Tool-making as an Intervention on the Accessibility of Interactive Data Experiences

Frank J. Elavsky

April 18, 2026

Human-Computer Interaction Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Dominik Moritz, Chair, Carnegie Mellon University HCII
Patrick Carrington, Co-Chair, Carnegie Mellon University HCII
Ken Holstein, Carnegie Mellon University HCII
Jennifer Mankoff, External, University of Washington
Tamara Munzner, External, University of British Columbia

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

This research has been partially or fully funded by the Human-Computer Interaction Institute, Apple Inc, Highsoft, Adobe, and the Chan-Zuckerberg Initiative.

Keywords

Visualization, accessibility, data interaction, tangible interfaces, tactile data representations, sonification, graph theory, human-computer interaction, personalization, disability, malleable interfaces, guidelines, standards, heuristics, evaluation, human-studies, qualitative research, quantitative research, assistive technology, data science, software tools, hardware tools, input modalities, voice-controlled interfaces, gesture-based input, natural language system design, programming tools, sensory substitution, software debugging.

Abstract

This thesis demonstrates practical advancements in making interactive data experiences accessible through a suite of tools and frameworks. Central to this work is the reframing of accessibility as a process that not only addresses the functionality of data visualizations but also considers how the underlying tools and methodologies that are used to build interactive data experiences can produce barriers, enable new interactions, and improve work for people with disabilities. Rather than framing research in accessibility towards correcting perceived deficits in users, this research posits that tools and toolmakers have a responsibility to enable designers, developers, and end-users towards more accessible outcomes.

This thesis introduces *Chartability*, a heuristic framework that enables designers, developers, and auditors to systematically evaluate data visualizations and interfaces for a wide range of accessibility barriers, considering people with visual, motor, vestibular, neurological, and cognitive disabilities. Through *Chartability*, practitioners—especially those with limited accessibility expertise—gain confidence and clarity in assessing and improving their work.

Complementing this, the thesis presents *Data Navigator*, a dynamic system designed as building blocks, which can be used to construct accessible navigation structures such as lists, trees, and graphs from underlying data. By supporting various input modalities—screen readers, keyboards, speech, and gestures—*Data Navigator* enables practitioners to construct data exploration experiences for users who rely on non-visual interaction, making complex datasets navigable and interactive for a wide range of people with disabilities.

Further, the concept of *Softerware* is introduced to address the tension between standardized accessible design and the diverse needs of real users with disabilities. This approach focuses on system design principles and empirical research to inform tool-makers to build interfaces that can empower end users with disabilities to customize and adapt interfaces to suit their data interaction preferences and needs.

This thesis also presents a blind-centered data analysis hardware tool, the *Cross-feelter*. This research project introduces a tactile input device that allows blind users to efficiently perform spatial cross-filtering tasks by providing simultaneous tactile input and output feedback. Evaluations with braille displays show that the device significantly improves interaction speed and exploratory query generation for blind users working with data.

(Proposed work) Finally, the work culminates in *Skeleton*, a development and de-bugging tool built on top of *Data Navigator*. *Skeleton* visually represents non-visual navigation and interaction experiences, which assists sighted designers in rapidly prototyping and fixing custom screen reader experiences for diagrams, maps, and bespoke visualizations. By enabling visual inspection and iterative design, *Skeleton* not only speeds up the development process but also

serves as an educational resource for understanding non-visual data interactions.

Together, these contributions provide both theoretical insights and practical tools that bridge gaps in current accessibility practices, ultimately enabling practitioners to build richer, more inclusive data experiences for all users.

Contents

Chapter 1
Introduction

This thesis is a body of research situated within the existing research area focused on making interactive data representations more accessible for people with disabilities. Much of the work in this existing area is situated within the context of making interactive data *visualizations* accessible, particularly (but not exclusively) for people who are blind and low vision. My work, contributed here in this thesis, is focused on using *tools* as a specific intervention and sub-area of study for making interactive data *representations* accessible for people with disabilities, broadly speaking. ("Representations" here is an intentionally broader term than "visualizations," which are exclusively visual representations of data.)

Before we begin, two things must be understood up front, or else the rest of this thesis could be interpreted with disruptive assumptions: we must interrogate the phrase "making visualizations accessible" and unpack why *tools* are a meaningful area of study.

1.1 Is "accessible visualization" really an oxymoron?

The first assumption that must be disrupted is perhaps the motivating cornerstone of this research, which is that the phrase "making visualizations accessible," while a noble goal, is not the semantically correct phrasing nor precisely what describes my work. This can be misleading. I do use the phrase "accessible visualization" but will admit that this seems to confuse certain people with very particular opinions about things. We will clear this up.

Villains of our field's past have written incendiary and ableist perspectives on why "no forms of data visualization, not just dashboards jam-packed with graphics, can be made fully accessible to someone who is blind," and that "[a blind man] will never be able to analyze data as I do visually, because many aspects of vision cannot be duplicated by his other senses" [73]. However, this position misunderstands what the goal of accessibility is, and arguably even what the goal of visualization itself is.

Making visualizations accessible *isn't* about the visualization, it's about making the outcomes of the visualization accessible.

Visualizations are ubiquitous and paramount for decision-making. However, the *artifact* that is a visualization is not even the goal of the act of visualizing: developing understanding, insight, confidence, and communication among and between human beings are the goals of visualization. Visualization is about making data easier to use for all kinds of things. Yes, our visual system enables us more than any other form of sensory cognition that we have [29, 85, 232]. But we aren't trying to make sight itself accessible. We are trying to make it possible for people to make meaningful decisions, gain valuable information, build conjectures, and effectively communicate with others.

Many, many people who I've spoken to over the course of my career, even before embarking on this thesis journey, misunderstand this simple fact: making a "visualization" accessible *isn't* about the visualization itself but rather making what the visualization is meant to *accomplish* accessible. It's about equal outcomes, not equal interactions with an artifact.

People with disabilities are no small portion of the world's population. In the United States, 27% of people self-report living with at least one disability that affects their daily lives [180] and all of us will eventually age into disability (if we are lucky to live a long life).

People with disabilities (again, that will be all of us *eventually*) deserve to participate fully in life. They deserve financial independence. They also deserve loving care and interdependence. People with disabilities have a right to make informed decisions, to know about the status of a global pandemic, and to have an understanding of local and national politics [71]. While we use visualizations to navigate all of these domains, the goal is not to make the charts and graphs themselves somehow equally useful to all people. That would be a false measurement of success.

Our goal then, is measured by the success of lives led by people with disabilities. Many other measurements are just metrics along the journey towards that goal. We then ask: Can people with disabilities also use data to live full lives? Can they make *fast* decisions based on data? Meaningful, careful, *slow* decisions? Communicate complex ideas? Crunch and clean data, develop models, find errors, and build hypotheses? Can they have memorable, immersive, beautiful, aesthetic experiences with data too [117]? Making "visualizations" accessible really is a misnomer. We are ultimately trying to make everything about what interactive data experiences *accomplish* for people equitable and accessible.

Again, if the goal of accessible visualization were about visualizations themselves, then the correct course of action would be one framed by the medical model [167]: that there is a normative state of behavior and capability (in this case, it would be "normal" to be able to read a visualization and make a decision) and any deviation from that norm must be corrected. This framing first assumes that the visualization should not be altered or improved. And then this framing puts the burden on the bodies of people with disabilities: that they must be "fixed" and given sight or brought to some equivalent state as someone who is "healthy," normal, and sighted. Plenty of scholars have already discussed why this framing is a problem, not only because it places undue burden on people with disabilities, produces pathologies and hierarchies of disability, but also because it is fundamentally not economically or ethically feasible.

So we then turn to other models of disability, such as the social model. The social model is heavily discussed by disability scholars and is not the end-game or last and total way of thinking about disability [167, 192, 195, 229, 272]. But the core motivation is that society, not medicine, is also a path towards solving problems that people with disabilities face. A few important concepts and concretely actionable things come from the social model that can help motivate the work of this thesis.

First, we look to the historical birth of the social model of disability: in the 504 sit-ins that took place in the United States in 1977. Cities had curbs and curbs are a barrier for people who use wheelchairs. So protests happened because decisions were being made without people with disabilities at the table. In this instance, people acknowledged that political power was an exclusive club and fought to ensure their cry "nothing about us, without us!" materialized.

And this leads us to the first and most-foundational philosophical framing for this thesis: that our *artifacts*, these things we've created from curbs to data visualizations, can become *barriers* for people with disabilities. And it is then the artifact, not the body of the person with a disability, where disability is produced in this model. Rather than a comparison to a normative state as a way to frame disability (the medical model), we instead must observe and evaluate material outcomes based on human-made problems.

So, the social model is framed around society "solving" inequities: we get involved and make political and legal change tangible. But a second model also emerges from within the social model: one where we can now frame *who is responsible* for repair: the curb designers and implementers.

And knowing who is responsible for access leads us into the moral and ethical imperative that motivates this thesis: the builders and makers of visualizations are ultimately the ones who provide exclusive value for only a subset of people: those *without* disabilities. **We must change how builders and makers do their work.**

So the phrase "accessible visualization" is really about recognizing that visualizations produce barriers for people. That means that it is our ethical imperative, as builders and makers, to fix them. And that act of fixing barriers leads us away from mere visual representations of data into a wide variety of other senses and interaction modalities. There are many paths forward towards fuller and more-equitable lives led by people with disabilities.

1.2 On *tools*, *tool-making*, and *human-tool* interaction

So the act of making becomes immensely important: we, the builders and makers of our world, need to get things right; there is a risk involved when making things that we will exclude people with disabilities. We need to make sure that we build a better world than the one we have now. We must care for new things we create and tend to the repair and maintenance of what we've already made. And this ethical imperative leads us to the topic of *tools* and *tool-making*.

So the second thing that must be understood before we embark on this thesis is that *tools* are not the same as *solutions* or *applications*. Sometimes tools can be used to *solve* things and are certainly, in ideal circumstances, *applied* in various contexts. But understanding the role of the "tool" in human-tool interaction is paramount for engaging in the work of making anything accessible for people with disabilities.

We use tools to shape our world, break old things, and make new things. But a tool, like the hammer (as an example), does not inherently *solve* something like homelessness. But a hammer can be used to build homes if there are social policies in place and proper resources allocated. This means that for the success of tooling, there is often a larger material, social, legal, and policy reality that supports and necessitates those tools. This thesis will not be focusing on changing the upstream dependencies, but optimistically operating as if they were true (or will be true in time).

However, in some cases, tools can *destroy*. The hammer has a claw and can easily pry apart boards and tear down homes. So tools carry potential to do all kinds of things, both good and bad, and how a tool is used is often open-ended, variable, and heavily dependent on socio-technical realities. Tools participate in personal and political agendas [259] and are sometimes, for this reason, regulated or made proprietary and controlled by powerful entities [90, 257].

So tools are not without any sort of ethics. We cannot just blame tool-users for outcomes when much of a tool depends on these larger systems and structures.Technologies (tools included) encode the assumptions and biases of their *creators* as much as, if not more than, their users. Tools that build things for others to use can be loaded with assumptions about what people are *able* to do [260] and also rules and guardrails about what anyone downstream from that tool's design *should* do [90, 256]. These assumptions, biases, and rules *limit*, *enforce*, *magnify*, *exclude*, and *enable* what a tool-user is capable of.

Tools for visualizing data are a perfect case study in this problem: virtually every major data visualization library, application, or software ever made was made entirely with the assumption that data should be transformed into visual representations. This is a reasonable assumption, since virtually all of the tool-makers are sighted and visualization is incredibly helpful to our cognition of and communication with data [79].

So data visualization, as a field, has focused its tool-making efforts on reducing the difficulty involved in visualizing data. Some visualization tools are concise [201], others are lower level but much more expressive [21]. Tool-making in visualization has focused on making it easier to scaffold a wide variety of interactions both with the visualizations as well as with their underlying data models [101].

However, as time has moved on, people began to speak out about color-vision deficiency in data visualization. Some people, primarily those with X/Y chromosomes (largely men) who are of European ancestry, have a deficiency in their ability to perceive certain colors. Then a plethora of research arose that began to look into the barriers that folks who are colorblind face in data visualization. As a result, our practices and tools improved. We began to educate practitioners, develop new color palettes, researched new methods for testing our designs, and built new systems for handling automatic color encoding. Our tools evolved.

But now data visualizations have arguably become ubiquitous in daily life. By comparison, we have far more tools now for making visualizations quickly and easily than we do for representing data in non-visual ways. We also have far more research, relatively speaking, into how sighted end users interact with visualizations.

So this thesis engages this gap: we need to educate practitioners on what accessibility barriers are in interactive visualizations, create building blocks that can assemble entirely new kinds of data interaction, develop software systems for end-user design and repair, explore new hardware that improves how people with disabilities can work with data, and design tools that enable practitioners to more easily identify and debug accessibility barriers as they work.

1.3 Overview

This thesis demonstrates practical advancements in making interactive data experiences accessible through a suite of tools and frameworks. Central to this work is the reframing of accessibility as a process that not only addresses the functionality of data visualizations but also considers how the underlying tools and methodologies that are used to build interactive data experiences can produce barriers, enable new interactions, and improve work for people with disabilities. Rather than framing research in accessibility towards correcting perceived deficits in users, this research posits that tools and toolmakers have a responsibility to enable designers, developers, and end-users towards more accessible outcomes.

This thesis introduces *Chartability*, a heuristic framework that enables designers, developers, and auditors to systematically evaluate data visualizations and interfaces for a wide range of accessibility barriers, considering people with visual, motor, vestibular, neurological, and cognitive disabilities. Through *Chartability*, practitioners—especially those with limited accessibility expertise—gain confidence and clarity in assessing and improving their work.

Complementing this, the thesis presents *Data Navigator*, a dynamic system designed as building blocks, which can be used to construct accessible navigation structures such as lists, trees, and graphs from underlying data. By supporting various input modalities—screen readers, keyboards, speech, and gestures—*Data Navigator* enables practitioners to construct data exploration experiences for users who rely on non-visual interaction, making complex datasets navigable and interactive for a wide range of people with disabilities.

Further, the concept of *Softerware* is introduced to address the tension between standardized accessible design and the diverse needs of real users with disabilities. This approach focuses on system design principles and empirical research to inform tool-makers to build interfaces that can empower end users with disabilities to customize and adapt interfaces to suit their data interaction preferences and needs.

This thesis also presents a blind-centered data analysis hardware tool, the *Cross-feelter*. This research project introduces a tactile input device that allows blind users to efficiently perform spatial cross-filtering tasks by providing simultaneous tactile input and output feedback. Evaluations with braille displays show that the device significantly improves interaction speed and exploratory query generation for blind users working with data.

This document culminates in a proposal for *Skeleton*, a development and de-bugging tool built on top of *Data Navigator*. *Skeleton* will be designed to visually represent non-visual navigation and interaction experiences, which we conjecture will assist sighted designers in rapidly prototyping and fixing custom screen reader experiences for diagrams, maps, and bespoke visualizations. We hypothesize that by enabling visual inspection and iterative design, *Skeleton* will not only speed up the development process but also serve as an educational resource for understanding non-visual data interactions.

In our proposal, we will outline our approach, evaluation process, timeline, and expected contribution of our project, *Skeleton*.

Together, these contributions provide both theoretical insights and practical tools that bridge gaps in current accessibility practices, ultimately enabling practitioners to build richer, more inclusive data experiences for all users.

Chapter 2
Background & Related Work

2.1 Practitioners and Tools

2.1.1 Understanding Builders, Makers, Designers, and Developers

Research investigating the practices and experiences of individuals who create with computers employs a range of high-level methods. Ethnographic studies, case studies, and design ethnographies are common approaches, allowing researchers to immerse themselves in communities such as the DIY/maker and assistive technology spaces [115, 120]. These methods capture the nuanced challenges practitioners face when engaging in new and unfamiliar work, including the transition from traditional to digital fabrication, coding, and tool creation [119]. By observing and interviewing practitioners in naturalistic settings, researchers uncover the social, cultural, and technical factors that shape how makers adapt and evolve their work practices.

Participatory design and co-creation are also central to this field [230]. These approaches encourage collaboration between researchers and practitioners or end-users, enabling a deeper understanding of the cognitive and creative processes behind design and development [94]. Such collaborative sessions reveal how designers shift their thinking when encountering novel challenges, embracing iterative processes that blend experimentation with reflection. Similarly, developers often modify their applications, tools, and even programming languages through feedback loops and community-driven innovation, highlighting a dynamic interplay between individual creativity and collective knowledge.

Additionally, design-based and case-study research methods explore how new practices can augment the existing work of practitioners [43, 126]. This involves not merely filling gaps or solving isolated problems but reimagining the possibilities for creative and technical expression. Researchers in this space envision systems that support continuous learning, adaptation, and innovation [92]. The focus is on enabling practitioners to extend their capabilities—providing scaffolds for experimentation, fostering environments where unconventional approaches are encouraged, and integrating new technologies in ways that amplify creativity and intelligence rather than simply addressing deficits [134, 237, 262].

Overall, the research methods used in this area are multidisciplinary, combining qualitative insights with iterative design practices to offer a holistic picture of the challenges and opportunities of builders, makers, designers, developers.

2.1.2 Approaches to Tool-making in Human-Computer Interaction

In human-computer interaction, tool-making research spans both the creation of entirely new capabilities and the enhancement of existing systems. One prominent approach involves piggybacking on current systems—leveraging their established functionalities to introduce improvements that streamline workflow or unlock new interactions [91]. This method often focuses on integrating with widely used platforms to amplify their usability, enabling users to perform tasks in more intuitive or efficient ways. By building on existing infrastructures, researchers can demonstrate how small, targeted modifications have the potential to transform user experiences.

Another significant approach centers on the notion of appropriation [54, 57, 196, 235]. Here, research examines how users adapt tools for uses beyond their original intent. Studies in this vein explore the creative processes behind such re-purposing, uncovering the latent functionalities and opportunities that emerge when practitioners modify systems to suit their unique needs. This perspective often leads to the development of modular, extensible tools that encourage experimentation and user customization, fostering a more personalized interaction with technology. In some cases, theory has been developed from the study of emergent and generative tool-use [14, 17], broadly informing future tooling projects as well as general theories of creative human interaction with technology.

Beyond these, tool-making in HCI also includes the development of systems designed to empower users by providing entirely new capabilities, sometimes explicitly named "toolkits" and other times generally just referred to for their ability to enable novel interaction and outcomes [142, 186, 197, 215, 216]. These projects may range from novel software environments that facilitate rapid prototyping and live programming to innovative hardware devices that bridge the gap between digital and physical interactions [104, 106, 186]. The emphasis is not solely on problem-solving but on enabling creative exploration, new possibilties, and even hacking the potential of technologies towards new ends [105]. Such projects often present their contributions through demonstrative prototypes and case studies that reveal potential applications, even if they are accompanied by minimal formal evaluations [64].

This body of work reflects a balance between novelty and practicality. While some projects aim to introduce groundbreaking new ways to interact with data and systems, others refine existing practices to improve efficiency and accessibility. Together, these approaches underscore a commitment to enhancing human capabilities, allowing users to not only solve problems more effectively but also to unlock new avenues for creativity and innovation.

2.2 Data, Accessibility, and Data *and* Accessibility

2.2.1 Advancements in Interactive Data Visualization and Data Science

Recent years have witnessed significant advancements in interactive data science and visualization, driven by innovations that enhance both the performance and usability of data tools. Cross-filtering, as a subtype of cross-linked interaction, has emerged as a powerful technique, enabling users to interact with multiple data dimensions simultaneously [15, 100, 150, 252]. By linking various filters, analysts can quickly build hypotheses and isolate patterns, trends, and anomalies in complex datasets, leading to more informed decision-making. Stress has been placed in recent years on developing fast systems that are optimized showing more and more data at once while reducing latency in user interaction as much as possible [101, 150, 263].

Automated data processing and cleaning have revolutionized workflows by reducing the time spent on manual data wrangling [67]. Sophisticated algorithms now automatically detect inconsistencies, fill missing values, and transform raw data into usable formats. These improvements enable researchers and practitioners to focus more on analysis rather than preparation.

Faster tooling has further accelerated data exploration. Enhanced computational frameworks and optimized libraries allow for real-time data manipulation, making interactive visualization more responsive [101]. Coupled with easier-to-use grammars and scripting languages, these tools lower the barrier to entry, empowering users with limited visualization, geometry, trigonometry, data, and graphics coding experience to generate complex, interactive, visual representations of data [201]. New visualization types and techniques—ranging from dynamic dashboards, faceting, to immersive 3D visualizations—offer novel ways to explore and interpret data [263].

Despite significant breakthroughs, current advancements have largely neglected the needs of people with disabilities. Innovations in data science and visualization have focused on sighted user populations, prioritizing visual clarity and interaction speed using direct pointer techniques (such as with touch or a mouse) [169]. This focus often overlooks accessibility requirements for individuals who are blind, have low vision, experience cognitive or vestibular challenges, or possess motor disabilities that limit traditional pointer use [258].

2.2.2 Accessibility and Assistive Technology in Research versus Practice

2.2.2.1 Research: Focus on Blindness and Computer Output

Research and standards are both somewhat limited by a strong bias towards visual disabilities. In *Chartability*, 36 of the 50 criteria related to accessible visualization considerations involve visual disabilities [63, 71]. Marriott et al. also found that visual disability considerations are the primary focus of data visualization literature [169], leaving the barriers that many other demographics face unstudied. Accessibility research broadly has traditionally concentrated on the experiences of individuals who are blind, investigating how they perceive and interpret computational output [161]. Studies in this area explore alternative modalities for conveying data, such as auditory representations (through synthesized speech), tactile interfaces, and sonification techniques. Researchers focus on identifying effective methods for transforming visual data into formats that blind users can easily comprehend. This body of work not only examines the perceptual challenges but also delves into cognitive processing differences, aiming to optimize the accessibility of complex information and interactive systems for users with visual impairments.

While research has made strides in converting visual outputs into auditory or tactile forms for blind users, interactive input methods remain underdeveloped. Most efforts have concentrated on optimizing screen reader navigation and information retrieval, leaving text entry and command execution cumbersome. Screen readers, as they currently exist, offer limited support for efficient input, making it challenging for users to perform complex interactions. Although tactile interfaces hold promise for providing more intuitive input methods, they are still in the experimental stage and have not been fully integrated into mainstream accessible computing solutions, perpetuating a critical gap in effective user interaction.

2.2.2.2 Practice: Focus on Standards and Specialization

In contrast, practical accessibility efforts are often centered on the implementation and adherence to established standards and guidelines, such as WCAG [243]. There has been a growing interest in developing guidelines for practitioners [59, 63] and even applying guidelines as a method of validation alongside human studies evaluations and co-design [71, 153, 156, 277]. Existing accessibility standards bodies like the Web Content Accessibility Guidelines do stress the importance of accurate, functional semantics in order for screen reader users to know how to interact with elements [247]. For interactive visualizations this means that button-like or link-like behavior should expressly be made using elements that are semantically buttons and links.

Accessibility professionals, who typically possess specialized expertise, act as intermediaries between the design and development of digital products and the strict requirements of accessibility standards. Their role involves translating abstract guidelines into concrete design solutions, ensuring that websites, applications, and services meet regulatory benchmarks. By focusing on a standards-based approach, practitioners help organizations navigate the complexities of legal and technical requirements, thus ensuring that accessible design principles are integrated into mainstream technology development. This dual focus on rigorous standards and specialized expertise ensures that accessibility is both technically sound and legally compliant across diverse digital environments.

However, accessibility standards are inherently reactive, often lagging behind rapid technological advancements by five, ten, or even twenty years (or more). This delay occurs because developing, vetting, and formalizing standards requires consensus among diverse stakeholders and extensive testing to ensure compatibility and compliance. In contrast, cutting-edge interfaces and computational capabilities evolve swiftly, driven by dynamic market forces and user innovations. Consequently, accessibility guidelines tend to reflect outdated technologies, creating a persistent gap between modern interactive systems and current best practices in accessibility.

2.2.3 Data and Accessibility

In parallel to Mack et al.'s "What do we mean by Accessibility Research?" [162] nearly all topics of study at the intersection of accessibility and data are focused on visualization and vision-related disabilities [258]. Largely, access issues other than vision that affect data visualization (such as cognitive/neurological, vestibular, and motor concerns) are almost entirely unserved in this research space. Kim et al. found that 56 papers have been published between 1999 and 2020 that focus on vision-related accessibility (not including color vision deficiency), with only 3 being published at a visualization venue (and only recently since 2018) [140]. Marriott et al. found that there is no research at all that engages motor accessibility [170]. We have found 2 papers that engage cognitive/neurological disability in visualization and 1 student poster from IEEE Vis, which are all recent (specifically intellectual developmental disabilities [267] and seizure risk [227, 228]). We found no papers that engage vestibular accessibility, such as motion and animation-related accessibility. We also found that there is no research specific to low vision disabilities (not blindness or color vision deficiency) unless conflated with screen reader usage in data visualization. Blind and low vision people are often researched together, but in practice may use different assistive technologies (such as magnifiers and contrast enhancers) and have different interaction practices (such as a combination of sight, magnification, and screen reader use) [234].

Since the 1990s, the most prominent and active accessibility topic in data has been color vision deficiency in data visualization [34, 148, 171, 177, 181]. Research projects that explore tactile sensory substitutions to charts have been a topic in computational sciences dating back to the 1983 [82], with tactile sensory substitutions being used for maps and charts as far back as the 1830s [93]. Sonification used both in comparison to and alongside visualization and tactile methods for accessibility dates as far back as 1985 [26, 47, 75, 164, 173, 275]. Some more recent work has explored robust screen reader data interaction techniques [87, 225], screen reader user experiences with digital, 2-D spatial representations, including data visualizations [202, 212], dug deeper into the semantic layers of effective chart descriptions [154], and investigated how to better understand the role of sensory substitution [42]. Jung et al. offer guidance that expands beyond commonly cited literature that chart descriptions are preferably between 2 and 8 sentences long, written in plain language, and with consideration for the order of information and navigation [133].

A wide array of emerging research projects investigate screen reader users needs, barriers, and preferences, and offer guidelines, models, and considerations for creating accessible data visualizations [41, 71, 153, 211]. Jung et al. offer guidance to consider the order of information in textual descriptions and during navigation [132]. Kim et al. collected screen reader users' questions when interacting with data visualizations, which could open the door for more natural language data interaction [137].

Data visualization accessibility has come far in recent years. But little work has been done to explore what disability scholars call "access friction" - a tension that arises when access must be negotiated [97, 117]. This friction is often a result of static barriers in shared spaces: one artifact or approach designed to include some people may end up excluding others.

Yet despite these resources, making data visualizations more accessible remains a difficult task for practitioners [130, 208]. Some accessibility guidelines even conflict, for example on the topic of patterns and textures used in charts. One side stresses that patterns are harmful to cognitive and visual accessibility [203] while another stresses that redundant encoding strategies are necessary [63].

