analyze** the group overview to detect the range of values and patterns. Given an element, you can **Retrieve** the aggregates that include the element. Given a selection of one or more aggregates, you can **Select** the elements that satisfy the selection. We do not differentiate *how* selection is actualized (i.e. highlighting or filtering). Lastly, given a selected element group, **Sync** is a global action from all elements to all aggregates to reflect underlying element characteristics. **Sync** action generalizes **Retrieve** for selected elements to enable **Analysis** within all aggregates. Sequencing these low-level actions on set list, degree and intersections allows expression of complex queries by creating flexible type-agnostic paths.

To exemplify the execution of this model, let me consider a movie dataset where each movie (element) has multiple genres (sets), an average rating, and a country of origin. What are the genres, the countries, and the range of ratings in the dataset (**Analyze** within aggregates)? What are the genres and the rating of the movie *Wall-E* (**Retrieve**)? What are the two most common genres (**Analyze** within genres, **Find**)? How many genres does a movie have at most (the maximum genre degree) and what is the degree distribution? (**Analyze** within genre degrees). Such overview reveals basic patterns. Then, exploration expands through selections. What are the drama movies? Movies that have at least three genres? Movies with highest ratings? Such exploration commonly starts with a **Select**, is followed by Sync that retrieves and aggregates selected element attributes, in order to Analyze data characteristics in multiple data dimensions. What is the rating distribution of children's movies (genre to rating)? What are the common genres of high-rated movies (rating to genres)? What other genres do documentary movies have (genre to genres - set relation)? Which genres have more multi-genre movies (genre degree to genres)? Which genre pairs are more common, which genre pairs include no movies (empty intersections), and which genres always appear together (are subsets) (Analyze within set intersections)? We can then compare different selections. How do ratings compare across horror vs. documentary movies (Select horror → Sync, repeat for documentary and Analyze for comparison within rating)? We can expand our inquiry by looking at intersections of multiple genres. AggreSet supports all such queries through its single aggregate-based exploration modelling.

Many exploratory questions depend on the **Select** action based on some criteria. Rich data exploration is only possible through flexible selection models, ideally with ease of expression. Selection for set-typed data can include multiple attributes (high-rated drama movies) and multiple set values can be selected using different modalities (family *and* comedy movies *without* action), representing intersection ($\cap$ - and), union ($\cup$ - or), and complement (\ - not).

Comparison of data characteristics under different selections is a more complex form of exploration. To support comparisons across different element selections, Select→Sync→Analyze pipeline needs to be executed under each selection, and the resulting distributions need to be saved and visualized. Exploratory comparison then follows visualizations of multiple distributions.



Rating	Movie
4.55	The Shawshank Redemption
4.52	The Godfather
4.52	Close Shave, A
4.52	The Usual Suspects
4.51	The Wrong Trousers
4.51	Schindler's List
4.5	Skipped Parts
4.5	The Bells
4.5	Dry Cleaning
4.5	The Inheritors
4.5	El Callejón de los Milagros
4.5	The Hour of the Pig
4.5	Mamma Roma
4.5	Dangerous Game
4.49	Sunset Blvd. (a.k.a. Sunset

Figure 14- Exploration of a movie dataset with multiple genres (sets) and ratings using AggreSet. Aggregate histograms are used for set-list and set-degrees, whereas the aggregate matrix (left) is used for set-pair intersections. The gray distributions visualize the number of elements per aggregate. The *Action* genre is selected by mouse-over 👆. Mouse click will filter. We compare *Romance* (black lines) to *Action* (orange areas). ❶Most movies (+2k) have one genre. 7 movies have maximum (5) genres. ❷*The Godfather* is the only Action movie in the rating-sorted movie list. ❸Of *Thriller*s, 133 have *Action* (orange bar) and few have *Romance* (black line). ❹More than 50% of *SciFi* and *Adventure* movies have *Action*, while very few have *Romance*. ❺*Thriller* is more common with *Action* than with *Children* movies (circle size). ❻There is no *Children* movie with *Crime* (empty intersection).

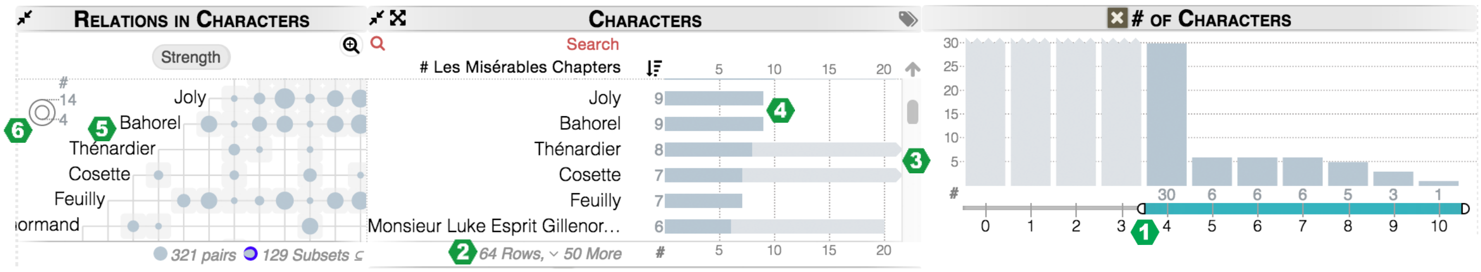


Figure 15- Character co-occurences in Les Miserables, with 80 characters (sets) in 356 book chapters (elements). ❶Data is filtered to chapters that have at least 4 characters. ❷The related 64 characters are reordered by the number of book chapters they occur. ❸ Thénardier and Cosette, have ghost-bars (gray extensions), showing that these characters also appeared in chapters with <4 characters while ❹Joly and Bahorel appear only in the chapters with ≥4 characters. ❺Thénardier is one of the common characters, yet he does not appear with Bahorel, Feuilly, and some other characters outside of the view (disjoint set pairs). ❻The legend shows circle size mapping.

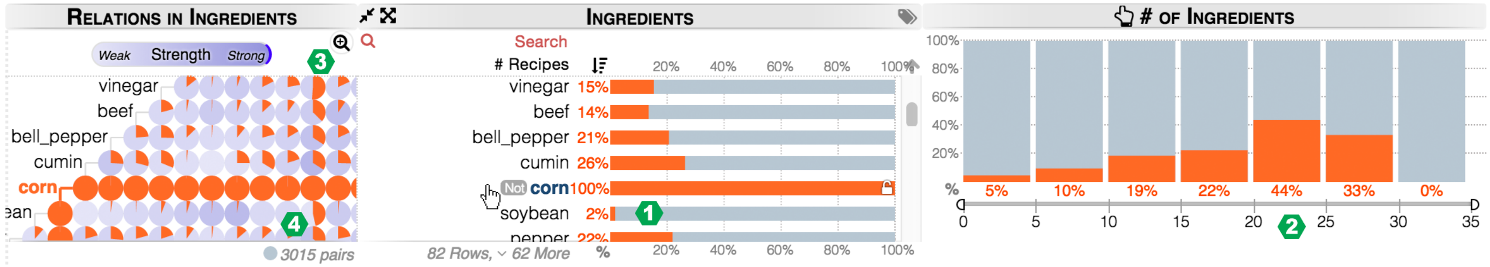
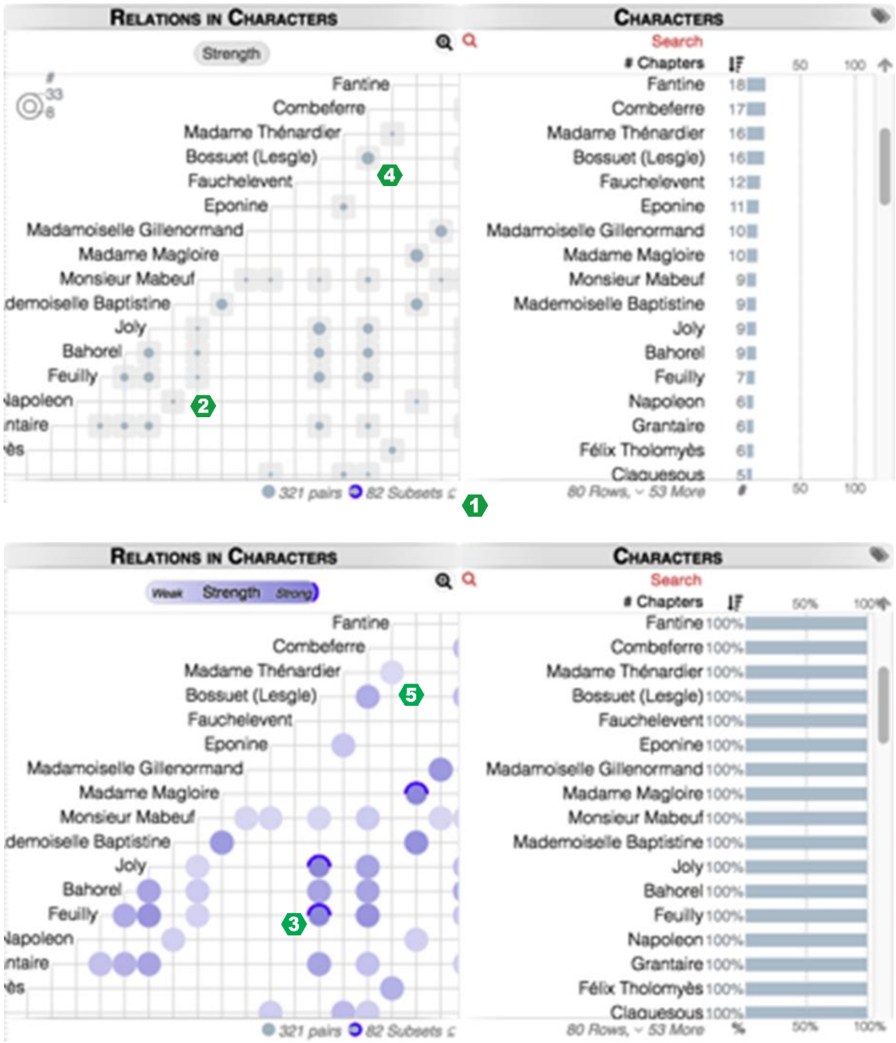


Figure 16- 313 ingredients (sets) in 5,000 recipes (elements). The *relative-mode* is active; each aggregate glyph is scaled to its maximum size (length or radius), creating a shared percent scale. The orange *result-preview* shows the distribution of selected corn among all aggregations in percentage. ❶Corn is rarely used with soybean; 2% of recipes with soybean have corn. Corn is popular in recipes with high number of ingredients. ❷At the peak, 44% of recipes with 20 to 25 ingredients have corn. In the matrix, recipes with the second rightmost ingredient (positioned above the view, tomato) frequently have corn, such as ❸half of vinegar-tomato, and ❹ half of soybean-tomato recipes.



Characters

Search

# Chapters		
Fantine	18	
Combeferre	17	
Madame Thénardier	16	
Bossuet (Lesgle)	16	
Fauchelevent	12	
Eponine	11	
Madamoiselle Gillenormand	10	
Madame Magloire	10	
Monsieur Mabeuf	9	
Mademoiselle Baptistine	9	
Joly	9	
Bahorel	9	
Feuilly	7	
Napoleon	6	
Grantaire	6	
Félix Tholomyès	6	
Claquesous	5	

80 Rows, ~ 53 More

Characters

Search

# Chapters	
Fantine	100%
Combeferre	100%
Madame Thénardier	100%
Bossuet (Lesgle)	100%
Fauchelevent	100%
Eponine	100%
Madamoiselle Gillenormand	100%
Madame Magloire	100%
Monsieur Mabeuf	100%
Mademoiselle Baptistine	100%
Joly	100%
Bahorel	100%
Feuilly	100%
Napoleon	100%
Grantaire	100%
Félix Tholomyès	100%
Claquesous	100%

80 Rows, ~ 53 More

Figure 17- Character co-occurrences in Les Miserables. ❶This dataset has 82 subset relations. **Top**: The circle area maps the number of chapters both characters occurs in. ❷Intersections with few chapters appear small and are hard to observe. **Bottom**: The circles are full and color denotes the character relation strength by the chapters they occur in together. The border is shown when one character always appears with the other character. For example, ❸all of Feuilly's chapters (7) also include Bossuet, who appears in 16 chapters. This suggests a proper-subset relationship, and the border is half. When two characters always appear together, their border is full (not visible in this cross section). We can also observe that while intersection of Madame Thenardier was one of the largest in number of chapters ❹, it is *not* one of the strongest ❺.

6.3 Set-typed Data Exploration with AggreSet

Set-typed data exploration with AggreSet encourages the overview-to-detail flow of the information seeking mantra [1]. Its approach can be explained in four levels with increasing depth and richness. (*i*) AggreSet displays sets as a linear list, aggregates elements within sets, and visualizes the distribution of elements. It orders sets with larger element counts first by default (Figure 16, and Figure 17-a). By selecting a specific set, the user can interactively explore (highlight, filter, compare) distributions of elements of the selected set, also revealing its intersections. (*ii*) AggreSet summarizes the set-degree of elements. Selections on this dimension can be used to reveal higher-order set relationships (e.g. intersections of >3 sets) (Figure 13). (*iii*) AggreSet introduces the set matrix to visualize the distributions in set-pair intersections and set relations (strength) using circle glyphs. The interaction design (highlight, filter, compare) seamlessly extends to this matrix. (*iv*) Intersections beyond second degree (set-pairs) are explored through selections. At all levels, the result list can show all, or filtered, elements (Figure 17), and other categorical and numeric attributes are presented with the same core design as set dimensions.

AggreSet uses element aggregation to scale on element count by design. Element are aggregated per set, per set-degree and per set-pair intersection, as modeled in Figure 11. Since set-pair aggregation is independent of the set order, the set matrix uses half of the matrix, and therefore avoids visual duplication. The intersections of a set are captured along two set-lines, one vertical and one horizontal. For example, in Figure 12, action movies are selected and two orange lines in matrix pass through the intersections of this set. The rows/columns are also highlighted when a cell is selected (pointed) in the matric (Figure 16). The empty half of the matrix displays set labels (for easy identification of sets involved in intersection circles) and visual legend for matrix.

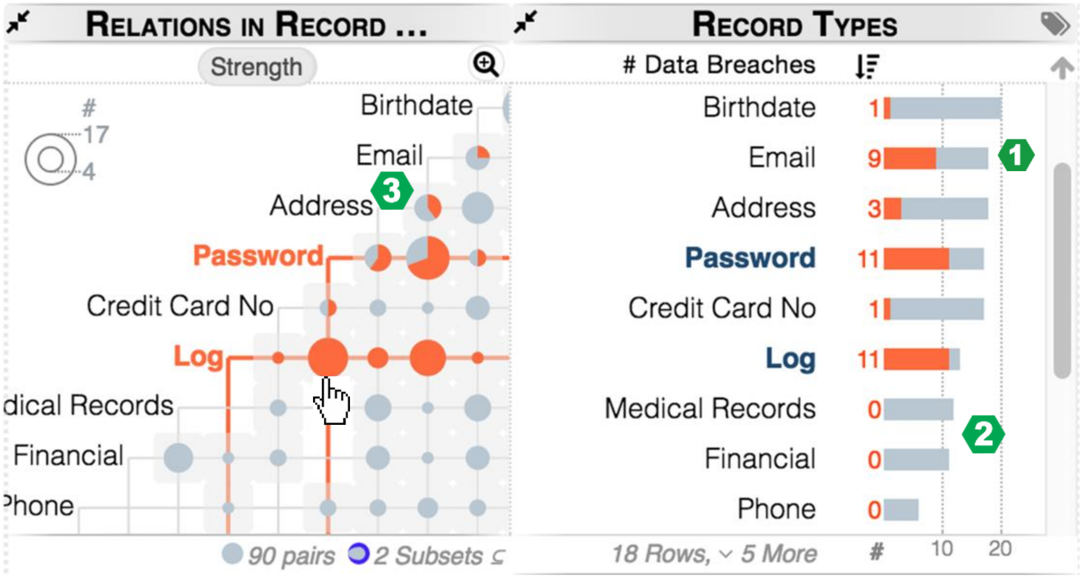


Figure 18- Record types (sets) compromised in 284 large-scale data breaches (elements). ☝ 11 Breaches with *log* and *password* record types are selected using result-preview. The large circle size shows these two record types were commonly compromised together. 3rd order intersections (∩'s of 3 sets) are shown on the set-list histogram. For example, ❶ *email* is commonly associated with the selected breaches (9 out of the 11 with *password* and *log*), and ❷ neither *medical* nor *financial* records were stolen with *passwords* and *logs*. We can also observe intersections of 4 record types. About 35% of *email* and *address* breaches also had *password* and *log* leaks ❸.

To explore a high number of sets that cannot fit within the linear and matrix view on a limited screen size, AggreSet matrix supports scrolling and panning, as shown in Figure 12, Figure 13, and Figure 14. Scrolling is a fluid interaction to observe limited parts of the dataset, compared to explicitly selecting active sets one-by-one such as applied in Upset [88], [121]. When the set-list is scrolled, the set matrix follows along its diagonal line so that for all the sets visible on the list, their intersections are also visible on the set-matrix. The intersections involving sets that do not appear in the set-list are outside the diagonal. AggreSet allows these intersections to be explored by panning the matrix view by mouse drag. Notice that the sets below the view cannot have any intersections within the matrix view by design. In addition, panning reduces the unused portion of the set-matrix view. AggreSet also supports adjusting the matrix cell size (zooming - 🔍 button) to make the circles easier to read, or to show more set-pair intersections in a single view Figure 17.

AggreSet enables exploration beyond set-pair relations by selection across set dimensions. Figure 16 shows that the *result-preview* selection on a set-pair enables analysis of intersections of three and four sets visually. Set-degree selection also enables higher order analysis. For example, to analyze intersections that involve 4 or more sets, one can filter to elements with degree 4+, as shown in Figure 13. Likewise, selection by an exact set degree will show set relations unique to intersections that only involve as many sets. Quickly iterating through different set-degrees by result-preview can provide a quick overview of higher order relations within the data.

6.4 *Details on Visual Encoding Design*

In the set-matrix, the *result-preview* is visualized with a sweeping arc on circles with 12 o'clock alignment, producing a pie chart with single pie ◔. Our design uses a sweeping arc (instead of radius mapping) to emphasize part-of relations within intersections. ◔ (¼), ◑ (½), and ◕ (¾) serve as easily recognizable visual anchors for comparison of previews to (filtered) element count. If radius mapping by area is used to reflect selection areas, such ratios are harder to perceive, such as ⊙ (¼), ◉ (½), and ● (¾). We notice that the visual distance between circles and the lack of a shared basis can be limiting factors for effective comparisons across set intersections within the matrix. The compare-selection visual encoding is an outline on the arc-swept circle, as shown in Figure 12. The 12 o'clock base line is not highlighted so that the line connecting from center to the arc is only used to show the value.

The *strength* of the relation, as defined in Section 6.1, is mapped to the circle color and border (Figure 15). Lighter color visualizes a weaker relation than darker color (○ vs. ●). The circle border visualizes subset relations. A full border (◉) shows the identity relation, while a half-border (◓,◑) shows the proper subset relation. The edge connecting the half-circle (upper or right) directs to the larger set. When the sets are ordered by element count, the containing set always appears above since it is larger. Yet, this property may not hold for other ordering approaches and the visual state encodes the direction. The total number of subset relations is also shown below the set-matrix, next to the total number of intersecting set pairs. To maintain design consistency, AggreSet re-computes the set strength metric after filtering. The *relative-mode* can be engage by clicking the Strength button on the set matrix summary. The strength button changes to Weak Strength Strong when relative-mode is enabled, describing the visualization of the strength relation with its gradient, and the blue border at the strong end. This design is limited for analysis of hierarchies of subsets, although hierarchies can be traced using the set matrix step by step.

When all circles (non-empty intersections) are scaled to full-size in the *relative-mode*, the disjoint-sets (of empty space) become visually more distinctive. The matrix layout creates a spatial context for observing sparseness of set intersections. In the absolute mode with varying circle size, AggreSet uses the grey cell background to help the viewer distinguish the small circles (few elements) from empty intersections (cells). Some sets may also be disjoint from *all* others (like disconnected network nodes). To distinguish such *isolated* sets, AggreSet removes their grid-lines, suggesting that there is no line to follow to uncover set-relations. This design reduces chart ink and makes existing lines easier to perceive.

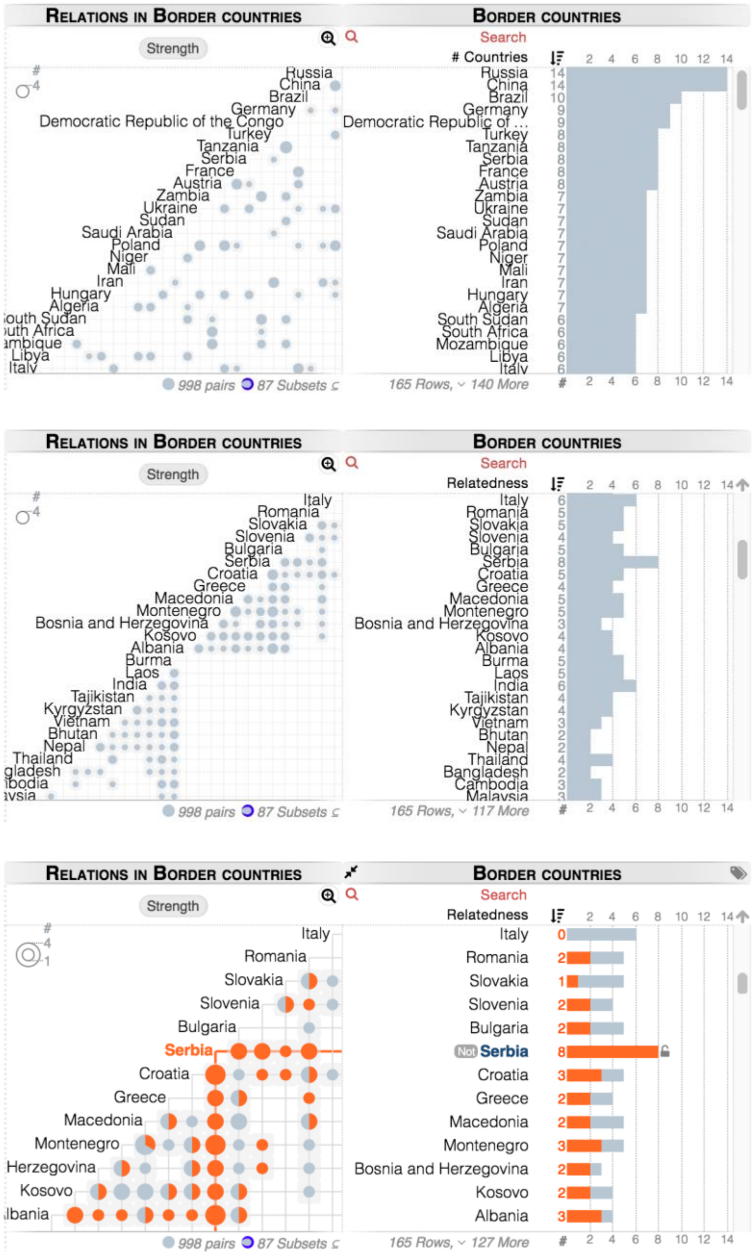
6.5 Perceptual Set Ordering for the Set Matrix

The Gestalt principles state that our perception is influenced by similarity, continuation, closure, and proximity. Jacques Bertin says “simplification is no more than regrouping similar things” [15]. Characteristics of set visualizations and visually emphasized patterns therefore depend on the set order. To reveal patterns among sets that are closely related, AggreSet includes a perceptual set ordering method aimed for the set-matrix layout. Figure 17 shows that ordering sets on element count may create salt and pepper pattern within the set matrix, and perceptual ordering can improve visual structure by placing connected sets along the diagonal.

Matrix reordering methods have been long studied [89]. Greedy heuristics and clustering are commonly used approximate solutions since ordering optimization is NP-complete in the general case given *#sets*! combinations. In AggreSet, set ordering is solved once as an approximate global layout optimization, since both matrix axes use the same order. AggreSet translates set ordering to the Minimum Spanning Tree (MST) problem by using sets as nodes, and set-pair intersections as undirected edges. The edge weight between two sets for MST is the total dissimilarity in their relation to all sets, such that $AB=XA\cap X-B\cap X$, where $A,B,X\in\mathbb{U}$. The intersection size $\alpha\cap\beta$ is used as the visual characteristic of the set-pair, i.e. the metric to optimize the matrix layout. To reduce the number of edges to be processed, only intersecting set-pairs, such that $A\cap B\neq\emptyset$, are considered. This edge weight is defined for the MST algorithm to optimize the layout globally, and is not exposed visually otherwise.

To generate MST(s) of the set-intersection graph, we used Kruskal’s algorithm, which greedily inserts edges with smaller weight (higher set similarity) to MST(s).

We generate the linearized set ordering by a breadth-first traversal of MST(s), starting with the largest tree in terms of the number of nodes (sets). To have a consistent linearization with larger sets within a tree appearing before smaller ones, larger nodes need to be traversed first. To achieve this, we modified Kruskal's algorithm such that when two nodes are connected, the node (set) with more elements becomes the new root. Our open-source implementation provides more details.



a) A zoomed-out view sorted by decreasing element (neighbor country) count. This view emphasizes countries with more neighbors. Notice the salt-pepper pattern in the set-matrix.

(b) Countries are reordered using a perceptual set ordering approach. The new ordering follows their geographical closeness, for many countries, and forms visual clusters along the diagonal.

(c) A group of 13 countries is focused by adjusting the matrix zoom. In this group, Serbia has the most neighbors, and is selected by mouse-hover. This selects the neighbors of Serbia, and the preview shows the neighbors of those countries.

Figure 19- Exploring country neighborhood relations. The list aggregate number shows the number of neighbors per each country (set).

6.6 Comparison of AggreSet and Other Set Visualization Techniques

This section presents a focused comparison of recent set exploration techniques, including AggreSet. Table 5 presents the comparison summary.

		AggreSet	**UpSet**	**RadialSets**	**OnSet**
ScSale	*# Elements*	Aggr.	Aggr.	Aggr.	100s
	# Sets	50+	20-50	30-40	N
	# ∩ (Intersections)	(#Row)^2	#Row	(#Set)^2	N
Data	*Elements* 👁	▯	▯	▯	▯
	Sets 👁	▯	▯	▯	▯
	Degrees 👁	▯	Group, filter	▯	▯
	Attributes 👁	▯	▯	▯	▯
	∩ *Degree*	2-4+	N	2-4+	N
	∩ *as*	Cell	Row	Arc/Circle	Set
Actions	*Retrieve*	▯	▯	▯	▯
	Analyze	Sets & Elements	Sets & Elements	Sets & Elements	Element focused
	Synchronize	▯	▯	▯	Partial
Features	⊂	Yes ~~hierarchy~~	▯	▯	▯
	∅	In-context	Remove	▯	▯
	∩ ∪ \	Mixed	Mixed	Mixed	Rich
	Similarity	▯	▯	▯	▯
	Compare Dist.	1-to-many	Tabular	Color	No
	Higher-Order	Preview, filter	Visible	Choose 2-4	Drag & drop
Design	*Matrix-View*	Set x Set	Set x ∩	N/A	Elements
	Element Aggr.	▯	▯	▯	▯
	Overlapping	▯	▯	Yes	Yes
	Animated	▯	▯	▯	▯
	Highlight-Select	Hover, brush	Within matrix only	▯	▯

Table 5- A comparison of interactive set exploration approaches. **Scale** group shows practical limitations in scale per data type. *Sets* shows active number of sets. ∩ shows number of intersections that can be visible on the screen. **Data** group shows the data dimensions explicitly shown. In *Degrees*, "Filter, Group*"* shows that degree is not a primary data type; it is explored by grouping and filtering in separate interface. **Actions** group shows low-level actions. *Partial sync* means not all components in the interface are connected. **Features** enable higher-order and set-specific exploration. ⊂ shows whether subsets are explicitly visualized; 1 denotes subset hierarchies are not explicit. ∅ (empty sets) can be highlighted in-context, or can be removed from display. *Similarity* of set-pairs includes deviation from expected values. *Comparison of distributions* can be enabled as *1-to-many*, in tabular form, or using color mapping. *Higher-Order* shows how intersections of many sets are explored. **Design** group lists design guidelines. *Matrix* row shows the matrix view construction.

UpSet [88] uses a combination matrix and table layout. In the matrix view, columns are (active) sets, rows are all possible intersections of these sets, and cells show the intersecting sets per row. Per each row (intersection), the tabular view shows the cardinality, deviation, and summary attribute statistics using sortable columns. Since UpSet explicitly shows all set intersections, it is effective for analysis of high-degree intersections as well as attribute characteristics per each intersection. UpSet answers ∩-∪-/ set queries by selecting and grouping intersections that satisfy the query. Grouping and sorting features for intersections extend its linear basis of design, yet these features apply view transformations that may not be intuitive on first use. As the active set count increases (more sets are inserted to the view), the combinatorial growth in number of rows and the widened matrix view reduces its visual scalability. Targeting sparsely connected sets, UpSet can reduce the number of rows by removing empty intersections. Set-attribute filtering is visually separated from filtering other attributes, while AggreSet uses the same selection modalities across data dimensions. UpSet does not visualize element degrees explicitly, although it offers a range filter and grouping by degree. In its element view, it also does not explicitly show, or link to, set memberships. Overall, when set exploration needs to focus on all possible set intersections and their characteristics given some chosen sets, the interactive tabular view of UpSet provides a rich visual exploratory space.

RadialSets [3] is based on the circular layout node-link diagram design, thus has the scalability limitations by intersecting edges. The distribution of element degrees is explicitly visualized by length encoding for each set (node), and revealed upon selection for set intersections (links). RadialSets can also visualize intersections of three or more sets using circular glyphs as hyper-edges. The positions of these glyphs are optimized to visually reduce overlaps, or placed in layers sorted by glyph sizes. Thus, understanding higher degree set relations relies either on tracing overlapping edges, or on selecting glyphs to see contributing sets. RadialSets also supports mapping other attribute characteristics to the color of set-intersection glyphs, allowing high-level overviews of differing characteristics of set intersections.

OnSet [121] visualizes elements as cells within set matrices. A matrix can represent a single set, or a set combination. Elements are located at the same cell positions across matrices, and can be spatially grouped by bounding boxes. OnSet matrices should be large enough to hold all elements, limiting scalability on element count. Sets can be dropped and merged with direct manipulation. Merge queries support ∩-∪-\ modalities with hierarchical compositions. When a matrix represents a set combination, cell (element) opacity/color shows the number of sets, of the combination, that the element appears under. Yet, the sets of the elements are not directly available. To visualize similarity across set matrices, OnSet supports a node-link diagram. This layer is visually limited in the number of (large) matrices because of occlusions. OnSet relies on pan-and-zoom interaction on a 2D zoomable canvas to explore non-trivial number of sets and relations. However, element context can be lost when zoomed out, and controlling the canvas can make the canvas space more complex to navigate and understand [14]. Its matrix design depends on the viewer's ability to understand which elements are located at which cells across matrices. Yet, element ordering and grouping structure is not explicit, and finding a specific element across multiple matrices with many rows and columns is a non-trivial task.

AggreSet supports a high number of sets, visualizes all set dimensions explicitly, enables the tasks consistently across data dimensions and attributes, supports rich, high-level exploratory goals, and avoids major design problems that may affect scalability and usability. It can be used to express the set exploration tasks proposed by Alsallakh et al. [4] through selections of five data dimensions (elements, set-list, set-degree, set-intersection and other attributes), except the three tasks relating to creating new sets from specific element selections, and analysis of inclusion (subset) hierarchies. AggreSet is also different from other multi-view visualization systems [118] with its novel combination of set-matrix view with element aggregations, set-exploration specific features (such as set-pair strength and perceptual set ordering), and interaction design with preview, filter, and compare models. The limitations of AggreSet can be discussed as the following:

(i) *Higher-order relations:* Exploring relations beyond set-pair are not immediately visualized and such exploration requires selection. In our overview-to-detail approach, this is presented as the final (fourth) level. Since explicitly visualizing higher-order relations increases the number of visualized data items, placing this information on demand through interaction allows our design to visually and seamlessly scale to overviews of more sets.

(ii) *Set intersection:* Element attribute characteristics cannot be shown within the set visualizations directly, while UpSet and RadialSets support such cases. Relations between sets and other attributes are explored through explicit selections in the minimalist design that consistently applies in both directions (set ⇔ attribute).

(iii) *Data density:* When aggregation glyphs are small, the visual mappings (size and color) can be hard to distinguish, especially for circles in the matrix view. To mitigate this problem, matrix zooming can be used to enlarge the glyphs, a tradeoff between space and number of data points. In addition, result-preview and set-pair strength uses the same visual channel (color) in matrix view, with the dominant being orange preview. While the strength is occluded on the circle, it is still available in the set-list view, right side of the matrix, in % value. This also highlights how set-list and set-matrix support one another.

(iv) *Scalability:* Given a laptop/desktop display (1280×800 pixels or more), AggreSet can accommodate on the order of 50 sets. Zooming out shrinks set and cell visualizations, and allows showing more data in a fixed display size. Panning allows exploring areas outside the visible matrix viewport. Perceptual ordering can improve the visual structure along the diagonal for some set relations and reduce information outside of the visible matrix area. Scaling to hundreds of sets with dense relations is still not practical, which would require techniques for aggregating sets and their intersections.

Chapter 7. User Evaluations of Keshif

"*In my experience, users react very positively when things are clear and understandable. That's what particularly bothers me today: the arbitrariness and thoughtlessness with which many things are produced and brought to market. Not only in the sector of consumer goods, but also in architecture, advertisement. We have too many unnecessary things everywhere.*"

Dieter Rams

In this chapter, we present user evaluations and applications of Keshif, which also include the underlying data exploration model, the set-typed data exploration technique AggreSet, and the evaluation based on Cognitive Exploration Framework presented in Chapter 3.

First, we describe two studies that include open-ended, self-driven data exploration, starting from raw data, authoring browsers, and exploring and communicating observed findings, and challenges. The first study of this kind focuses on the evaluation approach based on the Cognitive Exploration Framework, and aims to understand challenges of exploration using the proposed tool. The second study follows the insight-based methodology. Taken together, they present a complimentary overview: one of the barriers, and the other on the insights gained.

Then, we describe evaluations based on pre-defined browsers, and focusing on exploratory process rather than authoring. This evaluation is focused on the capabilities and usability of AggreSet, the set-based data exploration technique. We present an expert review, and a short case study with two domain experts in educational data analysis.

Finally, we present a summary of the public use of data browsers available on www.keshif.me, and other use cases with applications through collaborations in different organizations, and external use.

7.1 *Evaluation of Cognitive Barriers with Data Analytics Novices*

This study focuses on the application of the proposed user evaluation for cognitive activities and barriers (Section 3.3). Keshif was selected as the tool to demonstrate the protocol and gather input from the evaluation. As such, the goal was not to evaluate Keshif, but to evaluate the protocol and demonstrate the use of Cognitive Exploration Framework. Still, the observations from this study also shined light on the challenges of the first-time Keshif users, some of which were addressed in the follow-up research activities, such as improved design and providing part of the motivation for the help system. The recruitment of data analytics novices and the open-ended, exploratory and unguided nature of this study protocol are shared with the follow-up insight-based evaluation of Keshif. In this perspective, the study also provided early input about the behavior of novices, although the participants were asked to communicate different thoughts (challenges instead of insights).

This study was performed on early fall of 2015. At the time, Keshif did not have an option to modify the measure metric, the only option being the *count* metric. Other features of the tool, including authoring, were similar to otherwise described in this thesis.

7.1.1 Study Design

To detect cognitive activities and barriers in exploration, we designed a casual setting with a 15-minute exploration per dataset, and 5-minute training for using the tool. As existing knowledge and extensive training can reduce the barriers that the evaluation aims to detect, we aimed to recruit novices in data analysis, and offered limited training. The participants chose two multivariate, tabular datasets they would like to explore given five options: movies, traffic accidents, passengers of the Titanic, Lego sets, and foodborne disease outbreaks. The record (row) count ranged from 3.2k to 30k, and the attribute (column) count ranged from 8 to 16.

To encourage communication on exploration and emotional states, we also implemented an external strategy using printed cards. One group of cards described exploratory process: (i) “I am trying to find a question.” (Planning data analysis) (ii) “I am trying to answer a question.” (Planning interaction & visualization) (iii) “I have an insight.” (Assessing data analysis). Another group of cards focused on negative emotions: “I feel...” (i) confused, (ii) undecided, (iii) lost, (iv) bored, and (v) frustrated. The use of cards was not mandatory; the participants could talk on their observations and challenges without picking or pointing to cards.

Procedures and data collection. At the beginning of the study, the participants completed a background survey [2] on demographics (age, sex), existing knowledge in data analysis, visualization, and computer use/interaction, and overall motivation in data exploration, using a talk-aloud protocol. Then, they were trained with a 5-minute video tutorial [3], which described the tool features while demonstrating data analysis, and 20-slide printout [4] for future reference. After the training video, the facilitator presented the cards, and asked the participants to think aloud while exploring data, and use the cards if appropriate. To gain familiarity with the tool and the study process, the participants explored the training dataset for 5 minutes. Then, they explored two datasets of their interest, 15 minutes each. The facilitator answered questions about the tool based on the training material. While we encouraged self-driven exploration without external tasks, the participants could pick among five sample questions per dataset [5]. After each dataset, the participants

[2] docs.google.com/forms/d/e/1FAIpQLSd58tfmam5dw9ARW1tf4AKo3MDSZ_wiFyANqxuY0i2urqCH9g

[3] https://www.youtube.com/watch?v=3Hmvms-1grU

[4] docs.google.com/presentation/d/1beCw3KiFjWLdVfgp8EICFPNPiuu2UzX8PFbcirJFQVw

[5] docs.google.com/document/d/1HqK0fJOw2KSA_M59YxQj9PRLqftoHen8bc4yK1Wg5-c/

completed a survey [6] that encouraged recalling both positive and negative experiences, using ten Likert-scale questions based on [107]. The screen and the audio in the room were recorded during participation in the study. To detect the cognitive barriers, I watched the videos and took note of the problems faced by the participants, and their relevant verbal feedback, including feedback based on the surveys. I then classified them across the six cognitive stages.

Participants. We recruited participants using public message boards. The participants were non-experts in data visualization and analysis. The study included pilot-sessions with two participants and reported-sessions with three participants (P1, P2, P3). P1 was a male student in biology, age 18-24. P2 was a female professional in finance, age 40-49. P3 was a female student in food science, age 18-24. All participants were familiar with basic chart types (bar-charts, histograms, line-charts, maps), and none were familiar with advanced chart types (scatterplots, treemaps, node-link diagrams and ||-coords) by name. The self-reported computer skills were novice (P1, P3), intermediate (P2), and none advanced. All participants had experience with Excel, including basic visualizations, data entry (P1), formulations (P2), and none had experience with other data tools. Their motivation to join the study was curiosity (P1, P2, P3), and earning money (P2); $10 for their 1-hour

[6] docs.google.com/forms/d/e/1FAIpQLSeVdSGdQ1VaWeLabVeDUWxRddUbdB9lPVhs7AXu59K4FGiBQA

participation. While this reflects the demographics of the study location, a university campus, their data analysis experience were none (P1) or infrequent (P3), only P2 noting to frequently analyze data "to figure out the yield on investments." The participants were interested in the following domains: movies (P2, P3), traffic accidents (P1), foodborne outbreaks (P1, P3) and Titanic passengers (P2). Per each participant, the use of sample questions to bootstrap exploration was: P2-none, P3-1 question, and P3-multiple questions.

Next, we demonstrate the application of the *Cognitive Exploration Framework* for tool evaluation using the proposed protocol. We report exemplar barriers faced by the participants.

7.1.2 Barriers in Planning Data Analysis

Talking about his experience, P1 noted, "*Maybe I felt like I had too much control, but I wasn't ready for it*", and added, "*I wasn't quite able to figure out what I wanted to figure out*." He stated he was overwhelmed at points (by multiple views), noting, "*It's just a lot to take in. A lot of different elements to consider… I don't understand how to put (a lot of information) together*." P2 set some serendipitous goals, "*Let me see (filter) Clint Eastwood and see what happens*." When picking sample questions, P1 noted on his motivation, "*I want to find something… that I'd personally want to get the answer to*." In addition, to save the limited time, P1 did not want to pick questions that looked complicated to answer. Goals were also constrained by the content of data. P3 said, "*(the data) doesn't have enough criteria to give you a definite answer*", as she wanted to relate diseases from fish consumption to fish production per state. To address the information overload, the tool can be designed to offer simplified authoring interfaces, or to encourage step-by-step guided exploration. Sample goals can be provided from simple to complex as the user gets familiarity using the tool.

7.1.3 Barriers in Planning Interaction

After getting stuck in a question, P1 noted, "*The computer doesn't really know the question that I have (...) I am confused about how to go by answering that question, or if the method I'm using is actually the right way.*" P3 was confused after an ineffective sequence of actions—filtering, locking, and selecting the same histogram bin—where she noted, "*I don't know what exactly I'm trying to do.*" Participants also updated interaction plans and goals given the design and limitation of the tool. To search for specific values, P2 first wanted to alphabetically sort categories and records (not supported), then she used text search, a more appropriate strategy. When P2 wanted to sort few movies by year, which could be achieved using sorting dropdown, she hovered the cursor over movies to automatically highlight their year within summaries. Being satisfied with this approach, she discarded her original sorting plan. We also observed some learning challenges with contextual interfaces. P3 wanted to resort categories in reverse, however was not able to easily find the sorting button because it was hidden by default, and shown only on mouse-over in categories. She later suggested, "*If I had more practice with this, I would definitely be in more control.*"

To address the change-of-plan observation under sorting goal, we updated the design of the tool to include a sorting button within the summary in addition to the sorting option combobox. The tool can also be improved to identify repeated actions to reason about user intent, and suggest relevant actions to help the user plan for interaction. This idea is among those explored in the help system component of this dissertation (Chapter 8).

7.1.4 Barriers in Planning Visualization

With the selected tool, activities related to planning visualization include aggregate selection modes (highlight, compare, filter) and part-of/absolute mode. This contrasts to the charting tools that would require more careful planning to construct effective visualizations. Therefore, barriers in this stage were not frequently observed. In trying to find the most common food outbreak in different months, P3 filtered through multiple months, while highlighting would be more effective. Another barrier was that participants could not plan to execute part-of scale mode change, as no participant in our study used part-of scale. This may reflect that their questions may not have required such views, but also suggests that the limited knowledge about how this mode could be used effectively. The tool design may be improved to communicate and clarify the use of part-of scale mode to answer related questions.

7.1.5 Barriers in Assessing Interaction

Failing to consider filtering selections correctly was a common barrier leading to false conclusions about general, or targeted, populations. After unfiltering a selection, P1 said, "*I forgot that I had still filtered everything for the norovirus.*" When P2 wanted to analyze *survivors* of the *Titanic*, she highlighted *non-survivors* and reached a wrong conclusion about their ages. She realized and corrected her mistake shortly after. P3 interpreted the full bar length in a filtered summary to support her misunderstanding that the complete dataset was selected. P3 also misinterpreted how selections are linked across summaries, saying, "*If I lock (this bar), there's no way I could compare to (another summary) because they are two different things.*" Overall, tracking multiple selection states was found to be a non-trivial task for the novices in our experiment. The tool can be updated to offer simplified interactivity to reduce confusion on dynamic selection changes.

7.1.6 Barriers in Assessing Visualization

P1 was confused about what the numbers represent upon selection, saying "*Is this number representing fatal accidents, or just accidents or is it drunk vs. non-drunk... Ok, I didn't realize there are two different colors.*" P2 tried to understand linked highlighting selections by hovering on different bars, observing numbers, and making connections. P3 had trouble observing exact filtering range within the line chart because of its design. The rounding of histogram end-points also leads to wrong interpretations. With maximum duration of movies at 157 minutes, the high end-point of histogram was rounded to 300 minutes, an anomaly of the log scale used. With this view, P3 interpreted there were movies up to 300 minutes. Real maximum value could be observed by sorting movies in decreasing duration. We later improved the design of our tool by placing the maximum-tip on the scale to the real maximum value, instead of the maximum of the histogram bin range that may exceed true maximum. Filtering range can be more explicitly revealed in interval summaries, and information about what each number presents in the interface can be revealed dynamically.

7.1.7 Barriers in Assessing Data Analysis

Understanding data semantics was a common challenge. P2 asked, "*How do I find the definition of vote count?*" and later removed this summary from the browser. P3 asked, "*What is 'ethnic style, unspecified' (as food type)? That could be anything.*" and then noted, "*This doesn't really affect the program, it's just the data itself.*" Notice that these comments to not reflect to either visualization or the interaction design, and relates to data concepts related to analysis. Unexpected findings raised suspicions, with participants concluding, "*if I'm interpreting right (P1)*", and "*if I'm reading right (P2)*". Acknowledging an inappropriate strategy to reach answers, P1 said, “*I am merely associating these numbers with the question that I have.*” When only 10-20 outbreaks were selected after filtering, P3 concluded about statistical trends and did not discuss limitations of their significance. No participant recognized that some summaries did not include all records, e.g. there were movies without rating information. Another issue was potentially misleading inferences across summaries. When the filtered movies had high-ratings, and kids movies were common, P3 inferred that kids movies had high ratings based on univariate distributions, without querying further to confirm her intuition.

To address assessment challenges in data analysis, providing contextual information about metadata would be helpful. Warnings can be presented when few records remain to make statistical conclusions, or missing records can be highlighted explicitly.

7.1.8 The Factor of Existing/New Knowledge

Our participants were non-experts in visual data analytics. We further limited training and asked a casual short-term use to limit the factor of knowledge. We observed this approach influenced the experience and feedback of our participants. P1 said, “*It’s been a while since I looked at charts… You have to re-familiarize yourself with all the information it represents.*” P2 “*felt discouraged, just in the very beginning, as I was getting used to the tool.*” P3 added “*You never really learn it* *until you actually try to do it.*” These feedbacks point to the active learning experience of the participants during the use of the tool.

7.2 *Insight-based Evaluation with Data Analytics Novices*

In this section, we present an insight-based evaluation [122] of Keshif with visual analytics novices in a short-term, casual, open-ended data exploration study with short training. The goal is to understand insight characteristics and the exploration process, and how the proposed model relates to the process. We aimed to recruit visual analytics novices as they are most impacted by barriers in specifying visual encodings and unconventional visualizations, thus would benefit more from a streamlined exploration flow. The participants of this study used *only* the graphical interface of Keshif (not the API) to explore the data by authoring (creating and adjusting) data browsers. Thus, our participants did not use the JavaScript programming. Our results are comparable to the evaluation of Voyager [151] at high-level, showing that less-skilled participants could reach insights rapidly using Keshif, comparable to participants with more skills using tools that are more sophisticated.

7.2.1 Study Design

Participants. We recruited 6 participants using public message boards (4 female, 2 male, 5 aged 18-24 (4 of them students, all outside computer or information science departments), 1 aged 40-49). Participants were not skilled in visual data exploration, and had not received formal training on visualization. None had used Tableau or similar visual analytics environments. All had used Excel before. The five younger participants had created charts and analyzed data with Excel, and other tools they had used include SPSS (3), Stata (2), and Graphpad (1), showing their background in statistical analysis. They had not analyzed the studied datasets before, they were not domain experts, and they had not used Keshif before.

Datasets. We used two datasets (movies and bird strikes) for the study, also used in the evaluation of Voyager [151]. They are chosen for real-world interest to a general audience, of similar complexity and data types. The movies dataset includes 3,201 movie records with 15 attributes (7 categorical, 1 temporal, 8 numeric), including title, director, genre, sales figures, and IMDB / Rotten Tomatoes ratings. The bird strikes dataset is a redacted version of the FAA wildlife airplane strike database with 10,000 records and 14 attributes (8 categorical, 1 spatial region, 1 temporal, 4 numeric).

Training. The sessions began with a 6-minute video tutorial using a dataset on 5,000 companies, followed by a warm-up exploration of this dataset for 6 minutes. The participants were also provided with 23-page printed slides on the video training. The facilitator answered questions about tool features based on what is covered on the training material.

Study Procedure. We asked participants to explore a given dataset, and specifically to “get a comprehensive sense of what the dataset contains and verbally note interesting patterns, trends or other insights”. Their exploration started with the data imported to an empty Keshif browser. The participants performed an unguided, self-driven exploration without explicit tasks for 15 minutes for each of the two datasets in a think-aloud protocol. Half of the participants explored the movies dataset first, while the other half explored the bird strikes dataset first. After exploring a dataset, participants completed a survey focusing on insight-based metrics. Participants also completed a survey on demographics and data analytics experience.

We did not ask the participants to formulate any questions before the exploration, as doing so might have biased them toward premature fixation on those questions. However, we encouraged (i) changing the axis mode, (ii) changing the measure function, (iii) using compare selections, and (iv) using the map view (if available) so that they could form richer goals and reach wider insights. In our pilot studies, we observed these features were not utilized by novices in self-driven exploration. We did not enforce these recommendations so the participant remained in full control. Per the think-aloud protocol, the facilitator encouraged communication by asking questions such as “What are you thinking right now?” and “Can you explain in more detail?” when communication stopped or the feedback was vague.

Each study session took at most an hour. The participants were compensated with $10 cash. All sessions were held in a university lab using Google Chrome on a Macbook Pro with a 15-inch retina display, and a mouse for interaction. During the studies, the screen and the audio were captured. Surveys results on exploration experience and participant background are also part of the data collection.

The evaluation shares the structure of Voyager's study [151] in terms of datasets and the open-ended exploration task. However, (i) we recruited visual analytics novices instead of experienced participants, (ii) we limited exploration to 15 minutes per dataset instead of 30 minutes (a more casual use), (iii) we provided shorter explicit training (6 vs. 10 minutes), and (iv) we followed insight-based evaluation with think-aloud protocol instead of using bookmarked charts. Our protocol and analysis provide a thorough analysis of the exploration outcomes. We did not compare Keshif and Voyager side-by-side because the tools differ in visualization model, supported tasks, charts, and insights. For example, Voyager does not support interactive linked selections, and map views. Visualizations in Keshif are always aggregated, and do not include scatterplots and its variations. Keshif does not model data exploration as exploration of alternative chart types, but of aggregated summaries with linked selections.

7.2.2 Insight Coding

To detect the insights, I transcribed the verbal feedback of the participants. Using the transcripts, I identified statements that presented an insight on the data content as a single, cohesive proposition. I did not consider statements at a strictly visual level as an insight (such as "there is a peak"), unless participant related it to the data content. I also did not consider restatement of a previous insight as a new insight. Then, I coded attributes of each insight using two passes on the transcripts and the video captures. In the second pass, I extended insight categorizations, and confirmed existing codes. I also noted *hypothesis* statements as a question or an explanation of a trend that can neither be con-firmed nor denied within the dataset. A hypothesis commonly relates to participants prior experience and knowledge. The insight coding results are accessible and explorable as a Keshif browser at bit.ly/1Vbs40c.

I coded each insight on its insight-based characteristics and the interface state at the time of insight.

- Text: What is the insight? (transcription)
- Time: When was it noted? (seconds elapsed)
- Correctness: Was it correct?
- Feature: Was it describing a fact, min/max, distribution, comparison or correlation?
- Data types directly relevant to the insight (summary type (categorical, numerical, time, map), individual record, etc.).
- Selection state (the number of filtered, highlighted, compared summaries)
- Measure function (count, sum, average)
- Measure label (absolute, percent-age)
- Axis mode (absolute, part of)
- Dataset
- Participant ID
- Dataset order (First or second).

Next, we describe the insight categories and the data features they reflect.

- **Fact** describes a property of a record, an aggregation, or a basic observation that does not describe a trend. Examples include "*84 of them are causing minor damage*", "*That was Delta Airlines*", and "*it is an adventure movie*".
- **Min/Max** describes the most/least common feature in the data. Examples include "*B737-300 cause the most bird strikes*", "*Dramas typically make between 20 and 300M*", and "*[Movies were released] Mostly during this time period, between 2004 and 2007.*"
- **Distribution** focuses on the variations and trends within a data attribute. Examples include "*So, the comedy movie ratings.... it is kinda spread out, they are not that consistent.*", and "*It has a large variety of genres, from drama to action, horror.*"
- **Comparison** describes two or more specific aggregates, records, or selections. Examples include *"[Beloved] has a higher Rotten Tomatoes rating than it does IMDB rating."*, "*[After filtering] All of a sudden Dallas falls way down*", and "*So the average cost, is, I guess it's around the same [as the overall trend].*"
- **Correlation** describes relations across attributes in a dataset. The relation may be based on a subset of the data. For example, "*not many of that (highest grossing) were rated R*" relates gross sales with the R rating, describing a trend. "*It looks like they gave pretty good scores to original screen plays*" is another example.

The verbal statements may not reveal the details of observations and analytical thinking of the participant in the think-aloud protocol. Overall, the expectation from the participants is not articulation of the complete exploration state, but sharing important aspects of the insight clearly. While encoding insight correctness, we had a permissively positive basis. For example, when the participant noted, "the most strikes are in Pittsburgh region" on a filtered data, we consider it *correct*, even though the filtering criterion is not stated. An incorrect statement example is "*Portland has all their hits being the one species of bird*", because Portland has a variety of birds contributing to its bird-strikes. Some statements were encoded as partially correct when the trends could not be easily confirmed, or statements were vague. Examples include "*Comedies make that much out of that much money*", and "*the worldwide sales (...) definitely move*". Facts on personal experience are not coded for correctness. The confidence in the insights is assessed using post exploration survey.

The coding of the interface state (selections and visual modes) enables understanding how the tool is used and at which stages the insights were obtained/shared. However, the insight may not relate to all such states. For example, when there are multiple compare selections, the insight may describe one distribution rather than a comparison across multiple distributions. Lastly, an insight might relate to multiple data types. For example, "*Comedy was one of the top grossing in the US*" relates to both genre (categorical) and numeric (gross sales), while describing a min/max feature. The data type of an insight would be noted as "map" if the map view were used to describe the location in the insight.

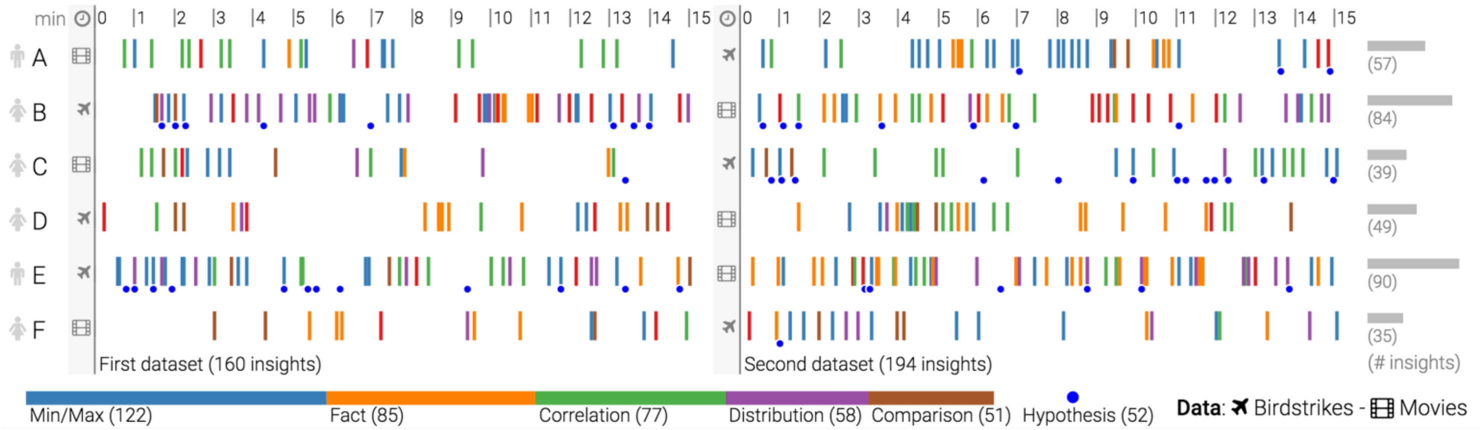


Figure 20- The timeline of the data insights of the participants in the user study. Each participant explored two datasets for fifteen minutes. Insights are color coded into categories: fact, min/max, distribution, comparison, correlation. Insights with multiple features are colored red. The colored legend shows the total count per each insight category. There are 354 insights and 52 hypotheses noted in the 180 minutes total.

7.2.3 Analysis and Results

The temporal overview and characteristics of the insights of our study participants is shown in Figure 18. Our participants reached 35 to 90 insights in total across two sessions, with ~2 insights/minute on average. During the studies, we noticed that personal differences were a big factor in the variances. To quote the participant with the lowest number of insights (F): "*I personally would have gained more from this experience if I was asked to perform specific tasks. (...) I'm not one who necessarily feels inclined to just play on my own. Some people are, some people aren't.*" Therefore, each individual may not be inclined to reach data insights or perform well when unguided, a challenge in broadening public use of data exploration.

In comparison, Voyager [151] reports 12.5 bookmarked charts in average per 30-minute data exploration session by skilled participants using the same datasets (and 10 charts in average for a drag-and-drop visual specification). Studying the effect of display size across two conditions (targeting large displays), Reda et al. [114] report about ~1.2 insights/minute. Their participants were mostly computer science graduate students. Liu and Heer's study [90] on the effect of 500ms interaction latency using imMens [91] system with 16 participants skilled in visual analytics (R and Tableau) report a throughput of ~1.9 insights/minute, based on observations or generalizations on two datasets explored 30 minutes each. The participants in this study had no visual analytics experience, and achieved a high insight throughput with short training.

Insights of the participants most commonly described the min/max features in the data attributes (34%). 79% of these insights were on categorical data, suggesting that auto-sorting influenced the exploration process. 24% of the insights included simple facts, 38% of which were on individual records (an individual movie). Correlations were also common (22%), as they also include statements that relate two attributes by first selecting an aggregate on one, and observing the trends in the other. Comparisons were the least common type of insights (14%). Note that an insight may have multiple types. 28% of the coded insights had more than one feature. The analysis shows the variation in the types of insights shared by our participants. Arguably, their experience in statistical analysis (through course and personal work) may have guided them to look for and report detailed insights, even though they were not skilled in visual analysis.

The participants had insights most frequently under the default settings that create a familiar faceted interface with absolute record counts and basic distributions. 96% of the insights were made under absolute axis mode, 92% were made with absolute measure label, and 90% were made under the count aggregate measure. Remarkably, the participant with the most insights (E) used the default settings throughout. In contrast, 78% of the insights, a high ratio, were reported with some active data selection. Highlighting was active for 34% of the insights, and filtering was active for 55%. However, comparison was less common, only 18% across all insights.

Our results show that non-default, less familiar settings for expressive richness are more likely to lead to incorrect statements. Insights made under average or sum measures were incorrect 24% and 20% of the times respectively, compared to only 5% for the default count measure. A substantial difference in accuracy was observed for compare selection as well. 35% of the incorrect or partially correct insights had at least one compare selection at the time the insight was shared, another significant trend in our data. The compare selection on locking interaction is an unfamiliar design compared to filtering and highlighting actions, which may explain the lower accuracy under its use.

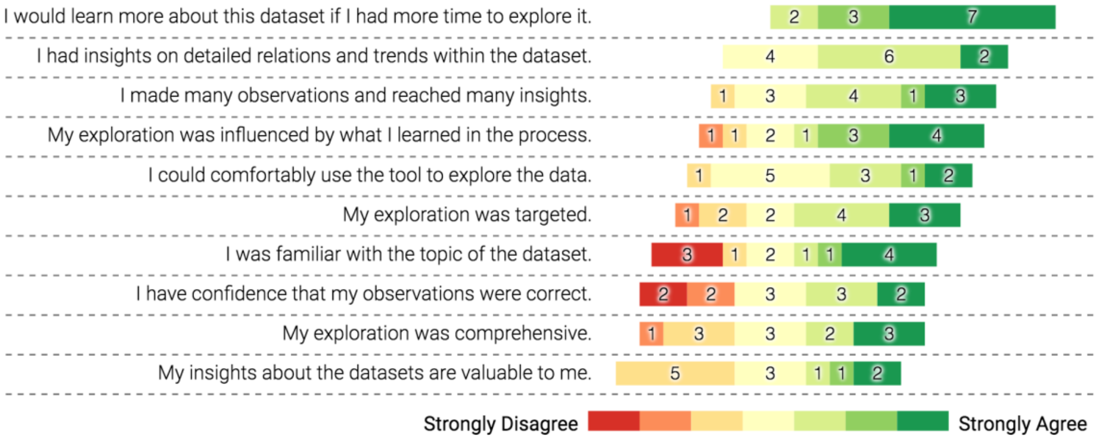


Figure 21- Post-exploration survey results focusing on the self-evaluation of data exploration experience. Each question includes 12 responses, across six participants on each dataset they explored. The color shows agreement, and the answers are aligned on neutral response, and sorted by mostly-positive agreements first.

The survey results are summarized in Figure 19. Participants collectively agreed they could reach more insights given more time using Keshif. Participants also positively noted they could observe detailed relations and trends, although not comprehensively. The least positive feedback was the perceived value of their insights. This follows that the participant's familiarity in the domains received the strongest negative ratings, as lower familiarity with datasets or domains is likely to lower the value of insights for the participant. The confidence and value in exploration also reflect (low) confidence in data samples. Participant C noted, "*I didn't know where the list came from, how the data was collected (...) I don't know how much value they have to me, because I don't know how much I can trust them [dataset]*." Participants responded more positively to their exploration being influenced by what they learn, rather than being targeted. The responses to comfortable usability of Keshif were among the positive feedback as well.

Our results suggest a learning affect over time with improvement of outcomes and satisfaction. More insights were reported in the second session compared to the first (194 vs. 160). Survey results (Figure 19) show that participants were more comfortable in using Keshif in the second session as well (4.3 vs. 5.3 average on 7-point Likert scale answers).

7.3 Evaluation of AggreSet

To evaluate the design of AggreSet, I conducted user studies with two complementary approaches. First, I conducted expert reviews to identify strengths and weaknesses of AggreSet as observed by visualization experts using multiple datasets. Expert reviews in visualization have been shown to help detect usability and design issues, and yield qualitative results [140]. Second, I conducted a case study where domain experts analyzed complex data, with the aim of uncovering the usability and usefulness of AggreSet and analysis strategies. In both evaluations, I collected qualitative feedback on usability and design features during the studies and in semi-structured post-study interviews. In both cases, I used the feedback to improve AggreSet design as presented in this dissertation, and to identify future work.

7.3.1 Expert Review

We recruited three visualization experts (senior researcher P., graduate student D., and industry professional F.) and asked for their honest feedback in 1.5-hour sessions. We first used the movie dataset to demonstrate set exploration in multiple dimensions and set-pair strength. We followed with the Les Miserables characters dataset to demonstrate subset relationships and perceptual set ordering. We encouraged the participants to think aloud, and interrupt at any point to ask questions, make, and share observations. The following summarizes some of their comments and observations.