Chapter 3
Chartability: Heuristics as a Tool and Resource

This chapter was adapted from my published paper:

F. Elavsky, C. Bennett, and D. Moritz, 'How accessible is my visualization? Evaluating visualization accessibility with Chartability', *Computer Graphics Forum*, vol. 41, no. 3, pp. 57–70, Jun. 2022.

3.1 Overview

26% of people in the United States self-report living with at least one disability [180]. Of those, 13.7% live with a mobility disability and 10.8% with a cognitive disability. Globally, the World Health Organization reports that 29% of the world lives with uncorrected or uncorrectable blindness, low vision, or moderate to severe visual impairment [183]. Access is a significant inclusion effort that has broad international impact, especially for data visualization.

Accessibility is the practice of making information, content, and functionality fully available to and usable by people with disabilities. As part of this process, practitioners need to be able to identify accessibility barriers. While general accessibility standards help, evaluating the inaccessibility of complex data systems can be a daunting and often expensive task. State-of-the art automated compliance checkers only find 57% of accessibility errors [53], meaning accessible experiences must still be manually designed and checked for quality. And following standards may only account for up to half of the needs of people with disabilities [189] anyway. Additionally, the intended wide applicability of these general standards means they fall short for information-rich systems, such as data visualizations (which use size, color, angles, shapes, and other dimensions to encode information). These specific contexts, communities, and libraries that deal with data visualizations and information-rich interfaces often have their own tools and guidelines for use, but they seldom include accessibility. Finally, research at the intersection of data visualization and accessibility has yet to meaningfully permeate data visualization tools and communities and primarily focuses on blindness and low vision, neglecting diverse accessibility needs of people with other disabilities.

Synthesizing evolving accessibility standards, research findings, and artifacts from communities of practice into usable knowledge for a specific, evolving domain is a wicked problem. To address this, we present Chartability. Chartability is an accessibility evaluation system specific to data visualizations and interfaces which aims to help practitioners answer the question, "how accessible is my data visualization?" Chartability organizes knowledge from disparate bodies of work into testable heuristics based on the functional accessibility principles POUR (Perceivable, Operable, Understandable, and Robust) [250] and 3 novel principles CAF (Compromising, Assistive, and Flexible), which we added to attend to the unique qualities and demands of data visualizations. We refer to these 7 heuristic principles as POUR+CAF. Chartability is a community-contributed project that leverages the governance strategies of open source projects as a way to address the complex dual-evolution of both accessibility and data interaction practices.

We additionally present an initial, light evaluation of Chartability from the experience of practitioners using it. We set out to see if using Chartability reduces the barrier of entry into this work for accessibility novices and if accessibility experts had any feedback to share about its use. We gave practitioners introductory material for Chartability and instructed them to use it according to their needs. We found that before using Chartability only accessibility experts believed auditing data visualizations to be somewhat easy or easy, while the other group believed auditing data visualizations to be somewhat hard or hard. All novice accessibility practitioners became more confident after using Chartability and believed auditing data visualizations for accessibility to be less difficult. Conversely while the expert accessibility practitioners were already confident in their ability to evaluate accessibility (and all unanimously had no change in their before and after evaluations), they were excited to adopt Chartability into their set of auditing resources.

Our work sets out to acknowledge that data practitioners face significant barriers when first making data visualizations, systems, and experiences accessible. While Chartability contributes to filling gaps and organizing knowledge, it also challenges visualization and data interaction researchers to explore new horizons of possibilities in this space. As such, we conclude with recommendations for future research at the crossroads of data visualization and accessibility.

3.2 Existing Work in Data Visualization and Accessibility

While recent works at the intersection of data visualization and accessibility are promising, they do not provide a consistent and unified methodology for designers to evaluate the accessibility of their work across the broad spectrum of disability considerations.

3.2.1 Research Advancements in Data Visualization and Accessibility

In parallel to Mack et al.'s "What do we mean by Accessibility Research?" [162] when we asked "What do we mean by data visualization accessibility research?" we found that nearly all topics of study were vision-related. Largely, access issues other than vision that affect data visualization (such as cognitive/neurological, vestibular, and motor concerns) are almost entirely unserved in this research space. Kim et al. found that 56 papers have been published between 1999 and 2020 that focus on vision-related accessibility (not including color vision deficiency), with only 3 being published at a visualization venue (and only recently since 2018) [140]. Marriott et al. found that there is no research at all that engages motor accessibility [170]. We have found 2 papers that engage cognitive/neurological disability in visualization and 1 student poster from IEEE Vis, which are all recent (specifically intellectual developmental disabilities [267] and seizure risk [227, 228]). We found no papers that engage vestibular accessibility, such as motion and animation-related accessibility. We also found that there is no research specific to low vision disabilities (not blindness or color vision deficiency) unless conflated with screen reader usage in data visualization. Blind and low vision people are often researched together, but in practice may use different assistive technologies (such as magnifiers and contrast enhancers) and have different interaction practices (such as a combination of sight, magnification, and screen reader use) [234].

Since the 1990s, the most prominent and active accessibility topic in visualization has been color vision deficiency [34, 148, 171, 177, 181]. Research projects that explore tactile sensory substitutions have been a topic in computational sciences dating back to the 1983 [82], with tactile sensory substitutions being used for maps and charts as far back as the 1830s [93]. Sonification used both in comparison to and alongside visualization and tactile methods for accessibility dates as far back as 1985 [26, 47, 75, 164, 173, 275]. Some more recent work has explored robust screen reader data interaction techniques [87, 225], screen reader user experiences with digital, 2-D spatial representations, including data visualizations [202, 212], dug deeper into the semantic layers of effective chart descriptions [154], and investigated how to better understand the role of sensory substitution [42]. Jung et al. offer guidance that expands beyond commonly cited literature that chart descriptions are preferably between 2 and 8 sentences long, written in plain language, and with consideration for the order of information and navigation [133]. We find all of this emerging work promising and foundational.

Despite this promising work emerging, we also want to acknowledge a spectrum of other work that exists at the intersection of accessibility and data visualization that does not serve the goals of our project. There is significant research that explores automatic or extracted textual descriptions [10, 35, 36, 38, 146, 178, 190, 210] and haptic graphs and tactile interfaces [3, 9, 20, 27, 30, 80, 82, 124, 125, 147, 205, 216]. These research projects produce artifacts that are high-cost for individual use, some are not robust enough to interpret complex visualizations effectively, and several have not included people with disabilities. Since our goal is to synthesize knowledge for practitioner accessibility work, we also acknowledge that some of these projects did not follow standards during their research project and in their output, such as using Web Content Accessibility Guidelines [121] or The American Printing House for the Blind and Braille Authority of North America [12]. All of these challenges are factors that limit the generalizability of these artifacts and knowledge for practitioner use [157, 174, 212]. We encourage work to continue at the intersection of accessibility and visualization, but stress the importance of practical, disability-led research that either builds on or explicitly challenges standards.

3.2.2 Accessibility Practices in Data Visualization Tools and Libraries

Our research goals are to find what is already being done in data visualization and accessibility and to see if we can enhance that activity. To this aim, our background investigation includes a broad and comprehensive exploration of the field of practitioner and non-academic artifacts.

Some open source and industry contributions have pushed data visualization and related accessibility efforts. Libraries like Highcharts [109] or Visa Chart Components (VCC) [241] and tools like the Graphics Accelerator in SAS [200] have broad accessibility functionality built in, but their documentation is technically specific to their implementation. While these relatively accessible libraries and tools can be helpful for inspiration, their specific techniques and guidance materials are not easily transferrable to other environments or applications where data visualizations are created. Practitioners must reverse engineer and deconstruct many of the methods employed by these libraries, and with the exception of VCC (which is open source), this task requires significant effort, given their primarily closed-box nature.

In common charting tools and libraries (apart from those already mentioned) accessibility engineering is often not present, limited in scope, or has only recently become an effort. More established visualization libraries like matplotlib, ggplot2, d3js, R-Shiny, and Plotly have left most accessibility efforts to developers, with varying levels of documentation and difficulty involved [32, 44, 77, 78, 255]. None of these major tools have a broad spectrum of accessibility options built in and documented.

Community contributors often must fight to make their tools and environments accessible (sometimes even against the design of the tools themselves) with little to no compensation for their contributions. For example, Tableau's first accessible data table was built by a volunteer community member Toan Hong as an extension [114]. Tableau users more broadly must resort to voting systems to gather attention to accessibility issues [51]. Semiotic's accessibility features were added by community member Melanie Mazanec [172]. For Microsoft's PowerBI, students have organized resources for how to make visualizations built with it more accessible [143] while non-profits like the City of San Francisco's data team have had to build features like keyboard instructions from scratch [179]. Mapbox GL JS is an example of a popular mapping library (over 400,000 weekly downloads) [165] that has no built-in accessibility support by default. The accessibility module for Mapbox GL on GitHub was created and maintained by volunteers but has had less than 10 weeks of work with any activity invested since its first activity in late 2017 [166].

Many community-driven efforts are under-utilized, must be discovered outside of the primary environment's ecosystem, have poor or no core, internal support, and are inconsistently and partially implemented. Accessibility is still an afterthought in data visualization and ad-hoc, specific solutions proposed have not led to widespread improvements.

3.2.3 Accessibility in Practice, Broadly

Accessibility in practice is largely motivated by standards work or assistive technology. We want to acknowledge that tactile and braille standards are robust [12], but have limited transferability to digital contexts currently. For example, whereas tactile graphics guidelines lend insight into information prioritization, layout, and fidelity, the assumption is they will be embossed onto paper or similar physical mediums [18, 157, 212].

In digital contexts, the most influential body for accessibility is the World Wide Web Consortium's (W3C) Web Accessibility Initiative (WAI). WAI's Web Content Accessibility Guidelines (WCAG) [121] influence accessible technology policy and law for more than 55% of the world's population [122]. WAI and WCAG outline 4 types of functional accessibility principles: Perceivable, Operable, Understandable, and Robust, abbreviated as POUR [250]. POUR is the foundation that organizes all 78 accessibility testing criteria in WCAG.

3.2.4 Using Heuristics to Break Into Under-addressed Areas

To summarize the complex problem space to which this paper contributes: Research in data visualization primarily focuses on visual accessibility, accessibility standards focus on a broad range of disabilities but lack deep contextualization for data visualization, and practitioners seem to build a wide array of solutions to fill these gaps, most of which are poorly maintained or adopted. Any time a practitioner wants to embark on a journey learning how to evaluate the accessibility of a data visualization, they must collect and synthesize this complex space of knowledge themselves. We have included (with permission) an exemplary field artifact as an example of this type of labor in our supplemental materials, which contributed to the United States Government's project, "Improving Accessibility in Data Visualizations" [239, 240].

After gathering information with this breadth and complexity, a heuristic evaluation model was chosen as a way to deliver useful but flexible knowledge. Heuristic evaluation models have a long history in HCI and are cheap to use and require little expertise. They have been shown to be effective methods for practitioners compared to user testing, focus groups, or other evaluative methods that require existing expert knowledge or recruitment, moderation, and compensation of participants [22, 24, 39, 129, 171, 175, 176, 184, 198, 222]. Heuristics are also not new in visualization [46, 76, 182, 206] even among topics related to accessibility (color vision deficiency, specifically) [181, 198].

3.3 Making Chartability

We next present Elavsky's work to develop Chartability as a real-world design process contribution to the larger research community. Our making process does not neatly fit into most design models that divide researchers from practitioners. In Gray's different models of practitioner-researcher relations, our work is some variation of bubble-up, practitioner-led research [89]. This project was initiated by Elavsky while they were an industry practitioner, deeply situated in this work already.

Thus, the following description of Chartability's 10-month creation is written from Elavsky's perspective. The supplemental materials include the data from this stage of the process, a preview of which is available in Table 3.1:

1. **Situate, Survey, and Select Problem Space**: I was situated within the context of accessibility evaluations of data visualizations. From personal experience, I recognized the prohibitively significant labor involved in ensuring I was effectively following accessibility standards while also attending to the complex design considerations of data visualizations. To improve this work both for myself and others in the future, I surveyed existing problems and challenges others faced and selected a solution that I felt equipped to address.
2. **Collect Existing Resources**: I set out to answer, "If evaluating the accessibility of data experiences is hard, what do existing standards miss?" I evaluated my seed knowledge (WCAG criteria) for shortcomings and gaps and collected other data relevant to my goal (academic and industry research, open-source libraries, tools, applications, data products, government guidelines, design guidelines, software documentation, university coursework, and practitioner articles).
3. **Code Resources**: After collating these resources (including relevant WCAG criteria), I loosely borrowed from thematic analysis [25] and qualitatively coded this data. I developed a set of 29 codes starting with WCAG's POUR principles and expanded the codes to account for other concerns that came up in the resources, including what type of acces-

sibility was being addressed (e.g., cognitive, visual), whether a solution was technology-specific or agnostic, and other categories (like "time-consuming" or "user-controlled"). I then divided the resources into codable segments with relatively distinct pieces of information and applied the 29 codes to the information segments. I grouped information with codes in common, resulting in a representative 45 groups of related information segments.

4. **Synthesize Heuristics**: Since auditing depends on measurable heuristics, I adjusted each of the 45 groupings that resulted from the qualitative analysis into phrasing that could be verified by an evaluation. I then augmented each heuristic with known testing procedures, resources, and tools necessary for applying them in practice. 10 critical heuristics (these were determined top priorities through user feedback) are previewed in Table 3.1, with the full version of this table (and more) provided in our supplemental materials.
5. **Group Heuristics into Higher-level Principles**: I linked each heuristic with relevant web accessibility standards and POUR principles to draw a familiar connection for users who might already be accessibility practitioners. 26 heuristics fit neatly back into Perceivable, Operable, Understandable, or Robust.
6. **Develop Remaining Themes into New Principles**: 19 remaining heuristics with complex codes and overlapping groups demanded new theorizing, as they either did not fit into POUR at all or could arguably belong to multiple principles at once. I analyzed these remaining complex heuristics and for similarities and organized them under 3 new themes, which we are contributing as new accessibility principles, Compromising, Assistive, and Flexible, defined below.

Table 3.1: Previewing Chartability's 10 Critical Heuristics

(Coding Categories are broken into two sections: first which POUR principles contributed to the heuristic while "Other" refers to how many additional coding categories were assigned.)

			Coding Categories	
Heuristic Title	Principle	Origin	POUR	Other
Low contrast	Perceivable	Standard	P	2
Small text	Perceivable	Research	P	2
Content is only visual	Perceivable	Standard	P, R	3
Interaction has only one input	Operable	Standard	O, R	3
No interaction cues/instructions	Operable	Standard	O, U	2
No explanation for how to read	Understandable	Research	U	1
No title, summary, or caption	Understandable	Research	U	1
No table	Compromising	Research	O, U, R	3
Data density inappropriate	Assistive	Research	P, U	4
User style change not respected	Flexible	Standard	P, O, R	6
... +35 non-Critical heuristics				

3.3.1 Compromising

Compromising is a principle that focuses on Understandable, yet Robust heuristics. These heuristics are based on providing alternative, transparent, tolerant, information flows with consideration for different ways that users of assistive technologies and users with disabilities need to consume information.

Compromising challenges designs that only allow access to information through limited or few interfaces or processes. These heuristics focus on providing information at a low and high level (such as tables and summaries), transparency about the state of complex interactions, error tolerance, and that data structures can be navigated according to their presentation. Compromising designs have both information and system redundancies in place.

3.3.2 Assistive

Assistive is a principle that primarily builds off the intersection of Understandable and Perceivable principles but focuses on the labor involved in access. These heuristics include categories that encourage data interfaces to be intelligent and multi-sensory in a way that reduces the cognitive and functional labor required of the user as much as possible.

The Assistive principle focuses on what Swan et al. refer to as "adding value" [233] and what Doug Schepers meant by "data visualization is an assistive technology" [204]. We visualize because it is faster and more efficient than munging cell at a time through data. Assistive heuristics ensure that both visual and non-visual data representations add value for people with disabilities.

3.3.3 Flexible

Contrasted with Compromising (which focuses on robust understanding), flexible heuristics focus on robust user agency and the ability to adjust the Perceivable and Operable traits of a data experience. Flexible heuristics all have a tight coupling between a data experience and the larger technological context the user inhabits. The preferences that a user sets in lower-level systems must be respected in higher level environments.

Self-advocacy and interdependent agency are important sociotechnical considerations that engage the conflicting access needs that different users might have in complex technological interactions like data experiences [16, 163]. Some users might want specific controls or presentation, while others might want something else entirely. Designs must not be rigid in their opinions and ability assumptions and should be designed to be moldable by and adaptive to user needs [145, 262].

3.4 Using Chartability

All of Chartability's tests are performed using Chartability's workbook [61] alongside various tools and software (linked in the workbook). For the scope of this paper, we are not including an explanation for how to perform all of these. Both the workbook and supplementary materials with this paper give more details.

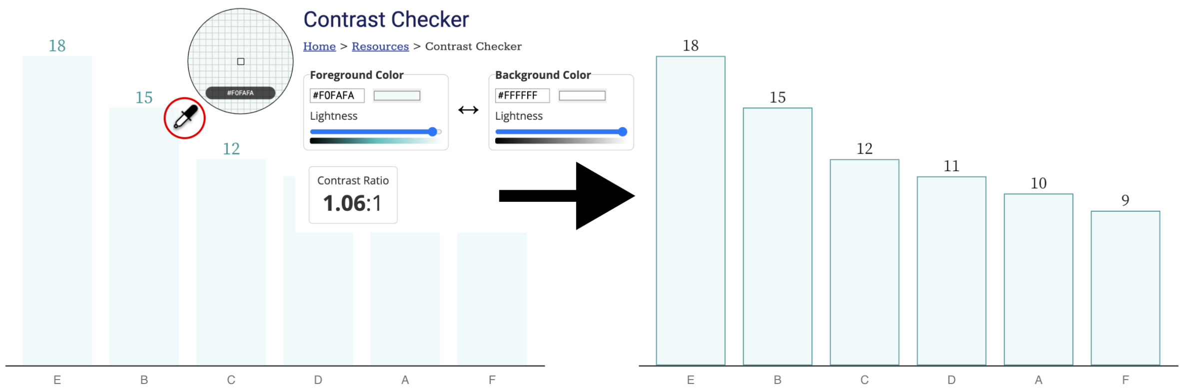


Figure 3.1: A low contrast chart (left) compared to a higher contrast version (right). A dropper tool is extracting the fill color of the bar and then a contrast ratio has been calculated. Note that the fill color is the same on both bars, but darker borders have been added to ensure the visualization passes contrast tests.

While a highly trained auditor may be able to casually evaluate an artifact in as little as 30 minutes or even hold heuristics in mind as they are doing their own creative work, those new to auditing may take anywhere between 2 and 8 hours to complete a full pass of Chartability. Professional audits, which can take weeks or months, often include multiple auditors and provide rigorous documentation and detailed recommendations for remediation, typically in the form of a report. Chartability is meant to serve both quick pass and deep dive styles of audits, so users are expected to leverage it as they see fit.

Below we give an example of what might be a quick pass audit, using Chartability. Which principles are applied in each of these stages are listed in parentheses in each heading.

3.4.1 Visual Testing (Perceivable)

Checking for contrast is the most common critical failure; 87.5% of tests (7 out of 8) from our user study involving this heuristic failed, which supports the WebAim Million Report's findings (83.9% of the top 1 million websites also fail contrast testing, more than any other WCAG criteria) [254]. In order to evaluate contrast, often a combination of automatic (code-driven) and manual tooling is performed. When manually auditing, practitioners typically use a dropper and a contrast calculator (Figure 3.1). Most auditors find this to be one of the easiest tasks to perform and accomplishes 3 different heuristics in Chartability: ensuring text/geometries have contrast, interactive states for elements have enough contrast change, and the keyboard focus indicator is easy to distinguish.

Perceivable heuristics also include tests and tools for color vision deficiency and ensuring that color alone isn't used to communicate meaning (like the redundantly encoded textures in Figure 3.8). And another common, critical failure from Perceivable is text size. No text should be smaller than 12px/9pt in size.

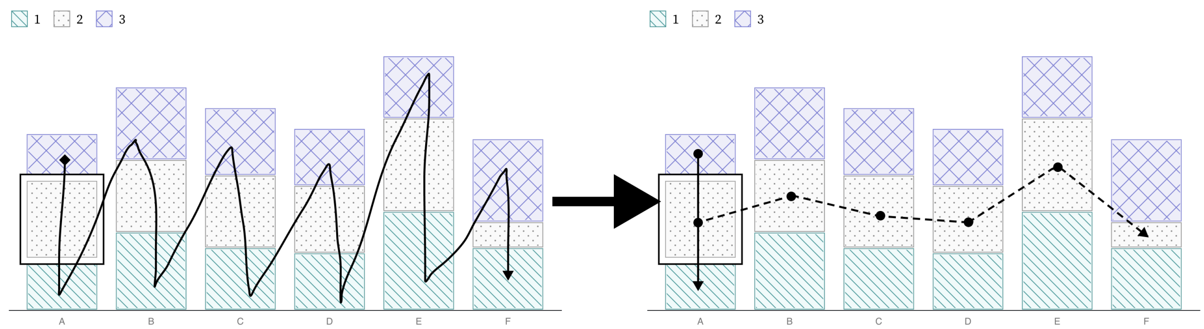


Figure 3.2: Keyboard navigation paths on a stacked bar chart. The left shows a serial navigation example, typically just a default of rendering order. The right shows both groups (the stack of bars) and categories (the color/texture shared among bars across stacks) as dimensions to explore laterally or vertically.

3.4.2 Keyboard Probing (Operable, Assistive)

The next practice that most auditors should become comfortable with is using a keyboard to navigate and operate any functionality that is provided. Most assistive technologies, from screen readers to a variety of input devices (like switches, joysticks, sip and puffs, etc) use the keyboard api (or keyboard interface) to navigate content. If a data interface contains interactive elements (Figure 3.2, Figure 3.3), those elements (or their functionality) must be able to be reached and controlled using a keyboard alone. Auditors should be critical of how much work is involved in keyboard navigation, especially (Figure 3.7). All that is required to start is the auditor begins pressing the tab key to see if anything interactive comes into focus. Arrow keys, spacebar, enter, and escape may be used in some contexts. Generally, instructions or cues should always be provided.

Using a keyboard provides an opportunity to evaluate many different heuristics: checking for multiple inputs (Figure 3.3), whether the data structure that is rendered is navigable according to its structure (Figure 3.2), and whether keyboard navigability across all elements in a data interface is even necessary (Figure 3.7).

3.4.3 Screen Reader Inspecting (Perceivable, Operable, Robust, Assistive)

Closely related to keyboard testing is testing with a screen reader. Some things may work with a screen reader that do not with a keyboard (and vice versa), so both must be evaluated.

Screen readers, unlike more basic keyboard input devices, read out content that is textual (including non-visual textual information like *alternative text*). Using a screen reader to audit is generally the hardest skill to learn. Keeping this in mind, testing whether the meaningful text provided in a visual (such as in Figure 3.4) is accessible with a screen reader is the easiest and most basic test that auditors should first perform.

Next, all valuable information and functionality in a data experience should be tested whether it is available to a screen reader. This includes the individual variables about a mark as well as

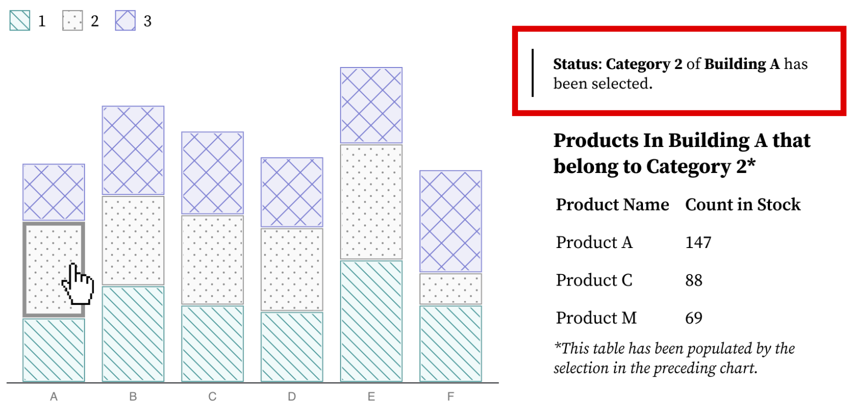


Figure 3.3: A mouse cursor is selecting a bar (left, shown with a thick indication border) in a stacked bar chart to filter a dataset (on the right). A system alert (red box) notifies the user of their interaction result. This selection capability must also be provided for the keyboard interface and the alert must be announced to screen readers.

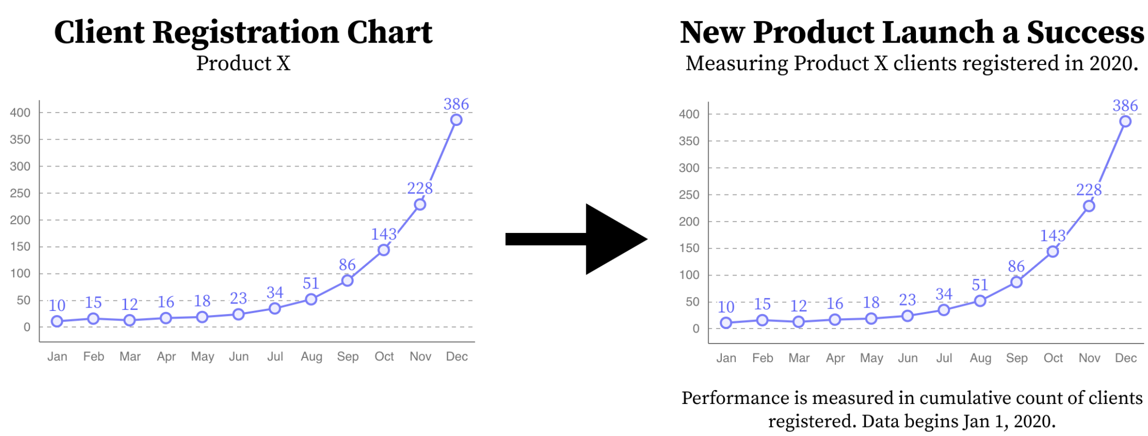


Figure 3.4: Charts must have a visually available textual explanation provided that summarizes the outcome. “Client Registration Chart” for “Product X” (left) is inaccessible while “New Product Launch a Success” (right) gives a clear takeaway.