Before introducing the matrix view, we asked D. which movie pair would have the biggest intersection, to which he replied, "*I cannot tell, I don't have the overview. If I knew which ones to compare, I'll use (selection), but I don't know. You need other ways to see which pairs are most interesting*". With genre matrix enabled and high-rated movies previewed, he said, "*The drama and war (movies) seems to be very good… I immediately found (the intersection). Now I want to see the release date of war and drama, and 4-star rating*". By filtering and selection, he found some movies he liked. This exemplifies the utility of set-matrix view.

The participants also developed strategies to effectively explore data using AggreSet. F. noted, "*The bar chart serves as a key to the matrix.*" He continued, "*For navigation, you have the matrix,… the 2D space you are maneuvering in… For interpretation, it is good to look back at the bar chart… That is two of them complementing each other*". Upon selecting a genre-pair intersection and analyzing the selections for a while, F. said, "*You are actually showing, out of the intersection of 2 things, multiple set of intersections… It is a little bit of a mind-bender*". D. commented likewise upon selecting comedy, "In other views, it tells me the percentage of *comedy* in those overlaps of the other movies… I am comparing three basically".

When explaining the potential complexity of the interface, F. said, "*It is a lot of information. Once the person masters it, and then they have at their fingertips a lot of information in a very little space. It is just that getting there takes some effort. I understand you are trying to minimize that effort so that the user can quickly master the way to interpret this chart*". This follows our suggestion that intersection characteristics should be queried after the set-list and set-degree, as part of overview-to-detail exploration. As F. notes, "*When you hover with your mouse on top of the matrix, showing (previewing) those intersections is when it is a little overwhelming*". Commenting on matrix readability, F. also said, "*Interacting with the matrix on the horizontal level and on the vertical level (for a single set), that takes some time. It is not something that comes to you immediately, like differences in (strength) colors do*".

The participants found the zoomed-out matrices dense overall; visualizations on small circles were not easy to observe. However, D. added, "*This makes sense. I start with the overview, and then I drill down to the area… It helps me… because I have made some observation based on the high-level small pie chart. I want to confirm, so I will drill down and see exactly how it looks like.*"

The relative mode with percentage distributions was favored among all participants; P said, "*I like this (percentage) view better for doing… complex queries*". Subset relations were found the most complex concept, although the participants could understand the relation and encoding through some exploration.

At the start, F. noted, “*I am trying to understand why (circles) have outline… Three states: Total outline, half outline, and no outline.*” After exploration, F said, “*This is one that I think some teaching aid would be great.*” And P. said, “*I like that I was able to do it, but it was hard.*”

We implemented several changes to our design following the expert reviews: (*i*) An earlier design visualized set similarity (strength) by mapping to circle size. This made understanding circle-size mapping harder as it overloaded the element-count mapping. We updated AggreSet design to use color-coding for strength metric as suggested, and to use circle size for element count only. (ii) We noted that color-coding was ineffective with varying and small circle sizes with the cell background. Thus, in relative-mode (strength), we chose to use full-size circles and remove cell-background. (iii) *Relative-mode* and strength metric are linked together, effectively encoding strength as a *relative* set-pair metric. This simplified AggreSet while making it easier to understand and use. (*iv*) An earlier design used a 3-second mouse point-wait to select an aggregate for comparison. D. stated, “Hovering means I am thinking, it doesn’t mean I want to compare”, and P. said, “I’d like to turn it off when I don’t want it.” Users converted to using their hands to point things instead of using mouse, changing their behavior to overcome the issues with the specific design. We then designed an explicit control using 🔒, which also visually reveals the selected aggregate. (*v*) An earlier visual design for comparing distributions (black lines) was an enclosing section (▭), which suggested stacked-charts semantics for some users when previews were enabled (▭), thus complicating the visual language of AggreSet. We changed the bordered design in favor of a simple bar extending from the baseline (▬).

7.3.2 Case Study

I conducted a case study with two assistant deans of the undergraduate studies department of a large public university analysing student degree and course enrollment data. First, the participants had access for a few months to a version of the visualization without the set matrix, but with histograms and the data preview and selection. This allowed them to look at categorical and numerical aspects of the multivariate student records, including set-typed data using set-lists and set-degrees. They used the tool a few times on their own during this period. After we developed the set matrix design, we performed data exploration including the matrix view in a 1.5hr session with the two participants together. The aim was to capture the cognitive and reasoning processes of novice visualization users with rich data in a limited time using AggreSet. Thus, we used pair analytics [9]. The participants collaboratively formed questions, observed data, and generated insight. I acted as “driver”, demonstrating features (from set-list overview to set-matrix detail) and expressing their queries.

First, the participants analyzed 175,000 students and the degrees they received, along with their birth year and gender to provide context. 131 most common majors with at least 100 students were the sets over students (elements). (*i*) Early in the exploration, the participants wondered why there were multiple majors on "*Math*". The driver performed a search within the degree-list to select all majors with "*Math*" (a ∪ query by text input). The resulting visualizations supported their hypothesis that one of the "*Math*" sets was "*Applied Math*". (*ii*) When the driver previewed the *Economics* selection, they observed the other degrees received by students in *Economics*. (*iii*) They wanted to explore students who did not receive a degree. First, they tried to generate hypotheses about their distribution trends and what the data represents, such as whether the declared yet unfinished degrees were included in the reported numbers. Upon selecting 0-degree students, they noticed these students were younger, suggesting many were possibly still taking courses. To improve their outlier analysis, they wished for more data context in the browser, such as entry term and majors declared. Upon selecting students with 1-degree, they noted, "Those (selected) are all the people that earned 1 degree… (The rest) are the ones with double majors". (*iv*) The driver then enabled relative mode. Upon selecting females, they noted, "67% of the sociology students are female. *It makes more sense this way*". Upon selecting 1-degree students again, they noted some majors had very few students with multiple majors, enriching their knowledge of the more demanding majors.

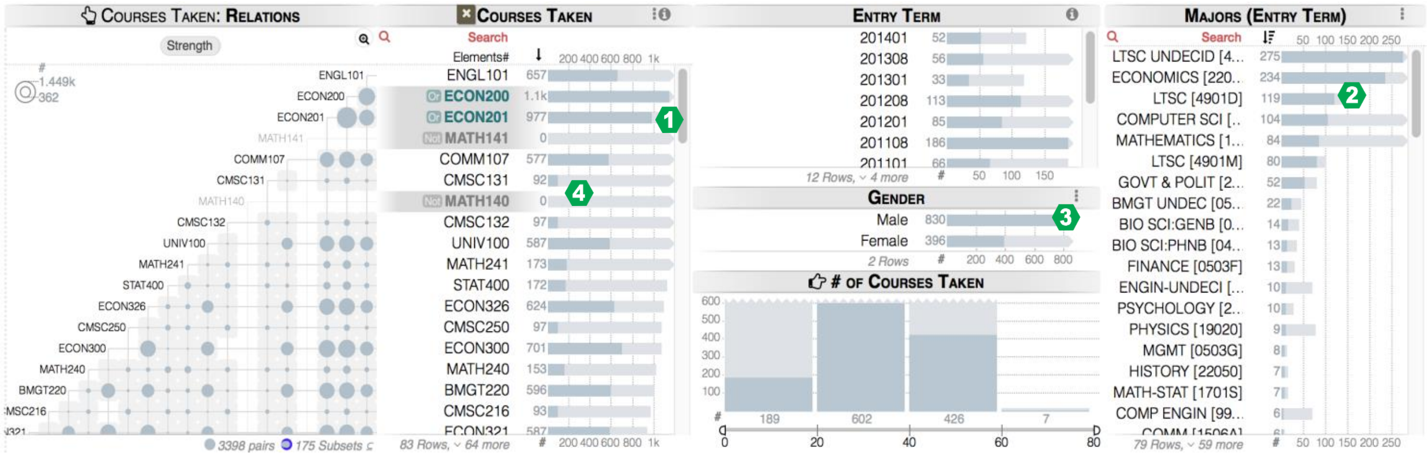


Figure 22- 83 courses taken by 4,300 students. ❶Students who took ECON200&201, but not MATH140&141 are selected. Updated distributions show the other courses taken, the total number of courses, gender, entry term, and the majors of these students. ❷ Most of these students are in LTSC (undecided), or in ECON, ❸ as well as with a male majority. We can notice emerging patterns in the top courses after filtering. ❹ In this filtered selection, CMSC courses are less common than other ECON and core courses.

Next, the driver showed the major (set) matrix. One participant immediately pointed out “this means there are more people that have accounting and finance. The bigger gray circle means there is more people”. When the driver asked about any trends they detected, one said “All those double majors with X… Department of X would be very interested to see this”. Since only a limited number of majors could be shown at once, one asked, “Does it ever get wider this way?” suggesting outside the triangle, at which point the driver panned the set-matrix. They explored various departments and their intersections through rapid result previews. Then, the driver enabled major-pair strength visualization. First, enlarged circles made it easier for them to see intersecting majors, as it was a stronger cue than the gray cell background in the default view. One noted, “Darker color means a higher percentage than the one next to it (lighter)”, while the other complimented this statement by saying, “When we looked at that gray view, it was actual numbers.” After further discussion, they concluded, “While there are a lot of marketing and finance (students), there is more accounting in finance, of the total numbers.” Few students received three or more degrees, limiting exploration of higher-order intersections in this dataset.

Next, they analyzed 4,300 students and the 83 most-registered courses (Figure 20). They noticed that few students took 50 or more courses. Note that the sets (courses) are densely connected, and the set-degree distribution has a wide range. By selecting those students, they explored their majors and courses, and generated insights regarding degree programs, and potential effects of course count on student success. They also noted “This isn’t showing courses they are taking *above* what they would have needed”. They needed a new form of set-summary that would show the additional courses the student is taking compared to declared major requirements, a more complex data setup. When the matrix view was shown, they noted large pair intersections of some common core courses (such as English), as well as courses that are prerequisites to others. Noting of their previous experience analyzing this data without the matrix view, one said, “*This view would have allowed us to do what we wanted to do more easily than what we did. What courses they take, and what they take together*”. When the strength metric was enabled, they noticed courses that had consistent colors among all its intersections, which meant that they had no strong relationships with others. They went on to analyze common properties among students that did not take some specific courses.

7.4 *Applications over Multiple Domains*

In this section, we describe how data authoring and exploration using Keshif aims to fit in a workflow of data analysis / exploration, how it is applied over multiple data domains with real-world datasets, and its impact and external use so far.

7.4.1 Sample Use Case Scenario for Data Journalism

We present a use case scenario for data journalism using an existing dataset. The goal is to give a clear demonstration of a sample, supporting workflow using Keshif, and various tasks it supports to enable data exploration and sharing.

A local newspaper wants to run a story on the homicide victims in the city to inform its readers and policy makers. The journalists track ten years of reported cases, describing the location, motive of the murderer, police investigation status, as well as the name, age, gender and race of the victim in a spreadsheet. They add the neighborhood of each homicide using its point-location in data pre-processing to reveal spatial trends as a regional overview. They also generate a GeoJSON file describing the neighborhood boundaries, indexed by neighborhood name. Then, to explore this structured data rapidly, they import it into Keshif.

Keshif first reveals the number of homicide victims (2,294) and the list of attributes with simple distribution previews. Interested in demographics, the journalists add age and sex summaries, which immediately reveal that 20-40 age is the most common range (1.4k), and that this population is significantly male (2.0k).

They change measure labels to percentages, and note that 62% was between ages 20-40, and 89% were male. To analyze if female victims had different characteristics, they filter to female, and notice that only 44% were between ages 20-40. The change of distribution shows that the age of female victims were higher overall. They confirm this observation by clearing the filtering, opening percentile chart of age, and highlighting the female victims. Using the distributions in a simple 1D chart with color-coding, they note that the median age of male victims was 26, while the median age of female was 31. They take a note of these numbers for their news story.

Next, the journalists are interested to see temporal trends in motives. They quickly preview most common motives by mouse-move: arguments, drugs, retaliation and robbery, and observe their temporal trends. Knowing that the city had been taking measures to reduce drug violence, they highlight drugs again, and find out that over ten year period, the number of drug homicides decreased 84% (49→8). However, they also notice an overall decreasing trend in homicides. Therefore, they lock-select drug related homicides for comparison, and change to *part-of* scale. This reveals that the relative ratio of drug related offenses dropped 21%→8%, a smaller, yet still significant 62% decrease. They note these trends may be due to the new drug policies and policing in the city.

They save the homicide victims browser with selected attributes, and share the link with another colleague. When she opens the link, she notices they may have not looked at manner of homicide, and its relation to neighborhoods. She adds the neighborhood summary and changes to map view to study spatial trends. She also adds manner summary, noticing that shooting was the largest motive by far (1.8k victims), followed by stabbing (246). To explore patterns, she moves the mouse across motives and observes changes on the map. She quickly notices homicides with stabbing have a different distribution than the overall: Central regions of the city have more victims of stabbing. She sends a note to her colleagues, along with a link of the updated browser to reproduce the result.

Collecting many insights over the process, along with other resources from public officials, interviews and high-profile cases, the journalists are ready to write their story. They create simple annotated charts with constrained interactions to highlight the individual trends they observe, and link them in their story. They embed the Keshif browser in the end of their report to make it freely explorable. They invite readers to look at their own neighborhood and to find the information valuable to them in Keshif browser. This sample workflow is based on the example at http://www.keshif.me/demo/dc_homicides, and can be reproduced live and online.

7.4.2 Public Web-based Data Exploration at www.keshif.me

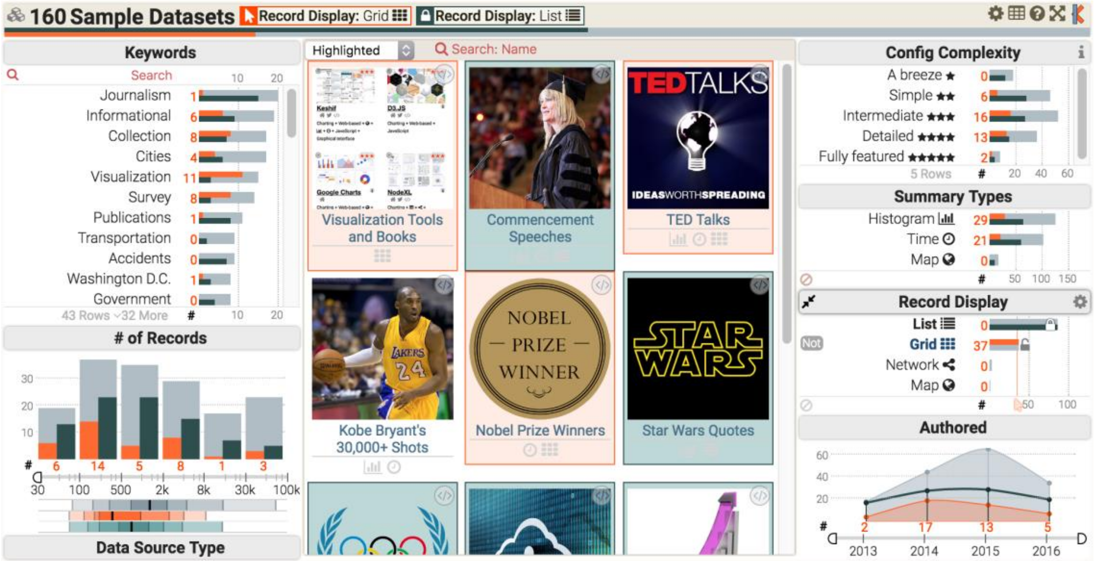


Figure 23- A Keshif browser that displays the list of public Keshif demos (sample datasets) on 160 datasets available at www.keshif.me.

I have created public Keshif browsers for 160+ datasets across many domains including journalism, surveys, transportation, cities, food, finance, entertainment, politics, and personal data, some of which are discussed and showcased in this paper. Figure 21 shows a screenshot from the collection of public data explorers made available online at www.keshif.me. The range of datasets demonstrates the generalizability and flexibility of our model and implementation. While importing, studying, and testing many tabular datasets with a wide range of data characteristics, structure and formats, we incrementally refined Keshif's design, features, and implementation over three years. These sample public data browsers are created and maintained using the JavaScript API by the lead author of this paper.

In one-year period, data browsers on keshif.me have been visited 100k+ times by 50k+ users in 62k+ sessions based on Google Analytics. Keshif source code repository on GitHub has been starred 400+ times, and forked 100+ times at the time of this writing. The browsers have been shared on social media posts 440+ times in the recent year as tracked by AddThis web service. The project page includes a list of mentions and references to Keshif, including paper citations, research proposals, talks, resources, and interviews.

7.4.3 Other Use Cases

In addition the targeted user evaluations and high-level overview of applications of Keshif above, I have worked with multiple partners and organizations across different domains to enable visual data exploration in new and emerging datasets. Below is a summary of these efforts.

- I collaborated with course coordinators and instructors at the Department of Communication to help them analyze course consistency across multiple sections and years of offerings. Our approach enabled them to explore course structure, student success, and grading and feedback practices. A paper published at the annual meeting of the National Communication Association was recognized as the Basic Course Division Top Paper [7].
- I collaborated with the National Socio-Environmental Synthesis Center (SESYNC) in their *"Data to Motivate Synthesis"* program, which aimed to provide

tools to build improved capacities to perform collaborative research on rich open government datasets across food, energy and water systems. My role was to help them build web-based user interface for data visualization and discovery using Keshif. Specifically, I build tools to explore multiple attributes (indicators) across multiple datasets, and then to explore spatial and numerical trends and characteristics across datasets. This effort has lead to the development and improvement of the spatial display capabilities of Keshif, and was further customized to meet their needs to explore both watershed and county boundaries on a single integrated map view with supporting histograms.

- I collaborated with the Teaching and Learning Transformation Center (TLTC) of University of Maryland to collect and analyze data relating to the online course information for teaching and learning.
- I am currently collaborating with the National Consortium for the Study of Terrorism and Responses to Terrorism (START), transforming their datasets on global terrorism events, foreign fighters, and narratives shared by terrorism and counter-terrorism organization, using rapid prototyping and continuous feedback and integration.
- In addition, I am also volunteering with CodeForDC, a local branch of Code For America (https://www.codeforamerica.org/), focusing on providing transparency for political campaign contributions in Washington DC races for council and mayors over the last 4 election cycles. Specifically, Keshif is used to provide an overview,

and allow rich exploration, of contributions across selected candidates or campaigns. This project is available at http://codefordc.org/dc-campaign-finance-watch/ .

The open-source release and public demos and demonstration of Keshif also allowed other people and organizations to use Keshif for their own datasets. Below is a quick summary of some of the public use and acknowledgements we have received.

- Society for Industrial and Organization Psychology (SIOP) created a Keshif browser for the 8-years of their conference programs, using the Gist-based sharing and graphical authoring, which received favorable feedback on their Twitter channels. More information is available on their website [133].
- A recent public feedback on the project mailing list [7] by a software engineer at School of Computer Science, University of Manchester notes that "(Keshif) is proving to be a really popular way of viewing data in the organization I am using it for."

[7] https://groups.google.com/forum/#!topic/keshif/FZNc81Hd7EY

Chapter 8. Integrated Contextual Help for Data Interfaces

"Combined with adequate presentation techniques, the old Bauhaus notion of doing more with less can come true."

Hans van der Meij and John M. Carroll,

in Minimalism Beyond the Nurnberg Funnel [28]

The influence of knowledge to visual data exploration can be modeled as a dynamic construct that can be extended with new knowledge of data and of the application over use, in addition to existing knowledge that steers the exploration process (See Cognitive Exploration Framework, Chapter 3). Even if a first-time user of a data analysis tool has analysis skills, they still need orientation and training on the go. Novices have an even greater need for training for the basics of visualization and interaction features. This chapter addresses the challenge of supporting a wide range of users with various backgrounds with various needs on the go, and providing this support as quickly as possible with minimal intrusion to original task on the data interface.

In this chapter, we first present contextual features in visual data interfaces, including data-driven features and the use of context for help seeking and help-comprehension. Then, we describe the design of *HelpIn*, including its help modes and implementation overview. We evaluated HelpIn with first-time users of Keshif who were mostly data novices, in comparison to a non-contextual version. While task completion/progress performance was similar under both conditions, one contextual help-seeking mode, Point&Learn, was found the most useful subjectively in participant feedback, and increased objective performance overall.

8.1 Context in Visual Data Interfaces

The overarching idea of contextual help is to use the current context of an application to provide customized and targeted help to facilitate comprehension. Specifically, contextual features can be used to filter and rank help material by relevance, and also to present dynamic and integrated answers. Below, we provide an overview of (i) data-driven, (ii) application-driven, and (iii) history-driven contextual features. In each category, we exemplify its use to find relevant help material, and to present integrated answers. Our contributions include identifying *data* as a first-class context category with multiple use cases, and identifying how context can be instrumental for both help seeking and comprehension.

8.1.1 Data-Driven Context

This context category describes the features of the underlying data, such as data types, distributions, and relations, and the states of data visualization and queries.

Help Seeking

Relevance of help topics can be defined by existing data types, features and query states. For example, topics concerning computing temporal characteristics, such as extracting month, would be relevant *only* when data has a temporal component. If the data is not filtered, a topic on clearing filters will not be relevant. Topics can reflect the existing data visualization types as well. For example, selecting data by geographical regions would be relevant only when a map is visible. Ranking topics on relevance can reflect frequency of data types as well. For example, if numeric data type is common, tasks on numeric data can have higher priority. (See Topic Relevance and Ranking section below for details.) To further support data-driven help seeking, data glyphs or visualization components can be directly selected to retrieve their contextual information. Topic names can also reflect visualization states. For example, if application allows two modes for visual scale, the topic name can reflect the alternative (target) setting.

Help Comprehension

To try to aid comprehension, help descriptions can highlight appropriate data types, features, or distributions. For example, linked selection would be best demonstrated by exemplifying a selection that reveals interesting features. A heuristic approach may select a data aggregate that includes about half of the data. To pick examples to describe the effect of actions, such as with tooltips and data information, similar heuristics can be applied. In addition, the descriptions of help can include information about the data distribution and features. For example, description of a record can include its sorted rank, or multiple encodings in a visual glyph can be clarified with legends and exact values. The answer can also respond to the visualization state and visual encodings. For example, in a scatterplot, requesting help information on a filtered-out dot (record) can describe *why* it is filtered out (i.e. which query it fails). If points are color coded by category, the description can describe the color mapping and the category of the point.

8.1.2 Application-Driven Context

This context category describes the application state, as well as UI components (such as widgets, buttons, menus, etc.) that are visible or are reachable through interaction.

Help Seeking

The help material relevance can reflect active application settings. For example, if none of the panels in the interface is collapsed, “uncollapse panel” topic will not be relevant. Relevant help material can be requested by interface components, either by direct interaction (pointing), or using a textual list of components. For example, pointing to a sorting icon can suggest “Change sorting criteria” and “Sort in reverse” topics. Considering help-system as part of the application status, presenting related help topics to a *selected help material* can expand user’s repertoire and provide supporting information. In addition, the position of help panels (and tooltips) can be adjusted to avoid, or minimize, overlap with highlighted components. Location-aware presentations has been shown to increase training performance [75].

Help Comprehension

Help can be presented by highlighting relevant interface components, such as where to click to change a setting, to minimize the distance between answer and action. The help *descriptions* can be responsive, describing the current state, and the role and use of alternative states.

8.1.3 History-Driven Context

This category is based on the actions performed by the user.

Help Seeking

The help topics can be ranked by recency or frequency of usage, emphasizing either more/less or most/least frequently used features. The action history information can be used to refresh the user’s memory or clarify most recent interactions, or to enable discovery of new (or unused) features. User actions can also be used to infer high-level behavior. For example, if the user scrolls frequently in a visualization panel, the help system may suggest maximizing it. Or, when user frequently highlights two categories, the system may suggest comparing the two by a locked selection.

Help Comprehension

The answer may exemplify the most recently used components if there are alternatives to achieve the task.

8.1.4 Topic Relevance and Ranking by Context

Ranking and filtering help topics using contextual information can offer more relevant options up-front and improve navigation. In HelpIn, each topic defines a list of context features (such as data, application, or UI state) that need to be satisfied in order to be relevant. The ranked topic relevance is computed using the following strategies:

- The weights of satisfied context features are added. The richer the required context features of a topic, the heavier the topic weights. The context feature weights are defined in help material, and reflect the importance and commonality of the feature within expected interface use.
- If a context feature returns a count of satisfactory features (such as the number of numeric attributes), this value can be used to adjust the context weight, so that more common features have higher ranking. HelpIn uses a $1.05^{(x)}$ multiplier, where x stands for the matching features, if relevant.
- If topics reflect multiple targeted UI components (such as by recognizing UI hierarchy), the topic that relate to more specific components are ranked higher by adding adjusted weight based on component specificity.
- If topics are ranked by recency of use (history), a score that reflects if and how recent the feature was used is added. When ranking for the most recent first, the score is inversely proportional to how recent the feature was used.

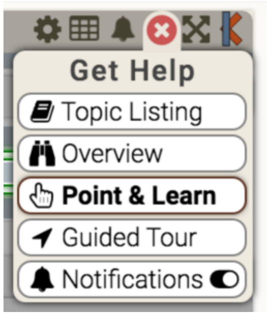
When ranking for unused first, the score is highest for topics that have not been used, and lowest for the most recently used.

- The topic self-weight (if defined) is added. This allows adjusting ranking per-topic irrespective of the context.

8.2 HelpIn – A Contextual In-Site Help System

HelpIn is a contextual help system designed as an overlay on top of a visual data interface. It blends a semi-transparent help overlay with the underlying interface in the background, supporting the user to stay oriented. HelpIn features a stencil approach [75] to highlight interface components that are selected by the user (Figure 24), or to present part of a help topic answer (Figure 22). To demonstrate HelpIn, we use a data exploration tool, Keshif [164], as the underlying visual data interface.

8.2.1 Seeking Contextual Help



To address different use cases for help seeking, HelpIn includes five modes: (i) Topic Listing, (ii) Overview, (iii) Point & Learn, (iv) Guided Tour, and (v) Notifications. Accessible by clicking a ❓ icon, these modes reflect a mix of push/pull approaches [67]. With the *pull model*, the user initiates the request to get help and pulls (searches for) for relevant help topics. The role of HelpIn is to evaluate the active context, identify relevant help, and rank them on relevance. With the *push model*, the role of HelpIn is to suggest help based on the inferred context, therefore pushing the help [98]. The push model may allow the user to carry out their tasks more efficiently through unintentional learning [124]. In HelpIn's design, respective to the push/pull models, ● *Topic Listing* reflects an explicit pull action with the user controlling the topic search by keywords and text, ● *Overview* presents a short narrative summary of active data analysis state, ● *Point & Learn* makes it easier to pull help based on the pointed interface area, ● *Guided Tour* is initiated (pulled) by the user, yet the sequence of material is pushed by the help system, ● *Notifications* reflect the explicit push mode by monitoring application use, and suggesting specific help directly.

8.2.2 Presenting Contextual Help Instruction

Once help material is selected, its answer is presented in-situ, that is, the material is fully integrated into the interface (Figure 4 and 5). The design of HelpIn highlights relevant components, uses tooltips to describe the actions, and provides descriptions that reflect the selected data glyph, component, and active application settings. After the user reads the answer, they can select another help topic or component, or activate an action by interacting with the answer.

8.2.3 Instructional Design

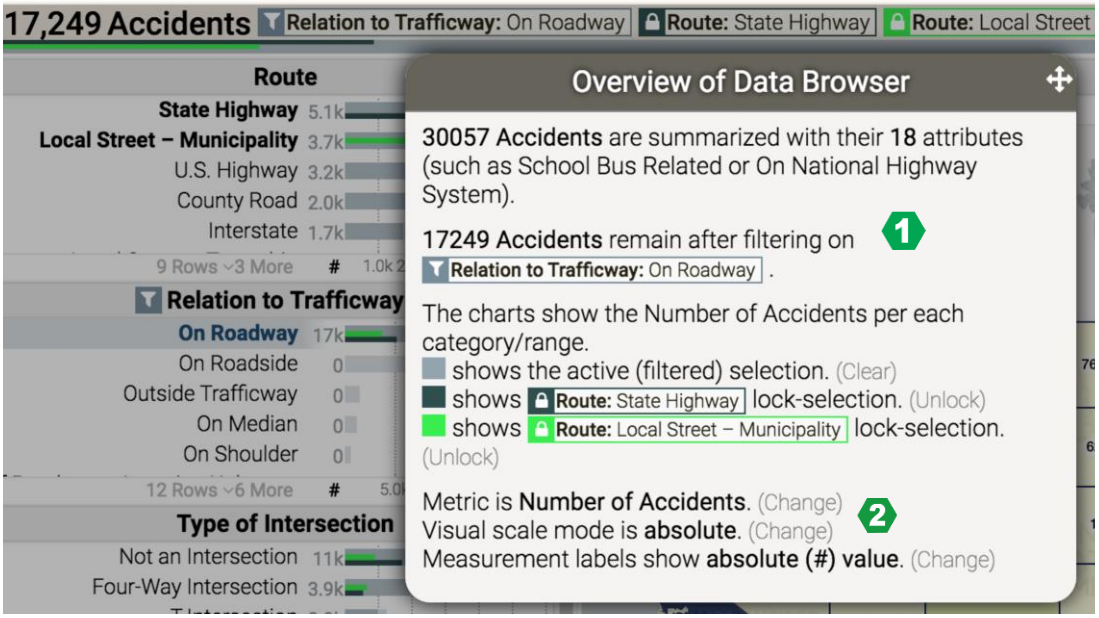
In our instructional design, topics reflect the unit tasks of the application, i.e. tasks that can be completed with very few actions and change only one aspect of the interface. We strived to achieve minimalism and simplicity in the help language, to reduce the number and complexity of the words, and to maintain consistency across all components [27]. Our material reflects the design language of the underlying application, such as using the same icons and color design, thus reducing the extraneous cognitive load [30] in translating help information to the current interface state.

8.3 *Modes of HelpIn*

In this section, we describe the design of five help-seeking modes and the topic answer, which provides instructions.

8.3.1 Overview

The *Overview* mode (Figure 22) shows a narrative high-level summary of the active data analysis and interface state. It orients the user in data analysis and exploration by describing multiple relevant features that affect the active view (such as active selections, and visualization modes). It also allows the user to see how these modes can be changed by linking to individual help topics.



Overview of Data Browser

30057 Accidents are summarized with their 18 attributes (such as School Bus Related or On National Highway System).

17249 Accidents remain after filtering on ❶
Relation to Trafficway: On Roadway .

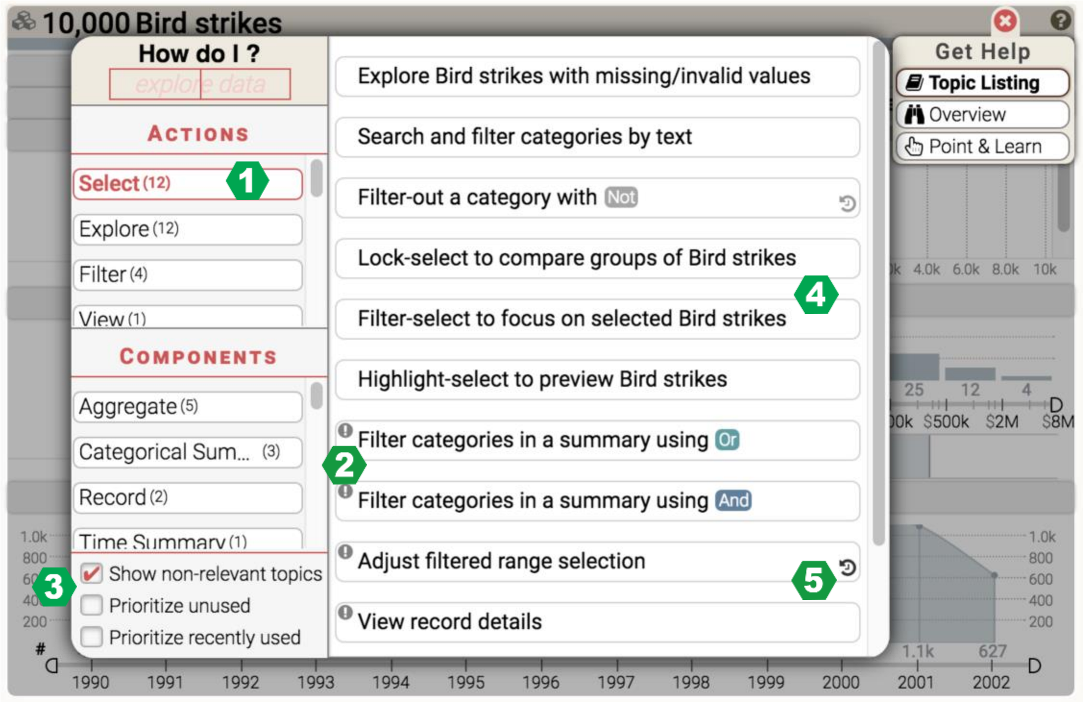
The charts show the Number of Accidents per each category/range.
▢ shows the active (filtered) selection. (Clear)
■ shows Route: State Highway lock-selection. (Unlock)
🟩 shows Route: Local Street – Municipality lock-selection. (Unlock)

Metric is Number of Accidents. (Change) ❷
Visual scale mode is absolute. (Change)
Measurement labels show absolute (#) value. (Change)

Figure 24. The *Overview* mode. ❶ The interface state is briefly described using the active settings and data features. ❷ The user can interact to learn how to change the related states (for example, changing selections or visualization modes).

8.3.2 Topic Listing

The *Topic Listing* mode (Figure 23) lists all help topics, ranked (and filtered) by relevance given the current context. While this mode reflects the traditional *pull* approach with tag-based filtering and text-search to navigate through help topics, our context-aware ranking improves upon the static help listings and navigation of topics. In addition, HelpIn provides contextual options to hide (or show) non-relevant topics, and to prioritize unused (or most recently used) features. Providing paths to topics that may be currently irrelevant can help users learn about extended tool capabilities.


10,000 Bird strikes

How do I ?

explore data

ACTIONS

- Select (12) ❶
- Explore (12)
- Filter (4)
- View (1)

COMPONENTS

- Aggregate (5)
- Categorical Sum... (3)
- Record (2)
- Time Summary (1)

☑ Show non-relevant topics ❸
☐ Prioritize unused
☐ Prioritize recently used

- Explore Bird strikes with missing/invalid values
- Search and filter categories by text
- Filter-out a category with Not ⟲
- Lock-select to compare groups of Bird strikes ❹
- Filter-select to focus on selected Bird strikes
- Highlight-select to preview Bird strikes
- ! Filter categories in a summary using Or ❷
- ! Filter categories in a summary using And
- ! Adjust filtered range selection ❺ ⟲
- ! View record details

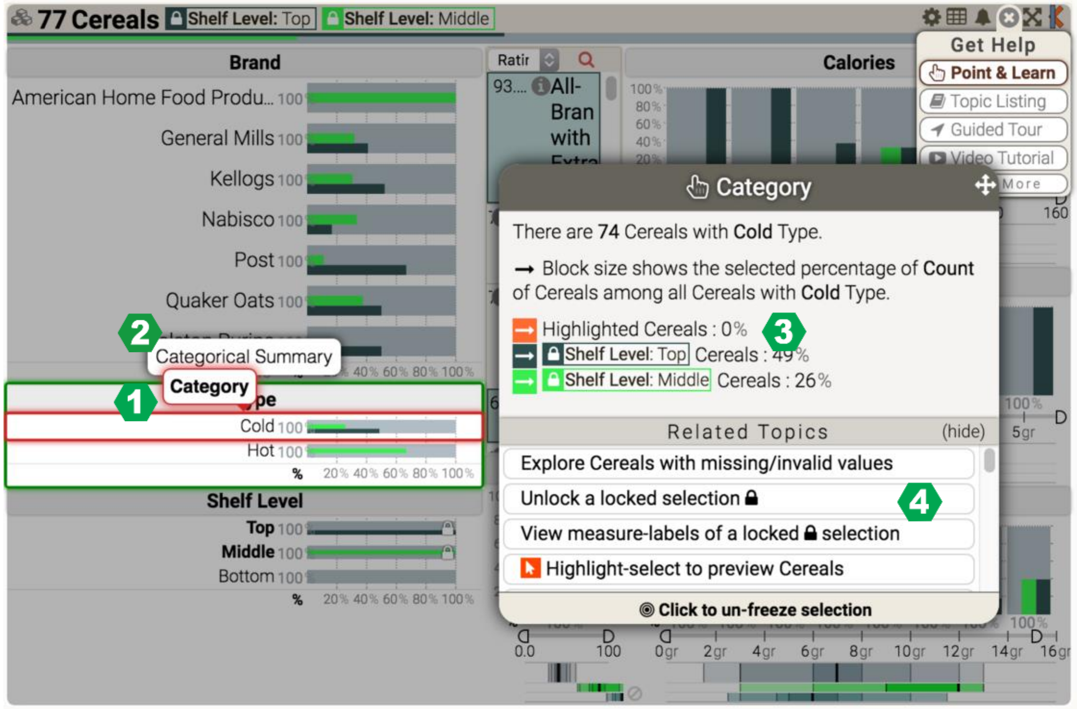
Get Help

- Topic Listing
- Overview
- Point & Learn

Figure 25. The *Topic Listing* mode. ❶ Topics are filtered to those relating to *Select* action, and ranked by contextual relevance. ❷ Non-relevant topics are shown with (!), and are ranked below relevant topics. ❸ Ranking options can be modified. Topic names reflect the dataset ❹ (Records are *Bird Strikes*) and application state (ex: absolute vs. relative visual scales - not visible in this screenshot). ❺ Topics that have been recently used are marked with a ⟲-icon.

8.3.3 Point & Learn

In the *Point & Learn* mode (Figure 24), the user selects an interface or visualization component by hovering their mouse over it. The help panel shows the information relevant to the pointed element, including its name, description (along with visual encodings and settings), and related help topics, while the pointed element is highlighted using a stencil window and tooltip in the semi-transparent overlay. The hover-action provides a fluid and responsive interaction design to quickly learn about multiple components. Clicking freezes the selected element, and enables interaction with the help panel (such as activating a related help topic). The freeze-action can also trigger updating help material, such as showing connected components of the selected item (such as a data record) on the interface. The selection can be unfrozen by clicking outside the help panel.



Category

There are **74** Cereals with **Cold** Type.

→ Block size shows the selected percentage of **Count** of Cereals among all Cereals with **Cold** Type.

Highlighted Cereals : 0%

Shelf Level: Top Cereals : 49%

Shelf Level: Middle Cereals : 26%

Related Topics (hide)

Explore Cereals with missing/invalid values

Unlock a locked selection

View measure-labels of a locked selection

Highlight-select to preview Cereals

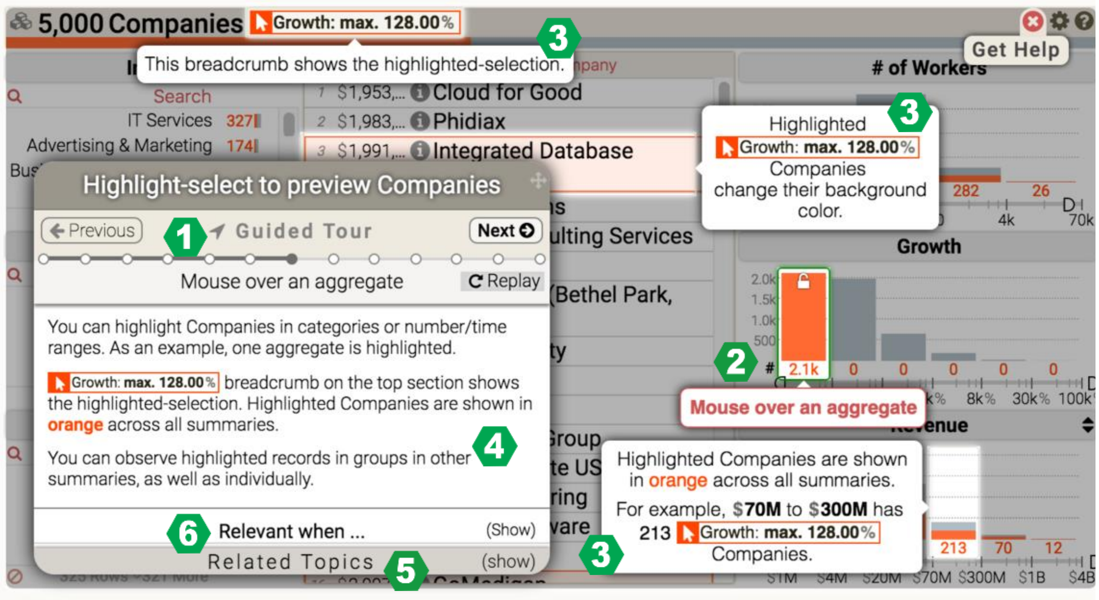
Click to un-freeze selection

Figure 26. The *Point & Learn* mode of HelpIn. ❶ A category is selected by pointing. ❷ Its parent, *categorical summary*, is also highlighted. ❸ Descriptions of the category is responsive to data, visualization and selection states. It includes a basic description, the visual encoding, and for each visual feature, describes the encoded value and how to read the interface. ❹ Related topics include those relating to the category component, as well as the categorical summary component.

HelpIn recognizes the hierarchical composition of UI elements on pointer-based selection. For example, a measure label appears inside a category (glyph), which appears inside categorical summary, which appears inside a panel (of the data browser). While the description follows the most specific element (such as the measure label), the help topics and stencils can reflect multiple layers in hierarchy. We limit the hierarchy to two components (self and parent), so that material is focused, and not overwhelming to the user.

8.3.4 Guided Tour

The *Guided Tour* mode (Figure 25) aims to quickly familiarize the user with the interface using a pre-determined sequence of help material (topics or interface components). The user controls the pace by explicitly stepping through the sequence. Related topics to the active step are available on request. HelpIn displays the progress through a dot-pattern, and clicking on a dot jumps to the tour to a specific step. If the user exits or changes help mode during the guided tour, they can later resume from the last active step.



This breadcrumb shows the highlighted-selection.

Highlighted Growth: max. 128.00% Companies change their background color.

Highlight-select to preview Companies

Previous | Guided Tour | Next

Mouse over an aggregate | Replay

You can highlight Companies in categories or number/time ranges. As an example, one aggregate is highlighted.

Growth: max. 128.00% breadcrumb on the top section shows the highlighted-selection. Highlighted Companies are shown in orange across all summaries.

You can observe highlighted records in groups in other summaries, as well as individually.

Relevant when ... (Show)

Related Topics (show)

Mouse over an aggregate

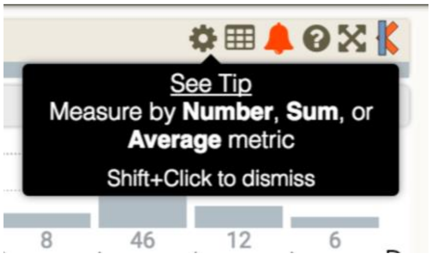
Highlighted Companies are shown in orange across all summaries.

For example, $70M to $300M has 213 Growth: max. 128.00% Companies.

Figure 27. The *Guided Tour* mode. ❶ The tour progress is visible, and user can control it forward, backward, or to a specific step. This step shows answer to a help topic, *Highlight-selection to preview Companies*. ❷The tooltip of the main action, which is a mouse-over on a visual glyph, is highlighted by **color**. ❸ Additional tooltips describe the effect of this action on other interface components. ❹ Detailed description of the topic presents an easy-to-read summary of tooltips and additional information. ❺ Related topics, and ❻ the context under which this topic applies can be viewed on demand as well.

8.3.5 Notifications

The *Notifications* mode suggests relevant help topics on the fly as an explicit push-model. To not disrupt to the user, we followed a subtle design that uses 🔔 on the corner to present incoming notifications. On mouse-over, the icon reveals the related task name, and allows for



See Tip

Measure by **Number**, **Sum**, or **Average** metric

Shift+Click to dismiss

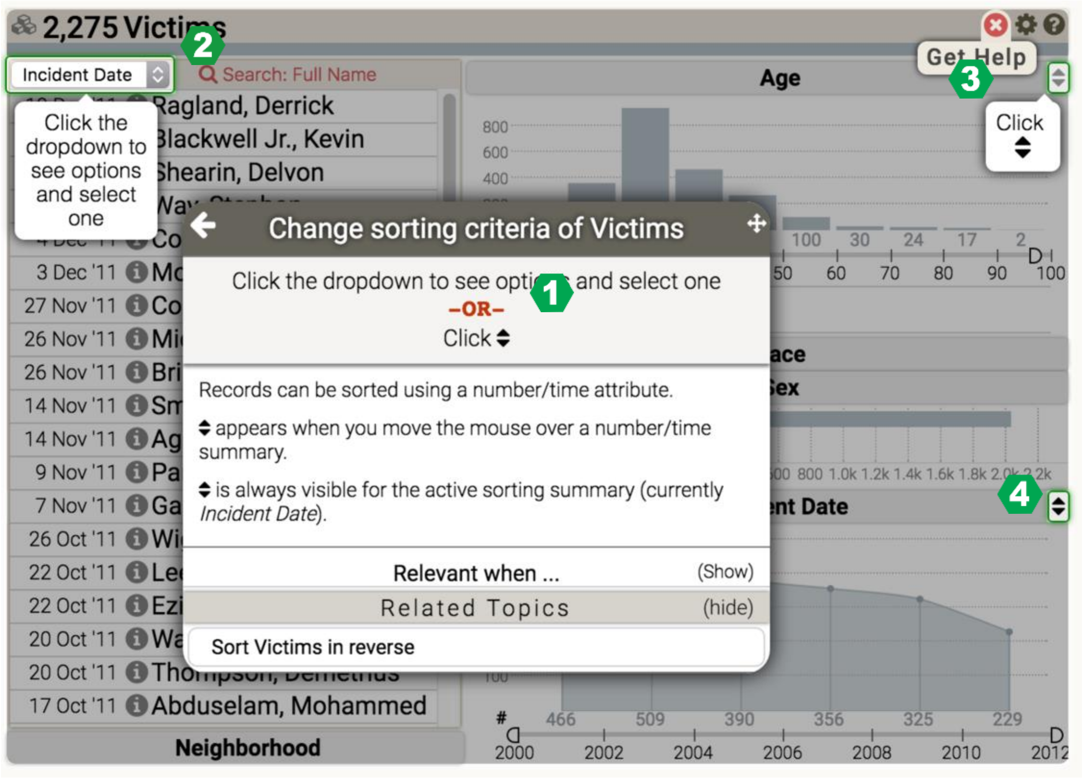
dismissing the notification. In our current prototype, we enable notifications on a per-topic basis over an extended period of time if the user has not used the relevant feature yet. The notifications can also be used as a tip-of-the-day feature to suggest new topics for re-visiting the interface.

Generating relevant notifications require detecting user behavior by tracing their actions, in addition to taking data and application context into account. While earlier *intelligent* help systems such as Microsoft's Clippy has not proven to be effective [100], finding the right content and presentation design for notifications can enable opportunistic learning. In other words, more semantics and less intrusiveness is desired [57]. To achieve unobtrusiveness and usefulness, the notifications should not be frequent (avoid false positives), and help the user when appropriate (avoid false negatives). We present Notifications as a design prototype that covers the explicit push model for help, and we claim no contributions on identifying when to raise notifications.

8.3.6 Topic Answers

A contextual topic answer aims to ease help comprehension (rather than help seeking), and can be reached through the topic listing or relevant topics of a pointed component. The topic answer is presented directly on the interface by highlighting all the UI components that can achieve or affect the task using a stencil window and tooltips (Figure 26). Help descriptions include not only how to perform the task, but also how it affects the rest of the interface, such as in coordinated-views design pattern for selection tasks [118]. When a dynamic demonstration of an answer is appropriate, HelpIn can present an animated sequence of steps, highlighting information relating to each step directly on the interface. The user can replay these animation sequences to better attend to interaction details and sequences. Clicking on a highlighted UI component passes the mouse-click through overlay and executes the action. The help overlay closes if there are no other actions to execute for the task, or shows the next action step if other steps remain. When multiple components can achieve the same task, a single tooltip is shown for each component group (Figure 26).

The contextual features of a help topic are shown under "*Relevant when…*" part of the help panel. The selected topic can be non-relevant contextually if one or more context features are not met in live interface (for example, when input data does not include the relevant data type, or when data is not filtered for a topic that is to modify an existing filtering selection) (Figure 27). The help descriptions are also available within help panel, providing training for application features and future use-cases. In our instructional design, we enriched answers and descriptions using screen captures and short animated sequences (implemented as animated GIFs) on-repeat to help users learn the tasks using visual media when live integration is not possible. This is similar to traditional approaches using static material to describe answers when contextual relevance fails.



Change sorting criteria of Victims

Click the dropdown to see options and select one
–OR–
Click ⬍

Records can be sorted using a number/time attribute.

⬍ appears when you move the mouse over a number/time summary.

⬍ is always visible for the active sorting summary (currently *Incident Date*).

Relevant when ... (Show)

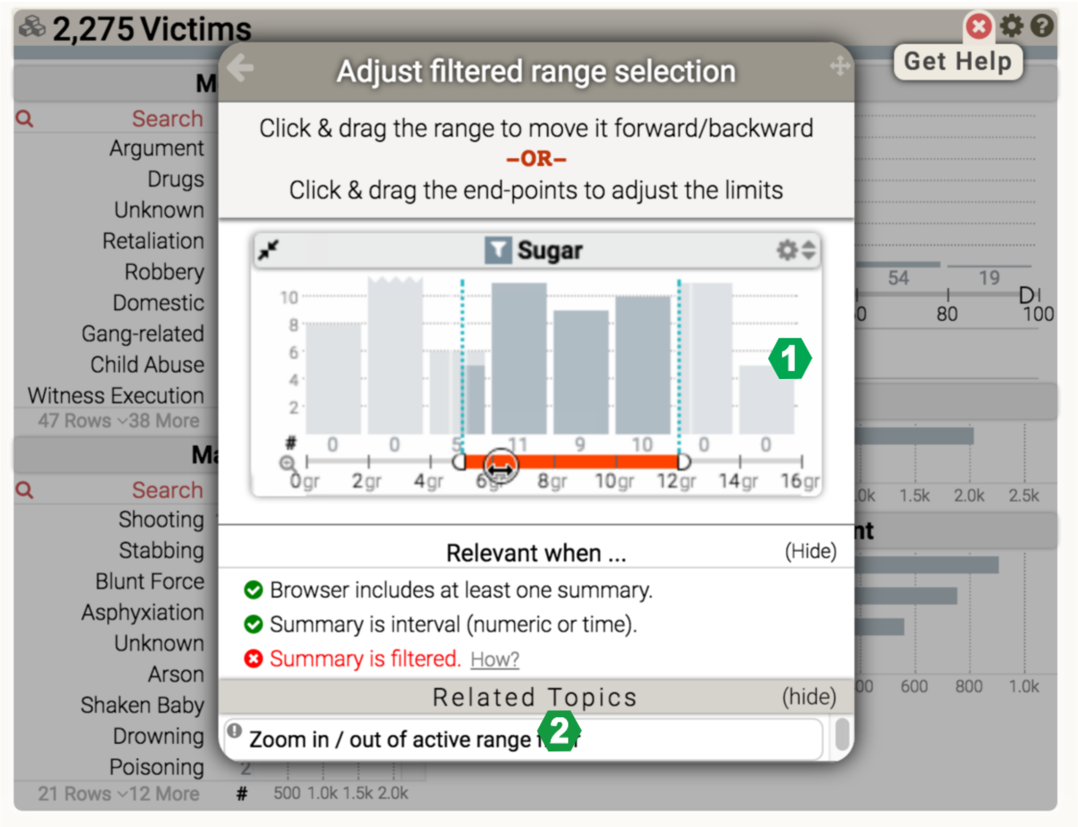
Related Topics (hide)

Sort Victims in reverse

Click the dropdown to see options and select one

Click ⬍

Figure 28. The *Topic Answer* mode ("*Change sorting criteria*" topic). ❶ Two distinct actions can satisfy this task, either by ❷ using the dropbox, or ❸ clicking an icon in a numeric summary. ❹ **Notice that** all relevant icons are highlighted, yet "*Click*" action tooltip is shown only for one. Clicking on any of these stencil boxes pass the mouse event to underlying interface element, action is executed and HelpIn closes.



Adjust filtered range selection

Click & drag the range to move it forward/backward
–OR–
Click & drag the end-points to adjust the limits

Relevant when ... (Hide)

- ✓ Browser includes at least one summary.
- ✓ Summary is interval (numeric or time).
- ✗ Summary is filtered. How?

Related Topics (hide)

Zoom in / out of active range f

Figure 29. A non-integrated (static) topic answer, including a screenshot or animated gif 1 using a fixed dataset. The reason why this topic is not relevant is highlighted 2 under *Relevant when…*, and a link to satisfy this criteria is available.

8.4 Implementation

We implemented HelpIn for web applications based on modern web standards. The program logic of HelpIn is implemented in JavaScript, and help material is also described as JavaScript objects. The material includes lists of contextual features, help topics, UI components (for Point&Learn mode), and guided tour steps. Our implementation, including help material for Keshif, is available on (GitHub).

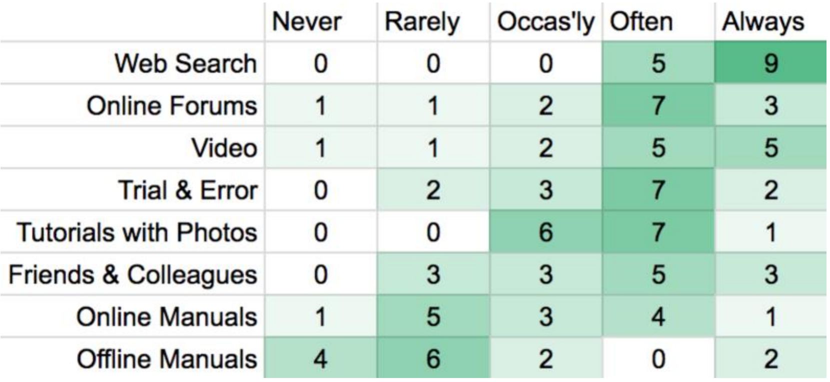
To evaluate the context, HelpIn accesses the DOM of the webpage and/or accesses the underlying application state and dataset in JavaScript directly. It can also modify the application state through this direct code access. The stencil areas used for answers and Point&Click components are expressed as DOM class names, which also enable detecting help topics for tracking historical context of use (for example, HelpIn can track a click on .summaryCollapse button to “Collapse summary” topic).

8.5 Evaluation

To understand how HelpIn influences the help-seeking and learning performance, behavior, and experience of first-time users for data analysis tasks, we conducted a laboratory experiment. For comparison to the contextual in-situ help system (*HelpIn*), we used non-contextual help topics with non-integrated topic answers (*Baseline*). We present a quantitative analysis on performance, and the interactive help system use to answer tasks. We also present subjective feedback of our participants regarding the observed usability and efficiency of the help system and help materials.

8.5.1 Participants

We recruited 14 participants (7 male, 7 female) using university public mail-lists and message-boards. They were university students in various departments (6 undergraduate and 8 graduate, departments including English, business, math, agriculture, computer science, information management, system engineering, and computer engineering). All participants were first-time users of the data analytics tool. Two participants had experience creating visual dashboards for other data analysts (using SAP or d3). Other participants did not have visual data analytics training beyond basis statistics courses, and most previous experience was related to coursework. They all had some experience with drawing charts using Excel. Thus, the majority were novices in visual data analytics, as well as in Keshif. We also asked their existing help seeking behavior. The feedback demonstrates a variety in personal preferences, including use of videos, online forums, tutorials, and trial and error (Figure 28).



	Never	Rarely	Occas'ly	Often	Always
Web Search	0	0	0	5	9
Online Forums	1	1	2	7	3
Video	1	1	2	5	5
Trial & Error	0	2	3	7	2
Tutorials with Photos	0	0	6	7	1
Friends & Colleagues	0	3	3	5	3
Online Manuals	1	5	3	4	1
Offline Manuals	4	6	2	0	2

Figure 30. Existing help seeking behavior of participants.

8.5.2 Study Design

We used a within participant design with the help system (*Baseline* vs. *HelpIn*) as the independent variable. The ordering of help systems shown to the participants was counterbalanced, i.e. 7 participants completed tasks with *Baseline* first, and the other 7 started with *HelpIn*. The system conditions were as the following:

- The *Baseline* condition included a traditional (non-contextual) topic listing with alphabetical sorting, and did not integrate answers into the interface, i.e. did not include stencil highlights or tooltips. The answers included static media (images, animated gifs) using samples from other datasets as traditional, non-contextual material.
- The *HelpIn* condition used contextual and integrated help with *Topic Listing*, *Point & Learn*, and *Overview* modes.

Baseline was created using a stripped-down version of *HelpIn* to eliminate other differences across the systems. The help material used across the system conditions were the same except the help modes, the use of context, and integration of answers. The material, with 32 topics and 50 components, focused on the exploratory use of Keshif (i.e. did not include authoring data visualizations). We disabled *Notification*s as its efficiency depends on inferring user behavior with minimal false positives/negatives, which is not among our contributions. We used the *Guided Tour* mode for training only for both conditions.

8.5.3 Collected Data and Metrics

We quantitatively measured the interactive use of help system (such as time spend on task and help, and the number of times the help modes are used), progress on task, and response time. One of the authors of the paper coded features of interaction with the help system and the task performance of the participants based on grading rubrics, using screen captures and verbal feedback on video captures of the study sessions. Our data collection also includes survey results and semi-structured interviews as well.

8.5.4 Training

In the beginning of the study, all participants received an ~8 minute training and introduction on data analysis with Keshif and the help system. First, participants completed a self-paced 12-step Guided Tour for Keshif. Then, the facilitator gave ~1-minute demonstration on how the help system can be opened/closed, and the three help modes: Topic Listing, Overview, and Point&Learn. If a participant had completed guided tour early, we allowed them to use the tool and help-system as they wished in remaining 8-minute training time. A separate training dataset (homicides in Washington D.C.) was used for training purposes. We did not provide any external help to complete the tasks during the experiments. In other words, the participants could get training or help on the underlying data interface (Keshif) using the help system *only*, i.e. not by asking the experiment facilitator.

8.5.5 Tasks

The participants were given 12 tasks across three task types and four datasets. The three task types (Explain, Retarget, Analyze) cover both understanding the data interface, and executing actions to achieve desired outcomes. Specifically:

T1-Explain: We asked the participant to "*Focus on the (specific) summary and explain the chart, including the meaning of each color, numbers, and trends you identify.*" This task is aimed to assess visual data comprehension. The charts included different data selections, visual modes, and measured different characteristics, across the datasets. If participants used overview or Point&Learn modes to give an answer, we asked the participants to paraphrase the descriptions after closing the help system.

T2-Retarget: We provided a current configuration of the data interface, and a targeted configuration as a screenshot. We asked the participant to "*Modify the page on the computer to exactly match the one shown in the screenshot.*" The target included 2-4 reconfigured settings and adjustments, different for each dataset. This task required understanding multiple differences across two configurations, finding relevant help topics to learn how to make necessary changes (if needed), and executing correct actions.

T3-Analyze: We asked the participant to answer a specific analysis question, such as finding the company with a minimum number of workers, or total number of illnesses in outbreaks within two states. The questions required interacting with the interface and changing multiple settings, which were different for each of the four datasets.

We used four datasets (companies, bird strikes on airplanes, foodborne outbreaks, and traffic accidents; all tabular datasets of comparable sizes and features) to limit the effects of learning the features of the underlying data. For each dataset, participants answered all three task types in the order noted above. Dataset were also presented in the order noted above. Participants had 2.5 minutes to complete each task. The interfaces included a timer to inform the participants. The tasks across different datasets used targeted features of the underlying tool (Keshif), and were of comparable difficulty based on our pilot studies and earlier experience evaluating Keshif. Specifically, the features that were seen as challenging and which would benefit from the use of help system included: linked selections, measure metric (count/ sum/ average), visual scale mode (absolute/ relative), label mode (absolute/ percentage), changing histogram axis scale (linear/log), and the use of percentile charts.

We created a grading rubric on a [0,5] scale for each task. Zero noted no progress, and five noted a correct answer with all expected outcomes. For example, a retargeting task required clearing all filters (2 pts), filtering on a selection (1pts) and highlighting another selection (2pts, 1pts –partial- for not including it for final response). In data analysis task, filtering categories with OR (2pts), changing the aggregate metric (2pts) and finding the right number (1pt) was required.

Since each task involved changing or describing multiple features of the interface, the rubric allowed us to focus on task progress and differences in performance with more granularity. 2.5 minute limit and task complexities created a challenging, yet inviting, setting where participants had to strategize on how to use help, and how to best answer tasks in short time.

8.5.6 Procedure

In the beginning of the study, each participant completed a background survey. All participants received an introduction to the analytics tool (Keshif) and the help system (*HelpIn*) in 8 minutes. Then, each participant received a sequence of 12 tasks, and the help system was changed half-way. We introduced each dataset briefly before tasks on a specific dataset. We did not enforce a specific use of help system, i.e. the participants were free to choose *when* and *how* to seek help and to interpret the material. However, we encouraged participants to use the help system for each task. If they have not used help system before an answer, we asked “if (they) would like to use help before finalizing (their) answer”. When participants felt stuck, we also encouraged them to use the help system. Beyond this, we did not aid participants in solving tasks. Each task was followed by a task survey on subjective task performance and usefulness of the help content and system features. After finishing all tasks, the participants completed an overview survey and a short interview on effectiveness of various help techniques and materials.

All sessions were held in a university lab using Google Chrome on a Macbook Pro with a 15-inch retina display, and a mouse for interaction. We recorded the screen and audio during the sessions for future analysis of the data. We compensated participants with $10 cash. Each session was completed in about 1 hour.

8.5.7 Pilot Studies

We ran pilot studies with 4 participants to develop the study protocol. We observed significant variations for help use and analytical reasoning between participants, which limited effectiveness of the between participant design protocol across help systems. Within participant design also allowed us to collect feedback on subjective preferences. We also noted that without a brief introduction to help system, the participants could not make informed decisions during the study since they were not aware of help system features.

8.6 Results

8.6.1 Task Progress Performance

We observed no performance differences, measured by task progress, across the attempts with *HelpIn* vs. *Baseline* conditions (total progress scores 252 vs. 248, given 84 attempts each). Likewise, we found no major performance difference across those who used *HelpIn* first, or *Baseline* first (i.e. order effect) (total progress scores 256 vs. 244, given 84 attempts each). However, we found that participants performed significantly better in attempts where they used Point&Learn, compared to those where they used Topic Listing (with average progress 3.15 vs. 2.51, sample sizes of 82 and 53 attempts). Note that Point&Learn was only available in *HelpIn* condition, and Topic Listing was available in both conditions. We base this analysis on the modes used to answer a task, and reported sample sizes above.

The total progress per participant ranged between 18 to 45 (of 60 points total), showing significant individual variations in how participants performed. The total score per task were distributed mostly in [43-57] range (for 9 tasks), while three outlier tasks had total scores 14, 18 and 34, showing that tasks were mostly of comparable difficulty.

8.6.2 Time on Help

Of 168 task attempts (12 tasks by 14 participants), only 30 (18%) were finished before time-out. In other words, participants used all allocated time in 82% of their attempts. Thus, we focus our time-analysis on the use of help system.