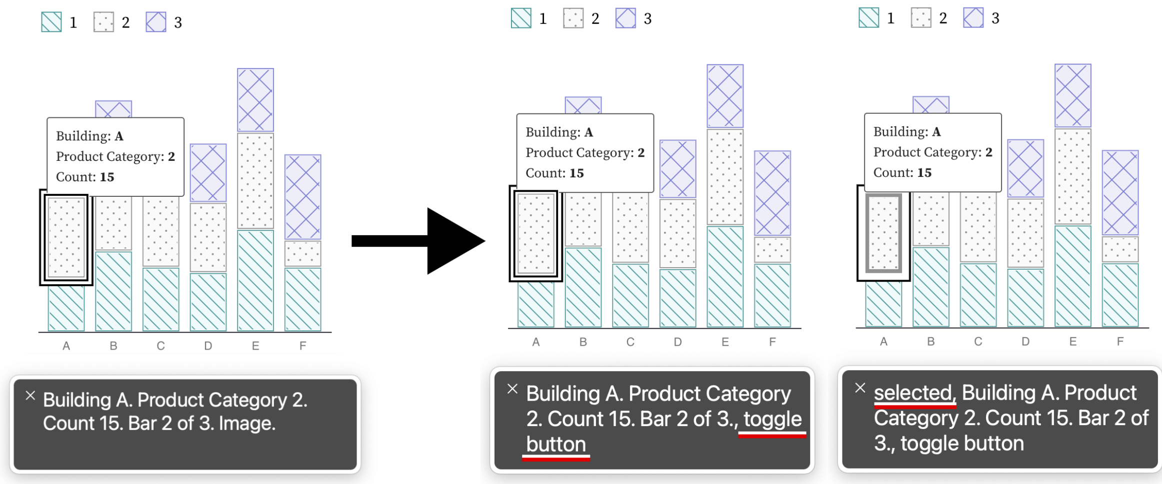


Figure 3.5: An interactive chart displaying only "Image" as semantic information with no feedback provided on selection. The robust semantics given to a screen reader, "toggle button" (middle) as well instant feedback, "selected" (right) are considered proper semantics for an interactive experience.

whether that mark is interactive (Figure 3.5), whether status updates that reflect context change provide alerts (Figure 3.3), and whether summary textual information is provided about the whole chart (Figure 3.4) as well as statistically and visually important areas of that chart (Figure 3.7).

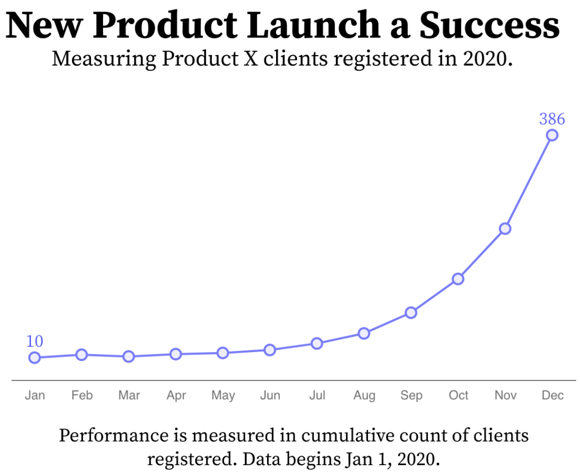
3.4.4 Checking Cognitive Barriers (Understandable, Compromising)

First, auditing for cognitive barriers generally involves checking the reading level and clarity of all available text using analytical tools. But Chartability also requires that all charts have basic text that provides a visually-available textual description and takeaway (Figure 3.4). This alone is one of the most important things to check for. In complex cases where a chart has a visual feature with an assumedly obvious takeaway, checking for annotations or textual callouts is important to help avoid interpretive issues [269] (Figure 3.7).

3.4.5 Evaluating Context (Robust, Assistive, Flexible)

The final series of checks an auditor should make involve thinking about the overall work in a design (as it intersects with other considerations) as well as the larger technical context where the user is situated.

Auditors should first try to change system settings (such as toggling high contrast modes) to see whether a data experience respects these settings (Figure 3.8), run automatic semantic evaluations as well as manually check for appropriate meaning (Figure 3.5), and check if dense or highly complex visuals have sonified, tactile, or textual summaries available (Figure 3.7).



Month	Registered Clients
Jan	10
Feb	15
Mar	12
Apr	16
May	18
Jun	23
Jul	34
Aug	51
Sept	86
Oct	143
Nov	228
Dec	386

Figure 3.6: A line chart (left) with a single line and an accompanying data table (right). This line chart would not provide enough low-level information about each datapoint without the table provided. A table alone however would also be inaccessible. Providing both can satisfy conflicting accessibility needs for different audiences.

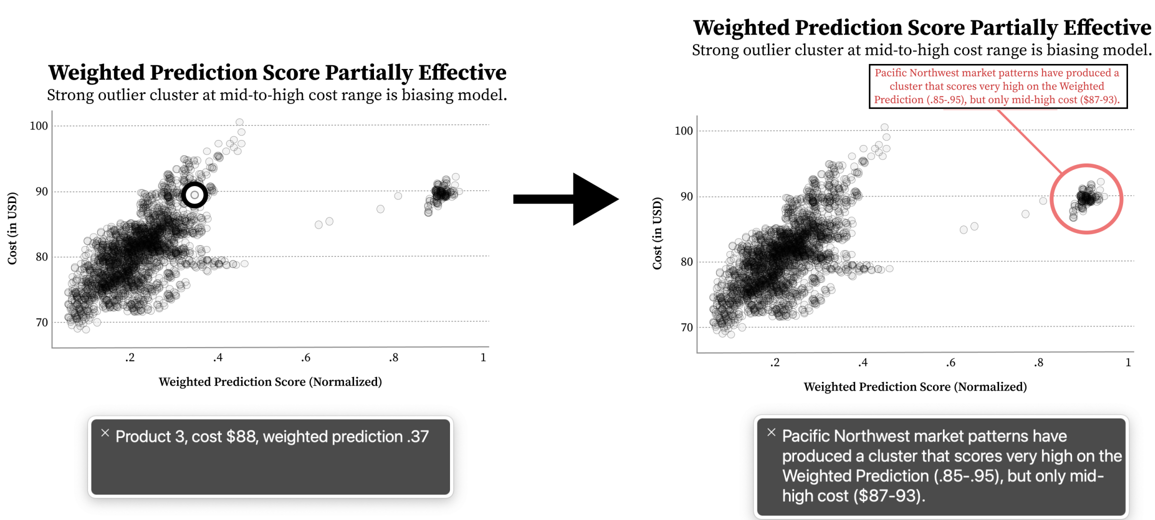


Figure 3.7: A scatterplot with many points, where a single point within the chart can be accessed by a screen reader (left). Navigating this data piece by piece is unnecessarily tedious, so an annotation callout is provided to help the reader focus on an outlier cluster (right). The callout is being accessed by a screen reader, which is displaying the annotation's summary as well.

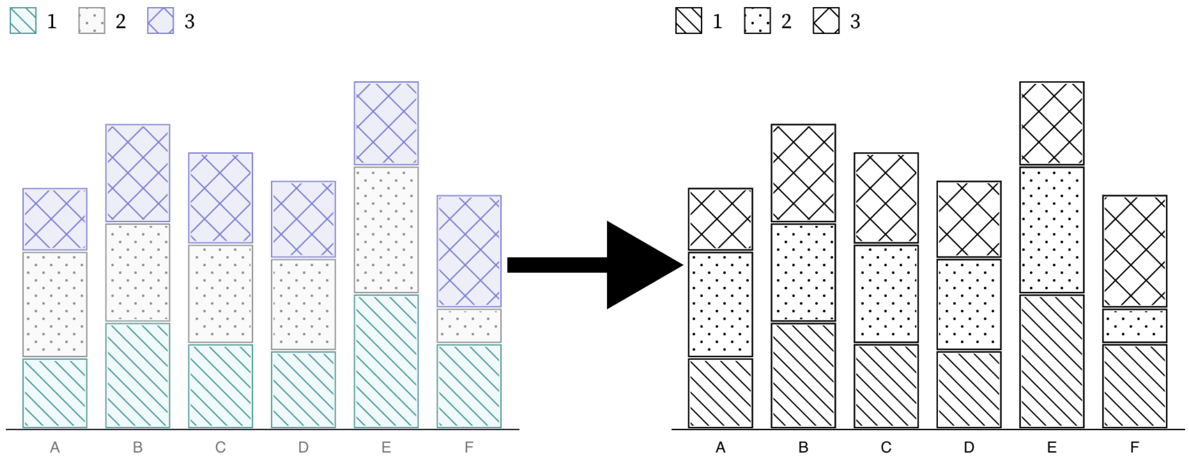


Figure 3.8: A bar chart with categories (left) shown not conforming to Windows High Contrast White Mode. High contrast mode on Windows requires limiting color palettes, using only black or white for most elements (shown on the right).

Auditors should also check whether system updates provide clear feedback textually (Figure 3.3) as well as checking if there are both high and low level representations of information available (Figure 3.6).

Auditors should be especially critical of static designs, such as those that either use textures by default or not (Figure 3.8), which are a high risk of compromising and assistive failure.

3.5 Validating Chartability

Next, Elavsky explains the preliminary user evaluation: I validated whether data practitioners felt more confident and equipped to make their own work accessible with Chartability. Additionally, I also wanted to interview expert accessibility practitioners (including those with disabilities) with the same questions, to see if Chartability had anything to offer in helping them understand and evaluate data experiences better.

My secondary goal was to present a tool that can be helpful even in the wild on real projects (with all the weird design and engineering quirks that come with that). I wanted Chartability to be usable on things built with a tool like Tableau and fully bespoke, hand-coded visualizations, like those made with JavaScript and D3. To this secondary aim, I intentionally solicited participants who were working on a variety of different projects, each of their own design.

3.5.1 Pre-Validations and Flipped Roles: Participants Question *Me*

I performed several early, light validations of my work before soliciting and involving participants formally. My early pre-validations #2-4 (below) all focused on practitioners asking me questions and giving feedback.

My 4 pre-validations happened during the process of making Chartability, as well as introducing short iterations back into the making process:

1. **Beta Testing**: I performed several beta tests of Chartability during the process of making. I audited using versions that only had POUR principles, tried versions of Chartability that focused only on standards, and also tried out different iterations of the heuristics as I was forming them. This testing was important to perform early in the process because it helped me test the limits of various possible directions for this tool (standards-only, against standards, building off of standards, etc).
2. **Early Advice**: After the first full pass of making Chartability was complete, I sent Chartability via email to 4 accessibility experts and 6 interested people with disabilities familiar with auditing in order to solicit open feedback.
3. **Professional Workshop**: I held a half-day professional workshop via zoom on auditing visualizations for accessibility and presented Chartability's heuristics to this select audience of 50 participants. I demonstrated how to audit and then had a chance for feedback and questions.
4. **Deep Feedback Session**: I presented Chartability to 14 experts on data visualization and accessibility, 5 of which are people with disabilities. I presented in two separate sessions through 2-hour video calls on zoom (roughly one hour was demonstration and one hour was discussion).

3.5.2 Discovering "Critical" Heuristics

These pre-validations helped me combine and divide some of the heuristics, adjust the language and phrasing, and label 10 specific tests as "Critical," which can be seen in Table 3.1. These critical tests were ones that community members stressed as an important priority for one or more of the following reasons:

- They are prohibitively expensive to fix late.
- The barriers they produce are too significant to ignore.
- They are among the most common type of accessibility failure.
- They affect many parts of a data experience.

All Critical heuristics are based on standards or research.

3.5.3 Selecting Participants and Projects

I was a practitioner and representing myself as a volunteer when I reached out to participants. At this stage in the project, I was still not affiliated with a research institution and was not interested in producing publishable knowledge. I intended to test Chartability in the wild and validate whether it achieved its aims. My priority was to collaborate with folks working on difficult problems or those who had a rare intersection of expertise between accessibility standards and interactive data experiences. To this end, I was highly permissive with potential collaborators in order to maximize the expertise of participants and breadth of environments for testing Chartability.

However, part of being permissive with participants meant that I was willing to collaborate on projects that I cannot share in a research publication and many of my participants must remain anonymous (including interview results that contain sensitive information about intellectual property). Given that auditing is a field of work about identifying failures, there was both a high demand for participation in the evaluation of Chartability in tension with a low motivation to make these failures known in a public venue.

A summary of our selection process:

- **Solicitation**: I reached out via email to 24 individuals in my network to participate in helping to evaluate Chartability. I mentioned that I wanted Chartability to be applied to a current project of theirs and was interested in performing some interviews about their experience before and after using Chartability. I mentioned up front that working with me would be uncompensated and potentially take multiple hours of their time (even multiple sessions) over zoom meetings.
- **Response**: 16 individuals were interested and shared their project details (2 would require an NDA to be signed).
- **Selection**: I selected 8, based either on the expertise of the individuals, on the robustness of their project, and/or on the opportunity to get feedback about Chartability in team environments (which I didn't anticipate, but 3 of the 8 represented team efforts).
- **Resulting Group**: I worked with 19 total participants across 8 environment spaces.
- **Publishable Group**: Due to intellectual property concerns, I can publish interview results from 6 participants and discuss the details of 4 audit environments.

Chris DeMartini: a multi-year Tableau Zen Master and recognized expert visualization practitioner. His dashboard of a coin flipping probability game dataset that he produced with his daughter was the subject of his audit [50]. His audit only included criteria labelled Critical in Chartability (which involves only 10 tests instead of the full 45) and his dashboard failed 7 of them. A full audit was later conducted on Chris's behalf. His full audit had a total 26 failures, 11 of which were considered non-applicable.[1]

Amber Thomas: a data storyteller and technologist credited on 30 of The Pudding's visual essays. Amber has had a growing interest in accessibility challenges related to her line of work designing and developing state of the art, bespoke visual essays. Her article The Naked Truth was still in the early design and development stages when it was fully audited [5]. It failed 22 out of 45 tests, including 6 out of 10 criteria considered Critical. 6 tests were considered non-applicable.[1]

Sam (self-selected pseudonym): a recognized design practitioner in the visualization community who lives with disability. They were collaborating on an interactive data project that would be specifically made to be used by international participants with a broad spectrum of disabilities. Their interactive infographic failed 21 out of 45 tests, 5 of which were considered Critical. 10 tests were considered non-applicable.[1]

Øystein Moseng: Core Developer and Head of Accessibility of Highcharts. Øystein was interested in taking one of Highchart's demo charts not specifically developed with accessibility

[1] "Non-applicable:" any test in the auditing process that that does not contain content relevant to the test, such as "Scrolling experiences cannot be adjusted or opted out of" for a visualization that does not a scrolling input control

features in mind [112] and testing it against a full Chartability audit to see how it held up. The demo failed 13 out of 45 tests, 3 of which were Critical. 10 tests were considered non-applicable.[1]

Jennifer Zhang: a senior accessibility program manager at Microsoft with expertise working on enterprise data products.

Ryan Shugart: a blind, screen reader user and disability subject matter expert at Microsoft who has a strong expertise in collaborative accessibility for interactive data systems.

Both Shugart and Zhang were interested in applying Chartability internally and testing its effectiveness and potential with various projects. Their application and use of Chartability (including audits) are not available for publication, but their valuable interviews and evaluations are included with permission.

3.6 Study Results

I asked the 6 participants a series of qualitative and Likert-scale evaluation questions:

1. Have you ever performed an audit of a data experience before?
2. What stage of production is your project in? Analysis, design, prototyping, development, maintenance?
3. How confident are you in your ability to perform an audit of a data experience for accessibility issues? (1-5, 1 being not confident at all, 5 being fully confident.)
4. How difficult do you perceive auditing a data experience for accessibility issues is? (1-5, 1 being trivial, 5 being very difficult.)
5. (After using Chartability) How confident are you in your ability to perform an audit of a data experience for accessibility issues? (1-5, 1 being not confident at all, 5 being fully confident.)
6. (After using Chartability) How difficult do you perceive auditing a data experience for accessibility issues is? (1-5, 1 being trivial, 5 being very difficult.)
7. (After using Chartability) Do you intend to continue using Chartability?

Each of these questions had an open-ended question attached, "Is there anything else you would like to add?" Every participant provided additional input on questions 3 through 7.

None of the 3 participants who only consider themselves expert data practitioners had performed an audit before. All 3 of them reported that they believed auditing to be easier and that they are more confident in their ability to evaluate the accessibility of data experiences after using Chartability.

Of the 3 accessibility experts (all of whom have performed audits of data experiences before), their opinions on these measurements were unchanged after using Chartability. All 6 participants noted that they plan to use Chartability in their own work and would recommend it to their peers.

Below we overview some of the key insights Elavsky received from the open ended responses.

3.6.1 Real Access has more Considerations than Colorblindness

Among the data practitioners, DeMartini wrote after his audit, “I have read a lot about color blindness and could provide meaningful feedback to visualization developers on that topic, but I have come to realize that accessibility is so much more than this and I basically didn’t really know where to start when it came to the true scope of accessibility.” He ended his qualitative feedback with, “I think this could be a great tool for the masses and really look forward to the impact it can possibly have on the (inaccessible) data visualizations which are being created in huge numbers these days.”

3.6.2 Audits are Slow, but Help me Focus

Amber Thomas wrote, “It still takes a while to do a complete audit, but it’s not hard! For someone new to the space, all the possible options that can be used to make visualizations more accessible can be overwhelming. [Chartability] helped me to focus.” She finished her feedback with, “There aren’t really guidelines (at least to my knowledge) that exist to help data visualization creators to ensure their work is accessible… [Chartability] helps to direct users to the most common accessibility problems with straightforward questions. It really helps to narrow the focus and prioritize efforts.”

3.6.3 Chartability Helps me Remember and Stay Consistent

Among the accessibility experts, Zhang wrote, “While I am skilled, depending on the day I might not remember everything I need to look at. I am more confident in consistency between different auditing sessions. For experts it’s a good reminder framework.” Moseng of Highcharts noted, “[Chartability] did a very good job of highlighting concerns that are often ignored or forgotten when auditing and designing/developing.” Shugart of Microsoft added along those lines, “I feel [Chartability] arranges a good set of questions in a user’s mind and makes it easier for them to determine if a visualization is accessible.”

3.6.4 Access is an Experience, not just Compliance

Zhang offered insight into the design intention of Chartability, “[it is] clearly going for above compliance and focusing on a good experience.” Sam expressed their need to make an excellent accessibility experience, “I am not just worried about compliance, but I want to make something really good. Nothing seems to help you go beyond? This is better than WCAG, I can already tell.”

3.6.5 Everyone wants More Evaluation Resources and Tools

For constructive feedback, all the data experts noted that they wanted more resources and materials related to learning the skills needed to conduct an audit. Shugart and Moseng both noted that they hope for more tooling and (in some cases) automated tests that can take the burden off the auditor and streamline the design and development process (much like Axe-core [52]). They both also agreed that automation and tooling would help novice practitioners perform this work faster and with more confidence. 2 of the 3 mentioned wanting more examples of failures as well as accessible data experiences. Sam wrote that they felt Chartability was overwhelming at first, but after focusing on just the Critical items, the rest of the framework "became easier."

3.6.6 Experts: "Novices will Struggle." Novices: "This was so helpful"

The accessibility experts all unanimously agreed that Chartability is helpful to their own work, but they are unsure how accessibility novices would do. They all believe that more training and resources are needed to help people who are new, with one noting that Chartability could even be "overwhelming" to someone who has not been exposed to accessibility work before. All of the novices remarked that Chartability was "so helpful," "made this work so much clearer than before," and "made a lot of hard problems not as hard."

3.6.7 What about Auditors with Disabilities?

Shugart's feedback was critical when discussing continuing to use Chartability, "I still feel as a screen reader user, the audit itself would have some unique challenges because I'd be missing a lot and would have problems determining things such as color." He continued, "Auditing anything accessibility-wise as a screen reader user poses challenges because you don't always know what you're missing. In many cases there are workarounds to this but datavis is one area where this is really hard to do now."

3.7 Extended Results

Following calls to ensure accessibility work has practical benefits that exceed publications [119], in April, 2021 Elavsky made Chartability openly available on Github. As new research and practices emerge and more community members get involved, Chartability will become an evolving artifact of consensus similar to existing standards bodies [251].

Projects like Turkopticon benefited from the discussion about how a community actually used their tool [123]. In the same vein, we are happy to report some valuable findings from within this last year that we think demonstrate (in a pragmatic way) that Chartability has some merit:

- **It is living and growing**: Chartability has received enough community feedback that it is now on Version 2, with more tests and background resources provided.
- **People are talking about it**: Chartability has been featured in 14 workshops, talks, and podcasts and at least 2 university courses.
- **People are using it**: Chartability has contributed to projects at Microsoft, Highcharts, Project Jupyter, Fizz Studio, FiveThirtyEight, Vega-Lite, UCLA, the City of San Francisco, the Missouri School of Journalism, a fortune 50 company, two Fortune 500 companies, and community groups (like MiR).
- **It has breadth**: Chartability has evaluated static and interactive data experiences made with Microsoft's Excel and PowerBI, Tableau, JavaScript (D3, Vega-Lite, Highcharts, Visa

Chart Components), Python (Altair, Bokeh, and matplotlib), R (ggplot2), as well as design sketches and low/medium-fidelity artifacts (Illustrator, Figma, Sketch).

When considering the analysis by Hurst and Kane about high abandonment rates in assistive technology, [119] we wanted to make sure that we created an artifact (assistive technology or otherwise) that would at least survive its first year of use in the real world.

The greater community feedback as well as new research before and after open-sourcing Chartability has also led to 5 new heuristics being added since our test users performed audits and gave evaluations. The current version of Chartability (v2) has a total of 50 heuristics.

It is important to note that the work of Chartability did not begin and does not conclude with the publication of this manuscript. We want Chartability to become a living, community-driven effort that will adapt and grow as more resources, tools, and research become available.

3.8 Discussion

From our presentation of Chartability and a preliminary user evaluation with data visualization and accessibility practitioners, we learned that Chartability reduced the perception that working on accessibility is difficult and increased the confidence of those new to this work. Chartability shows promise as a useful framework for expert accessibility practitioners because it serves to produce consistency in contexts like the evaluation of dashboards, data science workflows, and other complex, data-driven interfaces.

While our practitioners with novice accessibility experience were initially concerned about doing the audit correctly, most of their audit results were reasonably comparable with that of the authors (although their time to complete was much longer).

We agree with experts that beyond Chartability, more resources are needed which provide examples of both inaccessible and accessible data visualizations as well as how to perform some of the more difficult parts of the auditing process (such as evaluating with a screen reader). We hope that keeping Chartability on GitHub will inspire future improvements to address this gap in examples, and will address future limitations, as we discover them.

Chartability is a valuable tool for auditing. But we also hope that it can inspire researchers to:

1. Examine which heuristics (in our supplemental materials) could use more research attention, particularly those labelled "community practice."
2. Define constraints or requirements on novel projects, ensuring that new explorations still respects established standards, mitigating ethical risks.
3. Explore the intersections of disability in ways yet unaddressed in standards, such as the strong overlaps between understandability and operability (like keyboard navigation patterns across a data structure) or conflicts in understandability and flexibility (how some users need redundant encodings on charts while others find this overwhelming).
4. Consider access barriers in data experiences beyond those related to visual perception.
5. Engage the relationship between labor and access in computing, such as developing more measurements that demonstrate the imbalance of time and effort expected of users with disabilities (even in systems considered to provide "equal" access) and ways to evaluate

who is contributing to accessibility efforts in a project (core team members, contractors, or volunteers).

We also want to caution researchers who are considering developing heuristics or auditing tools for use in practitioner environments to consider the tradeoffs between evaluation in rich, authentic professional settings and concerns such as intellectual property and corporate branding. We were able to apply our work in rich and collaborative practitioner settings because we were permissive with our potential participants. However, much of this work exists behind closed doors, similar to the downsides of industry research settings. More work may need to be done in order to encourage rich, cross-industry research projects, such as helping to anonymize the content of intellectual property and not just participants, while retaining data and findings.

3.9 Summary

The demand for accessible data experiences is long overdue. The Web Accessibility Initiative's (WAI) Web Content Accessibility Guidelines (WCAG) are over 22 years old and yet little work has been done to synthesize this large body of existing accessibility standards with research and inclusive design principles relevant to the fields of data communication, data science, data analysis, and visualization. Chartability begins to address unique accessibility best practice gaps in these domains with specific heuristics. This synthesis is meant to empower researchers, analysts, designers, developers, editors, and accessibility specialists with a framework to audit the accessibility of data experiences, interfaces, and systems to produce more inclusive environments for users with disabilities. The goal of Chartability is to make this work easier in order to encourage practitioners to regard current practices and resources, some of which have existed for decades.

In addition, Chartability opens the door to more work that remains to be explored in this space. Additional research is needed into many of the topic areas within Chartability's heuristic principles (POUR+CAF) as well as resources, examples, and tools provided for practitioners to perform this work more confidently and efficiently.

The changing landscape of visualization techniques and alternative interfaces (such as sonification and dynamic tactile graphics) may increase the demands for accessibility considerations in this space. The growing technological divide will become an even greater human rights issue as time moves on and we believe that tools like Chartability are necessary for the community of data practitioners to ensure they are including people with disabilities.

Chapter 4
Data-Navigator: Accessibility Tooling for Visualization Toolmakers

---

This chapter was adapted from my published paper:

> F. Elavsky, L. Nadolskis, and D. Moritz, 'Data Navigator: An Accessibility-Centered Data Navigation Toolkit', *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–11, 2023.

4.1 Overview

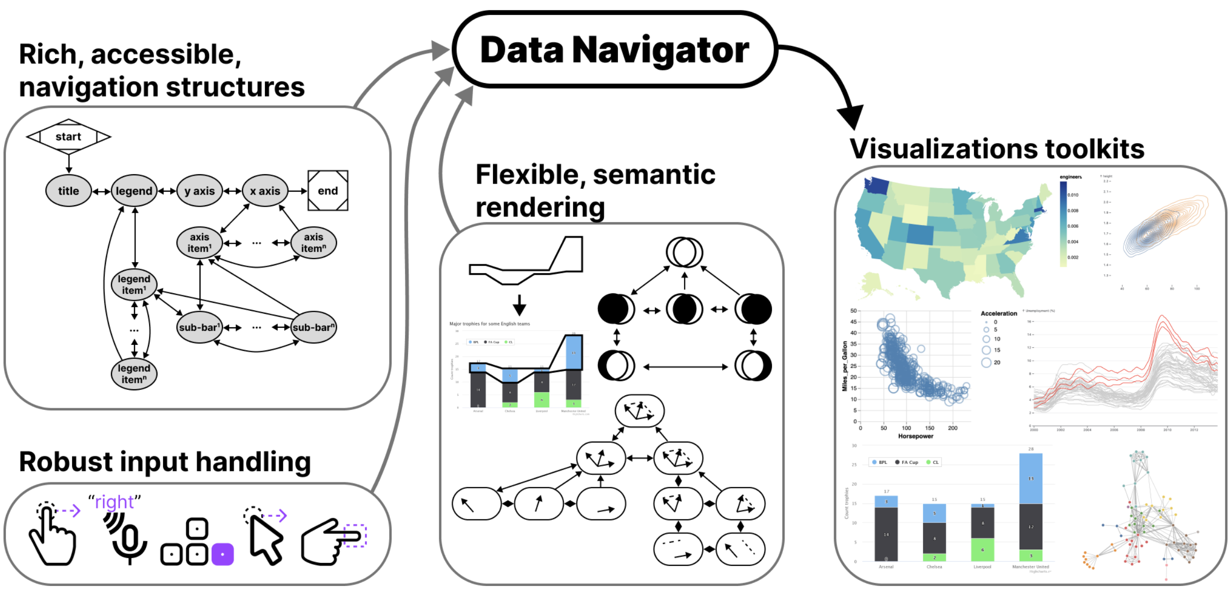


Figure 4.1: Data Navigator provides data visualization libraries and toolkits with accessible data navigation structures, robust input handling, and flexible semantic rendering capabilities.