Our participants spent significantly more time with *HelpIn* than with *Baseline* (52 vs. 30 second average, sample size: 84 attempts each). This was mainly contributed by Point&Learn (54sec average, based on 53 attempts where this mode is used), compared to Topic Listing mode (35sec average, based on 82 attempts with this mode). In addition, participants who used *Baseline* first spent significantly more time on help system compared to those who used *HelpIn* first (34 vs. 50 second in average, sample size 84). Therefore, using *HelpIn* first reduced total time spent on help, without major differences in task performance.

8.6.3 When Help Is Not Needed

Among 168 total attempts, 42 (25%) did not use help system, which also lead to higher average performance (3.57), compared to attempts with help use (2.77). This suggests that when participants felt confident in taking on the tasks, they did not seek help, and performed objectively better overall. In regards to not seeking help, a participant noted "*If it is slightly familiar system, and I feel I can get about exploring things on my own, I prefer that than the help.*" In other reasons, one noted, "*I wasn't sure it could really pin-point what I wanted.*" and another said, "*Because my time is so limited.*" Of the 30 attempts that were finished before timeout, 15 (50%) did not involve any use of the help system. 6 of the remaining cases of help use (40%) were to confirm an answer, rather than to search for answer.

8.6.4 The Characteristics of Help System Use

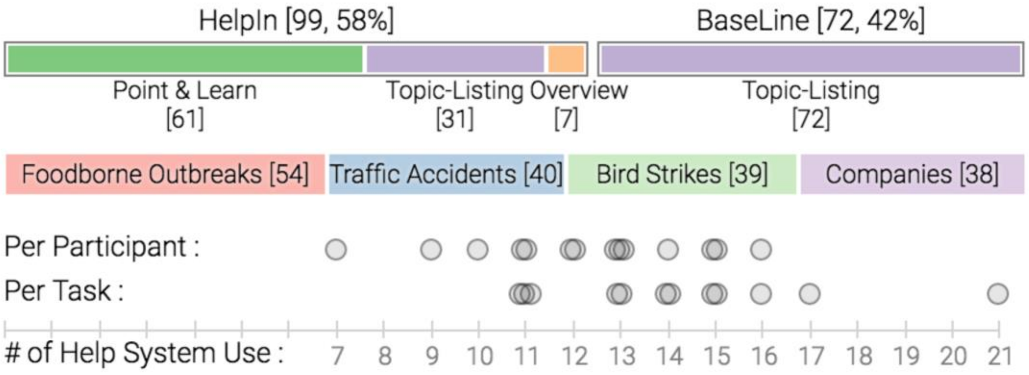


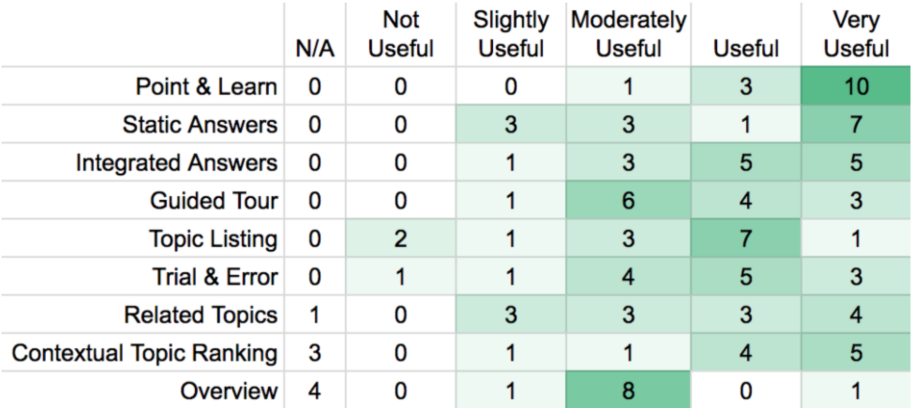
Figure 31. The distribution of the number of times the help system was used (of 171 total). *Top)* The distribution across systems (HelpIn vs. Baseline), and help modes. *Middle)* The distribution across datasets. *Bottom)* The distributions per 14 participants, and 12 tasks, shown with jitter on overlaps.

Figure 29 shows the distributions of the number of times the help system was used. Help was sought in *HelpIn* more than *Baseline* (58% vs. 42%). When all modes were available, Point&Learn was used significantly more than Topic Listing, and Overview was only used few times. Distribution of help use across different datasets shows that tasks on different datasets were of comparable challenge. Participants used the help system 7-16 times in total through the study. Help seeking per task is also distributed between 11 to 21 uses. The outlier task is where the participants could not find answers to necessary steps with ease.

We observed that help system was opened 20 times to confirm the answer or observation. 18 (90%) of these cases were with Point&Learn, while 2 were with Topic Listing. This demonstrates Point&Learn can also support the user to confirm or clarify the meaning of data visualizations.

8.6.5 Help Topic Listing Search Behavior

Of the 82 attempts that used Topic Listing, topics were searched by *tags* 41 times (50%) and by text 27 times (33%). Of all the tags selected (63), the majority (55) were action-tags (verbs), instead of component-tags (nouns). Our prototype used strict text matching with topic names, which frequently (22/35) did not return the relevant topics. The failed queries included names of the data attributes (9 cases, such as querying “workers” to find a topic that applies to “number of workers” attribute), as well as synonyms (10 cases, such as querying “combine” to add multiple filters, or “reorder” to sort). These interactions show that our participants preferred to search by action rather than component (potentially since component names may be unfamiliar), and that text query search needs to be flexible with synonyms, and match attribute names with components.



	N/A	Not Useful	Slightly Useful	Moderately Useful	Useful	Very Useful
Point & Learn	0	0	0	1	3	10
Static Answers	0	0	3	3	1	7
Integrated Answers	0	0	1	3	5	5
Guided Tour	0	0	1	6	4	3
Topic Listing	0	2	1	3	7	1
Trial & Error	0	1	1	4	5	3
Related Topics	1	0	3	3	3	4
Contextual Topic Ranking	3	0	1	1	4	5
Overview	4	0	1	8	0	1

Figure 32. Feedback on feature usefulness by the participants.

8.6.6 Subjective Preferences

Participants rated the help system features at the end of the study (Figure 30). *Point&Learn* was the feature found the most useful. One participant stated "(it was) my favorite part of the tool". Other participants shared similar feedback. When asked about preferences in static or integrated answer presentation, 8 preferred integrated, and 6 preferred static. A participant noted, "Integrated answer is definitely tremendously more useful as it showed you on the page itself where to be looking for (…) It was able to point you in the right direction". In favor of static answers, another noted, "*My attention is so concentrated over (main help box) that I just might miss out on (tooltips) (...) (On integrated answers) I don't know what the expected outcome would be (...) I really don't know if I did something right, or if that I am in a wrong state and I have to do something more*." Therefore, neither approach surpassed the other in our prototype. Preferences are also likely to be shaped by quality and content of help material and personal preferences. We noticed animated gifs within the help panel to be good demonstrations for most cases, and we noticed that some of Keshif's integrated answers could provide more animations on how the change would affect the interface.

In addition, we noticed one of the challenges of our participants was translating questions into relevant topics, i.e. what to get help about. About the language of the help material, a non-native speaking participant noted, "(English is) not a native language for me, so it's just a little bit too long, so it's just slightly helpful", while another one contrasted, “I don't know if it is possible for the text to be more succinct.”

8.7 Discussion

8.7.1 Experiment Results

The *Baseline* condition of our experiments was a non-contextual version of *HelpIn* with non-integrated topic answers. In order to create a shared basis of instructional material, we avoided using fully separated help material or videos which otherwise may lead to differences beyond help system design. Future studies may target evaluations across media types and designs. Our experiment also did not aim to measure long-term retention, or open-ended use.

The similar task performance across *Baseline* and *HelpIn* might be contributed by the shared instructional basis. In *Baseline*, our participants most strongly noted the absence of Point&Learn mode, and were less expressive on differences in the presentation of help topic answers and contextual topic ranking, although their final feedback was mostly positive. In addition, our participants showed more progress in tasks in which they used Point&Learn, compared to the tasks where they used Topic Listing. The highest performance was when they didn’t use help, where they mostly showed progress through learning and trial and error.

The experiment also helped us identify opportunities to improve help, as well as usability and help instructions for specific materials. Topic answers can include an option to present non-integrated, simple screen captures or animated gifs for cases where integrated answer would also apply, as the preference across the two appear to be personal. Based on participant feedback and system use, text query search can be improved to find more relevant topics by considering attribute names and synonyms, and Point&Click components can be narrowed down to individual glyphs (such as lines in line charts) across all visualizations.

8.7.2 Generalizing HelpIn

Our implementation presents as a proof of concept, and currently does not support targeting new interfaces (applications) easily. We believe that the implementation of HelpIn could be modularized so that it could be retargeted to other applications more easily. While we present strategies for the use of context and integration, the design of help content depends on the application. High-quality material requires careful design and iterative improvements on content and its integration, beyond what a modular implementation may provide out-of-the-box. Our design space and implementation provides a structured basis to undertake similar task for other visual data applications.

8.7.3 Help Material and Instructional Design

While we aimed to achieve a simple language consistent with underlying application for our study, we have not specified a target education level, and evaluated the terminology with a wide range of users. Our search and tagging system did not consider synonyms or alternative definitions. While HelpIn tries to facilitate learning the interface concepts in a rapid fashion, it does not fully facilitate translation of user goals to interface goals. Our topic model can be extended to define hierarchies for more complex applications. In addition, the help material for our prototype on Keshif does not comprehensively cover all features of the tool. While HelpIn offers a structure to express and present help material, it does not guarantee full coverage.

8.7.4 The Synergy Between Interface Design and Help Design

From the perspective of interface designers and developers, our integrated approach enables preparing and maintaining the training material along with the design and implementation of the interface. This can reduce time-consuming updates to existing material after interface changes, and can shift the preparation of the help material from post-implementation (waterfall model) to the course of interface development. In addition, the design of help material should build upon the design of the interface. While providing help and documentation is necessary to support the wide range of tasks or learning requirements, improving design of the underlying interface should be prioritized to *minimize* the need for help, and to push towards self-explanatory interfaces. In other words, the help material should not be the primary resource to enable the usability of an interface.

8.7.5 Limitation of Contextual In-Situ Help

Separated help materials, such as videos and manuals, can be the only viable option when the interface (application) is not immediately available to the user, for example when application requires purchasing or installation. Videos can also provide additional benefits in explaining interfaces by using spoken narratives, i.e. auditory channel, which may compliment visual channels. Future work in integrated help can integrate audio into explaining the live interface.

Chapter 9. Dense Visualization Design for Numeric Data

"This sounds bizarre, but I find it all too frequently – a complete bastardization of tools that were never meant (or validated) for the applications for which they are being used. You can't make this stuff up."

Alan Weiss

in Million Dollar Consulting [147]

Graphical perception is one of the key cognitive components to rapid, effective, and accurate visual data exploration (See Chapter 3). Evaluating graphical perception under various tasks and designing new visualization techniques is fundamental to data visualization research. Achieving dense data visualizations in limited screen space remains a challenge for many data types, including simple numeric data.

This chapter is motivated by our work in applying Keshif in multiple settings and observing existing visualization practices. In developing and evaluating Keshif, we noticed that categorical summaries with many categories result in long lists that need to be scrolled to get an overview, putting an interactive barrier. We also noticed a common practice of treemap use for non-hierarchical data. Since the technique was originally developed for hierarchical data, and studies were only focused in this case, we aimed to assess effectiveness of treemaps in non-hierarchical settings, and compare it with bar charts with multiple columns. In this chapter, we present the design objectives in this domain, alternative visualization techniques, including the novel piled-bars design, and a detailed evaluation in crowdsourced graphical perception and design characteristics across alternative visualizations of same data.

9.1 Design Objectives

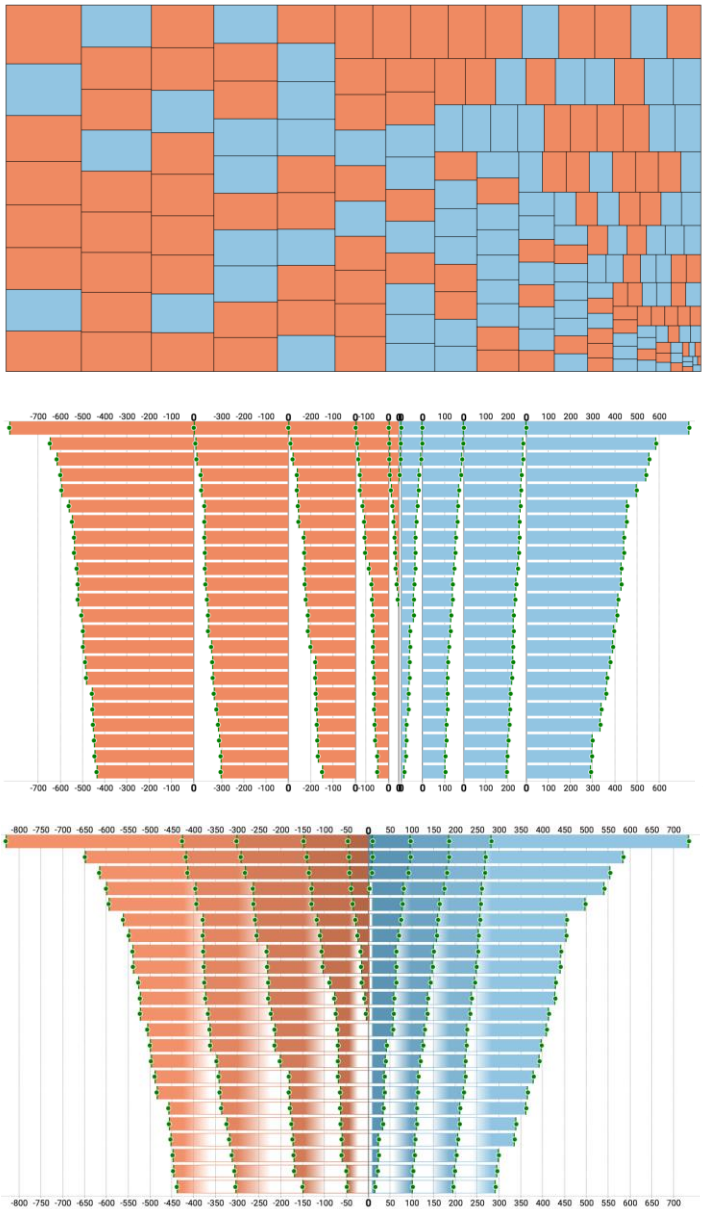
The visualization design space for sorted numeric data has the following objectives in this thesis:

(O1) Each record is *perceptually distinguishable*. All records must fit within the chart, and must be presented with their own visual glyph. This makes sure that all records can be observed, and compared, visually when needed.

(O2) An *overview* of all records is visible without interaction. This objective fits the use of visualization in static medium, such as in print and in social media image previews. While interaction can be used to reveal multiple perspectives and views over time, it is beyond the focus of graphical perception studies. In addition, a perceptual response to a visual data representation is more rapid and immediate compared to observation through interaction.

(O3) The records are visually *sorted* by value. This improves the visual structure, and simplifies assessing min/max, variance, and rankings. Without such order, the visual representation of data would be weaker in revealing data distribution characteristics.

The summary of three visualization techniques that meet these design objectives is presented in Table 6. *Treemaps* are a commonly used chart type to show many records that otherwise would not fit in a single-column bar chart, making use of all of chart pixels to encode the data by area. *Wrapped bars* and *piled bars* increase the number of visible items by utilizing a wide-aspect chart using multiple columns. The three chart designs use similar chart size and aspect ratio, and thus are directly comparable per our objectives. The number of records that the proposed alternative techniques can handle is larger than what a single-column bar chart can show; yet the chart area bounds the number of perceptually distinguishable records. The record count that can be effectively represented also depends on the chart size and the distance of the viewer to the display.



(a) Treemap

(b) Wrapped Bars

(c) Piled Bars

Figure 33- Three dense visualization techniques show 200 (+/-) numbers. (Left) Treemap, a space-filling design, shows the magnitude by the block size, and the sign by block color. (Middle) Wrapped bars are multi-column bars, and can organize +/- numbers across two sides. (Right) Piled bars use a shared basis across all columns. What are the design implications across the three chart designs? Which chart design can improve perception for comparison, ranking, and overview under varying data conditions?

Visualization	Treemaps (TM)	Wrapped Bars (WB)	Piled Bars (PB)
Visual Encoding	Space-filling rectangular area	Length by absolute value	
Gridlines	Not Available	Supports gridlines	
Baseline		One baseline per column	One baseline per chart
Horizontal (data) axis		Lower ↔ resolution	Higher ↔ resolution
Block order by value	↓ & → (Not guaranteed)	Columns first ↓ , then Rows →	
Overlaps	None	None	Along bars on a row
Filled pixels	All	Partial - Depends on distribution and variety of data	
More columns	N/A	Shrinking bar width ↔	Increasing overlaps ⧉
(-) Negative Values	Another visual variable (color) is required.	Bi-directional length encoding from the baseline can be used to separate negative values.	
Grouping Records	Color-coding blocks per group.	Color-coding bars per group.	Bi-directional axis only.
Label Display	Within blocks	Within or next to (more ↔ space) blocks	Within blocks
Other properties	♦ Emphasizes part-of relations. ♦ Area cannot visualize negative values.	♦ Supports visual gap (separation) between columns. ♦ Flexible display for labels and values.	♦ Gradient rendering is used to counter overlaps. ♦ Overlaps limit the use of color.

Table 6- Summary of three visualization techniques that satisfy the three design objectives in this paper.

9.2 Design Alternatives

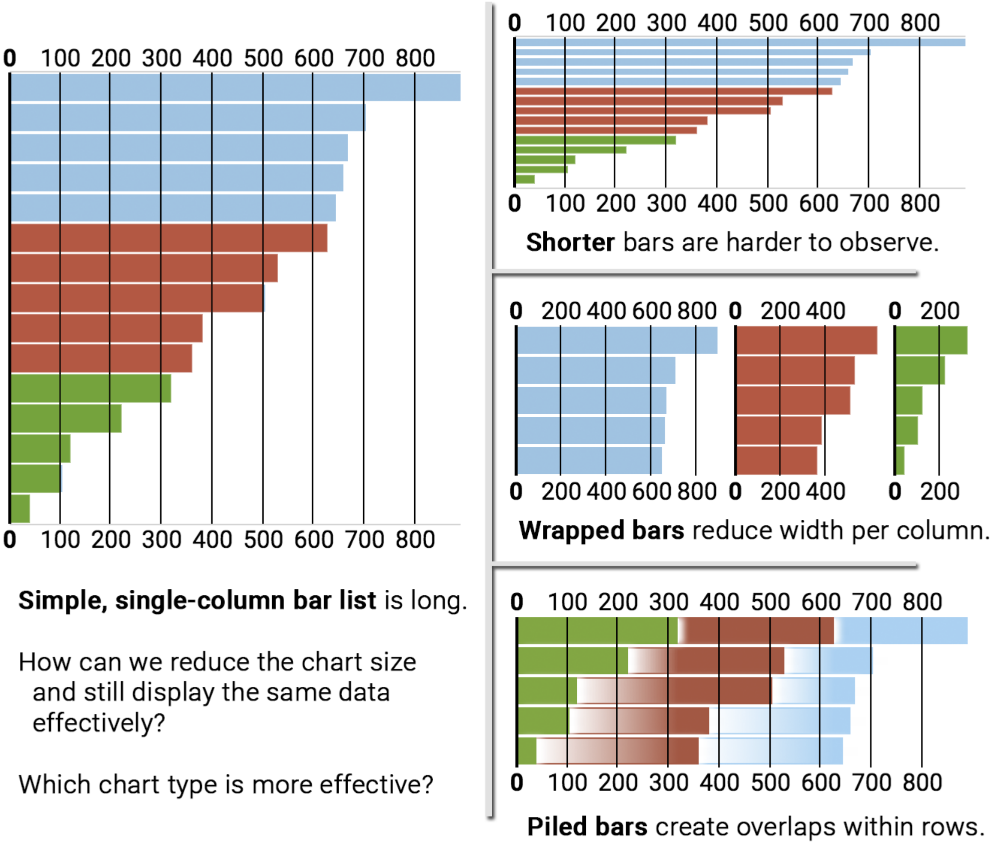
To motivate the objectives and their implications, let us also consider alternative techniques that do not meet the objectives. **(i)** Aggregated visualizations [42], such as histograms, violate O1 as they do not show each record individually. **(ii)** Single-column bar charts can be extended beyond the visible area with scrolling. This fails to show a complete overview (O2), and requires interaction to observe different sections of data. **(iii)** Single-column bar charts can show more data using shorter bars (Figure 32), however making individual records harder to observe (O1). **(iv)** A space-filling design could encode numeric data by color on fixed block size, instead of by area. However, the number of colors that can be effectively compared is fairly limited [109]. **(v)** Circular encodings, such as packed bubble charts, are weaker for perceptual comparison and use screen space less effectively.

Alternative contexts, such as visual analytics systems and interactive data reporting, may have different objectives that would benefit from the use of interaction, such as scrolling or more advanced focus+context views. In such cases, visualization designs that do not fit within the graphical-perception basis of this paper may be effective and preferable.

9.2.1 Treemap Technique

Treemaps are a space-filling visualization design where each data record is visualized using a rectangular block, and the rectangular area encodes the data value. Treemaps were originally designed to visualize hierarchical data groupings [71] using a nested block layout. Treemaps are also commonly used in practice to display *non-hierarchical* data scaling to more records on than possible with a single column bar chart.

The advantage of the space-filling design of treemap is that all pixels are used to visualize data. Treemap algorithms commonly aim to generate a layout with the largest block on the top-left corner, the smallest on the bottom-right corner, and blocks ordered along one direction (↓ or →) in decreasing size first. Yet, the optimized layout does not guarantee such order, thus relaxing the objective O3. The area encoding used by treemap has been shown to be perceptually less effective for comparison task compared to linear encodings of length and position on a shared baseline [32], [61]. Studies on the perceptual influence of rectangle aspect ratios report that rectangles with lower aspect ratios improve perceptual accuracy and extreme aspect ratios should be avoided [61], [81]. Squarified treemap layouts, which aim to avoid elongated rectangles, [23] is commonly preferred, and is the layout used in this study.



Simple, single-column bar list is long.

How can we reduce the chart size and still display the same data effectively?

Which chart type is more effective?

Shorter bars are harder to observe.

Wrapped bars reduce width per column.

Piled bars create overlaps within rows.

Figure 34- Transformations from a long single-column bar chart to dense bar charts. Coloring and gridline overlays are for demonstration.

9.2.2 Wrapped Bars

Wrapped bars [47] use multiple columns of aligned bars, which can effectively show more records than a single column bar chart. Where new bars would extend vertically beyond the chart area, they are wrapped to start a new column, similar to the two-column text layout of this paper. The bars are comparable across the columns since the length encoding has the same unit scale in all columns. The column width decreases as the column shows a lower end of the sorted data. The columns may be separated with vertical ↔ gap to emphasize separation, thus improving readability.

Given a fixed chart area and bar height, adding more records may insert new columns. To make space for new columns, existing bars must shrink vertically ⇆, in turn decreasing data resolution and perceptual accuracy. Increasing bar height ↕ under fixed record count may also have the same effects, i.e. as bars get taller, they get narrower (Figure 37). Thus, the column layout influences the aspect ratio of bars, and potentially creates a tradeoff in readability. We studied the effect of column layout in the graphical perception experiments.

9.2.3 Piled Bars

Piled bars are a multi-column bar chart on a single, shared baseline. This contrasts with multiple baselines, one per column, of the wrapped bars technique. Comparison across bars on different columns is expected to be more accurate since all bars are aligned by sharing the same 0-baseline. Furthermore, the chart width is fully utilized to scale the bars, i.e. the vertical data axis ↔ has higher resolution (Figure 32). Because of the shared scale, columns cannot be separated by vertical gap ↔ to improve readability, unlike wrapped bars.

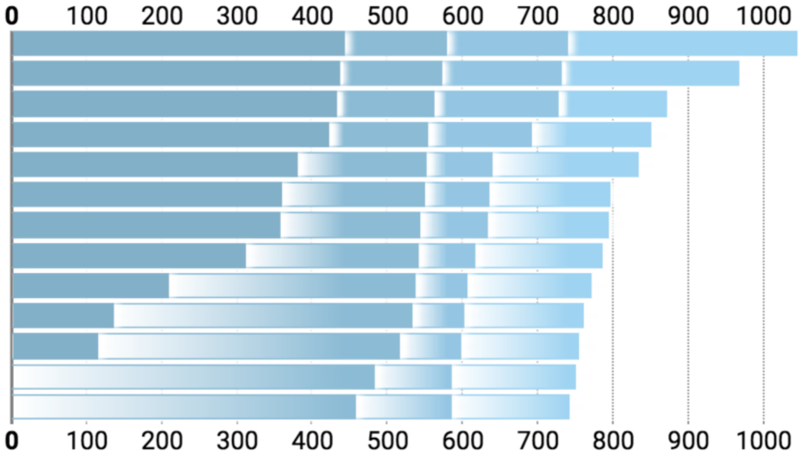


Figure 35- Piled bars rendering approach. Shorter columns (left) are darker than longer columns (right). The bar gradient starts from the smallest extent of the bar's column, and ends at the tip of the shorter bar on the same row. Each bar has a white shadow on its end so that bars on the top rows, which otherwise do not include gradients, are distinguishable. Each bar has borderlines on top and bottom to emphasize the row-based structure.

Piled bars are the only design, of the three, which has *overlaps* across records. Smaller bars in a row need to appear above the longer bars. To distinctly convey overlapping bars along a row, we designed a monochrome gradient coloring approach, presented in Figure 33. Our approach uses color brightness to differentiate overlapping bars. Alternative designs may adjust the use of color hue and luminance, and overlay bars with different shadows and minor layout adjustments along each row.

Occlusions across bars also limit the use of other visual encodings, such as color, to visualize additional data attributes. The readability of overlapping bars is hindered more as the bar-ends get closer within a row, either because of more columns, or because of the data distribution. Inserting records, or increasing bar height ↕, may increase number of columns, and thus increase overlaps.

While the visual design of *piled bars* is similar to *horizon charts* with multiple bands on each row, piled bars visualize records *sorted* by value, and do not follow a time-series like horizon charts. All columns of piled bars share the same scale and there are axis brake points, unlike horizon chart where columns represent different sub-bands of axis collapsed on top of each other with varying color luminance across different bands.

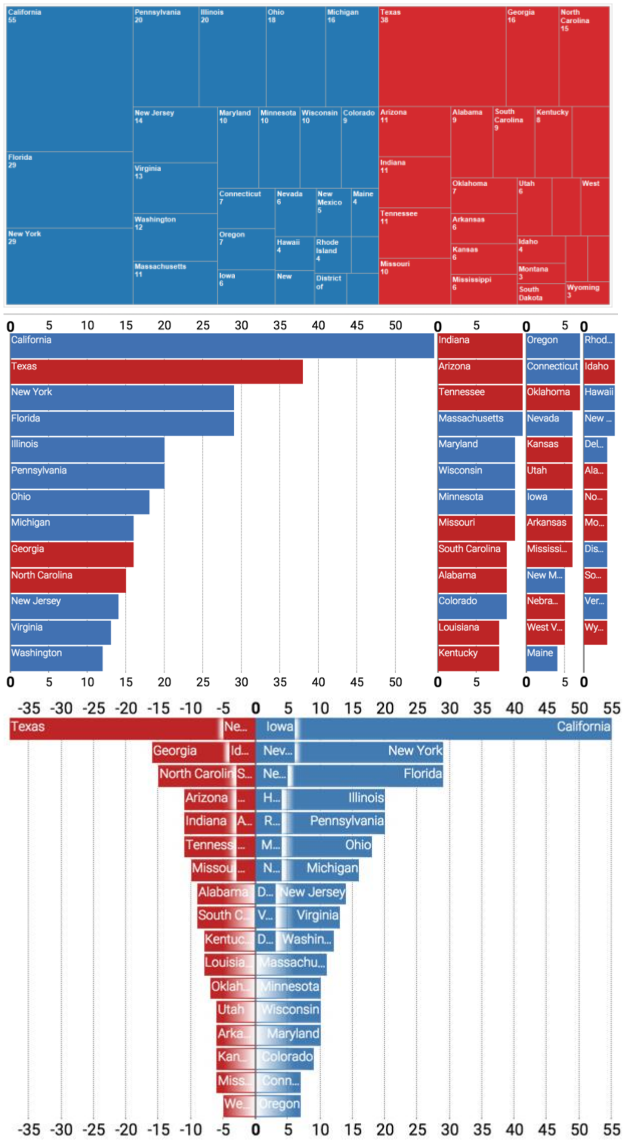
9.2.4 Labels, Use of Color, and Bi-Directional Bars

First, we considered grouping records by color and direction. Figure 34 shows data that represents two groups. The overlaps in *piled bars* limit the use of color-coding (i.e. Del... Utah Washing...). Instead, the sides of the baseline (←0→) can be used to organize two record groups to allow comparison. However, this approach is limited to two groups. In contrast, *wrapped bars* can display multiple groups with multiple colors, and *treemaps* that can effectively group (nest) multiple records spatially to represent distribution of group totals. In addition, *piled bars* reveal the difference in the number of records across two groups in opposing axes; the visual cues are the number of columns and the number of rows on the smallest columns on both sides. Aggregated sums of the records in two sides can be visualized using a supplementary chart. Piled bars also allow comparing the maximum absolute values on both sides (←0→) along the scale using the topmost row.

Next, we consider how to represent negative values (Figure 31). In *treemaps*, block area is implicitly positive, and the sign of the values is encoded by color. In multi-column bar charts (PB and WB), the baseline can be extended in both directions (←, →) to encode the sign, and color can emphasize the column of the sign flip. Therefore, treemaps are limited to use of color encoding to show sign, while piled bars require grouping +/- values along two sides of the baseline.

When the records are to be sorted by metrics other than their numeric value (such as alphabetically), the strategies and implications depend on the selected technique. Treemap layout algorithm may be adjusted to position nodes on the targeted order, although this is an uncommon case. Wrapped bars are the most flexible as they reflect single-column lists. However, arbitrary ordering would result in non-decreasing width of new columns and non-optimal use of vertical space ↔, unlike sorted order that ensures columns for the lower end of the list are narrower. Piled bars *require* the records to be sorted by value, such that overlaps can be resolved by layering from large to small records. Therefore, it doesn’t support arbitrary record sorting.

Lastly, we consider displaying record labels (Figure 35). As the visual layout of *treemaps* and *piled bars* strictly follow the data distribution, labels must be placed within the blocks, and smaller values offer smaller label space. *Wrapped bars* are more flexible. Labels can also be placed next to bars, as in single-column bars, they may also be shown for all columns or for a selected column [47]. Alternatively, record labels can be displayed as tooltips on mouse-over to individual records in interactive use of the studied visualizations.



(a) Treemap

(b) Wrapped Bars

(c) Piled Bars

Figure 36- Visualization of electoral vote results for the 50 states in the U.S. 2012 presidential elections. Each state has a number of electoral votes (block size) and a winning party (**Democrat** or **Republican**). (Left) Treemaps grouped by winning party (from "In Praise of Treemaps" by S. Wexler at http://www.datarevelations.com/in-praise-of-treemaps.html). The distribution across two parties is emphasized. (Middle) Wrapped bars with states ordered by electoral vote. Among the states with higher votes (leftm column), **Democrat**s are more frequent. (Right) Piled bars with records grouped by party. The leading states per party are available on the top row. **Democrat**s won in three more states than **Republican**s did (observable by comparison on the columns next to 0-baseline (←0→)).

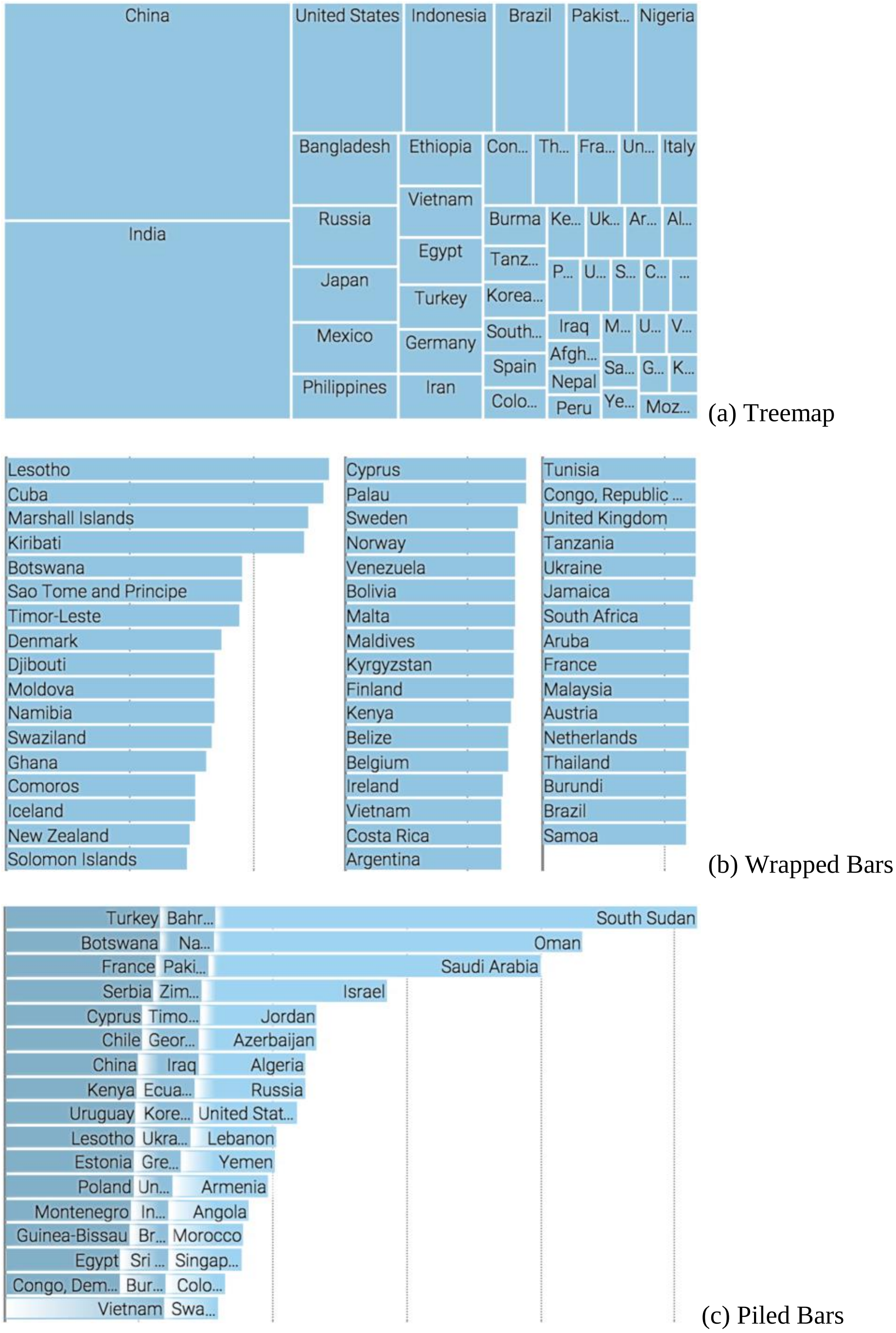


Figure 37- Selected strategies for displaying record labels across three techniques. (Left) Treemaps: Labels need to appear within the blocks. (Middle) Wrapped bars: Labels can appear within or next to blocks. (Right) Piled bars: Labels are at the tip of the bar within its visible area.

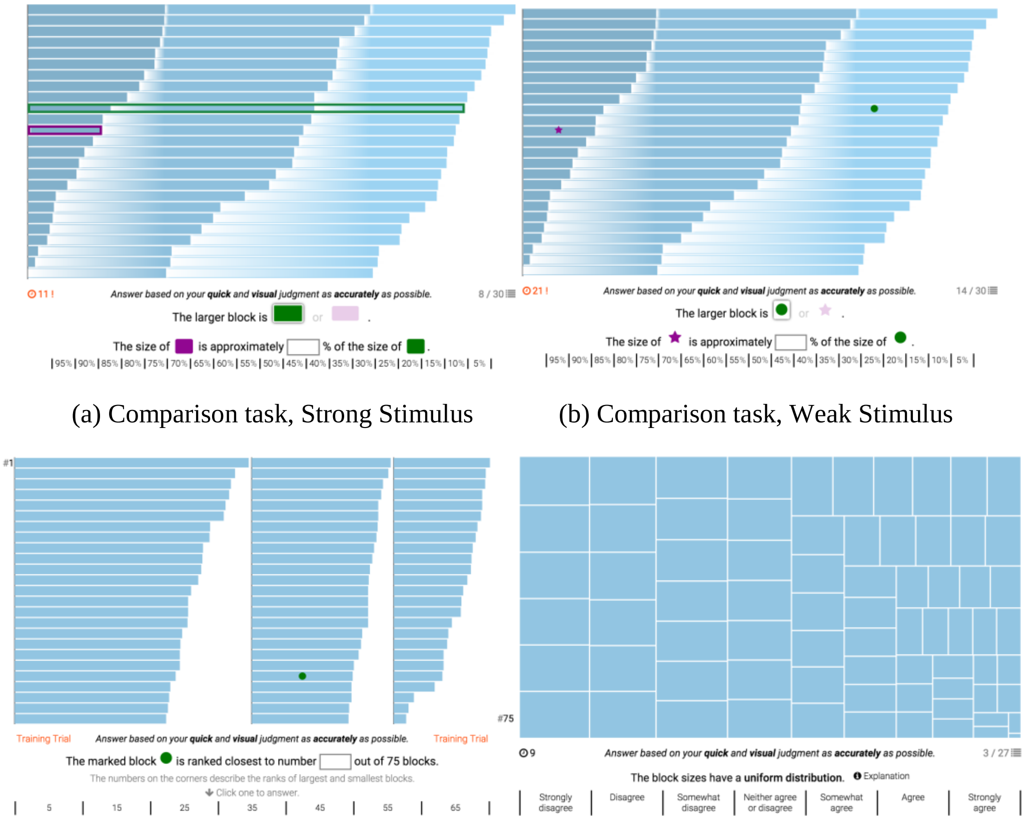
9.3 *Crowdsourced Perceptual Evaluation*

To evaluate graphical perception performance of the three visualization techniques, we designed online crowd-sourced experiments on three task types under varying data densities, chart layouts, and stimulus alternatives is appropriate. This chapter first describes the three tasks, and the shared settings and procedures in conducting these experiments. It follows with the detailed description and results per each task.

9.3.1 Tasks

To cover a wide range of perceptual characteristics of the alternative designs, we chose three graphical perception tasks (Figure 36) such that the answer would be *(i)* data-driven (i.e. changing data would predictably influence the answer), *(ii)* can be given within a few seconds following a quick impression in a casual use, *(iii)* based on a single chart. The tasks were designed to apply fairly to all chart designs. We present a summary of the three tasks below.

Comparison of two records: Two records (blocks) are highlighted. The participant determines which is larger and by how much. Comparison is the basis of visualization. However, this task focuses on two marks, and does not require reading the whole chart. This task is thus insufficient for assessing the perception of data distribution.



*Answer based on your **quick** and **visual** judgment as **accurately** as possible.*

The larger block is 🟩 or 🟪 .

The size of 🟪 is approximately □ % of the size of 🟩.

(a) Comparison task, Strong Stimulus

*Answer based on your **quick** and **visual** judgment as **accurately** as possible.*

The larger block is ● or ★ .

The size of ★ is approximately □ % of the size of ● .

(b) Comparison task, Weak Stimulus

*Answer based on your **quick** and **visual** judgment as **accurately** as possible.*

The marked block ● is ranked closest to number □ out of 75 blocks.

The numbers on the corners describe the ranks of largest and smallest blocks.

🡓 Click one to answer.

(c) Ranking task

*Answer based on your **quick** and **visual** judgment as **accurately** as possible.*

The block sizes have a **uniform distribution.**

(d) Overview Task

Figure 38- The graphical perception tasks of our experiments. (Top row) Comparison task, with *Piled Bars* sample. Compared blocks are highlighted using block background/border on the left (strong stimulus), and using dot-marks in the middle of visible portion of the block on the right (weak stimulus). (Bottom left) Ranking task, with *Wrapped Bars* sample. (Bottom right) Distribution overview task, with *Treemaps* sample.

Ranking of a record: The participant determines the rank of a highlighted record among all records. Ranking is a common task, such as finding the rank of a country or a university on an ordered list. This task requires observing the complete data distribution in relation to the focal record. While the rank of each record can be displayed by default (increasing chart ink) or on interaction (with a tooltip), graphical perception allows a quick assessment of the record ranks. When the data is visually sorted, the position of the record among all records suggests its rank. Thus, sorted visualizations avoid tedious size comparison across all records for ranking, and ranking becomes independent of the distribution characteristics.

Overview of all records: The participant is asked to assess whether a given statement on data distribution matches the displayed data. This task is solely based on interpretation of the overview of data. No individual records are highlighted, and the data is generated with specific targeted distribution characteristics. Our rationale is that understanding the overall distribution of data, without anchoring to a set of selected marks, is also an integral part of visual data comprehension.

Among other overview tasks, finding min/max is trivial in sorted data. While mechanical computation of *average* and *variance* is easy, such numeric characteristics are not naturally perceptible given many (50+) records, and can be easily annotated on the chart if necessary. We also avoided tasks that would require interaction within the chart to answer, such as clicking on a block that may best present the mean or the median. Specifically, the overlapping design of piled bars could introduce selection (motor-skill) errors that may negatively influence the measurements. As we aimed to assess how well the visualization, by itself, can communicate the data, we did not use the line-up protocol [66] which presents multiple charts with a presumable outlier for hypothesis testing. Charts are commonly shown in isolation to illustrate a single set of measurements, rather than with multiple alternatives that may serve as anchors to understand distribution differences. Overview tasks can also compare characteristics across data groups within a single chart, such as the moving average over time series [34], or differing glyphs per category in scatterplots [52]. We avoided such tasks since they require a design change, either using color or bi-directional multi-columns, which are not applicable fairly across all chart types in a similar fashion.

	Chart Type Between subjects			**Record #** Within subject	**Multi-Column (W&PB) Layout** Within subject		
Setting	**TM**	**WB**	**PB**		**Column #**	**Bar ↕**	**Bars/Col.**
Data Density	▯			**75#**	3C	**16px** Fixed	25
				150#	6C		
				300#	12C		
Column Layout	▯	▯		**75#** Fixed	**3C**	16px	25
					6C	32px	13
					11C	62px	7

Table 7- The factorial experiment design shared by all tasks. The settings include variations in data density (record count, 75#-150#-300#), and column layout (column count, 3C-6C-11C).

9.3.2 Experiment Factors and Design

Each participant answered multiple questions (trials) of a *fixed* graphical perception task on a *fixed* chart type with variations in data/chart configuration. Participants were randomly assigned to a trial group across *Data Density* or *Column Layout* settings, as shown in Table 7, and exemplified in Figure 36. The *Data Density* setting investigates the impact of data density (75, 150 or 300 records) across three chart types: treemap, wrapped bars, piled bars. In multi-column bar charts, bar heights were fixed and the column count was dependent on the record count. The

Column Layout setting investigates the impact of multi-column chart layout with three column count conditions (3, 6, 11 columns) given 75 records, with trial groups for wrapped bars and piled bars respectively. Bar height was dependent on column count. Since *column layout* setting does not apply to treemaps, the perception of treemaps can be studied under the *data density* setting only.

9.3.3 Chart Parameters

The charts had 800×450px size (16×9 aspect ratio). Treemaps were generated using the squarified layout of d3.js [18] defaults (v3.5.5) with 2px border between blocks. For multi-column bars, gridlines were hidden except the baselines. There existed 2-pixel ↕ gap between rows in wrapped bars and piled bars, and 5-pixel ↔ gap to separate columns in wrapped bars.

9.3.4 Participants

Each experiment condition was answered by 20 participants. We repeated comparison experiments with both strong and weak stimulus design to highlight selected records, as stimulus choice may influence the chart/block perception and required training. We recruited 100 participants for each task (and stimulus), totaling to 400 participants in our experiments.

The participants were recruited using Amazon Mechanical Turk. The qualification requirements were set to historical performance of at least 90% approval rate and at least 1,000 HITs completed. The participation was geographically limited to the U.S. following the IRB requirements of this project. Participation from mobile devices and screen sizes with less than 1280×800 pixel resolution was rejected to ensure the physical device can fully display the tasks. A participant could not partake in multiple experiments. The participants were rewarded with a targeted $8/hour rate, based on expected task durations.

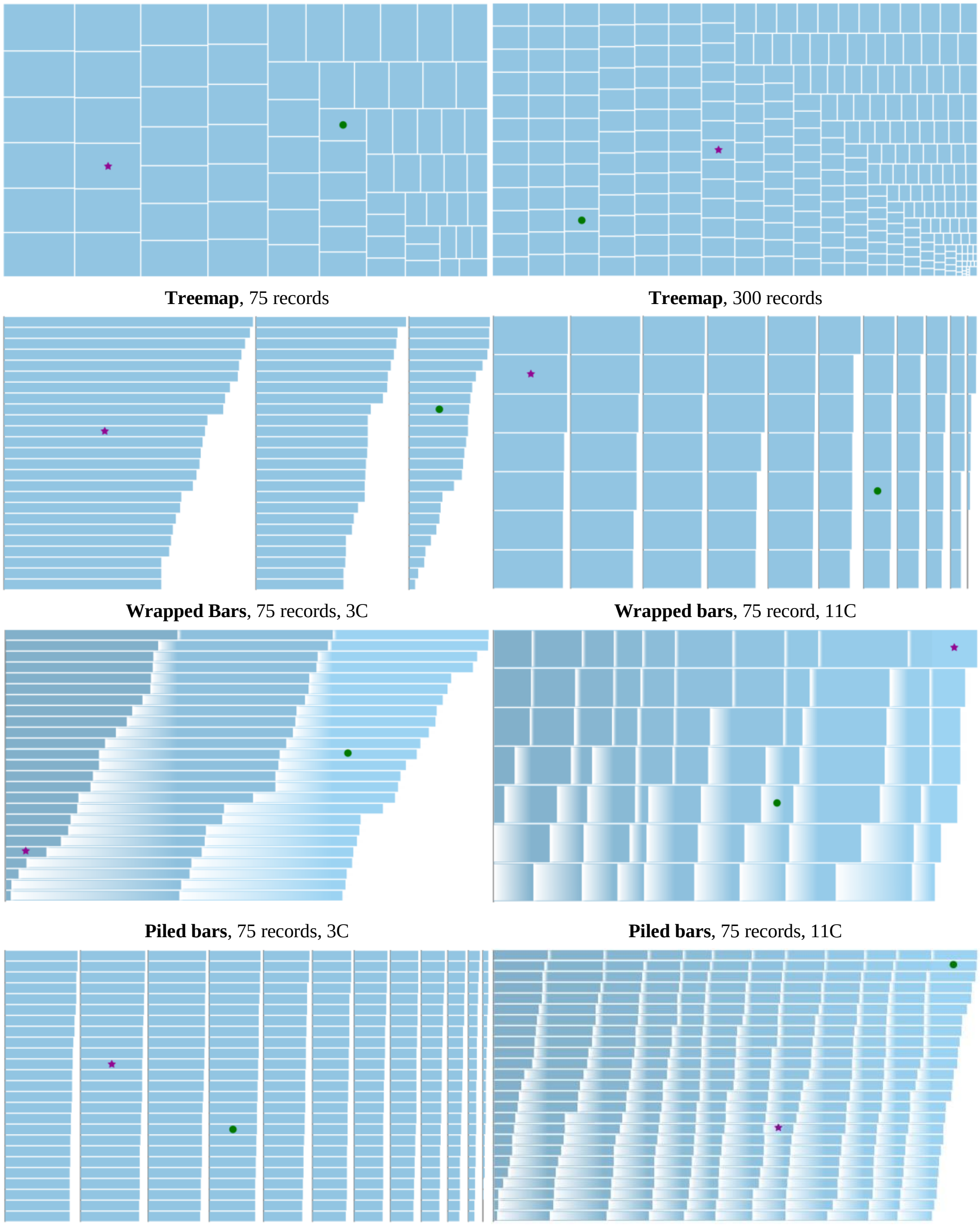
9.3.5 Training and Other Procedures

The experiments included multiple approaches to train the participants and to collect high quality data. All experiments included *training trials* using simpler versions of the task to ensure that the participants were able to understand the task. The participants could only proceed when they answered training trials correctly. They were allowed to repeat trials until they found the correct answer. In experiment trials, participants were not allowed to change their answers. To help participants stay focused while repeatedly answering the same task under different data and layout conditions, we presented a training trial after ⅓ and ⅔ of experiment trials. As in initial training, participants needed to answer these trials correctly to proceed, and they could repeat their attempts until finding the correct answer.

We also prepared animated training sequences to explain chart designs by transitions from single-column bar charts. In this sequence, the participant first saw 75 records in a single-column overflowing chart, with an animated scroll showing all the records. Then, on a button click, the single-column chart was transitioned to the chart type of the experiment with animation. The participant observed three data distributions and transitions, and could replay the sequences. The animated sequences were shown as the first step into the study. Since the strong border/background stimulus was self-explanatory for the chart design, we did not use animated sequences for this strong stimulus in comparison task.

When the participant selected an answer, the answer and response time were recorded, and the study progressed with a new trial. The marked block(s), if the task required, were visible until the task was answered. A time ticker was displayed next to the task. At 10 seconds, the ticker changed to display **10!** (note the exclamation point) to alert the participants of the passing time.

After running experiments, we confirmed that analyzed data correctly represents the experimental settings, with correct number of trials and variations per each participant, and the number of participants per each trial group. The experiment data, results and analysis scripts are accessible at **github.com/adilyalcin/chubuk.exp**.



Treemap, 75 records

Treemap, 300 records

Wrapped Bars, 75 records, 3C

Wrapped bars, 75 record, 11C

Piled bars, 75 records, 3C

Piled bars, 75 records, 11C

Wrapped bars, 300 records

Piled bars, 300 records

Figure 39- Sample charts from comparison experiments with varying record sizes and layouts.

9.4 Evaluation for Comparison Task

Each participant observed a chart with two highlighted blocks (Figure 37), and estimated what percentage the smaller block is of the larger block. Specifically, we first asked, “The larger block is A or B?” with random A-B order, where A and B represent the visual marks. After selecting an answer (e.g., B), we then asked, “The size of A is approximately [__] % of the size of B.” with A-B order based on the previous answer. The answer options were multiples of 5%, ordered from 95% to 5% under the question. Our design aimed to assist participants in focusing on their judgment at commonly expressed perception granularity (5x%) as reported in previous studies [81], [138]. Each participant answered 30 trials in randomized order on a single chart type with 10 conditions on true percent of difference, and 3 conditions on density or column layout.

Sixty uniformly distributed random data configurations were generated, as a combination of 10 true percentages (TP) and 2 settings (*Density, Layout*), with 3 conditions on each setting (75, 150, 300)# or (3, 6, 11)C. We selected 10 true percentages (TPs) at non-regular points in relation to 5% intervals (8, 17, 23, 38, 47, 53, 62, 77, 83, 92)%, such that the accuracy of an answer can be measured within 1%. The larger value was picked randomly among the top 25% of the sorted data. The smaller value was computed using the true percentage, and it replaced the smallest value. The same data configurations were used across all chart types. We used five training trials with (75)# records and (10, 30, 50, 70, 90)% for true-percentages and answer options.

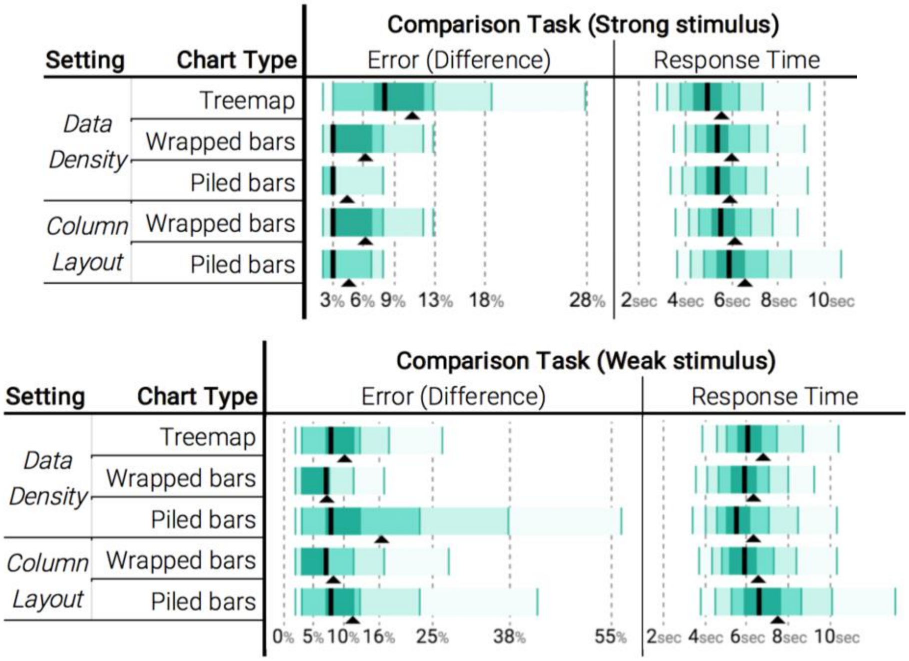


Figure 40- The high-level overview of graphical perception performance results for comparison task across three chart types in two settings and with two stimulus types (Top: strong stimulus with the outline. Bottom: weak stimulus with mark-type). Each box plot includes 20 participants, and 600 responses. The bars in box-plots show percentiles in 10% increments, ▌ shows the median, ▲ shows the mean of values within 10-90 percentile.

We ran two comparison experiments with two stimulus designs, as the stimulus can interfere with graphical perception of charts and comparison performance [138]. For the first stimulus, we highlighted the selected records (blocks) with colored background (█ - 🟩). Since overlaps in piled bars limit the use of background color, we highlighted the border in piled bars. The stronger background/border stimulus explicitly highlights the shape of the block, and the perception would focus on comparison of rectangular shapes. For the second stimulus, we highlighted the selected records with colored marks (●,★) placed in the middle of the visible portion of the block. This design is consistent across all chart types, and adapts to the visible portion of piled bars. This stimulus does not explicitly describe the shape of the record blocks. Comparison requires finding the small stimulus in the chart and understanding the total shape of the block. Thus, this weaker stimulus requires a deeper understanding of the chart design for correct evaluation.

9.4.1 Results and Discussions

To analyze the perceptual performance in comparison, we measured the error as the absolute difference between the response percentage and the true percentage difference of marked blocks. Figure 38 shows the overview of the responses in error ratio and response time across two stimulus types. To analyze the effect of data density (75, 150, 300)# and column layout (3, 6, 11)C across three chart types, we use the group means with 95% confidence intervals by bootstrapping [37] (Figure 39, Figure 40). Bootstrapping produces statistical estimates based on resampling the observations with replacement. It has been advanced in psychology [36] to address the shortcomings of significance testing and p-values, and we adopt it here for similar reasons. We also present significance results from statistical tests for comprehensiveness as appropriate. We first discuss results with the background stimulus, and then the mark stimulus.

<u>Results with outline stimulus:</u>

Based on the overview across five trial groups (Figure 38), *piled-bars* had the least error, while *treemaps* had the most. Bootstrapped confidence intervals of mean errors (Figure 39) show substantial differences across PB and TM, affirming that comparison is improved by the shared baseline of PB, and hindered by the area encoding of TM, in line with earlier studies [32], [61]. The higher accuracy of PB compared to WB (Figure 38, Figure 39) is also parallel to earlier reports on accuracy on aligned vs. unaligned bars. We also applied standard parametric statistical tests to responses in data density setting with mixed linear two-way, factorial model with interaction using the subject as random effect. Chart type was found as a significant factor (F(2, 1734) = 8.21, p < .001), while data density (F(2, 1734) = 2.68, p < .069) and their interaction (chart type x data density) (F(2, 1734) = 2.14, p < .074) were not significant. A Tukey HSD post-hoc tests found significant differences across PM vs. TM (p < .001), and WB vs. TM (p < .004), and no significant difference across PB vs. WB (p < .86).

Figure 39 shows that increasing the record count reduced the accuracy with TM (potentially due to smaller block sizes), slightly increased the accuracy with PB (potentially due to the overlapping gradients cueing on length differences), and slightly decreased the accuracy with WB (potentially due to smaller bar widths). Our results show no substantial effect of multi-column layout on the comparison accuracy, although PB outperformed the WB in all configurations. Lastly, only 62 responses (%2) misidentified the larger block. Among those, 35 were for (83 or 92)% true percentage (similar sized blocks). Only 92 responses (%3) had an error > 30%. The aggregated results (Figure 38) show small variation in response times, with TM leading by small, but not significant, margin.

Results with mark stimulus:

Wrapped bars had the least amount of error with significance under varying data densities (Figure 40). In line with results from the background stimulus, TM performed worse than WB, and there is no substantial and consistent effect of the column layout on accuracy. However, PB showed a significantly higher error rate when the weaker mark stimulus was used. The ratio of incorrect answer to larger block was 17% in PB, while only 1.4% in WB, and 2.8% in TM. This suggests that some crowdsourced participants may not have perceived the piled bars correctly. This may be due to a dominant perception of the marked blocks as *only* their visible portion, not including the overlapped section extending to the baseline. With mark stimulus, the confidence intervals of mean errors are wider (i.e. responses have more variation) and mean error is larger across trial groups compared to the background stimulus. Smaller mark stimulus may also have hindered finding highlighted blocks and observing complete block size. We also applied standard parametric statistical tests to responses in data density setting with mixed linear two-way, factorial model with interaction using the subject as random effect. Results confirm the significant effect of chart type on error (F(2, 1734) = 11.13, p < .001), and significant differences of error across PB v s. TM (p < .004) and PB vs. WB (p < .001), while no significant difference across TM vs. WB (p < .35). There was no effect detected for the data density (p < .14) or its interaction with chart type (p < .74).

The results acknowledge the effect of stimulus on comparison task, with two stimulus designs revealing different processes of perception. When the bars were more likely to be perceived in full with the stronger background stimulus, the shared baseline of aligned bars and the higher data resolution of *piled bars* improved comparison accuracy. On the other hand, the weaker mark stimulus made it harder to observe the complete size of blocks, leading to a reduced performance, and much more significantly for *piled bars* because of its overlapping design.

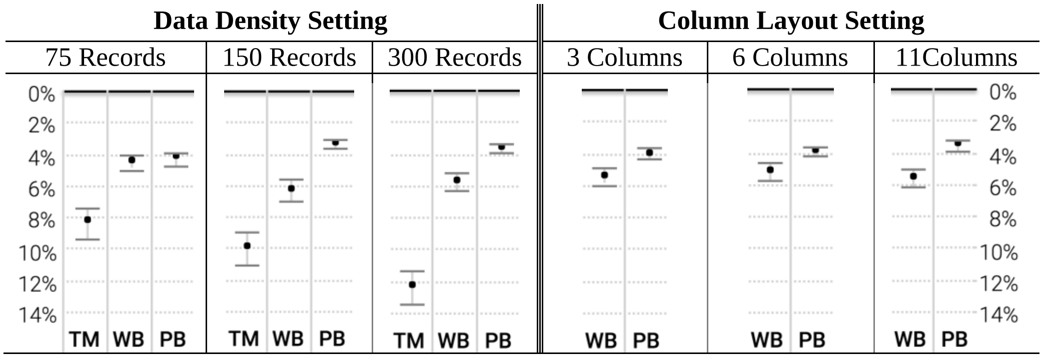


Figure 41- Analysis of accuracy (% error) in comparison task with outline (█-█) stimulus across data density and column layout settings. • shows the mean, the bars show 95 confidence intervals. Each column includes 200 responses (20 participants on 10 TPs).

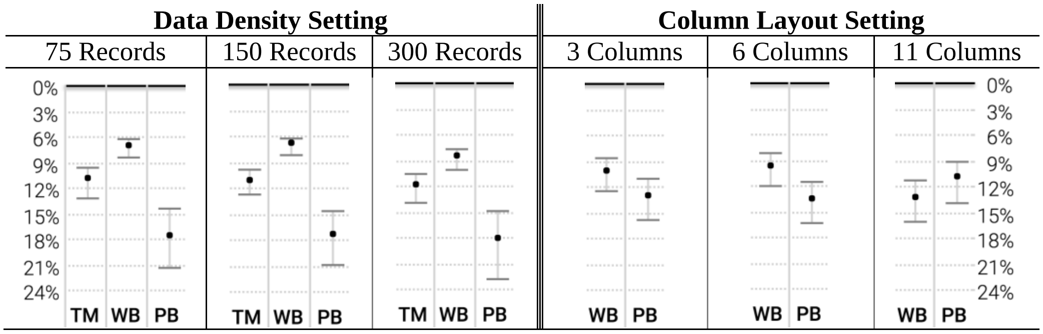


Figure 42- Analysis of accuracy (% error) in *comparison* task with mark (●-★) stimulus across data density and column layout settings. • shows the mean, the bars show 95 confidence intervals. Each column includes 200 responses (20 participants on 10 TPs).

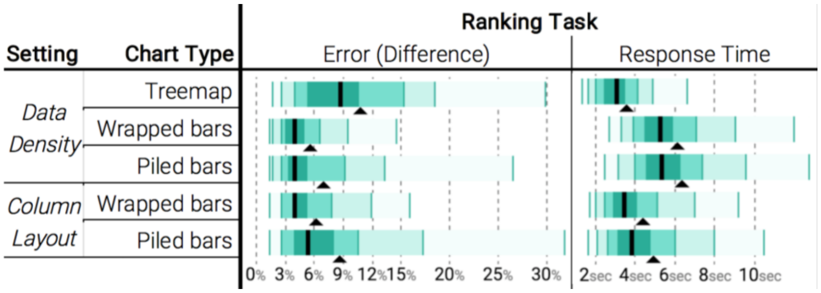


Figure 43- The overview of graphical perception performance results for ranking and overview tasks in two settings and three chart types. Each box plot includes 20 participants, and 600 responses. The bars in box-plots show percentiles in 10% increments, ▌ shows the median, ▲ shows the mean of values within 10-90 percentile.

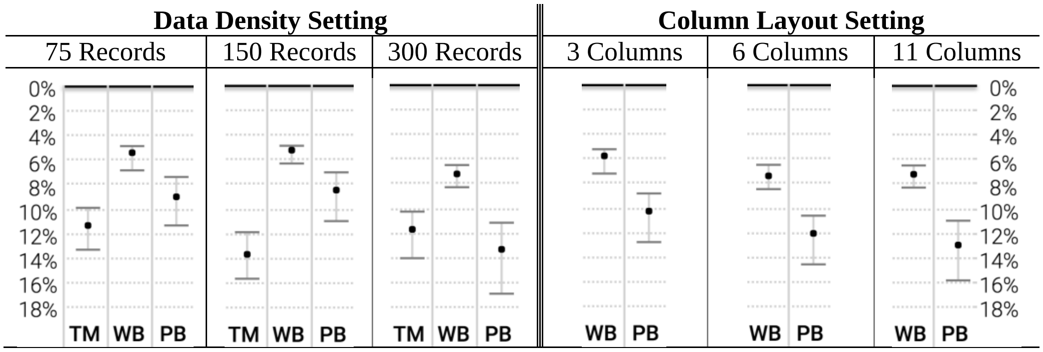


Figure 44- Analysis of accuracy (% error) in ranking task across data density and column layout settings. • shows the mean, the bars show 95 confidence intervals. Each column includes 200 responses (20 participants on 10 TPs).