While there is a growing interest in making data visualizations more accessible for people with disabilities, current toolkit and practitioner efforts have not risen to the challenge at scale. Major data visualization tools and ecosystems predominantly produce inaccessible artifacts for many users with disabilities. We believe this is largely a gap caused by a lack of underlying structure in most visualizations, failure to engage the input modalities used by people with disabilities, and over-reliance on visual-only rendering practices.

Users who are blind or low vision commonly use screen readers and users with motor and dexterity disabilities often do not use "pointer" (precise mouse and touch) based input technology when interacting with digital interfaces. Many users with motor and dexterity disabilities use discrete navigation controls, either sequentially using keyboard-like input, or directly using voice or text commands.

Most interactive visualizations simply focus on pointer-based input: they can be clicked or tapped, hovered, and selected in order to perform analytical tasks. This excludes non-pointer input technologies. These devices require consideration for the navigation structure and underlying semantics of a visual interface.

However, building navigable spatial and relational interfaces is a difficult task with current resources.

Raster images, arguably the most common format for creating and disseminating data visualizations, currently cannot be made into navigable structures. These are only described using alt text, which limits their usefulness to screen reader users.

Unfortunately, more accessible rendering formats like SVG with ARIA (accessible rich internet applications) properties are more resource intensive than raster approaches, like WebGL-powered HTML canvas or pre-rendered PNG files. SVG puts a burden on low-bandwidth users and a ceiling on how many data points can be rendered in memory.

In addition, ARIA itself has 2 major limitations. First, when added to interface elements, ARIA only provides *screen reader* access, which means that developers must build a solution from scratch for other navigation input modalities. Second, ARIA's linear navigation structure can be time-consuming for screen reader users if a visualization has many elements. This may impede how essential insights and relationships are understood [86, 132, 211, 224, 236, 277].

Some emerging approaches have sought to address this serial limitation of data navigation and provide richer experiences for screen reader users [86, 224, 236, 277]. However, these approaches rely on a tree-based navigation structure which is often not an appropriate choice for visualizations of relational, spatial, diagrammatic, or geographic data. Many visualization structures are currently unaddressed.

Zong et al. stress that in order to realize richer, more accessible data visualizations, the responsibility must be shared by "toolkit makers," the practitioners who design, build, and maintain visualization authoring technologies [277]. Our contribution is towards that aim, to make more accessible data experiences easier to design and implement within existing visualization work.

We present Data Navigator. Data Navigator is a toolkit built on a graph data structure, within which a broad array of common data structures can be expressed (including list, tree, graph, relational, spatial, diagrammatic, and geographic structures). Data Navigator also exposes an interface that supports interactions via screen reader, keyboard, gesture-based touch, motion gesture, voice, as well as fabricated and DIY input modalities. Data Navigator provides expressive structure and semantic rendering capabilities as well as the ability for developers to use their own, preferred method of rendering.

Data Navigator builds upon human-studies motivated work on accessible navigation [236, 277] towards a more generalizable resource for visualization practitioners. We contribute a high-level system design for our node-edge graph-based solution as well as an implementation of this system on the web, using JavaScript, HTML, and CSS. Through our case examples we also demonstrate that our generalized approach is suitable for replication of existing best practices from other systems, integration into existing visualization toolkit ecosystems, and development of novel prototypes for accessible navigation. We illustrate how Data Navigator's use of generic edges, dynamic navigation rules, and loose coupling between navigation and visual encodings provides practitioners robust, expressive, control over their system designs.

4.2 Related Work

Our contribution is an attempt to bridge the gap between research and practice more effectively across broad ecosystems in order to enable deeper and more expressive accessible data navigation interfaces. Below we outline the prior research and standards that inform our project, a breakdown of existing visualization toolkit approaches to data navigation, and then accessible input device considerations.

4.2.1 Accessibility research and standards in visualization

Research and standards are both somewhat limited by a strong bias towards visual disabilities. In *Chartability*, 36 of the 50 criteria related to accessible visualization considerations involve visual disabilities [63, 71]. Marriott et al. also found that visual disability considerations are the primary focus of data visualization literature [169], leaving the barriers that many other demographics face unstudied.

However, despite the heavy focus on visual disabilities, the work that does exist in the visualization community is deeply valuable and serves as an important starting point for our technical contribution.

4.2.1.1 Accessible navigation design considerations

Zong et al.'s research, which was conducted as in-depth co-design work and validated in usability studies involving blind participants, presented a design space for accessible, rich screen reader navigation of data visualizations. They organized their design space into *structure, navigation,* and *description* considerations and demonstrated example *structural*, *spatial*, and *direct* tree-based approaches [277].

Chart Reader also engaged these design space considerations in their co-design work on accessible data navigation structures [236]. We consider these design dimensions as the best starting point for our work, bridging the gap between research and toolkits.

There are additional research projects that have focused on accessible data navigation and interaction [86, 209, 213, 224]. These contributions explore a range of different interaction structures, including lists, trees, and tables of information as well as direct access methods such as voice interface commands and simple, pre-determined questions.

4.2.1.2 Accessible visualization: understanding users

A wide array of emerging research projects investigate screen reader users needs, barriers, and preferences, and offer guidelines, models, and considerations for creating accessible data visualizations [41, 71, 153, 211]. Jung et al. offer guidance to consider the order of information in textual descriptions and during navigation [132]. Kim et al. collected screen reader users' questions when interacting with data visualizations, which could open the door for more natural language data interaction [137].

4.2.1.3 Accessibility standards and guidelines

In the space of research, there has been a growing interest in developing guidelines for practitioners [59, 63] and even applying guidelines as a method of validation alongside human studies evaluations and co-design [71, 153, 156, 277]. Unfortunately, most accessibility standards and guidelines do not explicitly engage how to structure data navigation.

Despite this, existing accessibility standards bodies like the Web Content Accessibility Guidelines do stress the importance of accurate, functional semantics in order for screen reader users to know how to interact with elements [247]. For interactive visualizations this means that button-like or link-like behavior should expressly be made using elements that are semantically buttons and links. Our system should be capable of expressing meaningful semantics to users of assistive technologies.

4.2.2 Visualization toolkits and technical work

Unfortunately while many data visualization toolkits offer some degree of accessible navigation and interaction capabilities to developers, very few toolkits currently out there offer control over the important aspects of accessible data navigation design. Replicating existing research and strategies, remediating toolkit ecosystems, and building novel prototypes are all difficult or impossible to do due to the current lack of toolkit capabilities.

Existing data visualization toolkits have 3 major limitations that we wanted to address in the design of Data Navigator:

1. **Built on visual materials**: toolkits produce either raster or SVG-based visualizations, neither of which are focused towards designing navigable, semantic structures. As a consequence, many visualizations are simply entirely inaccessible.
2. **Lacking relational expressiveness**: When data navigation *is* provided, the navigation is based on either a tree or list structure (see Figure 4.2). The consequence of this limitation is that many other non-list and non-tree data relationships become difficult or impossible to represent without overly tedious navigation or inefficient architecture.
3. **Designed only for screen reader interaction**: When *accessible* data navigation is provided, it is generally only made possible through SVG with ARIA (Accessible Rich Internet Application) attributes. ARIA is primarily only leveraged by screen readers [248]. If a data element can be clicked and performs some form of function, only direct pointer (mouse and touch) and screen reader users are included. The consequence of this is that a wide array of other input devices, many used as assistive technologies by people with motor and dexterity disabilities, are excluded.

4.2.2.1 Rich, tree-based approaches

De-coupling rendered, visual structures from meaningful and effective navigation experiences can provide richer experiences for screen reader users [277]. Prior research and industry work,

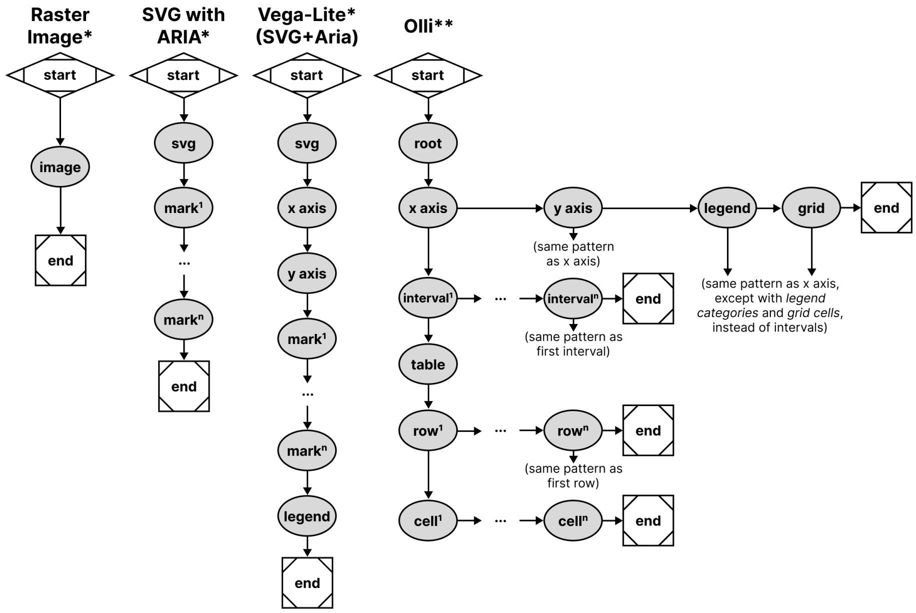


Figure 4.2: Existing accessibility trees and lists, shown using node-edge graph conventions. (*) Denotes only *screen reader* access. (**) Denotes *screen reader*, *keyboard-only*, and *pointer* access as well.

with the exception of the *Visa Chart Components* library [242], has relied heavily on a 1 to 1 relationship between structure (the encoded marks) and navigation. This emerging work is significant, because it paves the way for considering the design dimensions of accessible data interaction and navigation without dependence on a visually encoded space.

Olli's approach has been to build ready-to-go adaptors that automatically build multiple tree structures for a few ecosystems (*Vega*, *Vega-Lite*, and *Observable Plot*) and is entirely uncoupled from a data visualization's graphics. Their approach renders navigable tree structures *underneath* a visualization.

Other than *Olli*, *Highcharts* [111], *Visa Chart Components*, and *Progressive Accessibility Solutions'* visualization toolkits [86, 224] also primarily provide tree and list navigation structures across all of their chart types. These toolkits render their structures *upon* the visualization's graphic space. These tools also provide some degree of support for other assistive technologies and input modalities, although are limited exclusively to SVG rendering.

Unfortunately, these toolkits lack capabilities for dealing with graph, relational, spatial, diagrammatic, and geographic data structures.

4.2.2.2 Serial, list-based approaches

Toolkits like *Vega-Lite* [201] and *Observable Plot* only provide basic screen reader support through ARIA attributes when visualizations are rendered using SVG. These libraries do not currently provide additional access to other assistive technologies and input modalities.

Microsoft's *PowerBI* largely uses a serial structure, although it has tree-like elements as well. *PowerBI* generally provides the same access to keyboard users as it does to screen readers, although not completely.

4.2.2.3 No navigation provided

Other visualization tools, like *ggplot2* or *Datawrapper*, *Tableau*, as well as both *Vega-Lite* and *Highcharts* (when rendering to canvas), produce raster images and have no navigable structure available. Raster, or pixel-based graphics have been an accessibility burden since the early days of graphical user interface development [23]. Practitioners who use these toolkits can only provide alternative text.

4.2.3 Considering assistive technologies and input devices

Modern data visualizations may contain functional capabilities such as the ability to hover, click, select, drag, or perform some analytical tasks over the elements of the visualization space [201]. Virtually all of these analytical capabilities are designed for use with a mouse.

Input device consideration can roughly be organized as either *pointer-based* (such as a mouse or direct touch) or *non-pointer based* (which may employ speech recognition or sequential, discrete navigation such as with a keyboard). Assistive pointer-based devices, such as a head-mounted touch stylus, can typically perform any actions that a mouse can and are therefore served by current interactive visualizations. However, assistive non-pointer devices, such as a tongue, foot, or breath-operated switch, are not.

By only providing pointer-based interactivity, modern interactive visualizations exclude users who leverage non-pointer based input, who are most commonly people with motor and dexterity disabilities. And unfortunately, there is a complete lack of engagement with these populations in the data visualization research community [169].

By comparison, the broader accessibility and HCI research communities have rich engagement with interaction and assistive technologies for users with motor and dexterity disabilities. Most research either focuses broadly on physical peripheral devices or sensors [219], wearables [199], or DIY making and fabrication [118].

The DIY making space involves a broad spectrum of complex input devices and materials, such as fabricating with wood and sensors for children with disabilities [149], 3D printed materials for rehabilitation professionals [84], and even using produce-based input (such as bananas and cucumbers) for aging populations [194].

Broadly, both research and practical developments related to accessible, non-pointer input are much further ahead than data visualization research and practice. Our goal for Data Navigator is to provide a technical resource towards engaging this under-addressed space.

4.3 Data Navigator: System Design

We categorized our system design goals into design considerations for *Structure*, *Input*, and *Rendering*:

1. **Generic structure and navigation specification**: Human studies work has validated that lists, tables, trees, and even pseudo-treelike and direct structure types are all valuable to

users in different contexts and with different considerations. Our system must be able to work with all of these as well as less frequently-used structures (spatial, relational, geographic, graph, and diagrammatic).
2. **Robust input handling**: Blind and low vision users may use combinations of different assistive technologies, such as magnifiers, voice interfaces, and screen readers. Users with motor impairments may rely on voice, gesture, eye-tracking, keyboard-interface peripherals (like sip-and-puffs or switches), or fabricated devices. Both the developer and user should therefore be able to leverage and customize a broad range of input types, including the above as well as fabricated, adaptive, and future input modalities.
3. **Flexible rendering and semantics**: Visuals may or may not be necessary to render to demonstrate Data Navigator's structure. In addition, much of the latest research has shown that different screen reader users may prefer different orders of information and at different levels of verbosity. In addition, the context of tasks the user is performing as well as the nature of the data itself may influence the design of semantic descriptions and visual indications for elements. Data Navigator must provide a high degree of flexibility and control.

To help bridge the gaps between research and standards knowledge about best practices and building an effective toolkit for practitioners, we intend for Data Navigator to provide both *exploratory support* and *vocabulary correspondence* [159].

In particular, our ideal users are developers who specify data visualizations using code. To that aim, we intend to provide *exploratory support* through generic, dynamic, and flexible system design decisions. Our system is expressive and customizable, which encourages exploration of different options.

And we also want the API to include properties that have conceptual and *vocabulary correspondence* to our design considerations. Each design consideration (*Structure*, *Input*, and *Rendering*) are separately composable, modular subsystems of Data Navigator that can be used independently or in tandem with one another.

In this paper we present an implementation of our system using JavaScript, HTML, and CSS on the web. The demonstration of our system is best suited to the web due to the nature of existing, accessible building blocks (HTML), which resolve many of the semantic complexities and logic involved in enabling screen readers to programmatically navigate and announce meaningful information to users. In addition, many existing visualization toolkits target the web as an output platform and we believe that this is the best starting point for adoption and use of Data Navigator. However, this system design could be implemented as a toolkit in other environments with proper consideration for input device handling and screen reader semantics.

4.3.1 Structure

4.3.1.1 Beyond trees: towards an accessibility *graph*

The first major contribution in the design of Data Navigator is to use node-edge data as the substrate for our navigation system.

The most important argument in favor of using a graph-based approach is that a graph can construct virtually any other data structure type (see Figure 4.2), including list, table, tree, spatial, geographic, and diagrammatic structures. Graphs are generic, which enables them to represent structures both in current and future interface practices [81].

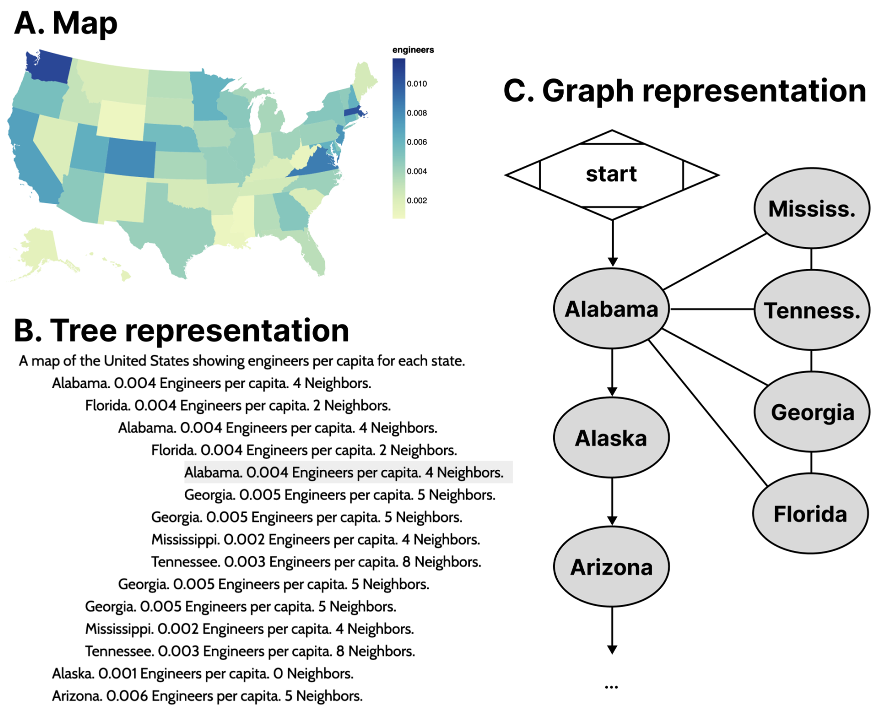


Figure 4.3: **A.** Map of engineers per capita of US states. **B.** Tree representation of the map data where states are listed alphabetically and also include links to neighboring states. The structure repeats itself if users navigate in a loop. **C.** Graph representation with the same navigation potential without redundant rendering.

To demonstrate our point, the most recent emerging work with advancements in accessible data navigation used node-edge diagrams to demonstrate their tree-like structures [236, 277] similar to Figure 4.2, Figure 4.9, and Figure 4.10. This is because trees are a form of node-edge graph, but with a root, siblings, parents, and children as sub-types of nodes that generally have rules for how they relate to one another.

Node-edge graph structures prioritize direct relationships. Examples of common direct relationships in visualization are boundaries on maps (see Figure 4.3), flows and cycles, data with multiple high level tree structures pointing to the same child datasets (such as *Olli* in Figure 4.2), or even just in diagrammatic, graph-based visualizations.

A graph structure allows for direct access between information elements that are not just part of the input data or 1:1 rendered elements, but may also have perceptual or human-attributed meaning. Examples of this might include semantic or task-based relationships, such as navigating to annotations or callouts, between visual-analytic features like trends, comparisons, or outliers. Spatial layouts such as intersections of sets or parallel vectors (see Section 4.4.3), or even relationships to information outside of a visualization and back into it (like in Figure 4.7) are enabled by a graph structure.

4.3.1.2 Graph structures are more computationally efficient

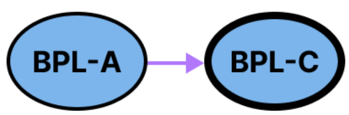
Data visualizations often portray information that becomes difficult to handle when using trees and lists. The distance users must travel between relational elements is significant in lists while redundancy when navigating relational elements in trees can be problematic.

As an example of this, often a data table or list of locations are used in conjunction to a map, such as listing all 50 states alphabetically along with relevant information. The list itself is expensive to navigate and may not provide any relationship information about which states border others, let alone ways to easily and directly access those states.

Part of the visual design justification of using a map instead of a table is for sighted individuals to understand how geospatial information may interact with a given variable. The spatial relationships matter. But when supplementing the list of states with sub-lists for each state's bordering states (see Figure 4.3), it produces redundancy in the rendered result. The rendered data contains circular connections between nodes but must render every reference, producing a computational resource creep and cluttered user experience that can be difficult to exit.

4.3.1.3 Specific edge instances and generic edges

```
// using movement
dn.move('right')
```




```
// an edge instance
'bpla-bplc': {
  source: 'bpla',
  target: 'bplc',
  navRules: ['right', ...]
}
```

```
// a navigation rule
right: {
  key: 'ArrowRight',
  direction: 'target'
}
```

Figure 4.4: An example of how a single edge instance references a navigation rule and can even have multiple navigation rules. A navigation rule can be referenced by multiple edges.

In Data Navigator, nodes are *objects* that always contain a set of edges, where each edge contains a minimum of 4 pieces of information: a unique identifier, a source, a target, and navigation rules. These properties are only accessed when a navigation event occurs on a node with an edge that contains a reference to a rule for that navigation event. Navigation rules may be unique to an edge instance or shared among other edge instances.

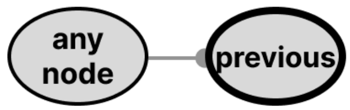
The source and target properties of edges are either ids that reference node instances (see Figure 4.4) or *functions* (see Figure 4.5). Because some edges in a graph may be directed or not, non-directed graphs can use source and target properties to arbitrarily refer to either node attached to an edge.

Generic functions for source or target properties can link nodes to other nodes based on changing content, structure, or behavior that may be difficult or impossible to determine before a user navigates the structure.

Function calling also allows some edges to be *purely* generic. An example of a reasonable use case of a purely generic edge is in Figure 4.5, where the source is a function which returns the present node and the target is whichever node the user was on previously. This single edge may then be part of every node's set of edges, enabling users to have a simple *undo* navigation control without creating an *undo* edge unique to every source node.

Using this pattern, it is possible to have fully navigable structures using only generic edges.

```
// using movement
dn.move('previous position')
```




```
// a generic edge
'any-return': {
  source: (_d, current, _p) => current,
  target: (_d, _c, previous) => previous,
  navRules: ['previous position', ...]
}
```

```
// a navigation rule
'previous position': {
  key: 'Period',
  direction: 'target'
}
```

Figure 4.5: A generic edge, such as "any-return" can be applied to any node. Function calls handle dynamically assigning the edge's source and target nodes on-demand.

4.3.2 Input

4.3.2.1 Abstracted navigation facilitates agnostic input

Navigation rules in Data Navigator (see Figure 4.4 and Figure 4.5) are created alongside the node-edge structure. Edges reference rules for navigation. However, these rules are generic and agnostic to the specifics of input modalities and can be invoked as methods by virtually any detected user input event (see Figure 4.6).

Navigation rules are objects with a unique name, ideally as a noun or verb in natural language that refers to a direction or location, a movement direction (a binary used to determine moving towards the source or target of an edge), and optionally any known user inputs that activate that navigation, such as a keyboard keypress event name.

It is important for a system to abstract navigation events so that inputs can be uncoupled from the logic of Data Navigator. This allows higher level software or hardware logic to handle input validation while Data Navigator is just responsible for acting on validated input.

Later in our first case example (Section 4.4.1), we demonstrate an application that handles screen reader, keyboard, mouse and touch (pointer) swiping, hand gestures, typed text, and speech recognition input. Abstract navigation namespaces can be called by any of these input methods.

Additionally, since navigation rules are flexible, end users can also supply their own key-bind remapping preferences or input validation rules if developers provide them with an interface.

Because calling a navigation method is abstract, users can even supply events from their own input modalities as long they have access to either a text input interface or access to Data

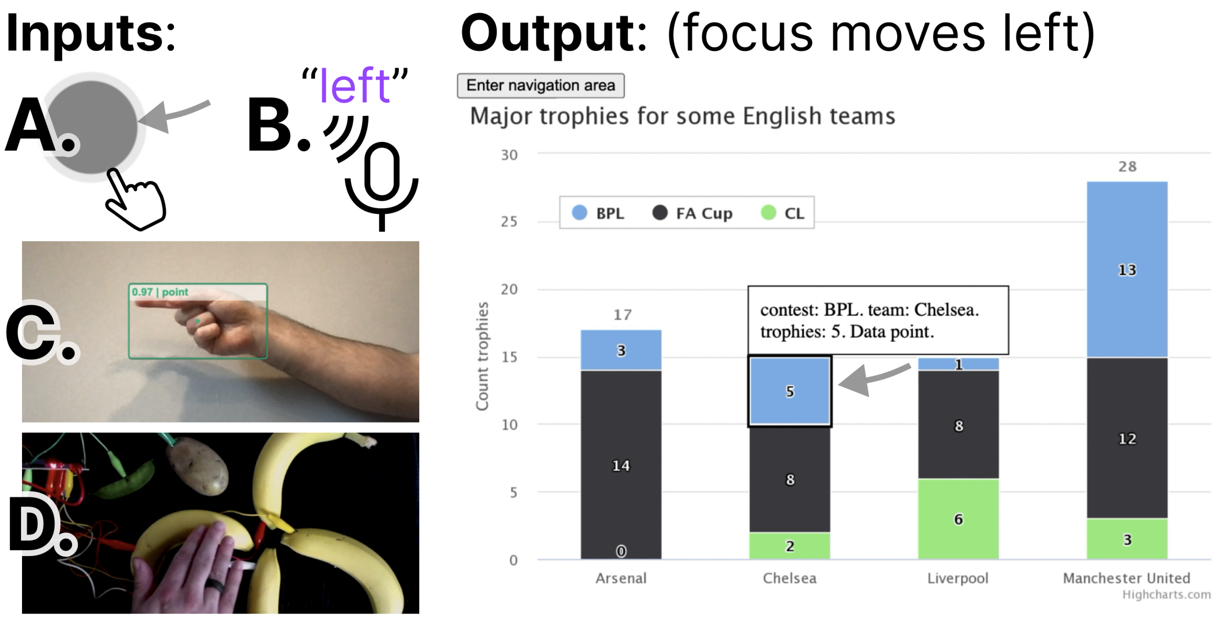


Figure 4.6: An example navigation rule to move "left" can be called as a method by an event from any input modality. Some examples include common modalities such as touch swiping (**A**) or speaking "left" (**B**). This also includes advanced or future modalities such as gesture recognition (**C**) or touch-activated, fabricated interfaces (**D**).