9.5 *Evaluation for Ranking Task*

The participant observed a chart with a block marked with ● placed in the middle of visible portion of the block. We asked, “The marked block ● is ranked closest to number [__] out of *N* blocks”, where *N* is the number of blocks. The marked blocks were generated using 10 percent-based rankings (8, 17, 23, 38, 47, 53, 62, 77, 83, 92)%, rounded to an integer. For example, a 23% ranked record across 150 records has rank 35. We presented 14 options, evenly spaced across all records and in absolute ranks since it is a natural form of interpreting ranks given a variety of scale. Each participant answered 30 trials in randomized order on a single chart type. Across five trial groups, 100 participants answered 3,000 rankings. The data was generated using random normal distribution with μ:2 and σ:0.8, with absolute values. We showed index labels for the first and last ranked records on the chart corners to help reading the chart structure. We used seven training trials with 75 records and (5, 15, 25, 35, 45, 55, 65) options for true-ranks and answers.

Results and Discussions

We measured the accuracy of a ranking response as a percent difference from true absolute rank normalized by the number of blocks (max rank). To analyze the effect of data density (75, 150, 300)# and column layout (3, 6, 11)C across three chart types, we used bootstrapping for group mean to generate 95% confidence intervals (Figure 42).

Based on the overview of the responses across five groups (Figure 41), *wrapped bars* had the least error and *treemaps* had the most with significance across confidence intervals. Based on bootstrapped averages (Figure 42), WB performed substantially better than TM and somehow better than PB under all settings. TM and PB have no consistently significant difference in accuracy under varying settings. We also applied standard parametric statistical tests to responses in data density setting with mixed linear two-way, factorial model with interaction using the subject as random effect. The results show significant effect of chart (F(2, 1734) = 3.54, p < .03), data density (F(2, 1734) = 7.41, p < .001), and their interaction (F(4, 1734) = 5.87, p < .001). The interaction effects can be observed across confidence intervals in Figure 42, and details are accessible at the result repository. Across chart types, the a significant effect was detected across TM vs. WB (p < .025), while the effects across other chart type pairs (TM vs. PB and WB vs. PB) had p > .17.

With the increase of record count (data density), the accuracy of WB and PB suffered, while accuracy of TM was not affected. With an increase in column count (effect of column layout), WB outperformed PB with wider margin of difference, while there was no substantial difference within a chart type across different layouts. WB and PB were slower in response time compared to TM. This suggests that given a multi-column bar chart (either WB or PB), the participants are likely to trace the rows and columns of the chart to give a more accurate answer, while still maintaining a six second response time on average. Our results also show that varying data density has a larger effect on the response time (slower performance) compared to varying column layout under fixed density.

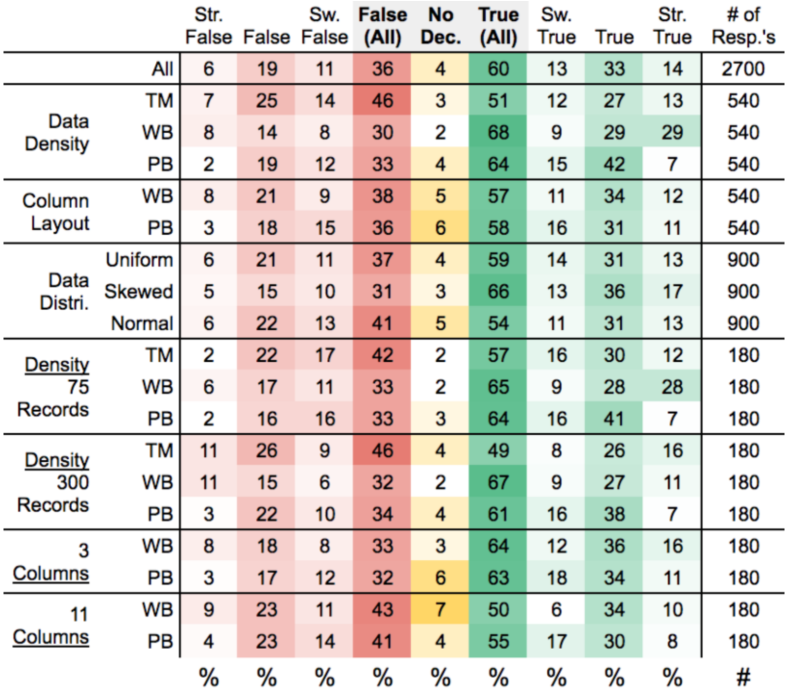
9.6 *Evaluation for Distribution Overview Task*

For the overview task, the participant stated their agreement to a data distribution statement given a chart, on a 7-point Likert scale as shown in Figure 36. The chart and the question of a trial were selected among three distribution characteristics, resulting in nine permutations. Each trial group is based on three conditions on data size *or* column layout, and each participant answered 27 experiment trials in randomized order. We generated 10 groups of random data distributions for 27 trials. Each data group was answered by two participants, totaling to 20 participants answering 540 trials.

The three data distribution characteristics of this experiment are as the following, with explanations presented in our experiments: (i) Uniform distribution, i.e. "There is a block of all possible sizes". (ii) Skewed distribution, i.e. "There are a few blocks that are substantially larger than all the rest". (iii) Normal distribution, i.e. "There are more medium-sized blocks than small and large blocks." In animated training sequence, we presented one sequence for each data distribution with a text describing the distribution characteristic of the data. After the sequence, three training trials were shown with "agree/disagree" options. Experiment advanced when the statement matched the data distribution.

Results and Discussions

We identified each response as *true*, *false*, or *no decision* based on agreement of the statement with the data distribution, and converted the scale from agreement to correctness. For example, a "*strongly agree*" response to a *uniform* statement for a *uniform* data distribution is "*strongly true*", and "*somewhat disagree*" response to a *normal* statement for a *skewed* distribution is "*somewhat false*". Figure 43 presents aggregated visual analysis of the responses across correctness, confidence, and different chart types under various density and layout settings.



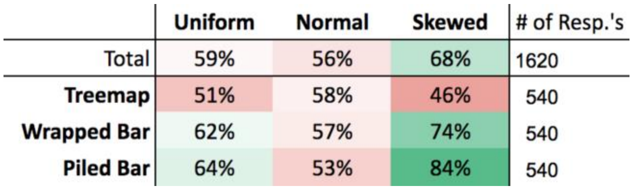
		Str. False	False	Sw. False	**False (All)**	**No Dec.**	**True (All)**	Sw. True	True	Str. True	# of Resp.'s
	All	6	19	11	36	4	60	13	33	14	2700
Data Density	TM	7	25	14	46	3	51	12	27	13	540
	WB	8	14	8	30	2	68	9	29	29	540
	PB	2	19	12	33	4	64	15	42	7	540
Column Layout	WB	8	21	9	38	5	57	11	34	12	540
	PB	3	18	15	36	6	58	16	31	11	540
Data Distri.	Uniform	6	21	11	37	4	59	14	31	13	900
	Skewed	5	15	10	31	3	66	13	36	17	900
	Normal	6	22	13	41	5	54	11	31	13	900
Density 75 Records	TM	2	22	17	42	2	57	16	30	12	180
	WB	6	17	11	33	2	65	9	28	28	180
	PB	2	16	16	33	3	64	16	41	7	180
Density 300 Records	TM	11	26	9	46	4	49	8	26	16	180
	WB	11	15	6	32	2	67	9	27	11	180
	PB	3	22	10	34	4	61	16	38	7	180
3 Columns	WB	8	18	8	33	3	64	12	36	16	180
	PB	3	17	12	32	6	63	18	34	11	180
11 Columns	WB	9	23	11	43	7	50	6	34	10	180
	PB	4	23	14	41	4	55	17	30	8	180
		%	%	%	%	%	%	%	%	%	#

Figure 45- Responses from the overview task. Accuracy values are shown in percentage and color-coded, with darker color showing larger value (True: green. False: red. No-decision: yellow). For example, of the 540 responses given for treemaps, 46% were false, while only 30% were false of the responses to the wrapped bars.

Treemaps had a higher percentage of false answers compared to *wrapped bars* and *piled bars*, which commonly show a similar accuracy. For example, for responses under *Data Density* setting, TM had 46% false responses, while WB had 30% and PB had 33%, given 540 responses in total for each chart type. Regarding the confidence level of the responses, WB has the highest ratio of "strongly" confident (false or true) responses in most settings. Under constant data density of 75 records, increasing column count (3 to 11) (and using thicker bars) increases the undecided or false responses. Using 3 columns (and bars with 16px height) performed better in comparison to 11 columns (with 62px height).

We also performed a standard statistical analysis based on a generalized linear mixed model for the binary outcome (with no-decision responses considered false). We detected significant effect of the chart type (F(2, 57) = 8.59, p < .001). A Tukey HSD post-hoc analysis reveals a high significant difference across PB vs. TM (p = .0042) and WB vs. TM (p = .0002), and no significant difference across WB and PB (p = .58), further supporting the analysis presented above on the frequency of response accuracy.

The accuracy effect across chart type vs. distribution characteristic is shown in Figure 44. Responses in column-layout setting are not included since this setting does not apply to treemaps. The charts show similar performance under normal distribution, however treemap performed significantly worse for skewed distribution, as well as uniform distribution. The results suggest that piled bars carry an advantage for observing skewed distributions, potentially because of shared alignment.



	Uniform	Normal	Skewed	# of Resp.'s
Total	59%	56%	68%	1620
Treemap	51%	58%	46%	540
Wrapped Bar	62%	57%	74%	540
Piled Bar	64%	53%	84%	540

Figure 46- Accuracy (ratio of true responses) across data distribution and chart types, based on the data density setting. Values are color coded from red to green, with the white midpoint at 61%, the accuracy considering all 1620 responses.

Analysis of the response time for overview task (Figure 45) shows that treemap was also the slowest in this task, compared to wrapped-bars and piled-bars under varying data density. A comparison across WB and PB shows that responses to variations in column layout were slightly slower compared to variations in data density. Given similar accuracy performance across PB and WB, our results suggest that piled bars may have a leading edge with shorter response time on a small margin while maintaining similar accuracy.

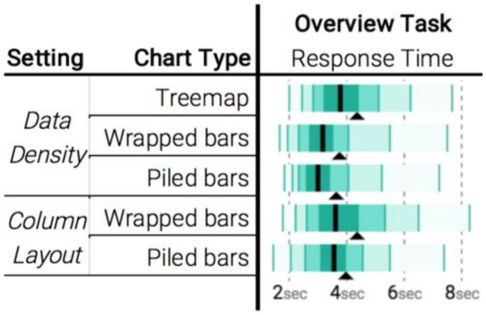


Figure 47- The overview of response time performance results for overview task. Each box plot includes 20 participants, and 540 responses. The bars in box-plots show percentiles in 10% increments, ▌ shows the median, ▲ shows the mean of values within 10-90 percentile.

9.7 Summary of Results

Overall, *wrapped bars* yielded a high perceptual performance among the three chart designs. Our results show that wrapped bars performed either as the best (comparison task with mark stimulus, ranking task), or as the second best (comparison task with border stimulus) of the three chart alternatives. For the overview task, it performed similar to piled bars, and better than treemaps. Its performance is likely due to its clean, easy to interpret, non-overlapping design. It strikes a balance between a single-column sorted bar chart and the complexity of multi-columns by explicit separation of columns. Given that its design can be extended by color and bi-directional encoding, and its flexibility to show labels in various forms, our analysis and evaluation shows that wrapped bars technique is a perceptually well performing design to present dense numeric data in wide-aspect charts.

Treemaps did not perform the best in any task in our experiments. It had the highest mean error under comparison task with background stimulus, and ranking task. Its lower performance for comparison is predictable since treemaps rely on area assessment instead of length assessment, and its lower performance for ranking task reflects its relaxed ordering/layout strategy. Results from overview task show that *treemaps* do not outperform multi-column bar charts either. Overall, our results suggest that treemaps are not a preferable design when records do not have an explicit hierarchy. Its visual design purpose of using all pixels in chart area does not increase perceptual accuracy under flat numeric lists, and common tasks of comparison, ranking and distribution overview.

Piled bars, a new multi-column bar chart design, performed the best for the comparison task with highlight stimulus (with its advantages in increased data encoding resolution and shared baselines), the second best for ranking task (after wrapped bars) and similar to wrapped bars for overview task, in terms of accuracy. However, when mark-type stimulus was used for the comparison task, its performance was significantly lower. This is potentially due to the perception of a piled bar not considering the overlapping portions of the block in our crowdsourced experiment setup. This bias may be countered with more training, or with inclusion of scale axis (ticks) to stress the shared baseline. With sufficient training and guidelines in reading its structure, our results suggest *piled bars* have the potential to improve data perception with its fully aligned design and higher resolution among data axis.

9.8 Limitations and Future Work

In this study, we focused on basic graphical chart designs without labels, legends or axis. The display of labels may impact readability of the chart. We did not evaluate designs with color or bi-directional axes, or display axis labels or gridlines in multi-column charts to maintain fairness to *treemaps*. Including such guides is likely to further improve accuracy for both *wrapped bars* and *piled bars*.

The results were reported from data densities of up to 300 records in 800×450 pixel chart area, with randomly generated uniform, normal, and skewed distributions. Figure 37 demonstrates that 300 records within the selected chart size create a dense setting for casual visualizations; doubling the scale would impact the size and readability of individual records. If this requirement is relaxed and experienced data analysts become the target audience, the record count may be increased further in future studies. Our findings may not extrapolate to higher data densities, smaller (mobile) or larger displays. Increasing data densities on highly skewed data may amplify the strength of treemaps with its non-overlapping, space-filling design, and emphasis of part-of-whole relations. The results are based on crowdsourced experiments that have limited training opportunities and cannot control for correct perceptual responses. Future studies may extend our results and analysis with variations in data size, distributions, and experiment setup, as well as chart design, such as different rendering strategies for the use of color and overlays (shadows) for piled bars technique.

Lastly, let’s consider how Keshif may be extended to support dense visualization of sorted numeric data. The categorical summary of Keshif is a single-column, dynamically sorted bar chart, and exhibits the problem of not visible overview. Wrapped bars is a natural extension for when the chart is positioned on a wide panel, or when a specific chart is enlarged to cover larger screen space (a potential future update). A wrapped-bar design would support multiple selections and interactions as noted earlier. However, one critical design issue to address would be the ability for continous scroll in wrapped bars. We assumed all the records would fit in a single display. However, in a generalized setting, the wrapped bars may extend beyond visible screen space, potentially towards right of the right-most column. Piled bars design would present even more challenges. The Keshif model requires that each glyph should be able to support multiple selections simultaneously, if possible. Given the overlapping nature of piled bars, further subdivisions may make the interface harder to read. Another challenge of generalized piled-bars is that it would not allow scrolling to see larger list of numbers. One potential interaction may be to *zoom-in* to the list, focusing on columns closer to zero-baseline. Given that it is unfamiliar, and potentially requires more training to increase perceptual accuracy, it may not be preferred over wrapped bars design for extending Keshif. Our results demonstrate that treemaps are not an analytically strong candidate, and they do not support visualizing negative values. Therefore, they are harder to generalize in a shared design basis, and not preferable over wrapped bars.

Chapter 10. Conclusion

"Innovation is not about alchemy. In fact, innovation is not about invention. An idea may well start with an invention, but the bulk of the work and creativity is in that idea's augmentation and refinement. The newer the idea, the coarser the granularity of most analysis, and the more likely people are to say, "oh, that's just like X" or "that's been done before," without any appreciation for how much work and innovation is involved in taking an idea from concept to wide practice."

Bill Buxton

in "The Long Nose of Innovation"[24]

This dissertation presented new approaches to improve upon rapid, effective, and expressive interactive visual data exploration. The contributions included (i) a new framework that brings a comprehensive structure to cognitive activities in data exploration (the Cognitive Exploration Framework), (ii) a new minimal yet expressive data exploration model (Aggregate Summaries and Linked Selections), (iii) its out-of-the-box, web-based, open-source implementation (Keshif), (iv) a contextual, in-situ help system to provide self-service training in visual data analysis (HelpIn), (v) a new visualization design for dense visualization of numeric data (Piled Bars) as well as an extensive evaluation of alternative designs, treemaps and wrapped bars, and (vi) multiple user evaluations, including insight-based analysis, barrier-based analysis, crowdsourced graphical perception, and numerous applications. Next, we present a summary of each aspect, along with future directions.

10.1 On Cognitive Exploration Framework

We first focused on the cognitive activities in open-ended visual data exploration, and presented the *Cognitive Exploration Framework* for visual data exploration. We used the framework to identify how established design guides potentially interact with cognition. We then demonstrated application of the framework in evaluating a data exploration tool by focusing on the failures and challenges. While our analysis exemplify a range of barriers tied to the framework (some of which are potentially addressable by incremental design improvements), it also raises questions about how to better support analytical goal formation and analytical evaluations by design. To move beyond the casual setting of our demonstrative user evaluation and to observe complex activities, future studies may increase training, motivation, domain knowledge and skills of the participants. Identifying the influence across cognitive stages and quantifying the differences in efforts can further guide better design of our tools, allowing us to explore data in depth more rapidly.

10.2 On the Data Exploration Model and Keshif, its implementation

We presented a minimal yet expressive model for rapid tabular data exploration using aggregate summaries and linked selections. This model constrains the search space for visualization through aggregate glyphs, and the search space for interactive querying through aggregate selections, enabling comparison of data distributions. Our implementation of this model, Keshif, is an out-of-the-box web-based tool that supports authoring visual data browsers from raw data, and interactively exploring relations in a unified, linked interaction across summaries and individual records. We validate our system by (a) presenting samples from 160+ public datasets imported to Keshif across many domains, (b) discussing a sample use case in journalism domain, and (c) results from an insight-based user study with visual analytics novices under a short-term casual use, supporting that Keshif can be rapidly learned and used to reach data-driven insights.

10.3 On AggreSet: Set-Typed Data Exploration

As a part of the proposed model, we presented AggreSet, an interactive visualization technique for exploring relations in set-typed and other attributes of multivariate datasets using a rich, scalable, clutter-free visual interface. AggreSet improves upon existing set visualization approaches using data aggregation that gracefully scales to larger set counts. The set-matrix improves the non-overlapping co-occurrence matrix design with advanced visual encodings for set-typed data, and with interactions that reveal higher order relationships. In the future, the data model and design of AggreSet can be extended to support set-dependent attributes by storing extra information along with the set membership relation. For example, the simple set-typed data model can encode the club memberships of a person, yet cannot encode the join-date and cost of each membership. Set memberships can also change in time, requiring focused, topological analysis through time dimension. Representing fuzzy set memberships is also another challenge. Finally, we are also interested in exploring how our mouse-based interaction model can be extended to other types of interaction, particularly multi-touch.

10.4 On HelpIn: Contextual In-Situ Help

To improve self-service training and help for visual data interfaces, we presented *HelpIn*, a contextual in-situ help system. HelpIn uses data and visualization features, in addition to application and action history context, to find relevant help material, and to present answers that are integrated and responsive to the active interface and dataset. We identified five modes to seek for help—Point & Learn, Topic Listing, Overview, Guided Tour, and Notifications—, as well as contextual approaches to support both help seeking and help comprehension. While our experiment with participants of majority data analytics novices show that full-featured HelpIn did not improve task performance overall compared to a non-contextual version of the same help material, both performance and subjective feedback highlights the utility of using Point&Learn, one of the modes, to seek help and to perform data analysis.

10.5 On Dense Visualization of Numeric Data And Piled Bars

Finally, we discussed and evaluated three alternative chart designs for dense visualizations of numeric data. We compared two multi-column bar charts, *wrapped bars* and *piled bars*, with *treemaps* as a non-hierarchical space-filling approach. We analyzed the design characteristics of these techniques in depth under various use cases and settings. We evaluated perceptual characteristics of the alternatives using crowdsourced graphical perception experiments based on comparison, ranking, and overview tasks. Our results suggest that *treemaps* is not an optimal choice, while commonly employed for non-hierarchical data outside its primary design purpose. *Wrapped bars* performed with high accuracy across all tasks. *Piled bars* did not outperform wrapped bars or treemaps except for comparison task with strong background stimulus. This is likely due to its unconventional and overlapping design, and the limited training opportunities in online crowdsourced experiments. Given its higher resolution data encoding design, it carries the potential to improve perceptual accuracy with trained perception and additional cues such as scale ticks and labels. In a broad sense, the results support that using treemaps, or potentially any visualization, outside the context for which it was originally designed, may be less effective compared to targeted designs that build upon the characteristics of data and visual perception.

10.6 Remarks

While this thesis spans over 250 pages with many chapters, figures and evaluations, there are some concepts not included in the work, and words not included in the body of this thesis. I provide some of the reasons below.

1) Some of the left-out concepts or ideas were not relevant within the purpose of this work. This relates most profoundly to the design of exploration model and its implementation. For example, Keshif, or its model, does not necessarily aim to support all chart types, some of which may not be analytically strong (such as wordclouds), or complex for the general needs (such as parallel coordinates). This is akin to a sculpture removing material that is not part of the core message s/he wishes to communicate, or for the viewer to experience. This thesis includes arguments as to why having such extraneous material, or unfounded choices in visual design, may have negative impact on the cognitive process in data exploration, either through the cost of decision making (bad decisions) and lower graphical perception accuracy. Task that were non-essential for data exploration, such as visualization for data presentation, were also left out of the targeted used cases. The presented, generic and systematic solution also does not restrict exploration spaces targeting a single dataset, domain, and audience.

2) While I worked on the ideas presented in this dissertation for over three years, this was still not enough time to explore all the potentials of what is proposed in this dissertation. Specifically, numerical, categorical, spatial (based on categories), and simple temporal data is not how we fully describe our world. There are time-series that describe an individual variable, spatial data that is not based on named regions (such as city names), but on a simple point on earth, relations across variables that need to be explored (such as movement patterns to/from different locations). My belief is the presented model can be extended to many of these other settings, but this thesis does not aim to present a formal proof. On a personal note, I found such aims to create formal design spaces ambitious yet impractical, and easily misguiding the researchers and the practitioners. Such formulations create crippled goals of filling in the blanks in some technical design space, where explicit, or implicit, assumptions on one side of an equation or diagram may not apply to other sides. From my personal experience, I can also note that each significant step in this thesis required taking a step back and refining the model or implementation, and challenging the assumptions of this work. Therefore, I claim no more than what is proposed in this thesis, but I suggest that the ideas can be extended to support new data types, and new tasks, one careful step at a time, with utmost care about the systematic consistency, maintaining minimalism and clarity in design, and the paths to failure when adding new features.

3) Another reason is the limits of my knowledge, experience, and perspective. While performing the work shared on this dissertation, I was on a journey where each dataset I studied, collaboration I engaged in, book or paper I read had the potential to transform some components of my work, or how I evaluated or communicated its value. As I know the transformations will continue without a doubt, and that others will hopefully build upon some of the presented ideas, this thesis will be a step towards the larger motivations and human-centered approach of this thesis. While this chapter concludes the body of this thesis, the material and ideas are not yet concluded, nor they may never be in my lifespan.

To summarize, this thesis contributed to our understanding on how to create effective visual and interactive data interfaces by focusing on human-facing challenges including design, cognition, perception, and the highly dynamic nature of data exploration. Particularly, our user studies on insight-based methodology (Section 7.2) suggests that novices using Keshif, a systematic, minimal yet expressive data exploration tool, can perform with similar insight throughput compared to more skilled audiences using more complex tools. The Cognitive Exploration Frameworks shows a high-level, comprehensive, new look at cognitive activities in data exploration, AggreSet demonstrated an improved and integrated set-typed data exploration model, HelpIn has shown how help material could be directly embedded contextually into data applications. Lastly, we have shown that transferring practices and solutions across different contexts (such as from hierarchical to non-hierarchical treemaps) may not lead to effective outcomes, and alternative, well-targeted solutions (such as wrapped bars or piled bars) would perform better

However, we have not yet reached the ideal future of *no-barriers* to make sense of data quickly and effectively. Our evaluation of the exploratory model for tabular data, and Keshif, was self-contained and high-level, and it did not include side-by-side comparison to other tools, with the rationale discussed in Section 7.2.1. We believe longer-term, real-life use and feedback will reveal more characteristics, strengths, and potential weaknesses of Keshif, and future improvements can make it applicable to wider data types, tasks, data sizes, and form factors, extending its systematic and minimalist design foundation. The Cognitive Exploration Framework does not propose new guidelines, although it suggest that high-level planning and assessment of data analysis activities are critical and currently not well supported or studied. Our evaluation of dense visualization for numeric data was constrained to fixed chart area, and a crowdsourced setting with lesser control and learning capabilities than a lab study. One of the broader challenges is enabling the broad public to truly understand, and analyze, data, with its strengths and limitations, which we believe remains a cognitive, design-driven, social, educational, and technical endeavor.

Glossary

Aggregate: A group of records that share a data characteristic / feature, such as the same categorical value(s), a numerical/temporal value within a specific range, a missing value, etc. Data selections (queries) also generate record aggregations.

Attribute: A measurement that describes an aspect of a record. An attribute may exist in the raw data, or may be *calculated* using existing raw data of a record.

Authoring: The actions that relate to creating and modifying a data browser (such as adding/removing summaries, adjusting panels, adding calculated attributes, customizing the style and presentation features, etc.).

Calculated Attribute: An attribute that is calculated using existing (raw) data attributes using a formal language specification (such as JavaScript).

Data Browser: Combination of interactive data representations (summaries and record display) in Keshif. Excludes available attributes panel of authoring mode.

Exploration: The interactive, dynamic dialogue between the user and the data in search for data-driven knowledge (insights).

Glyph: A visual object that represents a single record or a record aggregate.

Insight: An individual observation about the data by the participant, a unit of discovery [122], a data-driven knowledge.

Interaction: The communication (dialogue) between the data and the user.

Keshif: A data exploration environment (DEE). In contrast to Visualization Design Environments (VDEs), a data exploration environment offers a data exploration space with a fixed visualization and interaction design, rather than offering a highly flexible and customizable visualization and interaction design space.

Keshif API: The human *and* machine-readable representation of a Keshif Browser specification. It is based on JavaScript. The configurations can be stored in JSON (JavaScript Object Notation) format as well.

Measure Label: The textual representation of the computed measure metric of an aggregate. It can be presented in absolute or percent value (Measure Label Mode).

Measure Metric: The computed numerical value that represents a characteristic of an aggregate. For example, count of records, sum of a numeric attribute (such as $ cost), or average of a numeric attribute (such as age).

Record: A single observation, event, object, which can be composed of multiple attributes.

Record Display: Individual representation of records in the database. The records can be displayed in a list (grid), map, or as a node-link diagram.

Selection: A user-initiated query of an aggregate or a record. It includes filtering, highlighting, and comparison selections for aggregates, and mouse-over selection for an individual record.

Summary: A visual data representation that summarizes the distribution and characteristics of one data attribute or feature.

VDE (Visualization Design Environment): Software tools that offer a graphical environment to create pre-defined and custom data visualizations based on rich visual grammars, and supports interaction and data transformation pipelines.

Visual Scale Mode: The visual scale describes how the visual representation of the computed measure metric of an aggregate is visually scaled along the visual axis of the summary. Two modes are defined: Absolute, and part-of-filtered.

Visualization: The purposefully organized representation of data in an abstract visual language.

List of Publications

[Under Review] **M. A. Yalçın**, N. Elmqvist, and B. B. Bederson, "Evaluating Multi-Column Bar Charts and Treemaps for Dense Visualization of Sorted Numeric Data", *IEEE Transactions on Visualization and Computer Graphics*

[Under Review] **M. A. Yalçın**, N. Elmqvist, and B. B. Bederson, "Keshif: Rapid and Flexible Data Exploration using Aggregate Summaries and Linked Selections", *IEEE Transactions on Visualization and Computer Graphics*

M. A. Yalçın, N. Elmqvist, and B. B. Bederson, "AggreSet: Rich and Scalable Set Exploration using Visualizations of Element Aggregations," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 688–697, Jan. 2016. [154]

M. A. Yalçın, N. Elmqvist, and B. B. Bederson, "Cognitive Stages in Visual Data Exploration," in *Proceedings of the BELIV Workshop: Beyond Time and Errors - Novel Evaluation Methods for Visualization*, New York, NY, USA, 2016. [153]

M. A. Yalçın, N. Elmqvist, and B. B. Bederson, "Keshif: Out-of-the-Box Visual and Interactive Data Exploration Environment," in *Proc. of IEEE VIS 2016 Workshop on Visualization in Practice: Open Source Visualization and Visual Analytics Software*, 2016. [155]

M. A. Yalçın and C. Plaisant, "Information Visualization," in *Big Data and Social Sciences*, Chapman and Hall/CRC, 2016. [156]

C. Y. Ip, **M. A. Yalçın**, D. Luebke, and A. Varshney, "PixelPie: Maximal Poisson-disk Sampling with Rasterization," in *Proceedings of the 5th High-Performance Graphics Conference*, New York, NY, USA, 2013, pp. 17–26. [69]

M. A. Yalçın, K. Weiss, and L. De Floriani, "GPU Algorithms for Diamond-based Multiresolution Terrain Processing," in *Proceedings of the 11th Eurographics Conference on Parallel Graphics and Visualization*, Aire-la-Ville, Switzerland, Switzerland, 2011, pp. 121–130. [152]

M. Stroila, **M. A. Yalçın**, J. Mays, and N. Alwar, "Route Visualization in Indoor Panoramic Imagery with Open Area Maps," in *2012 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, 2012, pp. 499–504. [134]

Software Packages

Keshif

Website:	www.keshif.me
Source code:	http://github.com/adilyalcin/keshif
License:	BSD License 3-clause
Technology:	JavaScript, HTML, CSS, D3
Lines of Code:	~12k JavaScript, ~4k LESS (CSS preprocessor)
Maillist:	http://groups.google.com/forum/#!forum/keshif
Twitter:	http://twitter.com/keshifme
Facebook:	http://facebook.com/keshifme

Chubuk

Website:	http://adilyalcin.me/chubuk.js
Source code:	http://github.com/adilyalcin/chubuk.js
Experiments:	http://adilyalcin.me/chubuk.exp
License:	BSD License 3-clause
Technology:	JavaScript, HTML, CSS, D3
Lines of Code:	~750 JavaScript, ~500 LESS (CSS preprocessor).

Bibliography

[1] C. Ahlberg and B. Shneiderman, “Visual Information Seeking: Tight Coupling of Dynamic Query Filters with Starfield Displays,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York, NY, USA, 1994, pp. 313–317.

[2] B. Alper, N. H. Riche, G. Ramos, and M. Czerwinski, “Design Study of LineSets, a Novel Set Visualization Technique,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2259–2267, Dec. 2011.

[3] B. Alsallakh, W. Aigner, S. Miksch, and H. Hauser, “Radial Sets: Interactive Visual Analysis of Large Overlapping Sets,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2496–2505, Dec. 2013.

[4] B. Alsallakh, L. Micallef, W. Aigner, H. Hauser, S. Miksch, and P. Rodgers, “Visualizing Sets and Set-typed Data: State-of-the-Art and Future Challenges,” presented at the Eurographics Conference on Visualization (EuroVis), Swansea, Wales, UK, 2014, pp. 1–21.

[5] R. A. Amar and J. T. Stasko, “Knowledge precepts for design and evaluation of information visualizations,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 11, no. 4, pp. 432–442, Jul. 2005.

[6] E. W. Anderson, K. C. Potter, L. E. Matzen, J. F. Shepherd, G. A. Preston, and C. T. Silva, “A User Study of Visualization Effectiveness Using EEG and Cognitive Load,” *Computer Graphics Forum*, vol. 30, no. 3, pp. 791–800, 2011.

[7] L. B. Anderson, A. D. Wolvin, R. Kirby-Straker, M. A. Yalçın, and B. B. Bederson, “Incorporating learning analytics into basic course administration: How to embrace the opportunity to identify inconsistencies and inform responses,” in *Proceedings of the of the National Communication Association*, Las Vegas, NV, 2015.

[8] O. D. Andrade, N. Bean, and D. G. Novick, “The Macro-structure of Use of Help,” in *Proceedings of the 27th ACM International Conference on Design of Communication*, New York, NY, USA, 2009, pp. 143–150.

[9] R. Arias-Hernandez, L. T. Kaastra, T. M. Green, and B. Fisher, “Pair Analytics: Capturing Reasoning Processes in Collaborative Visual Analytics,” in *44th Hawaii International Conference on System Sciences (HICSS)*, 2011, pp. 1–10.

[10] R. Baecker, “Showing Instead of Telling,” in *Proceedings of the 20th Annual International Conference on Computer Documentation*, New York, NY, USA, 2002, pp. 10–16.

[11] N. Banovic, T. Grossman, J. Matejka, and G. Fitzmaurice, “Waken: Reverse Engineering Usage Information and Interface Structure from Software Videos,” in *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*, New York, NY, USA, 2012, pp. 83–92.

[12] F. Beck, S. Koch, and D. Weiskopf, “Visual Analysis and Dissemination of Scientific Literature Collections with SurVis,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 180–189, Jan. 2016.

[13] B. B. Bederson, “Interfaces for Staying in the Flow,” *Ubiquity*, no. September, pp. 1–1, 2004.

[14] B. B. Bederson, “The Promise of Zoomable User Interfaces,” in *Proceedings of the 3rd International Symposium on Visual Information Communication*, New York, NY, USA, 2010, p. 2:1–2:1.

[15] J. Bertin, *Graphics and Graphic Information Processing*. Walter de Gruyter, 1981.

[16] 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*, New York, NY, USA, 2014, pp. 17–24.

[17] Bloomberg Visual Data, “Which companies are biggest?” [Online]. Available: http://www.bloomberg.com/visual-data/industries/q/biggest-companies. [Accessed: 22-Sep-2015].

[18] M. Bostock, V. Ogievetsky, and J. Heer, “D3: Data-driven documents,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2301–2309, Dec. 2011.

[19] N. Boukhelifa, J. C. Roberts, and P. J. Rodgers, “A coordination model for exploratory multiview visualization,” in *International Conference on Coordinated and Multiple Views in Exploratory Visualization, 2003. Proceedings*, 2003, pp. 76–85.

[20] J. Boy, L. Eveillard, F. Detienne, and J. Fekete, “Suggested Interactivity: Seeking Perceived Affordances for Information Visualization,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 639–648, Jan. 2016.

[21] J. Boy, R. A. Rensink, E. Bertini, and J.-D. Fekete, “A Principled Way of Assessing Visualization Literacy,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 1963–1972, Dec. 2014.

[22] M. Brehmer, J. Ng, K. Tate, and T. Munzner, “Matches, Mismatches, and Methods: Multiple-View Workflows for Energy Portfolio Analysis,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 449–458, Jan. 2016.

[23] M. Bruls, K. Huizing, and J. J. van Wijk, “Squarified Treemaps,” in *Data Visualization 2000*, D. ir W. C. de Leeuw and ir R. van Liere, Eds. Springer Vienna, 2000, pp. 33–42.

[24] B. Buxton, “The Long Nose of Innovation,” *Bloomberg.com*, 02-Jan-2008.

[25] D. E. Caldwell and M. White, “CogentHelp: A Tool for Authoring Dynamically Generated Help for Java GUIs,” in *Proceedings of the 15th Annual International Conference on Computer Documentation*, New York, NY, USA, 1997, pp. 17–22.

[26] S. K. Card, J. D. Mackinlay, and B. Shneiderman, “Readings in Information Visualization: Using Vision to Think,” S. K. Card, J. D. Mackinlay, and B. Shneiderman, Eds. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1999, pp. 579–581.

[27] J. M. Carroll, *The Nurnberg Funnel: Designing Minimalist Instruction for Practical Computer Skill*. Cambridge, MA, USA: MIT Press, 1990.

[28] J. M. Carroll, *Minimalism Beyond the Nurnberg Funnel*. MIT Press, 1998.

[29] J. M. Carroll and C. Carrithers, “Training Wheels in a User Interface,” *Commun. ACM*, vol. 27, no. 8, pp. 800–806, Aug. 1984.

[30] P. Chandler and J. Sweller, “Cognitive Load Theory and the Format of Instruction,” *Cognition and Instruction*, vol. 8, no. 4, pp. 293–332, Dec. 1991.

[31] E. H.-H. Chi and J. T. Riedl, “An operator interaction framework for visualization systems,” in *IEEE Symposium on Information Visualization*, 1998, pp. 63–70.

[32] W. S. Cleveland and R. McGill, “Graphical Perception: Theory, Experimentation, and Application to the Development of Graphical Methods,” *Journal of the American Statistical Association*, vol. 79, no. 387, pp. 531–554, Sep. 1984.

[33] C. Collins, G. Penn, and S. Carpendale, “Bubble Sets: Revealing Set Relations with Isocontours over Existing Visualizations,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 6, pp. 1009–1016, Nov. 2009.

[34] M. Correll, D. Albers, S. Franconeri, and M. Gleicher, “Comparing Averages in Time Series Data,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York, NY, USA, 2012, pp. 1095–1104.

[35] A. Crahen, “Human Trafficking #MakeoverMonday.” [Online]. Available: http://tabsoft.co/1NK9GoY. [Accessed: 28-Apr-2016].

[36] G. Cumming, “The New Statistics Why and How,” *Psychological Science*, vol. 25, no. 1, pp. 7–29, Jan. 2014.

[37] T. J. DiCiccio and B. Efron, “Bootstrap confidence intervals,” *Statistical science*, pp. 189–212, 1996.

[38] K. Dinkla, M. J. van Kreveld, B. Speckmann, and M. A. Westenberg, “Kelp Diagrams: Point Set Membership Visualization,” *Computer Graphics Forum*, vol. 31, no. 3pt1, pp. 875–884, 2012.

[39] M. Dork, S. Carpendale, C. Collins, and C. Williamson, “VisGets: Coordinated Visualizations for Web-based Information Exploration and Discovery,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 6, pp. 1205–1212, Nov. 2008.

[40] W. Dou, C. Ziemkiewicz, L. Harrison, D. H. Jeong, W. Ribarsky, X. Wang, and R. Chang, “Toward a Deeper Understanding of the Relationship Between Interaction Constraints and Visual Isomorphs,” *Information Visualization*, vol. 11, no. 3, pp. 222–236, Jul. 2012.

[41] M. Ekstrand, W. Li, T. Grossman, J. Matejka, and G. Fitzmaurice, “Searching for Software Learning Resources Using Application Context,” in *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, New York, NY, USA, 2011, pp. 195–204.

[42] N. Elmqvist and J.-D. Fekete, “Hierarchical Aggregation for Information Visualization: Overview, Techniques, and Design Guidelines,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 3, pp. 439–454, May 2010.

[43] N. Elmqvist, A. V. Moere, H.-C. Jetter, D. Cernea, H. Reiterer, and T. J. Jankun-Kelly, “Fluid Interaction for Information Visualization,” *Information Visualization*, vol. 10, no. 4, pp. 327–340, Oct. 2011.

[44] K. A. Ericsson and H. A. Simon, “Verbal reports as data.,” *Psychological review*, vol. 87, no. 3, p. 215, 1980.

[45] J.-D. Fekete and C. Plaisant, “Interactive information visualization of a million items,” in *IEEE Symposium on Information Visualization, 2002. INFOVIS 2002*, 2002, pp. 117–124.

[46] S. Few, “Time on the Horizon,” *Visual Business Intelligence Newsletter*, 2008.

[47] S. Few, “Wrapping Graphs to Extend Their Limits,” *Visual Business Intelligence Newsletter*, 2013.

[48] J. Fuchs, F. Fischer, F. Mansmann, E. Bertini, and P. Isenberg, “Evaluation of Alternative Glyph Designs for Time Series Data in a Small Multiple Setting,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York, NY, USA, 2013, pp. 3237–3246.

[49] R. J. Gerrig, P. G. Zimbardo, A. J. Campbell, S. R. Cumming, and F. J. Wilkes, *Psychology and life*. Pearson Higher Education AU, 2011.

[50] M. Ghoniem, J.-D. Fekete, and P. Castagliola, “A Comparison of the Readability of Graphs Using Node-Link and Matrix-Based Representations,” in *Proceedings of the IEEE Symposium on Information Visualization*, 2004, pp. 17–24.

[51] P. Ginns, “Integrating information: A meta-analysis of the spatial contiguity and temporal contiguity effects,” *Learning and Instruction*, vol. 16, no. 6, pp. 511–525, Dec. 2006.

[52] M. Gleicher, M. Correll, C. Nothelfer, and S. Franconeri, “Perception of Average Value in Multiclass Scatterplots,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2316–2325, Dec. 2013.

[53] M. Golemati, C. Halatsis, C. Vassilakis, A. Katifori, and G. Lepouras, “A Context-Based Adaptive Visualization Environment,” in *Tenth International Conference on Information Visualization*, 2006, pp. 62–67.

[54] L. Grammel, M. Tory, and M.-A. Storey, “How Information Visualization Novices Construct Visualizations,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 6, pp. 943–952, Nov. 2010.

[55] T. M. Green, W. Ribarsky, and B. Fisher, “Visual analytics for complex concepts using a human cognition model,” in *IEEE Symposium on Visual Analytics Science and Technology*, 2008, pp. 91–98.

[56] S. Greene, E. Tanin, C. Plaisant, B. Shneiderman, L. Olsen, G. Major, and S. Johns, “The end of zero-hit queries: query previews for NASA’s Global Change Master Directory,” *International Journal on Digital Libraries*, vol. 2, no. 2–3, pp. 79–90, Sep. 1999.

[57] T. Grossman, G. Fitzmaurice, and R. Attar, “A Survey of Software Learnability: Metrics, Methodologies and Guidelines,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York, NY, USA, 2009, pp. 649–658.

[58] H. Guo, S. Gomez, C. Ziemkiewicz, and D. Laidlaw, “A Case Study Using Visualization Interaction Logs and Insight,” *IEEE Transactions on Visualization and Computer Graphics*, vol. PP, no. 99, pp. 1–1, 2015.

[59] L. Harrison, D. Skau, S. Franconeri, A. Lu, and R. Chang, “Influencing Visual Judgment Through Affective Priming,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York, NY, USA, 2013, pp. 2949–2958.

[60] J. Heer and M. Agrawala, “Software Design Patterns for Information Visualization,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 5, pp. 853–860, Sep. 2006.

[61] J. Heer and M. Bostock, “Crowdsourcing graphical perception: using mechanical turk to assess visualization design,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York, NY, USA, 2010, pp. 203–212.

[62] J. Heer, F. van Ham, S. Carpendale, C. Weaver, and P. Isenberg, “Creation and Collaboration: Engaging New Audiences for Information Visualization,” in *Information Visualization*, A. Kerren, J. T. Stasko, J.-D. Fekete, and C. North, Eds. Springer Berlin Heidelberg, 2008, pp. 92–133.

[63] J. Heer, N. Kong, and M. Agrawala, “Sizing the Horizon: The Effects of Chart Size and Layering on the Graphical Perception of Time Series Visualizations,”

in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York, NY, USA, 2009, pp. 1303–1312.

[64] J. Heer, J. Mackinlay, C. Stolte, and M. Agrawala, “Graphical Histories for Visualization: Supporting Analysis, Communication, and Evaluation,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 6, pp. 1189–1196, Nov. 2008.

[65] J. Heer and G. Robertson, “Animated Transitions in Statistical Data Graphics,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1240–1247, Nov. 2007.

[66] H. Hofmann, L. Follett, M. Majumder, and D. Cook, “Graphical Tests for Power Comparison of Competing Designs,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 12, pp. 2441–2448, Dec. 2012.

[67] E. Horvitz, “Principles of Mixed-initiative User Interfaces,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York, NY, USA, 1999, pp. 159–166.

[68] D. F. Huynh, D. R. Karger, and R. C. Miller, “Exhibit: Lightweight Structured Data Publishing,” in *Proceedings of the ACM Conference on World Wide Web*, New York, NY, USA, 2007, pp. 737–746.

[69] C. Y. Ip, M. A. Yalçin, D. Luebke, and A. Varshney, “PixelPie: Maximal Poisson-disk Sampling with Rasterization,” in *Proceedings of the 5th High-Performance Graphics Conference*, New York, NY, USA, 2013, pp. 17–26.

[70] W. Javed, B. McDonnel, and N. Elmqvist, “Graphical Perception of Multiple Time Series,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 6, pp. 927–934, Nov. 2010.

[71] B. Johnson and B. Shneiderman, “Tree-maps: a space-filling approach to the visualization of hierarchical information structures,” in *IEEE Conference on Visualization, 1991. Visualization ’91, Proceedings*, 1991, pp. 284–291.

[72] D. Kahneman, *Thinking, Fast and Slow*, Reprint edition. New York: Farrar, Straus and Giroux, 2013.

[73] H. Kang, C. Plaisant, and B. Shneiderman, “New Approaches to Help Users Get Started with Visual Interfaces: Multi-layered Interfaces and Integrated Initial Guidance,” in *Proceedings of the 2003 Annual National Conference on Digital Government Research*, Boston, MA, USA, 2003, pp. 1–6.

[74] D. R. Karger, “The Semantic Web and End Users: What’s Wrong and How to Fix It,” *IEEE Internet Computing*, vol. 18, no. 6, pp. 64–70, Nov. 2014.

[75] C. Kelleher and R. Pausch, “Stencils-based Tutorials: Design and Evaluation,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York, NY, USA, 2005, pp. 541–550.

[76] B. Kim, B. Lee, and J. Seo, “Visualizing Set Concordance with Permutation Matrices and Fan Diagrams,” *Interact. Comput.*, vol. 19, no. 5–6, pp. 630–643, Dec. 2007.

[77] J. Kim, P. T. Nguyen, S. Weir, P. J. Guo, R. C. Miller, and K. Z. Gajos, “Crowdsourcing Step-by-step Information Extraction to Enhance Existing How-to Videos,” in *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems*, New York, NY, USA, 2014, pp. 4017–4026.

[78] D. Kirsh, “Interaction, External Representation and Sense Making,” *Proceedings of the 31st Annual Conference of the Cognitive Science Society*, pp. 1103–1108, 2009.

[79] G. Klein, B. Moon, and R. R. Hoffman, “Making Sense of Sensemaking 2: A Macrocognitive Model,” *IEEE Intelligent Systems*, vol. 21, no. 5, pp. 88–92, Sep. 2006.

[80] A. Kobsa, “An empirical comparison of three commercial information visualization systems,” in *IEEE Symposium on Information Visualization*, 2001, pp. 123–130.

[81] N. Kong, J. Heer, and M. Agrawala, “Perceptual Guidelines for Creating Rectangular Treemaps,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 6, pp. 990–998, Nov. 2010.

[82] B. C. Kwon, S. H. Kim, S. Lee, J. Choo, J. Huh, and J. S. Yi, “VisOHC: Designing Visual Analytics for Online Health Communities,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 71–80, Jan. 2016.

[83] B. C. Kwon and B. Lee, “A Comparative Evaluation on Online Learning Approaches Using Parallel Coordinate Visualization,” in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, New York, NY, USA, 2016, pp. 993–997.

[84] B. chul Kwon, B. Fisher, and J. S. Yi, “Visual analytic roadblocks for novice investigators,” in *IEEE Conference on Visual Analytics Science and Technology (VAST)*, 2011, pp. 3–11.

[85] H. Lam, “A Framework of Interaction Costs in Information Visualization,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 6, pp. 1149–1156, Nov. 2008.

[86] S. Lee, S.-H. Kim, Y.-H. Hung, H. Lam, Y. Kang, and J. S. Yi, “How do People Make Sense of Unfamiliar Visualizations?: A Grounded Model of Novice ’s Information Visualization Sensemaking,” *IEEE Transactions on Visualization and Computer Graphics*, vol. PP, no. 99, pp. 1–1, 2015.

[87] A. Lex and N. Gehlenborg, “Points of view: Sets and intersections,” *Nature Methods*, vol. 11, no. 8, pp. 779–779, Aug. 2014.

[88] A. Lex, N. Gehlenborg, H. Strobelt, R. Vuillemot, and H. Pfister, “UpSet: Visualization of Intersecting Sets,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 1983–1992, Dec. 2014.

[89] I. Liiv, “Seriation and matrix reordering methods: An historical overview,” *Statistical Analysis and Data Mining*, vol. 3, no. 2, pp. 70–91, Apr. 2010.

[90] Z. Liu and J. Heer, “The Effects of Interactive Latency on Exploratory Visual Analysis,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 2122–2131, Dec. 2014.

[91] Z. Liu, B. Jiang, and J. Heer, “imMens: Real-time Visual Querying of Big Data,” in *Proceedings of the 15th Eurographics Conference on Visualization*, Aire-la-Ville, Switzerland, Switzerland, 2013, pp. 421–430.

[92] Z. Liu, N. Nersessian, and J. Stasko, “Distributed Cognition as a Theoretical Framework for Information Visualization,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 6, pp. 1173–1180, 2008.

[93] Z. Liu, J. Stasko, and T. Sullivan, “SellTrend: Inter-Attribute Visual Analysis of Temporal Transaction Data,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 6, pp. 1025–1032, Nov. 2009.

[94] Z. Liu and J. T. Stasko, “Mental Models, Visual Reasoning and Interaction in Information Visualization: A Top-down Perspective,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 6, pp. 999–1008, Nov. 2010.

[95] J. Mackinlay, P. Hanrahan, and C. Stolte, “Show Me: Automatic Presentation for Visual Analysis,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1137–1144, Nov. 2007.

[96] K. Madhavan, N. Elmqvist, M. Vorvoreanu, X. Chen, Y. Wong, H. Xian, Z. Dong, and A. Johri, “DIA2: Web-based Cyberinfrastructure for Visual Analysis of Funding Portfolios,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 1823–1832, Dec. 2014.

[97] J. Maeda, *The Laws of Simplicity*. MIT Press, 2006.

[98] J. Matejka, T. Grossman, and G. Fitzmaurice, “Ambient Help,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York, NY, USA, 2011, pp. 2751–2760.

[99] W. Meulemans, N. H. Riche, B. Speckmann, B. Alper, and T. Dwyer, “KelpFusion: A Hybrid Set Visualization Technique,” *IEEE Transactions on*

Visualization and Computer Graphics, vol. 19, no. 11, pp. 1846–1858, Nov. 2013.

[100] Microsoft, “Farewell Clippy: What’s Happening to the Infamous Office Assistant in Office XP.” .

[101] T. Munzner, “A Nested Model for Visualization Design and Validation,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 6, pp. 921–928, Nov. 2009.

[102] B. A. Myers, D. A. Weitzman, A. J. Ko, and D. H. Chau, “Answering Why and Why Not Questions in User Interfaces,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York, NY, USA, 2006, pp. 397–406.

[103] C. Nguyen and F. Liu, “Making Software Tutorial Video Responsive,” in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, New York, NY, USA, 2015, pp. 1565–1568.

[104] D. A. Norman, *The Design of Everyday Things*. Basic books, 2002.

[105] C. North and B. Shneiderman, “Snap-together visualization: a user interface for coordinating visualizations via relational schemata,” in *Proceedings of the working conference on Advanced visual interfaces*, New York, NY, USA, 2000, pp. 128–135.

[106] D. G. Novick and K. Ward, “Why Don’t People Read the Manual?,” in *Proceedings of the 24th Annual ACM International Conference on Design of Communication*, New York, NY, USA, 2006, pp. 11–18.

[107] H. L. O’Brien and E. G. Toms, “The development and evaluation of a survey to measure user engagement,” *Journal of the American Society for Information Science and Technology*, vol. 61, no. 1, pp. 50–69, Jan. 2010.

[108] V. Papanek, *Design for the Real World: Human Ecology and Social Change*, 2 Revised edition. Chicago, Ill: Chicago Review Press, 2005.

[109] S. N. Pattanaik, J. A. Ferwerda, M. D. Fairchild, and D. P. Greenberg, “A Multiscale Model of Adaptation and Spatial Vision for Realistic Image Display,” in *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, New York, NY, USA, 1998, pp. 287–298.

[110] A. Perer and B. Shneiderman, “Systematic Yet Flexible Discovery: Guiding Domain Experts Through Exploratory Data Analysis,” in *Proceedings of the 13th International Conference on Intelligent User Interfaces*, New York, NY, USA, 2008, pp. 109–118.

[111] P. Pirolli and S. Card, “Information foraging,” *Psychological Review*, vol. 106, no. 4, pp. 643–675, 1999.

[112] C. Plaisant and B. Shneiderman, “Show Me! Guidelines for producing recorded demonstrations,” in *2005 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC’05)*, 2005, pp. 171–178.

[113] S. Pongnumkul, M. Dontcheva, W. Li, J. Wang, L. Bourdev, S. Avidan, and M. F. Cohen, “Pause-and-play: Automatically Linking Screencast Video Tutorials with Applications,” in *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, New York, NY, USA, 2011, pp. 135–144.

[114] K. Reda, A. E. Johnson, M. E. Papka, and J. Leigh, “Effects of Display Size and Resolution on User Behavior and Insight Acquisition in Visual Exploration,” in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, New York, NY, USA, 2015, pp. 2759–2768.

[115] D. Ren, T. Hollerer, and X. Yuan, “iVisDesigner: Expressive Interactive Design of Information Visualizations,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 2092–2101, Dec. 2014.

[116] N. H. Riche and T. Dwyer, “Untangling Euler Diagrams,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 6, pp. 1090–1099, 2010.

[117] N. H. Riche, B. Lee, and C. Plaisant, “Understanding Interactive Legends: a Comparative Evaluation with Standard Widgets,” *Computer Graphics Forum*, vol. 29, no. 3, pp. 1193–1202, Jun. 2010.

[118] J. C. Roberts, “State of the Art: Coordinated Multiple Views in Exploratory Visualization,” in *Fifth International Conference on Coordinated and Multiple Views in Exploratory Visualization, 2007. CMV ’07*, 2007, pp. 61–71.

[119] P. Rodgers, “A survey of Euler diagrams,” *Journal of Visual Languages & Computing*, vol. 25, no. 3, pp. 134–155, Jun. 2014.

[120] S. F. Roth, J. Kolojejchick, J. Mattis, and J. Goldstein, “Interactive Graphic Design Using Automatic Presentation Knowledge,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York, NY, USA, 1994, pp. 112–117.

[121] R. Sadana, T. Major, A. Dove, and J. Stasko, “OnSet: A Visualization Technique for Large-scale Binary Set Data,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 1993–2002, Dec. 2014.

[122] P. Saraiya, C. North, and K. Duca, “An insight-based methodology for evaluating bioinformatics visualizations,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 11, no. 4, pp. 443–456, Jul. 2005.

[123] A. Satyanarayan and J. Heer, “Lyra: An Interactive Visualization Design Environment,” *Computer Graphics Forum*, vol. 33, no. 3, pp. 351–360, Jun. 2014.

[124] R. K. Sawyer, Ed., *The Cambridge Handbook of the Learning Sciences*, 1 edition. Cambridge ; New York: Cambridge University Press, 2006.

[125] B. Schwartz and K. Kliban, *The paradox of choice: Why more is less*. Ecco New York, 2004.

[126] V. Setlur and M. C. Stone, “A Linguistic Approach to Categorical Color Assignment for Data Visualization,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 698–707, Jan. 2016.

[127] K. Sherwin, “Pop-ups and Adaptive Help Get A Refresh.” [Online]. Available: https://www.nngroup.com/articles/pop-up-adaptive-help/. [Accessed: 24-Aug-2016].

[128] B. Shneiderman, “Dynamic queries for visual information seeking,” *IEEE Software*, vol. 11, no. 6, pp. 70–77, Nov. 1994.

[129] B. Shneiderman, “The eyes have it: a task by data type taxonomy for information visualizations,” in *IEEE Symposium on Visual Languages*, 1996, pp. 336–343.

[130] B. Shneiderman, “Direct Manipulation for Comprehensible, Predictable and Controllable User Interfaces,” in *Proceedings of the 2Nd International Conference on Intelligent User Interfaces*, New York, NY, USA, 1997, pp. 33–39.

[131] B. Shneiderman, C. Plaisant, M. Cohen, S. Jacobs, N. Elmqvist, and N. Diakopoulos, “Documentation and User Support (a.k.a. Help),” in *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, 6 edition., Boston: Pearson, 2016, pp. 446–475.

[132] Y. B. Shrinivasan and J. J. van Wijk, “Supporting the Analytical Reasoning Process in Information Visualization,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York, NY, USA, 2008, pp. 1237–1246.

[133] E. Sinar, T. Poeppelman, B. Armstrong, N. Blacksmith, and J. Thornton, “SIOP Technology Launch: A New Way to Explore our Conference Programs,” *SIOP News*, 17-Aug-2016. [Online]. Available: http://www.siop.org/article_view.aspx?article=1562. [Accessed: 05-Oct-2016].

[134] D. Skau, L. Harrison, and R. Kosara, “An Evaluation of the Impact of Visual Embellishments in Bar Charts,” *Computer Graphics Forum*, vol. 34, no. 3, pp. 221–230, Jun. 2015.

[135] B. Steichen, C. Conati, and G. Carenini, “Inferring Visualization Task Properties, User Performance, and User Cognitive Abilities from Eye Gaze Data,” *ACM Trans. Interact. Intell. Syst.*, vol. 4, no. 2, p. 11:1–11:29, Jul. 2014.

[136] C. Stolte and P. Hanrahan, “Polaris: A System for Query, Analysis and Visualization of Multi-Dimensional Relational Databases,” in *IEEE Symposium on Information Vizualization*, Washington, DC, USA, 2000, p. 5–.

[137] M. Stroila, M. A. Yalçın, J. Mays, and N. Alwar, “Route Visualization in Indoor Panoramic Imagery with Open Area Maps,” in *2012 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, 2012, pp. 499–504.

[138] J. Talbot, V. Setlur, and A. Anand, “Four Experiments on the Perception of Bar Charts,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 2152–2160, Dec. 2014.

[139] J. J. Thomas and K. A. Cook, Eds., *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. Los Alamitos, CA: National Visualization and Analytics Ctr, 2005.

[140] M. Tory and T. Moller, “Evaluating visualizations: do expert reviews work?,” *IEEE Computer Graphics and Applications*, vol. 25, no. 5, pp. 8–11, Sep. 2005.

[141] E. Tufte, *The Visual Display of Quantitative Information*, 2nd edition. Cheshire, Conn: Graphics Pr, 1983.

[142] L. Tweedie, B. Spence, D. Williams, and R. Bhogal, “The Attribute Explorer,” in *Conference Companion on Human Factors in Computing Systems*, New York, NY, USA, 1994, pp. 435–436.

[143] J. Underwood, “Data Visualization Best Practices 2013,” 2013. [Online]. Available: http://www.slideshare.net/idigdata/data-visualization-best-practices-2013. [Accessed: 23-Apr-2016].

[144] 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*, vol. 13, no. 6, pp. 1121–1128, Nov. 2007.

[145] J. Walny, S. Huron, and S. Carpendale, “An Exploratory Study of Data Sketching for Visual Representation,” *Computer Graphics Forum*, vol. 34, no. 3, pp. 231–240, Jun. 2015.

[146] C. Weaver, “Building Highly-Coordinated Visualizations in Improvise,” in *IEEE Symposium on Information Visualization, 2004. INFOVIS 2004*, 2004, pp. 159–166.

[147] A. Weiss, *Million Dollar Consulting*, 1 edition. McGraw-Hill Education, 2009.

[148] S. Wexler, “In Praise of Treemaps,” *In Praise of Treemaps*, 01-Sep-2015. [Online]. Available: http://www.datarevelations.com/in-praise-of-treemaps.html. [Accessed: 22-Jul-2016].

[149] J. J. van Wijk, “The value of visualization,” in *IEEE Visualization*, 2005, pp. 79–86.

[150] W. Willett, J. Heer, and M. Agrawala, “Scented Widgets: Improving Navigation Cues with Embedded Visualizations,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1129–1136, Nov. 2007.

[151] 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*, vol. PP, no. 99, pp. 1–1, 2015.

[152] M. A. Yalçin, K. Weiss, and L. De Floriani, “GPU Algorithms for Diamond-based Multiresolution Terrain Processing,” in *Proceedings of the 11th Eurographics Conference on Parallel Graphics and Visualization*, Aire-la-Ville, Switzerland, Switzerland, 2011, pp. 121–130.

[153] M. A. Yalçın, N. Elmqvist, and B. B. Bederson, “Cognitive Stages in Visual Data Exploration,” in *Proceedings of the BELIV Workshop: Beyond Time and Errors - Novel Evaluation Methods for Visualization*, New York, NY, USA, 2016.

[154] M. A. Yalçın, N. Elmqvist, and B. B. Bederson, “AggreSet: Rich and Scalable Set Exploration using Visualizations of Element Aggregations,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 688–697, Jan. 2016.

[155] M. A. Yalçın, N. Elmqvist, and B. B. Bederson, “Keshif: Out-of-the-Box Visual and Interactive Data Exploration Environmen,” in *Proc. of IEEE VIS 2016 Workshop on Visualization in Practice: Open Source Visualization and Visual Analytics Software*.

[156] M. A. Yalçın and C. Plaisant, “Information Visualization,” in *Big Data and Social Sciences*, Chapman and Hall/CRC, 2016.

[157] K.-P. Yee, K. Swearingen, K. Li, and M. Hearst, “Faceted metadata for image search and browsing,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York, NY, USA, 2003, pp. 401–408.

[158] T. Yeh, T.-H. Chang, B. Xie, G. Walsh, I. Watkins, K. Wongsuphasawat, M. Huang, L. S. Davis, and B. B. Bederson, “Creating Contextual Help for GUIs Using Screenshots,” in *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, New York, NY, USA, 2011, pp. 145–154.

[159] “USPTO PatentsView.” [Online]. Available: https://www.cssipdata.org/patentsview/#visual. [Accessed: 22-Sep-2015].

[160] “Chardin.js.” [Online]. Available: https://heelhook.github.io/chardin.js/. [Accessed: 20-Sep-2016].

[161] “Natural Language Generation | Tableau - Automated Insights, Inc.” [Online]. Available: https://automatedinsights.com/tableau. [Accessed: 20-Sep-2016].

[162] “Narratives for Tableau | Free Narrative Extension for Tableau | Narrative Science.” [Online]. Available: https://www.narrativescience.com/tableau. [Accessed: 20-Sep-2016].

[163] “The Design Ethos of Dieter Rams,” *San Francisco Museum of Modern Art*. [Online]. Available: http://www.sfmoma.org/about/press/press_exhibitions/releases/880. [Accessed: 10-Feb-2015].

[164] “Keshif: Data Made Explorable.” [Online]. Available: http://keshif.me/. [Accessed: 20-Sep-2016].