Navigator's navigation methods. Our demonstration material (in Section 4.4.1) also includes handling for DIY fabricated interfaces, which are important in accessibility maker spaces. We chose a produce-based interface [194], since it was an easy and low cost proof of concept.

We believe that enabling agnostic input provides a rich space for future research projects. In addition, browser addons and assistive technologies could both leverage this flexible interface for end users.

4.3.2.2 Discrete, sequential input opens new avenues

The *keyboard interface* is considered foundational for many assistive devices, which leverage this technology for discrete, sequential, non-pointer navigation and interaction [249]. Desktop screen readers are the most common example of an assistive technology device that leverages the keyboard interface, however single or limited button switches, sip-and-puff devices, on-screen keyboards, and many refreshable braille displays do as well. Support for the keyboard interface by default in turn provides all discrete, sequential input devices with access as well.

However by basing Data Navigator's foundational infrastructure on a keyboard-like modality, this also provides designers and developers new avenues to imagine how existing direct, pointer-based, or continuous inputs can map to discrete, sequential navigation experiences.

For example, with mobile screen readers this already happens: screen reader users swipe and tap on their screen to sequentially navigate, but the exact pixel locations of their swiping and tapping generally does not matter. Their current focus position is discrete and determined by the screen reader software.

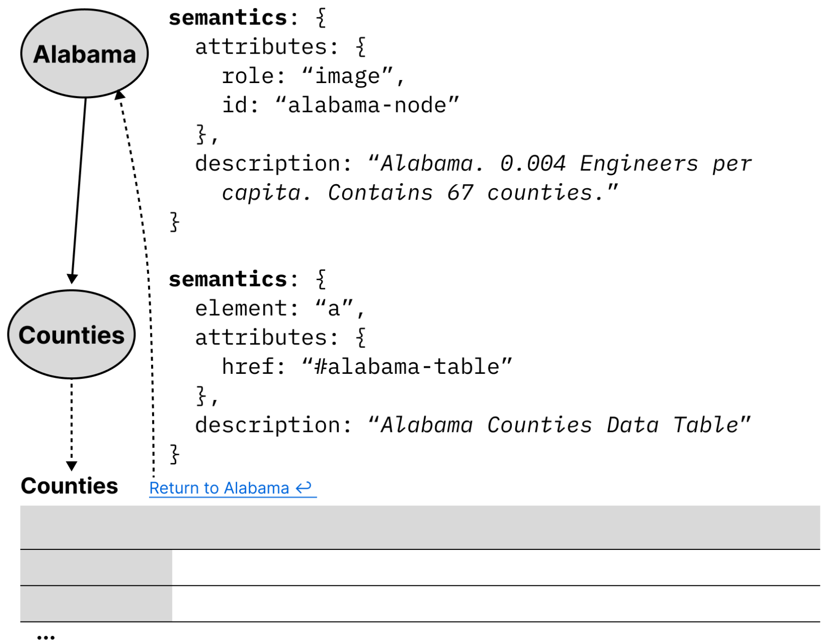


Figure 4.7: An example of how navigation within Data Navigator could use semantic nodes as hyperlinks to provide access to other areas in an application. Alabama has a child node “Counties” which is a semantic HTML link element pointing to a table of counties, outside of Data Navigator’s graph structure. A link is provided to return.

Data Navigator therefore allows for many new possibilities. One possibility is that sighted mouse and touch users may now also swipe their way through dense plots or use small interfaces (such as on mobile devices) that may otherwise be too hard to precisely tap. Data Navigator optionally removes the accessibility barriers sometimes posed by precision-based input in visualizations.

Data Navigator does not have to be in conflict with precision-based input, either. A discrete, sequential navigation infrastructure can be used in tandem with precision-based pointer events as well as instant access when coupled with voice commands and search features.

4.3.3 Rendering

4.3.3.1 Flexible node semantics provide freedom

Nodes in Data Navigator are semantically flexible. This is because the marks in a data visualization may represent many things, that are either dependent on the data or the user interface materials.

Since our toolkit implementation is in JavaScript and HTML, our map example from Figure 4.3 might use image semantics for states, alongside a description of the data relevant to that node. However since semantics are flexible in this way, Data Navigator could also be used to integrate into a larger ecosystem, with nodes rendered as hyperlinks to tables or other elements such as in Figure 4.7.

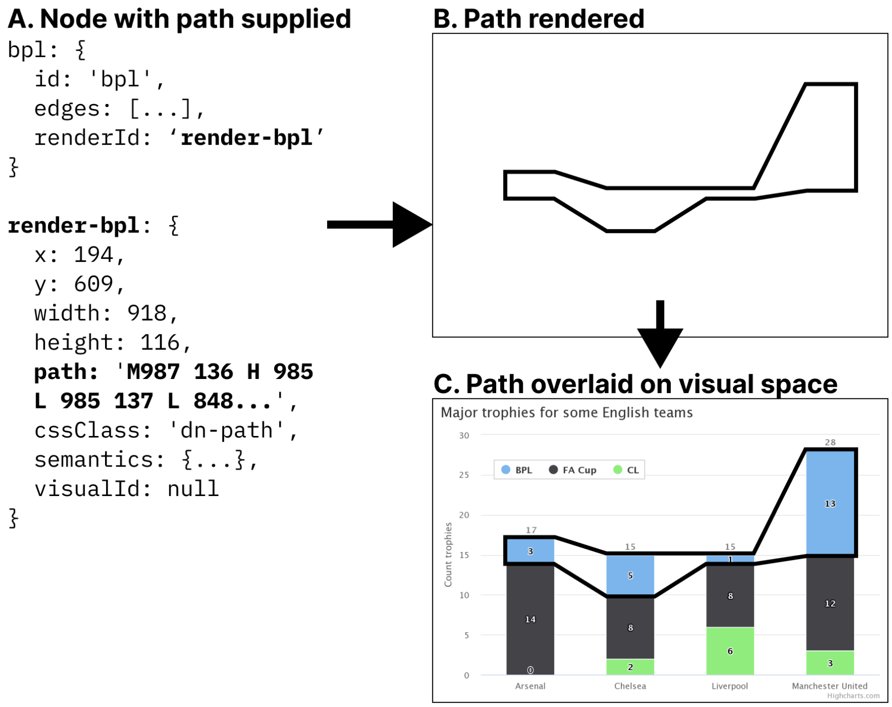


Figure 4.8: **A.** The data specified for a node with a reference to separate data that is used to render that node. **B.** The node will render as a path at the specified Cartesian coordinates. **C.** This rendered node may then be placed over a visual.

The concept of using node-edge graphs can even extend to have "nodes" that are entirely different parts of a document or tool, as well as integrated into the explicit structure provided by Data Navigator. In some accessibility toolkits, nodes are geometries without functional semantics [201] or list items nested within lists [19]. But in Data Navigator, nodes can semantically be buttons, links, or any HTML element. Interactive data visualizations sometimes demand more flexible node semantics than geometries or lists.

4.3.3.2 Loose-coupling to visuals enables expressiveness

One of the most significant technical limitations of existing data visualization toolkits with regards to accessibility is that they rely on visual substrate, or visual materials, in order to produce data visualizations. In the case of static, raster images such as png files or WebGL and canvas elements on the web, there are no interface properties at all exposed to screen readers for programmatic exploration and interaction.

If raster images are used, they generally cannot be changed after rendering. However, according to web accessibility standards, elements must have a visual indicator provided when focused [246].

Since Data Navigator navigates using focus, an indicator must be rendered alongside the node semantics. But *what* is focused visually and *where* it is depends on different design needs.

In *Visa Chart Components*, chart elements can be *selected*, so the focus indication is visible

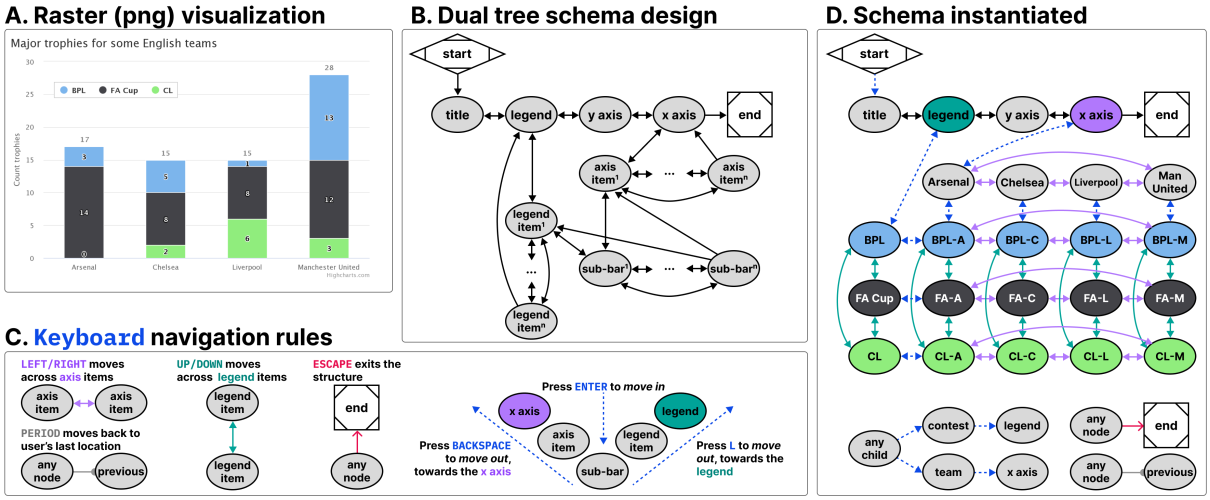


Figure 4.9: **A.** A raster (png) visualization of a stacked bar chart showing how 4 English teams performed across 3 major trophy contests. **B.** An example navigation schema that allows children nodes to have 2 parents (two tree structures intersecting), one for contests and one for teams. **C.** An example of Data Navigator's navigation logic abstraction, which allows edge types to have programmatic sources, targets, and rules, such as a single rule that gives all nodes a edge to exit the visualization. **D.** An instantiation of the schema, showing all corresponding rendered nodes and their edge types according to the schema design and navigation rules.

over the existing elements in the chart space. The design choice to have interactive visual elements located within a chart or graph is also common in other toolkits that provide accessible focus indication, such as *Highcharts*, *PowerBI*, and *SAS Graphics Accelerator*.

However, some visualization toolkits create accessible structures entirely uncoupled from visual space [19], so focus indication is provided beneath or beside the chart, not over it.

Due to the different ways that accessibility might be provided, Data Navigator enables developers to have complete control over the rendering of which focus elements they want, in what styling, and where. This can accommodate both un-coupled and visually-coupled approaches to focusing and more.

Data Navigator's focus is *uncoupled* by default and may even be used independent of any existing graphics at all. Rendering information may be passed to Data Navigator for it to render (like in Figure 4.8) or developers can provide their own rendered elements and simply use Data Navigator to move between them.

Because of Data Navigator's approach to rendering focusable elements, designers and developers can provide fully customized annotations, graphics, text, or marks that may not be not part of the original visual space or elements. One example of this might be adding an outlined path to a collective cross-stack group of bars in a stacked bar chart (see Figure 4.8).

Loose-coupling in this way provides robust flexibility to designers and developers to handle navigation paths and stories through a data visualization, even in bespoke or hand-crafted ways.

4.3.3.3 On-demand node rendering is efficient

Practitioners care about performance and so do users. Practitioner toolkits often focus on lazy-loading techniques where accessibility elements are rendered on-demand rather than all in-memory up front [19, 60, 277].

Data Navigator's nodes are rendered *on-demand* by default. Data Navigator only renders the node that is about to be focused by the user and after it is focused, the previously focused node is deleted from memory. This technique has advantages in cases where datasets are large or users have lower computational bandwidth available. However, there are cases where practitioners may want to render all of Data Navigator's structure in memory, such as server-side rendering or equivalent. Pre-rendering may be optionally enabled.

4.4 Case Examples with Data Navigator

We built example prototypes using our JavaScript implementation of Data Navigator, available open source at our GitHub repository.

Our first two prototype case examples represent some of the most powerful parts of Data Navigator as a system while reproducing known and effective data navigation patterns from existing industry and research projects. We provide a final case example as a co-design session that demonstrates how Data Navigator may be used to rapidly build new designs.

4.4.1 Augmenting a Static, Raster Visualization

The first case example (shown in Figure 4.9) builds on an online JavaScript visualization library, *Highcharts*. *Highcharts* already provides relatively robust data navigation handling out of the box for screen reader, keyboard, and even voice recognition interface technologies, such as *Dragon Naturally Speaking*. However, these capabilities are only provided when the chart is rendered using SVG. Developers have several other rendering options available, including WebGL, which is significantly more efficient [113]. We wanted to demonstrate that Data Navigator can provide a navigable data structure even if the underlying visualization is a raster image.

For our case example, we exported a png file using the built in menu of a sample stacked bar chart retrieved from their online demos [110]. We selected a stacked bar chart because it allows us to demonstrate how two tree structures may interact and share the same children nodes.

We recorded the data and hand-created all of the geometries and their spatial coordinates using *Figma*, by tracing lines over the raster image's geometries (see samples of the data and traced geometries in Figure 4.8). While this method was efficient for building an initial prototype, Section 4.4.2 engages deterministic methods for extracting and producing the nodes, edges, and descriptions required by Data Navigator automatically and at scale.

The visualization we selected represents 4 English football teams, *Arsenal, Chelsea, Liverpool,* and *Manchester United* and how many trophies they won across 3 contests, *BPL, FA Cup,* and *CL*.

We chose a schema design that arranged the *contests* to be navigable across one dimension of movement (*up* and *down*) while the *teams* are navigable across a perpendicular dimension of movement (*left* and *right*). This 2-axis style of navigation is used by *Highcharts* (when rendering as SVG) and *Visa Chart Components*. We also chose these directions because it is coincidental that their visual affordance is closely coupled with the navigation design (the x axis is ordered *left* to *right* and since the bars are stacked, *up* and *down* can move within the stack). These directions can also be applied to the axis categories and legend categories as well, moving *left* and *right* across the entire *team's* stacks or up and down across the entire *contest's* groupings.

Using a keyboard, a user might enter this schema and navigate to the legend, where they could press *Enter* to then focus the legend's first child, pictured in Figure 4.8. Pressing up or down navigates in a circular fashion among the *contest* groupings. Pressing *Enter* again then focuses the first child element of that *contest*, all of which are in the *Arsenal* group, since it is the first group along the x axis. A user can then navigate *up, down, left,* and *right* among children. Pressing *L Key* moves the user back up towards the contest while pressing *Backspace* moves the user up towards the x axis. The x axis and *team* groupings represent the second tree which intersects the first (the *contests*).

Our first case example includes handling for additional input modalities beyond screen readers and keyboards, including a hand gesture recognition model, swipe-based touch navigation, and text input (which can be controlled using voice recognition software).

4.4.1.1 Discussion

Our first case example demonstrates several of the most important capabilities of Data Navigator, namely that practitioners can add accessible navigation to previously inaccessible, static, raster image formats and that a wide variety of input modalities are supported easily.

Widely-used toolkits like *Vega-Lite*, *Highcharts*, and *D3* [21] allow practitioners to choose SVG and canvas-based rendering methods. Data Navigator's affordances help overcome the lack of semantic structure in canvas-based rendering, allowing developers to take advantage of its processing and memory efficiency.

Notably in addition to these capabilities, the visual focus highlighting added was entirely bespoke (as in Figure 4.8) and the navigation paths through the visual were based on our design intentions, not an extracted view or underlying architecture such as render order. This demonstrates that our system provides a significant degree of freedom and control for designers and developers.

As a final discussion point, the resulting visualization contains no automatically detectable accessibility conformance failures according to the W3C's Web Accessibility Initiative's accessibility evaluation tool, *WAVE* [253]. It is important for any technology developed to also meet minimum requirements for accessibility [63, 153, 156, 277], even when following best-practices and research.

4.4.2 Building Data Navigation for a Toolkit Ecosystem

Our second case example, shown in Figure 4.10, builds on *Vega-Lite*. As shown in Figure 4.2, *Vega-Lite* offers basic screen reader navigation but provides no navigation at all when rendered using canvas.

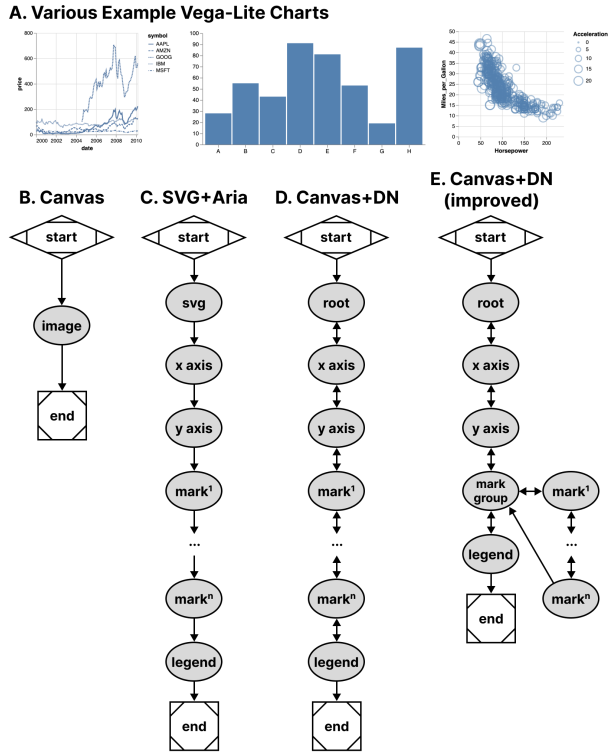


Figure 4.10: **A.** Various charts from *Vega-Lite* share the same general structures with each other when rendered using canvas (**B**) or SVG (**C**). **D.** With Data Navigator, we replicated the existing SVG navigation pattern (**C**) but used a canvas-based rendering for the visualization. **E.** We also improved the navigation scheme to nest marks within a mark group to allow users to skip them, if needed.

While it might be a tedious design choice to allow every mark in a visualization to be serially accessible to screen reader users, we nevertheless set out to build a generic ingestion function that would take a *Vega-Lite View* object and deterministically recreate their existing SVG navigation structure in Data Navigator. This way users would have the same experience between SVG rendered charts and all current and future rendering options that *Vega-Lite* offers to developers.

Notably, *Vega-Lite* does not explicitly manipulate the navigation order at all when rendering with SVG. ARIA is simply provided to allow screen reader users to access each mark in the visualization in the order the mark appears in the DOM (which is the order it was rendered). The legend appears after the marks in our schema for this reason because *Vega-Lite* renders the legend after marks. This choice of ordering is for visual reasons: z-axis placement is currently based on render order in SVG and *Vega-Lite* wants their legend visually on top of the rendered marks.

In addition to mimicking their existing SVG navigation strategy, we also created a way to nest all of the marks within a group so that users can skip past them and drill in on-demand, which is a valuable pattern when dealing with situations where providing a mark-level fidelity of information may not be relevant to a user's needs by default [217, 277].

In order to deterministically supply Data Navigator with accurate information about any given

Vega-Lite visualization, we built 3 functions: one that takes a *Vega-Lite View* as input and extracts meaningful nodes, one that produces edges based on those nodes, and one to describe our nodes in a meaningful way for screen reader users. These generic functions technically work on all existing *Vega-Lite* charts, however some are more useful out of the box than others due to the type of marks involved.

4.4.2.1 Discussion

This case example demonstrates that ecosystem-level remediation and customization is not only possible for toolkit builders but Data Navigator offers robust potential. Data Navigator's structure, input, and rendering capabilities are all flexible and can be adjusted to suit the needs of a specific toolkit's design and intended use.

Many visualization libraries may not even provide screen reader accessible SVG using ARIA-based approaches but do have a consistent underlying architectural pattern. Some libraries have a consistent method for converting data into visual formats, readable text labels, and interaction logic. Strong contenders would be visualization libraries popular in online, web-based data science notebooks like *ggplot2* in R or *matplotlib* for Python, which typically only render rasterized pngs or semanticless SVG.

Toolkits with consistent underlying architecture would allow toolkit developers, not just developers who *use* toolkits, to remediate and customize their navigation accessibility using a generic approach.

Enabling accessibility at the toolkit level allows all downstream use of that tool to have better defaults, options, and resources available for building more accessible outcomes for end users.

Many libraries and toolkits provide users with a level of functional defaults and abstract conciseness so that users don't have to worry about low-level geometric considerations [201].

Data Navigator allows toolkits developers to also provide their users with abstractions and defaults for accessibility that make sense for their ecosystem.

Despite our schema recreating a screen reader experience based on SVG (and improving it), Data Navigator's additional features also apply: users are able to leverage a much wider array of input modalities.

Vega-Lite provides many ways to make marks clickable and even perform complex actions using mouse-based input. While Data Navigator does not engage accessible brush and drag-based inputs, it does provide keyboard-only access by default, which can be used to make events previously only accessible to mouse clicking available to many other technologies. This is an improvement over *Vega-Lite*'s SVG + ARIA rendering option.

When measuring performance across test datasets containing 406 and 20,300 data points in a scatter plot, Data Navigator increases initialization time by $\sim$0.45 to $\sim$1.5ms respectively. Our extraction functions specific to *Vega-Lite* increase initialization between $\sim$4.8 and $\sim$8.5ms respectively. Given that our benchmark testing for *Vega-Lite's* SVG rendering initialized in $\sim$1,800ms for 20,300 data points and canvas in $\sim$700ms, we do not anticipate that Data Navigator will have a negative impact on performance in most visualization contexts.

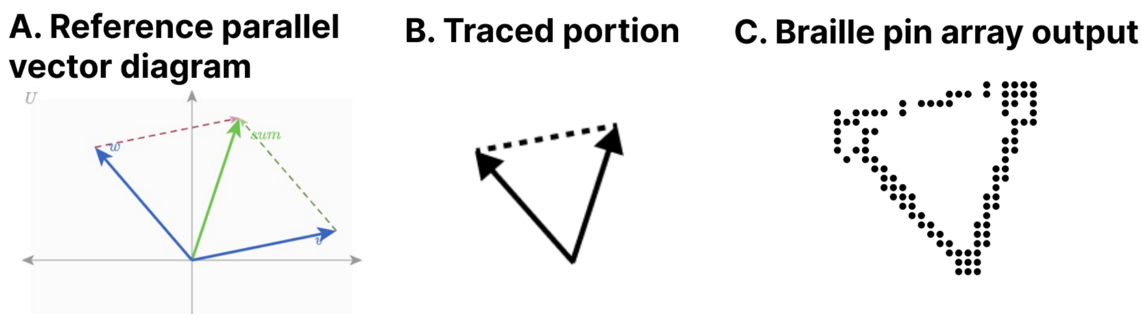


Figure 4.11: Our material preparation process involved taking a reference (**A**), tracing it (**B**), and rendering it on a tactile display (**C**).

4.4.3 Co-designing Novel Data Navigation Prototypes

Recent projects in accessible data navigation have involved extensive co-design work with people with disabilities, ranging on the magnitude of months with as many as 10 co-designers at a time [153, 156, 236, 277].

However many visualization experiences may be authored in smaller scales, with fewer designers, and less time such as the development of a prototype or demonstration of an emerging idea. In practical or industry contexts, co-design sessions (and design sessions in general) may be much shorter. The goal of these co-design sessions is simply to create an artifact with the artifact's intended users.

Since our paper is contribution towards practical outcomes, we simulated a light co-design session with the aim of producing low-fidelity prototypes of novel data interaction patterns.

4.4.3.1 Co-design Session Methods and Setup

Authors Frank Elavsky (sighted) and Lucas Nadolskis (blind) set out with the goal of developing screen-reader friendly prototypes that can explore geometric and mathematical models produced by the math diagramming tool *Penrose* [271].

Nadolskis is a neuroscience engineer who is a native screen reader user and uses both mathematical concepts as well as data-related tasks in his research. Elavsky proposed a series of possible math-based visualization types produced by *Penrose* to build prototypes for, and Nadolskis selected *set* and *vector* diagrams as the two worth exploring first. The justification for this selection is that understanding these two concepts is important for work in data science, programming, and more advanced math concepts.

In particular we grounded the context of our contribution in a hypothetical classroom setting, where a screen reader user who is a student will have access to the equations in both raw text and *MathJax*. We want to provide an experience that does not replace the existing resources screen reader users have to learn in classrooms but rather supplement.

At our disposal for our co-design session was a *Dot Pad* [55], which is a refreshable tactile braille display. Our *Dot Pad* enabled Elavsky to produce something visual and then translate it into the display for Nadolskis. Similar to de Greef et al. [49], we used a tactile interface as an intermediary to help us get a shared sense of the meaningful spatial features of our figures.

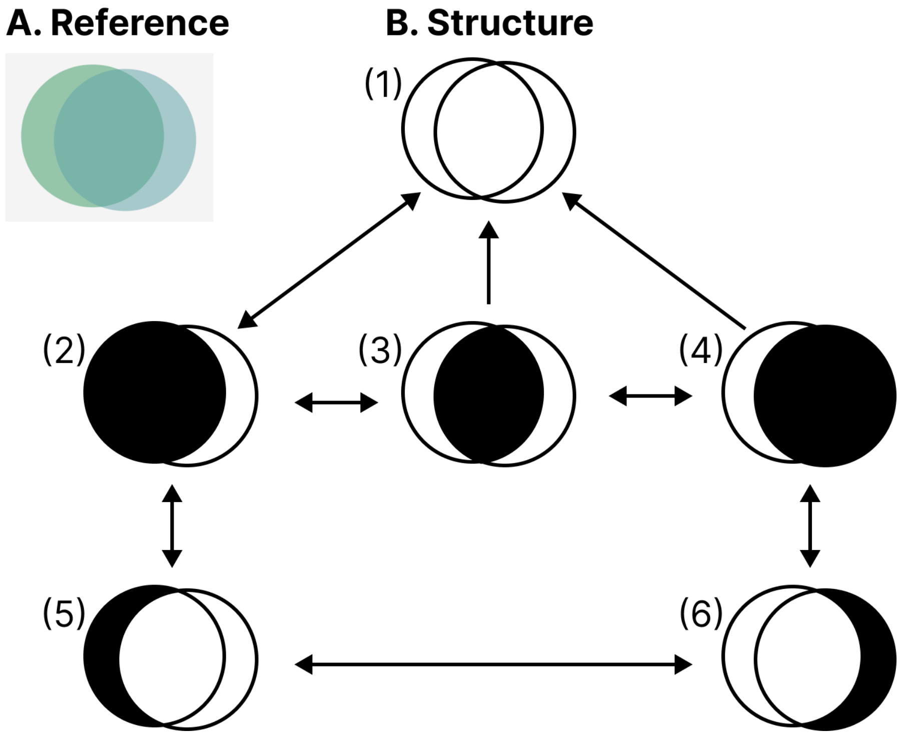


Figure 4.12: **A.** A reference image from *Penrose* of a set diagram containing two sets intersecting. **B.** A diagram of our proposed structure, with three levels of information.

Elavsky started with a reference diagram and then traced a wide variety of every possible node that might be worth navigating to in the diagram (see Figure 4.11).

We selected which nodes were most important in each diagram, how to navigate between them, and how we wanted to render their visuals and semantics.

The selection of our problem space, scope of solutions, context of contribution, general discussion, and preparation of materials took approximately 12 hours of work over 2 weeks. The exploration of our prototype design space for our 2 prototypes took 1 hour. Building the prototypes took 2 hours.

4.4.3.2 Creating a Navigable Set Diagram

Our first prototype was a set diagram (see Figure 4.12). For our structure, we decided that it has 3 important semantic levels: the high level, the inclusion level, and the exclusion level. The inclusion level is first and the siblings are all sets or subsets that include other sets. The exclusion level is beneath and contains sets or subsets that are exclusive to the sets they belong to, which are accessed by drilling down from a set.

Our schema design starts with a user encountering the root level (1) and may optionally drill in to the first child of the next level (2) using the *Enter* key. The user may navigate siblings at this level using *right* and *left* directions, but this level is not circular (like in Figure 4.9) to maintain the spatial relationships. The user may drill in on either set again to view the non-intersecting portion of that set. Any node can drill up, towards the root, using *Escape* or *Backspace*.

4.4.3.3 Creating a Navigable Parallel Vectors Diagram

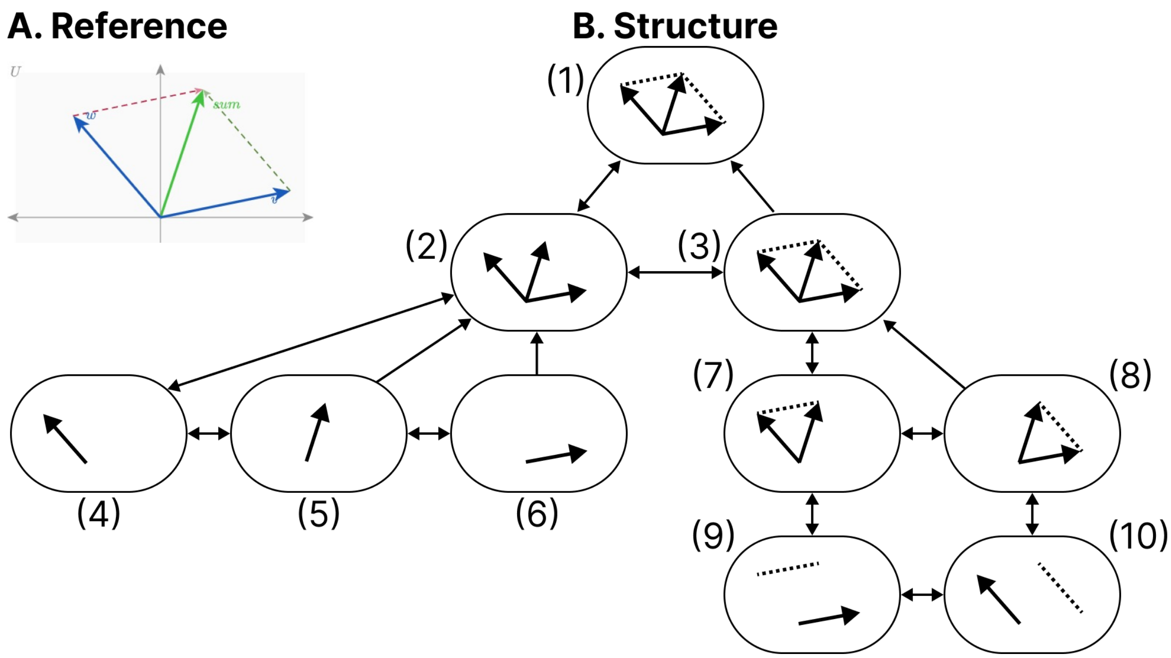


Figure 4.13: **A.** A reference image from *Penrose* of a parallel vectors diagram. **B.** A diagram of our proposed structure, with two main sub-categories of information: understanding the vectors and their parallels.

Our second prototype was a parallel vectors diagram (see Figure 4.13). For the structure of this diagram we created a first level group that contains each vector and vector sum. The sibling to this grouping is another group which organizes sub-equations related to calculating each parallel vector. The sub equations each contain children that pair the sub equation with the vector it is parallel to.

Similar to Figure 4.12, this figure maintains spatial relationships along the x dimension, does not have circular navigation, and allows drilling in and out.

4.4.3.4 Discussion

After our co-design sessions, our visual materials and navigation structures were used in the creation of functional prototypes. We additionally hand-crafted the descriptions and semantics for each node.

Accessibility work often takes a long time, from co-design to building to validation. But we believe that a well-articulated and useful design space, with tools that provide expressiveness and control over the dimensions of that design space, can improve how this work is done. The above case example demonstrates how builders who are thinking about data navigation design can rapidly scaffold prototypes for use in Data Navigator.

In particular, Data Navigator's design as a system gave our co-design sessions *vocabulary correspondence*. Data Navigator's language helped us focus on the *nodes, edges,* and *navigation* *rules* for our *structure* while we also explicitly discussed the *rendering* details of *coordinates, shapes, styling,* and *semantics* for each node. The vocabulary of our design space directly corresponded with code details required to create a functional prototype.

We note that this co-design work is not intended to contribute a *validated* set of designs. Rather, our contribution with this case example is to demonstrate that within the larger ecosystem of a research venture, Data Navigator is an improvement over designing and building navigable structures from scratch.

4.5 Limitations and Future Work

Data Navigator is a technical contribution, a system designed for appropriation [56] and adaptation [261] in different applied contexts. It is, as Louridas writes, a *technical material*: a technology that enables new and useful capabilities [152]. While beyond the scope of the current paper, a critical next step for future work is to conduct separate studies with both practitioners and end users to evaluate Data Navigator's affordances.

Unlike toolkits that provide an end-to-end development pipeline for accessible visualization, Data Navigator serves as a low-level building block or material (like concrete). As such, one potential limitation of the framework is that it can be used to build both curbs (which are inaccessible) as well as ramps and *curb-cuts* (which may be more broadly accessible).

Even when building more accessible curb-cuts, we stress the importance of actively involving people with disabilities in the design and validation of new ideas, in line with prior work [153, 156, 192, 277]. For example, while our first two case examples replicate co-designed and validated existing work, our third case example's co-designed prototypes would need to be validated with relevant stakeholders before wider implementation. Our system does not *guarantee* any sort of accessibility on its own.

The diverse array of modalities supported by Data Navigator opens an immediate line of future work in engaging people with a correspondingly diverse set of disabilities. While recent explorations into accessible data visualization have been inspiring, this trend has primarily focused on the experiences of people with visual disabilities [63, 139, 169]. More research should be conducted with other populations, particularly people who leverage assistive technologies beyond screen readers, to understand how interactive data visualizations can be better designed to serve them.

Finally, there are significant opportunities to improve the efficiency of our approach, including developing deterministic and non-deterministic methods to generate node-edge data and navigation rules from a visualization. Ma'ayan et al. stress in particular that reducing tedious complexity can contribute to the success of a well-designed toolkit [159]. Future work should identify areas where graphical interface tools or higher-level specifications can improve the experience of working with Data Navigator.

4.6 Summary

Practitioners at large continue to produce inaccessible interactive data visualizations, excluding people with disabilities. We believe that the burden of remediation first starts with the developers who build and maintain the toolkits that practitioners use.

However, the challenges faced by toolkit builders are significant. Most toolkits lack an underlying, navigable structure, support for broad input modalities used by people with disabilities, and meaningful, semantic rendering.

To engage these limitations we present Data Navigator, a technical contribution that builds on existing work towards a more generalizable accessibility-centered toolkit for creating data navigation interfaces. Data Navigator is designed for use by practitioners who both build and use existing toolkits and want a tool to make their data visualizations and interfaces more accessible.

We contribute a high-level system design for our node-edge graph-based approach that can be used to build data structures that are navigable by a wide array of assistive technologies and input modalities. Data Navigator is generic and can scaffold list, tree, graph, relational, spatial, diagrammatic, and geographic types of data structures common to data visualization.

Our system is designed to encourage both remediation of existing inaccessible systems and visualization formats as well as help scaffold the design of novel, future projects. We look forward to further research that explores the possibilities enabled by Data Navigator.

Chapter 5
Softerware: Building Tools for Accessibility and Personalization

---

This chapter was adapted from my paper, currently under review with CG&A:

> F. Elavsky, M. Vindedal, T. Gies, P. Carrington, D. Moritz, and Ø. Moseng, 'Towards *softer*ware: Enabling personalization of interactive data representations for users with disabilities', *Computer Graphics and Applications* (to appear at *IEEE VIS 2026*), 2025.

5.1 Overview

There is a significant and relatively unacknowledged problem in emerging work on accessible visualization: a single design cannot satisfy all users. People with disabilities, even those who share the same category of disability, often have different experiences, capabilities, and needs. As experienced practitioners and researchers who have been working to make data visualizations more accessible (some of us for more than a decade), we have each observed this persistent problem in our own practice.

Data visualizations that are produced by a designer for an audience tend to be designed in a way that is relatively *unchangeable*. As a material, we use the metaphor that the creator of a visualization manipulated their design while it was in a *softer* state, like clay. And eventually, the clay is *hardened* into a state that is presented to the user. Often visualization design artifacts cannot easily be altered by an end-user after they are created. Pixels cannot be moved, graphics cannot be re-embedded. The clay has been fired and the visualization is now *baked*. We intend to make visualizations easier to change and manipulate for end-users. We want a material that is softer than software. But we also don't want a fully malleable interface, like in end-user programming, either. We propose to call this space of design "softerware."

We wanted to explore material softness *with constraints*. We conjecture that advancements in malleable interfaces and end-user programming are too system-centric and open-ended. End-user programming puts too much burden on end-users to know the language and symbols of interface-building in order to build their own interfaces. Instead, we chose to enable end-users to have agency over a data visualization interface by exposing a preferences-driven menu of options built from our existing knowledge of visualization and accessibility.

Exploring the *softerware* gap in data visualization became our primary research focus, which led us to formulate the following qualitative exploratory research questions:

- **R1**: What constraints and capabilities should we provide end-users to give them meaningful agency over interactive data visualizations?

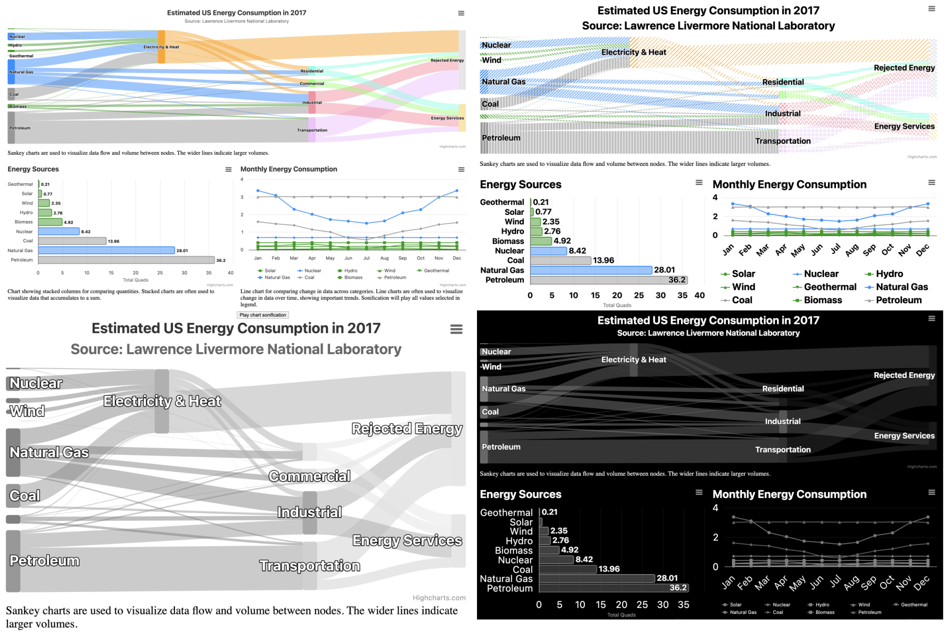


Figure 5.1: Sometimes one design is not enough. Our design (upper left) and three different designs by low vision users. All low vision users chose larger text, but then diverged: redundant-encoding enabled (upper right), high zoom and greyscale on white (bottom left), and then dark mode (enabled externally) with greyscale (bottom right).

- **R2**: What qualities, challenges, and design opportunities do designers and engineers envision for a data visualization *softerware* system?
- **R3**: What qualities, challenges, and design opportunities do blind and low vision users envision for a data visualization *softerware* system?

We contribute our findings from this research to the larger accessibility and visualization communities in hopes that we can inform and inspire future work that investigates *softerware* systems, end-user design, preferences-based user experiences, and fluid and malleable interfaces focused on end-users with disabilities. Our future work will focus on completing a finished version of our prototype and deploying it at scale within Highsoft's Highcharts ecosystem [108].

5.2 Related Work

Our contribution is an attempt to bridge the gap between the knowledge we have on accessibility for visualization (as a complex space of design and engineering) with research and practice that centers on users with disabilities being able to adust, change, or control the interfaces they interact with. We intend to frame our work towards the benefit of data visualization designers, system engineers, and end-users of data visualizations. We believe that more flexible data visualization systems that enable user preferences will require a careful approach to architecture and thorough consideration for the burdens placed on end-users.

5.2.1 Data Visualization and Accessibility

Data visualization accessibility has come far in recent years. But little work has been done to explore what disability scholars call "access friction" - a tension that arises when access must be negotiated [97, 117]. This friction is often a result of static barriers in shared spaces: one artifact or approach designed to include some people may end up excluding others.

In general, accessibility concerns itself with a broad spectrum of barriers that people with different disabilities face. And while most literature focuses on visual disabilities [168, 258], there are growing resources on areas such as cognitive/intellectual disabilities [264, 266], neurodivergence [238], and both research and systems exploring epilepsy and vestibular/motion inaccessibility in visualization [221, 226].

Yet despite these resources, making data visualizations more accessible remains a difficult task for practitioners [130, 208]. Some accessibility guidelines even conflict, for example on the topic of patterns and textures used in charts. One side stresses that patterns are harmful to cognitive and visual accessibility [203] while another stresses that redundant encoding strategies are necessary [63]. Understanding how to make the correct design decisions may sometimes be impossible. Either existing guidelines are incorrect or it is possible that access friction becomes inevitable the more we know what different barriers look like for different people with disabilities.

5.2.2 Systems that Adapt

One angle of exploration that has been engaging this issue already focuses on systems that can adapt. Work on adaptive systems for people with disabilities, such as in *ability-based design* [260], stresses the importance of design alleviating burdens placed on users. Users who don't fit initial system designs are often expected to adapt to fit the system. This means that they may have to acquire an assistive technology, learn a peripheral skill, hack the system, or wait on a design fix. This places the burden on the user to fit the system. Ability-based design instead stresses that systems should be capable of automatically adapting, in order to reduce these burdens placed on the system's users.

However, building data visualizations that automatically adapt to users via some form of data collection often do so through means such as monitoring live biometric data and input patterns, collecting a user's self-declared conditions and cognitive ability, parsing a user's history, and sensing a user's environmental or situational context [270]. We argue that these methods for an adaptive system raise questions of end-user agency, trust, privacy, and awareness in regards to the system decision-making [185]. They may not be sufficient for addressing a user's needs while also preserving their privacy and agency.

5.2.3 Personalization and Accessibility

Lastly, we researched broader spaces where users have more design agency and explicit awareness of a system that is built to be adapted. We were interested in literature and projects that explore ways end-users can enact meaningful change on an interface, with special attention paid to accessibility and disability.

One specific project has emerged at the intersection of accessibility, visualization, and customization which focuses on screen reader users adjusting the content of textual tokens when navigating data visualizations [127]. While this is excellent work, we still have larger questions about when preferences, options, and customizations are appropriate and in what contexts as well as other ways of conceptualizing end-user agency over a system. It remains unclear when, why, and how customization and personalization can be used effectively when designing a system.

In the field of meta-design, meta-designers consider these end-user manipulations of a system to be one facet of "end-user design" and "continuous co-design" between a system and a user [160], which helps give us some meaningful language to refer to our system goals.

Recent work on the influential factors for personalization and adoption of accessibility settings [265] also informs our work in 2 key ways: conceptual mismatching between a system and user can contribute to a system's under-use while features that propose value, are time-saving, or reduce cognitive load for a user can contribute to positive perception and use of personalization of a system.

5.3 Presenting: *Softerware*

Softerware is a vision for software design that is not just based on giving a user the ability to set preferences or personalize. *Softerware* is about the intentional design of a software system that enables people with disabilities to have meaningful, opinionated, and persistent agency over that system.

We contribute the concept of *softerware* to the larger community of researchers and practitioners because we argue it is a useful construct that can help us categorize past work, improve existing projects, and inspire new directions. *Softerware* systems have been part of existing work for decades, but we lack a cohesive way to refer to designing and engineering experiences that enable end-users to have agency over malleable interfaces without entering into the territory of end-user programming and end-user development.

5.3.1 Defining *Softerware*'s Principles

Here we present the principles that define *softerware* before demonstrating an example instantiation in the context of online, interactive data visualization.

5.3.1.1 Principle: Has Reasoned, User-centered Constraints

An important aspect of *softerware* is that it is *softer* than software (which is already-baked) but not quite as *malleable*, free, and potentially low-level as systems that facilitate fully realized end-user programming and development [31, 144].

End-user programming is still a form of *programming* in the end. It focuses on taking constructs, functionalities, and reasoning from software programming and development and presenting these elements to users in ways that may suit a user's natural language or mental models, such as through no-code, visual-only, or low-code approaches. We anticipate that many users, especially those experiencing accessibility barriers, will have difficulty interacting with software paradigms based on end-user programming and development.

Instead *softerware* engages this limitation through reasoned constraints that leverage conceptualizations and language focused on overcoming anticipated user barriers. Providing constraints and then framing and presenting those constraints in ways that have vocabulary correspondence to user needs is what separates *softerware* from existing work and literature on end-user programming and development.

To accomplish this, the *softerware* system designer must work to anticipate not only what their system should do in a default or beginning state but also which ways that system will potentially fall short and require fitting by the end-user. The system designer should motivate all of the capabilities of a *softerware* system based on what they anticipate users will want to change, how users can discover that change is possible, and then how best to enable users to enact that change easily.

5.3.1.2 Principle: Facilitates End-user Agency

Softerware is ultimately about the process of architecting and implementing a system that enables an end-user to be able to easily express meaningful changes to that system's appearance and behavior.

Accessibility has been framed as a tension between fit and scale [107], where *fit* refers to a system that is perfectly complimentary and synchronized to a user and *scale* refers to a system that is capable of reproducing functionality for many different users. We believe that the tension between fit and scale, in addition to *access friction*, can both be alleviated when a system is designed to facilitate end-user agency.

The cornerstone goal of a *softerware* system is an attempt to facilitate *self-fitting* at a minimum, and in ideal circumstances also facilitate social methods of sharing fitting (such as loading profiles or ingesting metadata from others).

5.3.1.3 Principle: Demonstrates Value

Existing literature makes one thing particularly clear when it comes to personalization and end-user design: it has to be worth it [265]. Users must be able to recognize barriers, issues, or shortcomings of a system and then discover and utilize capabilities provided to them to eliminate or alleviate those barriers.

This entire process must not be too burdensome and the payoff should establish an expectation that future use of the system will be improved. The time and effort it takes for a user to fix a problem should be less than the time and effort generated by that problem. This means that *softerware* systems can likely be optimized and improved significantly over time, as better techniques are developed to perform tasks as quickly and easily as possible.

The user should also be able to validate the value of their interaction with a *softerware* system through the continued use of that system. If something was a painful experience and they took action to alleviate that issue, they should be able to observe the effects easily.

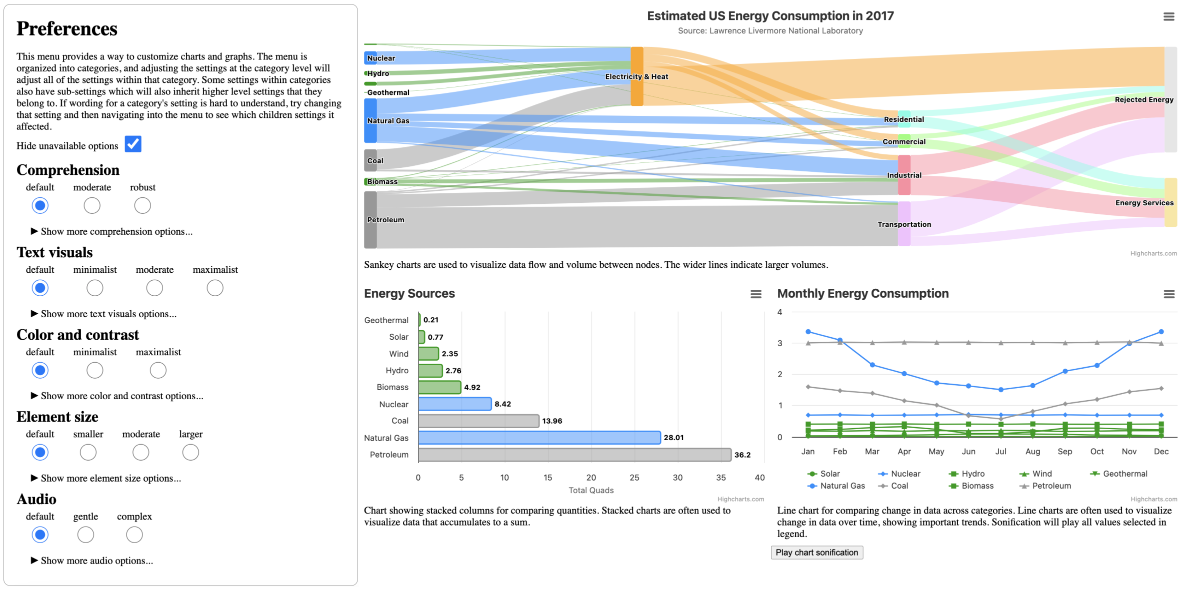


Figure 5.2: Our dashboard design on US Energy Consumption in 2017 with a sankey, bar chart, line chart, and preferences menu on the left.

5.4 Prototype: Visualization *Softerware*

We first built a data visualization dashboard (see **Figure 2**) that would allow us to build a *softerware* system prototype that we could demonstrate to designers, engineers, and evaluate with end-users with disabilities. You can view and interact with our prototype, including view our open source code and dataset on user preference options, on github.

We chose a relatively clean and standard dataset that would be relevant to our target end-users, who we would recruit from the United States, on 2017 US energy consumption. This dataset afforded us enough complexity to build multiple data visualizations in one dashboard (including an uncommon type like the sankey). This choice also allowed us to explore more ideas for user interventions and investigate broader questions in our eventual study. Our dashboard was built in JavaScript and laid out in a wide format for interacting with on a desktop machine.

We designed the dashboard to contain some interactivity, but nothing highly complex. Each chart has tooltips and visual filtering provided on hover/keyboard focus and the line chart has data filtering through the legend as well as sonification.

5.4.1 *Pretty Accessible* by Default

In order to really test *access frictions*, we wanted our dashboard to be considered accessible in its default state. We ran manual and automated tests according to accessibility standards as well as a 50-heuristic manual accessibility evaluation specific to interactive data visualizations [63, 243]. In addition, we chose to use Highcharts to create the visualizations in our dashboard because existing work has demonstrated the breadth of their accessibility capabilities [141].

We ensured that text contrast was strong, text size was well above guidelines, interaction targets were of a minimum size, screen reader access was descriptive and interactive, our DOM was structured into a hierarchy, we provided semantic HTML data tables for each chart, and there were no *critical* issues detected when running automated tests using *axe DevTools* software. Many of the capabilities that made our dashboard accessible were provided out of the box by Highcharts.

5.4.2 Reasoned Constraints: 195 Accessibility Options for Interactive Data Representations

After building our initial dashboard (see Figure 2), our research team collaborated and discussed potential access frictions that might arise from the use of our dashboard. We used our existing experience from accessibility work, more than a decade in the case of the Highsoft and Elsevier co-authors, to assemble a list of concrete, expected user barriers that would be hard to resolve in a single design, and organized those barriers based on common themes.

We then used these themes to brainstorm how we anticipated end-users would identify barriers and then what language they might use to describe an identified barrier and overcome it. One example might be under the theme "hard to see": "I can't read this text" as a barrier with the final language for them overcoming that barrier as an expression similar to, "I wish this text size was larger" or "make this text bigger."

From these solutions, we re-framed the language into interactive option categories. All categories were given binned options and not more than 7 choices, to avoid overwhelming users. For example, "text size" was an option category and had 7 binned choices from "small" to "large" available. We then created a hierarchy of option categories underneath these higher-level categories which could allow for more element-level specificity within a data visualization. So "text size" as a higher level category with options also contained children that had more specificity, such as "title text size," "legend text size" and so on. The final stage of our language and options design was in line with prevailing industry wisdom, which was to avoid organizing the language of our configurations based on categories of disability [7] and instead focus on higher-level categories of identifiable elements of a user's experience, such as "text visuals," "audio," and "color and contrast."

At the end, we produced 195 option categories and 774 total option choices. Using the combinatorial rule of product, we calculated 6.83e114 possible unique end-user design configurations from these choices, which is more than the estimated number of atoms in the universe.

However, to ensure the scope of our user study was feasible, we reduced our initial working option categories down to 33 with 137 total options (and 9.35e19 possible design combinations), all focused on options we believed would be most relevant for users who are blind or low vision. These options and our subset are viewable in our live, interactive demo online.

5.4.3 Preferences Menu Design

We then iterated on visual and functional designs to allow users to actually interact with and enact these design configurations. Our early ideas included a natural language interface (since we used a relatively "natural language" centered process to develop these categories), direct manipulation of the elements in a visualization (through focus, hover, click, or selection methods), and eventually settled on the user interface of a separate, visually nearby menu with nested options (see Figure 2). We designed our menu so that manipulating higher level options in the hierarchy would enact downstream options to follow suit, but any manipulation to downstream options would override higher controls, following common patterns used in systems that implement hierarchical specificity.

We justified our user interface as a menu for our final choice because it provides a place for metadata from the other design ideas (natural language and direct manipulation) to live, in case we develop those down the line as well. We anticipated that a menu not only provides a means of interaction but also storage of the state of a system. In addition, this type of user interface is common and relatively recognizable.

5.5 Evaluation

Our first research question for this project ("What constraints and capabilities should we provide end-users to give them meaningful agency over interactive data visualizations?") focused on our thematic collation and compilation of anticipated access frictions, but our following two research questions would require outside evaluation: "What qualities, challenges, and design opportunities do designers and engineers envision for a data visualization *softerware* system?" and "What qualities, challenges, and design opportunities do blind and low vision users envision for a data visualization *softerware* system?"

5.5.1 Preliminary: Visualization Practitioners

The preliminary step in our evaluation was to investigate what qualities, challenges, and design opportunities data visualization engineers and designers envision for a *softerware* system.

5.5.1.1 Recruitment

We recruited 4 data visualization practitioners, each with roles as a current or former visualization software engineer (3) or designer (1). We recruited participants from our existing network of engineers and designers, requesting participation via email. Our practitioners were not compensated for their participation and we asked them up front if they would be willing to volunteer their time for us.

5.5.1.2 Procedure

We conducted 30-minute, semi-structured, qualitative interview sessions either over Zoom or in-person. The session consisted of a 5 minute explanation, 5 minute demo of our prototype's capabilities, and a series of open-ended, semi-structured questions for 20 minutes. Our questions started with getting their thoughts on the idea, what they anticipated other developers and designers would think, what aspects of a visualization they believe end-users will want control over, issues they believed end-users would face, and what new opportunities they envision our prototype and underlying design concept of *softerware* enables.

Table 5.1: Study Participants

PID	Age	Gender	Disability
P1	39	F	Totally Blind
P2	38	F	Totally Blind
P3	46	M	Legally Blind
P4	28	F	Low Vision
P5	34	M	Legally Blind
P6	52	M	Totally Blind
P7	56	F	Low Vision
P8	36	M	Low Vision
P9	55	M	Totally Blind

5.5.2 Study: Blind and Low Vision Users with Accessibility Expertise

Our primary study was focused on our third research question on the qualities, challenges, and opportunities that users with disabilities, in this case users who are blind and low vision, envision for a *softerware* experience of interactive data visualizations. To explore this, we used our prototype dashboard as a design probe to stimulate concrete feedback and ideation on both the details of our prototype as well as our larger design concept of *softerware*, borrowing from Noor Hammad's method used when exploring accessibility preferences of users in novel streaming software [96].

Our study is intended to contribute qualitative knowledge, largely because we believe that statistical generalizations or controlled experiments about a particular group or subgroup of people with disabilities may actually produce knowledge that reinforces the existing problems we are trying to address. Instead, we are explicitly interested in knowledge and experiences that might exist on the margins, even knowledge as specific as a single individual's preferences. We want to explore ways that broader guidelines and design knowledge are capable of producing artifacts that still retain barriers for some individuals with disabilities. This larger challenge (of general guidelines that do not provide a meaningful fit for individuals living with disabilities) is not new to accessibility and assistive technology research [188].

And to this end, we designed our study to maximize the production of knowledge that is considerate and careful of individual differences, challenges, preferences, and envisioned opportunities.

5.5.2.1 Recruitment

Our study involved 9 total participants who are blind or low vision (see **Table 1**), all of whom are also professionals with accessibility expertise (either currently or formerly employed in an accessibility-specific role as subject matter experts). 5 of our participants self-identified as male, 4 as female. Average age of our participant group was 42.67 (SD = 9.99). We initially recruited 6 participants using an existing, compensated research relationship between Highsoft and an external consultancy. In addition, we recruited 3 more participants from our existing network of accessibility consultants, who were each compensated 100 USD for their time.

We anticipated that recruiting participants who not only have lived experience with a disability but also are subject matter experts in accessibility would contribute to the depth of our qualitative study as well as general breadth of considerations. We wanted to maximize the value of feedback on our work.

We reached out to all participants via email with a call for participation and participants were screened according to whether they are blind or low vision. Participants were notified in advance of compensation and that consent to participate is voluntary.

5.5.3 Procedure

Our qualitative study sessions were recorded and conducted over zoom in 3 primary phases (plus a break) during one 90 minute session. Our phases were: early interview, task-evaluation of our dashboard (menu hidden) with discussion, a break, and task-evaluation of our dashboard (menu shown) with final discussion.

5.5.3.1 Introduction and Early Interview [20min]

Our session opened with an introduction to the research team and gathering verbal consent from participants for participation. We gathered demographic information from participants and asked them about their current assistive technology use. We followed up with questions related to whether or not they customized their technology in any way, through adjusting settings, modifications, adding scripts, getting extensions, or equivalent. We then ended the opening session with ice-breaker questions about whether they can recall a chart or graph they have experienced in the past and what their favorite way to experience a chart is.

5.5.3.2 No Menu Prototype, Tasks, Discussion [30-35min]

The next phase of our session involved showing participants our demo environment (see Figure 2), except that our preferences menu was hidden. We explained what the dashboard was and entailed, including explaining each chart type shown (sankey, bar, and line) and how to read them. We gave users a short amount of time to explore the dashboard, and then notified them that in order to evaluate the effectiveness of our technology, we would be asking them to perform 2 data tasks, one elementary and one synoptic [1]. Our intention for performing tasks was not to measure speed or accuracy of the participants, but simply as a probe for eliciting feedback on the usability and effectiveness of our prototype and design.

Our first question was to answer an elementary analytical task (direct or indirect lookup), "Does petroleum or nuclear contribute the most to Electricity and Heat?" ("nuclear" was correct). Our second was a synoptic task (pattern identification or multi-value comparison), "Which energy type has the highest use in Dec *and* Jan?" ("natural gas" was correct). We gave participants a limited time (5mins total) to answer the questions and upon answering, we gave them the correct answer and asked them to explain their process of finding their answer, step-by-step.

Our final step in this process was to interview them about their perceived challenges and frustrations with the dashboard and whether anything could be changed or adjusted in order to help them complete their tasks. We followed this phase with a 10 minute break.

5.5.3.3 Menu Prototype, Tasks, Discussion [30-35min]

We opened the final phase of our study by sending our participants a new link to a version of our online dashboard that included our preferences menu. We explained the purpose of the menu and gave them 5 minutes to explore the available options.

After participants explored the menu and its effects, we repeated our tasks procedure. Participants were given 2 tasks to complete in five minutes. First, an elementary analytical task, "Where does most coal go?" ("electricity and heat") and then a synoptic task "In the summer, June through August, which energy type has the highest consumption rate?" ("petroleum").

Our final discussion focused on investigating our participant's thoughts on our prototype, the idea of preferences and customization, why they chose the customizations that they did, whether they had any new or additional ideas, considerations for other users with disabilities, and any other concerns, challenges, or feedback. We asked them specifically to consider both their personal, lived experience with their disability and assistive technology in addition to their professional expertise in accessibility.

5.6 Results

We performed two analyses from our studies, first analyzing our findings from our preliminary study from practitioners and then analyzing our results from our study with end-users. We collated our notes and transcript materials, coded them thematically, and then used affinity diagramming to group the themes that emerged from our data [98].

5.6.1 Preliminary Findings

To avoid repeating information between our preliminary study with visualization practitioners and final study with blind and low vision participants, any findings from our end-users that are echoed by our practitioners will be mentioned later. Only the findings unique to our preliminary study will be included here.

5.6.1.1 Alleviating Situational Barriers

3 of our 4 practitioners spoke about the potential benefits of end-user manipulation of a visualization for situational or contextual reasons. One participant gave the example that when giving a presentation using an existing dashboard, having the ability to manipulate features to suit layout, flow, and interactions on-demand would be valuable. Another example given was that at times a user's viewing device (such as a smartphone) can cause barriers, so softerware would be useful to have available.

5.6.1.2 Creating Potentially Harmful Visualizations

The second theme from our practitioners (3) was the concern that end-users would be able to create a misleading or harmful data visualization. For example, we have studies on how encoding area size [151] and aspect ratios [33] can be misleading or deceptive, yet being able to manipulate these for accessibility and contextual barriers (such as viewing a chart designed for desktop on a mobile phone) are important design considerations. Users may accidentally adjust features of a visualization while self-fitting that actually create problematic designs. Being able to design a system to avoid this would be important.

5.6.1.3 Designing via *Softerware*-first

The final theme from our practitioners was around authoring and design-tuning via *softerware*, where 3 participants discussed using a direct-manipulation or LLM-based *softerware* interface to author data visualizations and 1 of the 3 also mentioned that large-scale, privacy-preserving data collection from users could be used to create smarter design defaults in the future. We believe that both suggestions mirror existing work that speaks of the benefits of "design-through-use" and "continuous-co-design" [160].

5.6.2 Prototype-level Feedback

The advantage of a study with participants who had accessibility expertise in addition to their lived experience as people who are blind or low vision is that we were able to get feedback on our existing prototype as well as on our larger idea space for *softerware*.

5.6.2.1 Navigation Structure Options

While not a theme across participants, P2 mentioned that they would like to be able to navigate a data visualization using headings with their screen reader, rather than via regions. ("Regions" are a type of semantic markup used to create programmatically recognizable organization for screen readers.) This was a suggestion that immediately led to our team iterating in parallel on ideas the next day. More than 71% of screen reader users navigate information via headings when first encountering a new web page [244]. This suggestion made sense to explore as a sensible default.

5.6.2.2 Previewing Change

Our low vision users (P4, P7, P8) requested a feature that we had originally designed but not implemented, which was to directly show what different options would look like in the preferences menu itself. For example, the "text size" options would either have a preview of the text size shown for each option in the menu (like showing "Large" in the actual resulting large text size) or with a nearby preview window that would show the result of a selection as it is being selected. Low vision users in particular often use high levels of magnification and zoom, so the live results shown in the visualization space required users to go back and forth between the menu once an option was chosen and into the chart space to find what had been affected.

5.6.2.3 Language Re-consideration

Some of our participants (P1, P2, P4, P6, P7, P8) noted ambiguity or lack of clarity in the wording we used for our menu's higher level options, such as "Audio" having options for "default," "gentle," and "complex" while lower level options that inherited these were hard to connect to. For example, “Sonification order” under “Audio” had the options “default,” “sequential,” or “simultaneous.” “Gentle” in “Audio” would set the child setting for sonification order to “sequential,” but this was unclear initially.

Other participants (P1, P2, P3, P6) noted that while the menu’s focus on functional categories was helpful, it might be nice to also have a way to customize the menu itself or view it from a “disability” perspective, so they could get all the screen reader options in a single place. Users were interested in looking at all options relevant to “screen readers” or “low vision” together.

5.6.3 System-class Accessibility Findings

The next set of themes that emerged were considerations that both our end-users and our visualization practitioners shared, which we are calling *system-class* considerations, using the phrase from Chris Fleizach and Jeffrey Bigham [74]. In order for *softerware* to function at scale, certain technical and infrastructure considerations would have to be prioritized to make things possible. This theme emerged thanks to the accessibility knowledge and expertise of our end-users and engineering concerns of our practitioners.

5.6.3.1 Persistence

Every participant (P1, P2, P3, P4, P5, P6, P7, P8, P9) as well as all of our practitioners noted that the ability to create some sort of “profile” or persistent state of their customizations would be one of the most important features that would make *softerware* actually useful.

> “What if I come back to this? Will I lose this? Do I need to do it again?”—P6

We followed up by asking whether certain contexts would make persistence more or less important. We asked users whether a random website or news article with a chart in it would be worth their time, to which most users replied, “no.” However, P5 noted that “This is so fun that if it was there I still might play around with it and use it, especially if I had the time.” P4 related this issue to an existing frustration with video games, noting that having to set up repeated options for every game was time consuming. It would be nice if they could “do this once and forget it.”

5.6.3.2 Profile Sharing

Following this theme of establishing a profile, most of the participants (P1, P3, P4, P5, P6, P8) and 2 of our practitioners also expressed interest in being able to share their own profiles or ingest settings from others, in order to save others or themselves time. In phase 1 of our procedure, we asked users about their existing levels of modification, customization, and preferences setting in their existing use of technology. While all of our participants (except for P9) customize, personalize, or modify their technology to some degree, those who were most interested in customization or spoke the most about it (P3, P4, P5, P8) were also the most passionate about being able to save *other* people time and not just themselves.

> “I customize my tech a lot. If I use something for the first time and it feels off, I find a way to fix it. But most people aren’t like that; it takes too long. So I love when I can share [my modifications and customizations] with others.”—P3

5.6.3.3 Cross-system Interoperability

Closely related to *persistence* and *profile-sharing* was an idea expressed by several participants (P3, P4, P5, P7, P8) that they wanted to be able to use these settings outside of Highcharts and even outside of the web. "Will this work in Microsoft Excel?" and "I use salesforce for analytics a lot and would love this there," remarked P4. However, cross-system interoperability would require multiple charting libraries being intelligent enough to ingest user settings, when most are currently incapable of even recognizing a system's "high contrast" settings being active. In addition, all 4 of our practitioners suggested that there would need to be a system in place, either at the operating system level or as some kind of service hosted by a platform, where these settings could be recognized and ingested. For cross-system interoperability to be made possible, it would require establishing standards for customization, standards for preserving user privacy, and coordination with the larger community of visualization practitioners and software providers.

5.6.4 User-Centered Findings

The last major set of themes is related to the considerations of end user experience of a *softerware* system applied in practice, including our observations about the differences between users and their choices when personalizing a data visualization interface.

5.6.4.1 Frictions in User Differences

Our first major user-centered finding was that no participant chose the same set of preferences as another. Every user discussed different reasons for justifying their choice of options. Users even chose options that others specifically emphasized were inaccessible to them. An example of this was a tension in preference for and against use of "dark mode" designs.

> "If anything has dark mode? That's great. I wish everything used dark mode."—P4

P4 mentioned that they had "night blindness" (*nyctalopia*), which is why dark mode designs are helpful for them. However, P7 also mentioned that they had progressive nyctalopia, but dark mode makes an interface "virtually impossible" to them.

> "Oh, I can't use dark mode at all. I hate when websites have [dark mode] because it can be virtually impossible to use."—P7

Any one of the designs chosen by a low vision participant would have been insufficient for providing access for any of our other low vision participants (see Figure 1).

Our blind participants also had different justifications and preferences for their text and audio customizations. Some justified their differing preferences with similar justifications, such as cognitive accessibility and text description length. For example, P9 stressed that "I prefer to keep things simple" to "avoid overwhelm" while P2 said, "more information is better than less, when it comes to data." Both P9 and P2 preferred "accessible defaults," but disagreed on what length of textual descriptions should be default.

5.6.4.2 Accessible Defaults are a Necessary Prerequisite

Several participants were concerned that this approach would allow designers and developers to continue to make inaccessible charts (P2, P6, P7) if users have the ability to *self-fit*. Participants emphasized how important it is to have strong accessibility *before* customization is introduced (P2, P4, P6, P7, P9). Even 3 of our 4 of our practitioners expressed worry that *softerware* could put a design burden on users.

P9, our only participant who almost exclusively uses default settings (and avoids mods and extensions) with their current assistive tech, stressed the importance of well-thought out defaults. It is clear that for users like P9 in particular, strong defaults are much more important than customization. Although less common, some assistive technology users are not interested in the work involved in personalization and would prefer technology to suit their needs out of the box.

This leads us to argue that there is a line between ethical use of *softerware*, which is built on top of already-accessible material, and *softerware* that is filling gaps in poor design. Designs that are lacking access that wouldn't cause any friction for someone else if they were present, such as simply having alt text, aren't in need of *softerware*, they're just in need of accessibility.

5.6.4.3 Effort-to-Outcome Ratio

As a playful rephrasing of the visually-centric (and controversial) *data-to-ink* ratio [48], we observed an *effort-to-outcome* ratio among our participants, in line with previous results from existing work [265]. Nearly all participants (P1, P2, P3, P4, P6, P7, P8, P9) noted that the work required to interact with this menu wouldn't be worth the effort if they had to do it every time they interacted with a data visualization.

Most of our participants who used screen readers (P1, P2, P3, P6, P9) also mentioned that the menu itself was too cumbersome for navigating within and back and forth with the dashboard. Setting options was quite slow, and observing the output of a given option change, such as text verbosity or sonification type, was hard to do with accuracy. Keeping what a previous state was like in memory was hard.

One participant was interested in different ways that this process could become easier, suggesting

> "What if I could just tell it what to change while I'm listening? Like right here [navigating a chart element] what if I could just say "keep it short" or maybe "wait, tell me more."—P6

This suggests that there may be a space to explore non-visual direct manipulation *softerware* strategies.

5.7 Summary

In an idealized world, designers do their best to produce useful and accessible interfaces. They're concerned with making software as accessible as possible by default. But no single design is capable of perfection. *Access frictions* between accessible defaults and the needs of real individuals might always be present in software interfaces. To that aim, we hope to contribute knowledge that can inform future designers and developers to not only build accessible artifacts, but build *systems* that enable end users with disabilities to have interactive agency over their software experiences.

Our vision of data visualization *softerware* demands more involvement from research and industry. In order to offer as much value as possible to end-users, we need standards set for accessibility profiles and we need data visualization software, libraries, and applications to respect and be able to contribute to those profiles. We want to encourage researchers to investigate further the needs of people with disabilities, designers to imagine new interfaces and interaction paradigms for end users, and engineers to build robust systems that are capable of not only respecting a user's preferences and customizations, but providing persistence, interoperability, and system-class infrastructure.

Chapter 6
Cross-feelter: Tool-making for Blind Data Science

This chapter was adapted from my paper, currently under review with IEEE VIS:

F. Elavsky, Y. Li, P. Carrington, and D. Moritz, '*Cross-feeltering*: A principled tactile design approach for blind linked data interaction', *IEEE Transactions on Visualization and Computer Graphics*, 2025.

6.1 Overview

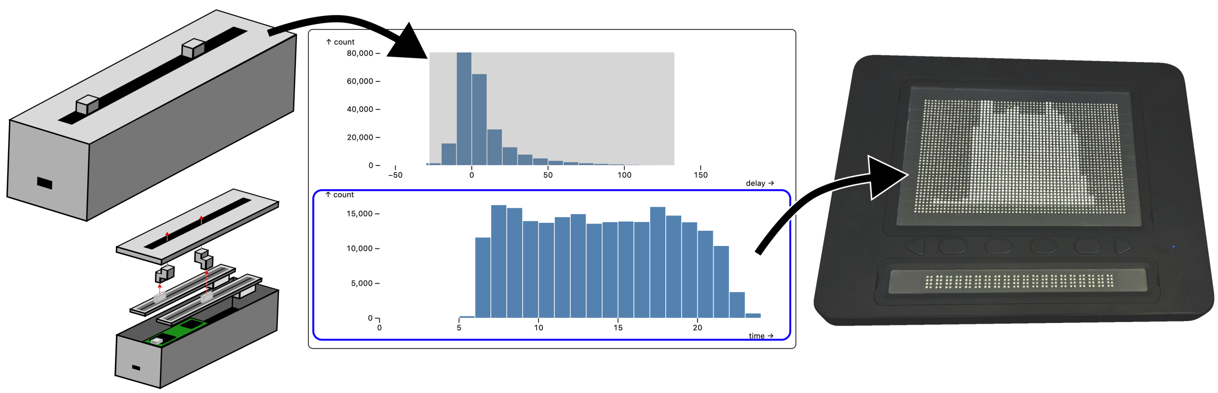


Figure 6.1: Interaction and perception in one space while being able to perceive output in a separate space is the cornerstone of cross-filtering. Our prototype *cross-feelter* (left) can manipulate a visual cross-filter on one visualization (middle) in order to produce output in a separate visualization, as a tactile graphic (right).

Cross-linking is a powerful and flexible interaction in data visualization. In this paradigm, a user will perform some visual input interaction over a chart space, such as filtering, selecting, and hovering, in order to produce a simultaneous output in another chart space. The output change is dynamic and conditional based on the user's interaction parameters from the first chart space.

Cross-filtering, in particular, is a form of cross-linked interaction where a visual filter on one chart space actually performs a filtering operation over a shared dataset, data cube, or model. This filtering action then produces coordinated output in linked charts. Cross-filtering helps users rapidly build hypotheses in data science because it supports exploring correlations and rudimentary relationships between variables within a multidimensional data space. One reason cross-filtering is effective is due to the speed that it provides users, offering significant interaction improvements compared to serially querying data.

However, making cross-filtering and other cross-linked interactions accessible to blind data scientists and analysts remains an unaddressed problem. Very little research on data accessibility focuses on blind people who are data practitioners, leaving many open questions about how to improve their work [2, 43, 126, 187]. We are interested in addressing this gap.

For some background, people who are blind, which is about 4% of the population in the United States [180], predominantly rely on an assistive technology called a *screen reader* to interface with computational devices, from personal computers to smartphones. Screen readers parse content in computer interfaces one element at a time and either produce audio output that semantically describes each element, or produce textual output for display on a refreshable braille device. Screen readers perform a variety of actions, such as clicking (selecting), opening links and menus, navigating using shortcuts, and more.

Unfortunately, screen reader technologies carry a fundamental interaction flaw: they only provide perceptual and interactive support for one element at a time. And human auditory perception and cognition in general struggles with multiple tasks [83]. Screen readers are designed with this cognitive limitation in mind, intentionally maintaining a serial output of only a single unit of information at a time. It is no surprise that screen reader users often ramp up the speed of their synthesized speech, so that they can consume as much information as possible in less time [158].

The limitations of both screen readers as well as human auditory perception make it particularly difficult to design an accessible alternative to cross-filtering that doesn't eventually put significant burden on blind data practitioners. Visualization interfaces with cross-linked interaction such as cross-filtering will inherently demand tedious back-and-forth interaction by a screen reader user. We conjecture that the increased cognitive load and slow interaction time when using a screen reader ultimately defeats the purpose of fast, cross-linked data interaction.

In this paper, we explore the following research questions:

- **R1** (Exploratory) How might we envision a tactile interaction design approach that maintains the analytical principles and capabilities of cross-filtering?
- **R2** (Quantitative: Objective) In what measurable ways can we observe improvements to blind data interaction using our tactile interaction approach? (Speed, accuracy, precision, and quantity of data queries.)
- **R3** (Quantitative: Subjective) What are the self-reported benefits, opportunities, and challenges that blind people experience when using our tactile interaction approach? (Anxiety, cognitive load, stress, and enjoyment)

We propose a tactile interaction design approach that supports simultaneous perception of input and output across multiple interfaces, adapting principles of cross-filtering for blind interaction. We instantiate our design approach using a novel hardware device we are calling the *cross-feelter*. Our device has two linear, motorized slide potentiometers, referred to as *motorized faders* in sound hardware, which are arranged with angled knobs along a single railing in order to place the two sliders in the same interaction dimension. Our design provides a way to perform tactile input that spatially filters a data representation to produce coordinated change in a separate, linked data representation, parallel to the visual metaphors and elements provided to sighted users.

We present an evaluation of our design approach with professionals who are blind and actively employed or previously employed in work that involves data analysis in addition to people who are blind and do not have prior professional data interaction experience. We compared our device coupled with a tactile display to a method that leverages a screen reader and a tactile display as a baseline.

For our objective measures, we find that our approach substantially improves the completion rate of data tasks within a timed exercise, significantly increases the quantity of computational data-queries performed, and greatly increases user data-queries spoken aloud during an exploratory session of the data interface. We also analyzed the subjective experiences of our participants and found that both data experts and non-experts self-report less stress during task completion, less anxiety when speculating about future use of our approach, and greater enjoyment of the overall data analysis experience, with results stronger for our participants without data expertise.

We conclude our contributions with future innovative use cases of both our tactile design approach and applications of our *cross-feelter*.

6.2 Related work

Making both static and interactive data interfaces accessible to users who are blind is a growing area of focus in HCI research. It remains a challenge. Approaches to accessibility vary, but some of the dominant research threads in this space are focused on textual descriptions, structured and interactive text experiences, tactile data representation, and tactile-audio multi-modal approaches.

Before we begin, we want to clarify our terminology in this paper in terms of accessibility and data. "Data experiences" are any human experience with data. "Data representations" are rendering and output methods for spatial, tactile, auditory, text-linguistic, and visual data experiences. "Data interfaces" refer to data experiences that may, in some way, provide a user with interactivity such as querying, filtering, or sorting. Note that while some researchers have argued that *any* representation of data is technically *interactive* because our perceptual systems or assistive technologies must parse and process them [276], when we use the term "data interfaces," we are only referring to systems that produce computational *change* as a result of human input (explicit or otherwise). "Data visualizations" are a visual data representation and are often interactive, while "sonifications" are auditory data representations. "Tactile charts" are tactile data representations (sometimes referred to as "tactilizations" [102]). However, we prefer primarily call them "tactile data representations" here.

While our research is focused on making interactive data visualizations more accessible, we use broader terms such as "data experiences" and "representations" in cases where a visualization is not the only representation we are referring to (such as a simple case where a visualization also has an accompanying non-visual interface).

6.2.1 Limitations of screen readers as a technology

We conjecture that screen-reader-based navigational and interaction capabilities are a significant bottleneck for blind data analysis. At the beginning of this research project, we set out to explore the scope of capabilities of modern screen readers: what are they currently used for and how might we improve data experiences to suit their limitations?

For context, screen readers are assistive technologies that audibly read out the semantics and content (text) of visual and non-visual interfaces and are the primary technology leveraged by blind users. Blind and deafblind users may also use a refreshable braille device. However, both screen readers and braille displays have the same fundamental interaction limitation: they only ever navigate to a single element at a time. This means that input and output are discrete and determined entirely by individual elements in a computational interface [70, 138].

By default, screen readers will navigate one element at a time and along a single, serial axis (forward and backward). Due to this, research has looked for ways to enhance the directionality of screen reader navigation, to enable multiple dimensions and directions (such as up, down, left, right, drill in, drill out, etc.) [64, 213, 236, 276]. We have empirical evidence that changing the order and arrangement of navigation experiences of charts and visualization dashboards provides screen reader users with a richer understanding of the data at both a high and low level [131, 231, 276].

In addition, guidelines and older work both argue that well-structured tables (such as a semantic HTML table) can also provide screen reader users with a richer data navigation experience, since most modern screen readers are capable of navigating across rows and up and down columns using special commands [63].

However, augmenting the navigation capabilities of screen readers and braille displays may ultimately always be insufficient because these devices are still fundamentally just tools that create text- and language-based experiences for users [231].

6.2.2 Limitations of text- and language-based approaches

We conjecture that language may never be fully sufficient as an independent alternative or primary modality for data interaction. Language alone may not be enough for blind users to have fast, immersive, deep, and exploratory data experiences.

Despite this, text- and language-based approaches are the most common and long-standing area of focus in research on accessible data experiences. Human-authored descriptions are "manual" summarizations of a data visualization (commonly referred to as "alternative text") and are the de-facto standard for making visualizations accessible [63, 131, 138]. Empirical work has established guidelines for how to describe charts that are simple in contexts with blind students [28] and how to describe novel or strange chart types [135]. We also have existing methods to provide longer descriptions of visualizations in cases where more detail is merited, such as in scientific contexts [273].

Understandably, the difficulty of describing data visualizations has been offloaded in recent years to a wide variety of automated methods such as leveraging large-language models [268], computer-vision approaches [37], and other algorithmic methods [72].

One limitation of descriptions are that they are static and must assume the context and intentions of the reader [62]. Text descriptions are not inherently interactive, they are simply an output method. To this end, recent work has explored how to create question-based and conversational interfaces of data [116, 137] as well as preferences-driven approaches to description customization [128] in order better fit language-based approaches to user needs and circumstances.

However, visualization is a practice of turning the linguistic-symbolic form of data into the spatial-geometric form of charts and graphs. We visualize data to *escape* from language, because that part of our brain processes information more slowly [79]. For this reason, there are advantages to leveraging non-linguistic parts of our cognition.

6.2.3 Limitations of sonification and auditory cognition

In addition to text- and language-based solutions, work in accessible data representation has focused on providing some form of sensory substitution for existing visualization [13, 40]. The most common form of sensory substitution is sonification [103].

Projects in sonification are varied and the field of work has robust tooling of its own, separate from visualization [136]. Some work has even explored how to de-center visualization using a data-centric approach instead of sensory substitution, where the output of data interaction is both visualization and sonification [278].

However, sonification has a harsh limitation in practical contexts: it competes with the synthesized speech of a screen reader, which has been noted by existing projects in this space [136, 236]. At a technical level, it is difficult to coordinate the timing of the speech of a screen reader without being able to communicate to it programmatically, which is not something current standards bodies are interested in exploring [6]. And this technical limitation means that sonification-based interfaces must either control synthesized speech (and thus recreate their own screen readers from scratch) [136] or wrestle with the fact that sonifications may play at the same time as synthesized speech [236].

This is not just a technical limitation with minor downsides, but a foundational problem for blind users. Human auditory cognition has a relatively short working memory and easily becomes overwhelmed when encountering multiple pieces of information simultaneously [103]. Human auditory cognition is notoriously poor in even "dual task" paradigms (two tasks at once) [83, 95].

And finally, sonification is only an output method. There are common input methods that work alongside sonification, such as brushing and scrubbing or using simple play/pause controls [103]. However, we believe that the primary gap for blind analytics rests between the breadth and power provided to *interactive* visualizations, ones where explicit user input is a driving force.

6.2.4 Looking to tactile interaction in accessibility and HCI

Given present limitations of other approaches, we look to tactile data representations and tactile and haptic interaction methods. Relatively speaking, there are fewer projects specific to data interaction or accessibility compared to tactile interaction broadly across the field of human-computer interaction. We searched for relevant work in all these spaces to inform our research.

First, the intersection of tactile data representation and accessibility is where some of the oldest research can be found [82]. More recent work has formalized tactile alternatives for basic chart types [66, 155, 191], experimented with simultaneous tactile and synthesized speech output [8], leveraged lower-cost, lower-resolution single-line, refreshable braille displays [207], and provided voice-input-with-tactile-output mixed-modality interfaces [193]. Other contexts have explored haptic and tactile methods, sometimes coupled with sonifications or synthesized speech audio [69, 220].

However, we are specifically interested in research that uses tactile *input*, and especially those that could inform a brush-based interaction approach that is commonly used in cross-filtering.

We only found one emerging project and one more-established in the domain of accessibility with an input modality relevant to our work. In the emerging work (a poster), they use the touchpad of a laptop as an input method [214]. This could be used to map the notion of brushing a 2-handled filter across a single dimension (x or y), however the current position of the brush (when not in use) is difficult to determine, making this method insufficient for our uses. The more-established work (a full-paper project), the authors propose "interactables" which can be used to augment the use of a smartphone for blind users with an array of different tactile inputs [274]. Within this project, one of the input elements they use caught our attention: a slider. Their current use only has a single slider and it is not motorized, so it isn't necessarily ideal. We want a device capable of manipulating at least a 2-handled filter as well as providing live synchronization (output) with data interfaces that also come pre-filtered.

So we look broadly to HCI and found a poster [68] as well as 2 highly relevant projects that leverage motorized "faders" (a slider component used in sound hardware) [45, 223]. Both projects propose dual faders for precise tactile control of axes in AR/VR displays of data, the MADE-Axis in particular even presents a cross-filtering use case for their device [223]. We believe that improvements could be made to the dual-track device for the specific purpose of cross-filtering, however these projects serve as a strong foundation for our work.

Yet, our research gaps still remain: we must explore and justify our design approach based on principles from cross-filtering and accessible tactile interaction (**R1**). In addition, we lack empirical observation of the value of a dual-fader device when used for blind cross-filtering analytical work (**R2**, **R3**).

6.3 Design approach to enable tactile cross-feeltering

To explore our first research question, "**How might we envision a tactile interaction design approach that maintains the analytical principles and capabilities of cross-filtering?**" we formulated our design goals as-follows, which we unpack in the following subsections:

- **G1** Maintain and reflect system state through tactile location
- **G2** Allow for multiple channels of perception
- **G3** Correspond tactile-to-analytical functional semantics
- **G4** Provide linked interaction of multiple dimensions
- **G5** Support fast, dynamic, reversible filtering
- **G6** Facilitate user-driven exploration

6.3.1 Principles of blind data and information interaction

Sometimes research motivations come about through real-life observation. And the entire motivation for this project came from prior work that preceded a previous project [REDACTED]. REDACTED, the primary author on this paper and in that work, was observing their blind friend, soon to be co-author and co-designer, REDACTED explain how they read an embossed academic research paper. (This story was shared with permission by our former collaborator REDACTED.)

For context, embossed paper is a type of "high resolution" tactile rendering where an ordinary piece of paper is sent through a printer that leaves punched holes at a fidelity of up to 100 dots per inch. This method of embossing can be used to create readable braille documents and high-fidelity tactile graphics affordably (since no ink is used, the cost is typically 0.01 to 0.03 USD per page of paper).

And REDACTED employed two advanced tactile interaction strategies when reading the paper that are informative for our work which we are referring to as *fuzzy search* and *dual task comparison*.

First on *fuzzy search*, REDACTED was trying to explain to REDACTED how a particular equation changes throughout a paper he was reading. He is a neuroscience engineer, so REDACTED wasn't really familiar with the details as he was explaining. So REDACTED grabbed his stack of pages and thumbed a few pages in. He touched near the top of the paper and immediately knew he was one page off. He flipped to the next page and again touched near the top. Then his finger quickly moved to the more specific area where the equation began. He explained the equation to REDACTED, one step at a time.

Then, REDACTED kept that page to his left and flipped through the remaining stack of pages, repeating his fuzzy tactile search process to find the equation as it was mentioned later in the paper. He found it and then performed a *dual-task comparison*: he skimmed his fingers quickly across both equations at the same time, one hand on each. He stopped, mentioned to REDACTED that this is where the equation diverged, and then explained the evolution of the equation.

We first establish our foundational principle of tactile interaction based on this experience, which is that we must **maintain and reflect system state through tactile location**. We colloquially refer to this principle as "objects with permanence" playing on the visual concept of *object permanence*, where a subject remains aware of the persistence of things even though they may not actively see them fully or at all. *Objects with permanence*, then, became a motivation for us to ensure our system's tactile state remained in a location where users last perceived it being. REDACTED was able to remember where the equations were in his paper because their location persisted on the embossed paper.

In existing empirical work on tactile interactive systems, it is common for people to use multiple tactile locations at once to approximate relationships, such as resting a pinky on a "known" barrier or tactile location while "searching" relative to that barrier using another finger [88]. To enable searching and tactile comparison, we need a tactile system that has a persistent and meaningful location state of its tactor elements, especially for barriers and inputs.

The next principle we used to define our design goals is that our system must **allow for multiple channels of perception**. In Section 6.2.3, we have already established that audio *alone* is insufficient. However, a system that leverages tactiles can still use audio output, as many do [8, 193]. We want to establish that tactile interactions can take place simultaneously, but audio output must always be handled in a serial manner to avoid overwhelming or confusing users.

Our final tactile principle is that we want to design our system to **correspond tactile-to-analytical semantics**. What we mean by this is actually informed by web standards and practical accessibility contexts: things should have names and verbs that meaningfully describe *what* they can do analytically and *how* they are used [245]. This means that in the context of cross-filtering, which is done by *sliding* the *handles* of a spatial filter, a good tactile equivalent will also have *sliding* capabilities that are used when manipulating a *handle* over the spatial dimensions of a filter over a tactile graphic. There is a shared tactile-to-*visual* metaphor with handles and sliding, but the important part of the action is that it is fundamentally *spatial* and performs an *analytical* task. While we acknowledge that not all spatial interaction metaphors will easily transfer to tactile interactions, it is an important principle to avoid introducing an unfamiliar device that also uses interaction metaphors and terms that require translation.

6.3.2 Principles of cross-filtering

Cross-linked interactions, particularly cross-filtering, are foundational interactions for exploratory data analysis in visualization, commonly referred to in early literature as "brushing" [15]. Cross-filtering was more formally established methodologically for coordinating filtering interactions over multi-dimensional datasets, specifically [252]. In order to provide value to blind analysts, it is important to understand which principles of cross-filtering can guide our tactile design space.

Our first principle for cross-filtering is **linked interaction across multiple dimensions**. Prior work established that coordinated brushing views with underlying linked data enables users to see relationships between variables by propagating selections dynamically [100]. This work underscores the importance of multi-dimensional coordination, where filtering in one view (e.g., a histogram) directly constrains another (e.g., a scatterplot).

Additional work further formalized cross-filtering as a means to support **fast, dynamic, and reversible filtering**, empirically observing the inverse relationship of "latency" (or lack of interaction-to-rendering speed) on accuracy in cross-filtering analytical tasks [150]. Speed enables more forgivable and usable interactions, which has motivated more recent research to develop more powerful, scalable cross-filtering [101]. This principle of fast, dynamic, and reversible filtering was actually our initial motivation for looking into alternatives to screen reader interactivity, which is slow and serial.

The last principle of cross-filtering valuable to our work is **user-driven exploration**, which positions cross-filtering as a "direct manipulation" paradigm where users (not algorithms) control the analytical flow [65, 218]. The fundamental value of cross-filtering is that the user, not a model, agent, or algorithm, can gain a sense of the relationships present in a dataset.

Our work intends to adapt these principles to tactile interaction, enabling blind users to engage in the same iterative, hypothesis-driven analysis that sighted users perform visually.

6.4 Prototype: The cross-feelter

We contribute the design and schematics of our prototype hardware device, the *cross-feelter* as well as a prototype cross-filtering environment for testing our device. We have made all of our code open source at our Github repository [REDACTED] in addition to our supplemental materials: the Arduino (C++) code for the hardware, Bluetooth and USB serial protocol interfaces (in JavaScript), as well as our visualization interface code (also JavaScript). Our binary STL files for 3D printing our case, knobs, and rail cover are also available in our supplemental materials and on thingiverse [REDACTED]. We also intentionally designed the *cross-feelter* to be relatively

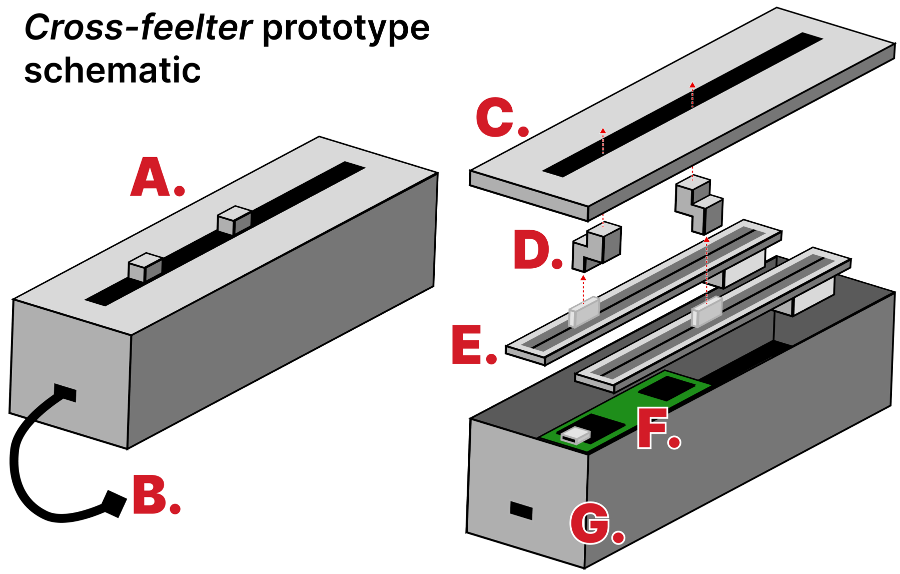


Figure 6.2: A. Our prototype *cross-feelter*. B. USB output (to computer). C. 3D printed rail cover. D. 3D printed knobs that consolidate rails into one. E. 2 motorized linear potentiometers. F. Arduino board. G. 3D printed casing.

cheap and easy to produce, costing no more than $30 to 50 USD for all the necessary hardware components.

Our *cross-feelter* was designed independent of knowledge of existing projects like the *MADEAxis* and *Embodied Axes* projects [45, 223], however the core design is nearly the same: placing two straight-rail, motorized, linear potentiometers (commonly called "motorized faders" or "sliders" in sound design equipment) side-by-side (see Figure 6.2). We designed two separate housing configurations for our rails, one which exposes each rail independent of one another and a second that uses angled knobs in order to place the two rails along a single rail. (This addition was a novel part of our design which diverges from existing work.) Our merged-rail design ensures that the left handle of a spatial filter remains to the left while the right handle remains to the right, supporting more consistent functional semantics and easier use because they do not cross over one another (**G3**).

We chose to use rail-based faders for several reasons, largely based on our design goals. Firstly, the input is absolute, meaning the location of the primary tactor (the knob on the rail) directly corresponds to the resistance generated by the potentiometer. The knob then is an *object with permanence* and will remain in a meaningful position set by the user until acted on by the user or system (**G1**). This allows the user to easily query the system state through touch.

In addition to the knob location, the rails have bounded ends; they *terminate*. The housing we designed for the rails to rest in have a strong edge, which is inset with enough distance to be easily felt (guidelines recommend a minimum of 4mm [11]). The tactile terminals again allow for a user to not only know the state of the knob, but the state of the knob roughly relative to each end of the rail (**G1**).

Within our prototype system design (see Figure 6.1), we use `aria-live` announcements to give auditory feedback when our *cross-feelter* is used with a screen reader. `aria-live` is a standard that will provide asynchronous announcements to a screen reader to notify them of changing content that may or may not be their currently focused element. We set our `aria-live` to use the `polite` setting so that it will not interrupt existing screen reader announcements that might be present. This enables us to provide our user with both tactile approximation of their input (as it relates to the spatial filter) as well as auditory (precise) feedback of their input location (**G2**).

In our system design, we used *Mosaic* [101] to scaffold an environment that handles the construction of an efficient multi-dimensional cross-filtering environment (**G4)**. *Mosaic* also automatically constructs an interaction data cube, which essentially prepares interaction results in advance, to support fast interaction over large and complex datasets (**G5**). In addition, *Mosaic* has an easy to use interface for specifying and leveraging a brush filter interaction layer over the dimension of a visualization that is relatively simple to set up to handle "undo" actions as well as programmatically controlling (**G5**).

Our prototype system relies on a refreshable tactile braille display for tactile output, the *Dot Pad* [55], which has an array of 60x40 pins that can raise or lower in a binary on or off state (**G2**). This allowed us to create fast rendering on-the-fly during user interaction, since our *Dot Pad* could render a new tactile arrangement in 800 to 1200ms from the moment our *cross-feelter* was used. Compared to printing with a higher-fidelity braille embosser (which takes 3 to 4 minutes per page), we chose speed over resolution to suit our goals (**G5**).

In our fully-built analytical environment (see Figure 6.3), we have a visual representation shown of the target "pixels" that are sent to our *Dot Pad* (these are not actually pixels, but the metaphor of a bitmap of pixels neatly applies to a bitmap of braille pins). This not only allowed us to debug the system but also illustrates how much more progress could be made for the resolution of refreshable braille displays.

Our device uses an *arduino* chip to communicate via the *usb-serial* interface that is part of common operating system architectures. We wrote an API for our device that writes and reads every 5ms. Our device emits a message whether the rail is being touched and the rail's current position. Our API then would listen to commands that it should move and, if not being touched currently, would slide into position. This simple API enabled full user control over the device and was capable of up to 1024 units of precision, which maps well to most uses in a cross-filtering analytical context (**G6**).

When initialized, our interface would set the visual filter between 0% and 100% and emit this to our *cross-feelter*, which then set the left knob to 0% and right knob to 100%.

6.5 Evaluation

Our evaluation engages our final two research questions:

- **R2** (Quantitative: Objective) In what measurable ways can we observe improvements to blind data interaction using our tactile interaction approach?
- **R3** (Quantitative: Subjective) What are the self-reported benefits, opportunities, and challenges that blind people experience when using our tactile interaction approach?

We set out to evaluate differences in the performance of cross-filtering tasks using our *cross-feeler* prototype versus not. To engage these research questions, we designed a 2x2, within-subjects, Latin square evaluation study where we asked participants to perform two tasks: a *structured data-relationship task* and *open-ended data exploration*. We conducted our study in one-on-one, single, 90-minute sessions in-person either in our research lab or at the location of the participant's choosing, as approved by our IRB.

Our overall procedure was roughly as follows (for our A-group participants): (**1**) tech set-up, initial interview, introduction to tactile graphs and cross-filtering, and a single Likert-scale question (*20min*), (**2**) an introduction to using our *cross-feelter* device to manipulate a video feed, followed by a short semi-structured interview (*15min*), (**3**) while using our *cross-feelter*: a cross-filtering data task of a dataset, an exploratory session of that dataset, and short Likert-scale interview (*20min*), (**4**) without using our *cross-feelter*: a cross-filtering data task of a different dataset, an exploratory session of that dataset, and short Likert-scale interview (*20min*), (**5**) lastly, we ended with a semi-structured interview about our participant's challenges, opportunities, and feedback when using our prototype device (*10min*).

For our B-group participants, their procedure was (**1**), (**4**), (**2**), (**3**), (**5**) instead. All participants were also divided in half again: C-group participants used a generic flight dataset followed by our custom local dataset, while D-group participants used our local dataset followed by the flight dataset. Alternating both the order of exposure to our *cross-feelter* and the datasets allowed us to test the effect of our device.

6.5.1 Participants

We conducted our study with 15 total participants, 8 who did not have existing professional expertise working with data and 7 who did. 7 of our participants identified as male, 8 as female.

All 8 of our participants without existing data expertise, in addition to 2 of our participants with existing data expertise, were recruited locally through existing relationships and social networks. We reached out to local participants via email and in-person at community events.

For our remaining 5 participants with data expertise, we coordinated our studies to be conducted in person at mutual conferences. We found it difficult to recruit blind data scientists, analysts, and engineers locally; however, we have a wider network of connections outside of our local area. We took advantage of travel and reached out to our remaining participants via email and in person.

Participants were screened with a short survey for eligibility: we recruited only participants who identified as blind, had screen reader expertise, were 18 or older, and US citizens or permanent residents. The chosen participants were informed of the risks associated with our study, notified in advance that they would be compensated with a $50 gift card for participation, and that consent to participate is voluntary.

6.5.2 Data-preferences workshop

We held 2 preliminary study sessions to test our methodology, polish our process, and find bugs before running the study in full.

However, we noticed that our preliminary participants were not particularly interested in the data we were using, so we then held a community workshop at our local library of accessible media (a braille and talking book library that services our state). In attendance were 21 members of the community and we discussed which possible datasets they would be most excited to interact with.

We broke into small groups and discussed things from local to state politics, economics, sports teams, weather, climate, barber shops, rent prices, and more. At the end, members voted on which datasets were most interesting to them and they enthusiastically chose rent prices across the city by neighborhood and "fish fry locations" across the city, also organized by neighborhood. Both of these were among the highest voted datasets that are readily available. For context, on what a "fish fry" is: our city has a tradition of frying fish during the Catholic lent, during which local businesses and even non-Catholic groups participate. There is a well-curated dataset made available every year.

6.5.3 Introduction to tactile histograms and cross-filtering

When our session with a participant began, we again gathered consent from them (this time spoken) to record our session. We had zoom running with us to coordinate recording audio, the computer's shared screen, and the video feed from our webcam, which captured a top-down view of the desk or working surface that we had available. We also made sure to capture our participant's hands as they interacted with our prototype environment. We provided a laptop for participants to use, but if they had an internet connection on a preferred device and had it present, they could connect and opt in to using their device for recording and conducting the experiment instead. All participants except for 3 used our provided laptop.

We then asked our participants a single 5-point Likert-scale question, "On a scale of 1 to 5, 1 being very confident and 5 being very anxious, how anxious are you to try exploring data?" We followed up with a short interview about their experience working with data (professional or otherwise), familiarity with charts, and familiarity with tactile graphics.

Next we had a scripted introduction where we explained how to read a tactile chart and introduced our participants to a histogram, as well as the statistical concept of a histogram. We described how multiple histograms could share a single source dataset but represent different variables. We then showed participants 3 histograms of different variables from the same dataset (a fake, example dataset about fruits with "price to grow," "distance from market," and "price to purchase" as variables in histograms).

Lastly, we held 2 sheets of paper over one of the histograms to introduce the concept of "spatial filtering," and the idea of "cross-filtering," where keeping a narrow perceivable section of a data set could be used to change the shape of the other 2 histograms. We explained that filtering is a way for people who are analyzing data to ask questions more than it is to find answers, that it is a way to "look for possible relationships that might exist in the data." We discussed different filtering scenarios and showed participants different results before and after filtering such as, "if we filter on the price of buying fruit to look at the cheapest fruits to buy, we can compare the before and after of the fruit distance histogram." We showed them how to read a histogram that changes from one shape to another.

Just to gauge their interest and look for more opportunities to help teach them how to cross-filter, we asked them two follow-up questions, "why do you think fruits from far away would cost less?" and "what other possible relationships would you want to explore in this data?" We ended with an open discussion with participants about filtering, data, and any additional questions they had before proceeding with our study.

6.5.4 Preliminary video interaction

Before participants used our *cross-feelter* device on data, we had them get used to how it worked by manipulating a video. We linked our device to the video, gave them simple instructions about how the device worked and how to operate it, and then let them explore the video. We asked them a simple information-retrieval task about the name of the person in the video recording, who announced their name early on. We then used this simple information retrieval task to ask them some qualitative questions about how their past experiences interacting with video and audio media using a screen reader compared to using our device, specifically when trying to find information.

For the video interaction, we did not use our merged-rail casing but instead exposed both rails. Our top rail corresponded directly to the volume of the video, and tapping the knob on the top rail would start or stop the video. The bottom rail would track the video's progress when not being touched. Otherwise, if the user touched the rail, they could quickly track the video to a different position.

6.5.5 Dataset-focused evaluation

The first major activity participants performed was to cross-filter a given dataset and give an answer to a question within three minutes. This activity was followed by a chance to openly explore the data using cross-filtering for 10 minutes. These exercises were accompanied by a graphical interface of 3 histogram visualizations derived from different variables in a shared dataset as well as a refreshable tactile braille display. The braille display would only display one chart at a time, but users could change or select the output of the display using a screen reader.

Participants were then given a series of 5-point Likert-scale questions in the following order about their experience, each question starting with the prefix, "On a scale of one to five..."

1. "How stressful was it to try to answer the data question earlier?"
2. "How much cognitive load, or mental effort, did you feel you had to dedicate to answering the question and then exploring the data after?"
3. "How much anxiety do you have about your own ability to do a timed task again in the future using the same tools you just used?"
4. "How enjoyable was it to do this task and explore the data?"

After this, we informed participants of whether or not they correctly answered the question given to them during their task and we repeated the data task and exploration, but with a second dataset. We randomized our participants so that half used our prototype for the first dataset while half used our prototype for the second dataset. Without our prototype, participants used a screen reader and our refreshable tactile braille display.

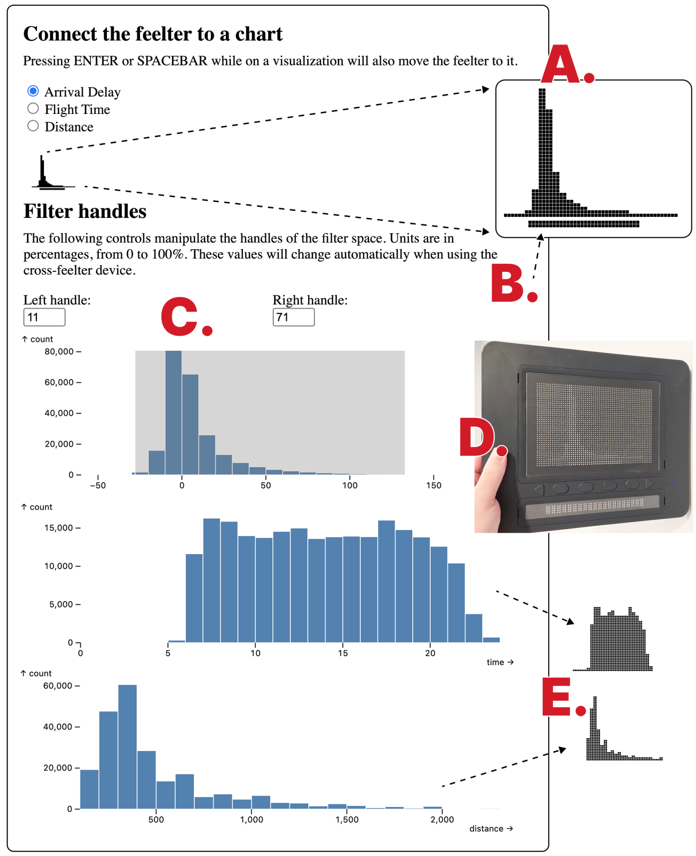


Figure 6.3: Our analytical environment. A. a 1 to 1 translation between visual and tactile that represents a 60x40 pixel-to-pin array. B. A tactile version of the filter location (if on the chart being focused). C. Cross-filtering controls, including text inputs for screen-reader only manipulation. D. Refreshable tactile display output. E. Example renderings of the other 2 charts.

6.5.5.1 Structured data-relationship task

For the structured data-relationship task, we told our participants that they would be asked to perform a specific filtering task and answer a multiple choice question verbally within three minutes. We told participants that we would only repeat the question and options if they asked, but otherwise we would not intervene or help them in any way during the test of our interface.

For the dataset of flight information, we asked participants, “What is the relationship between filtering for flight distances and the arrival time of flights?” The correct answer was “longer flights tend to arrive earlier compared to all flights.” Their options were, (**a**.) longer flights tend to arrive later compared to all flights, (**b**.) longer flights tend to arrive earlier compared to all flights, and (**c**.) longer flights tend to arrive at a similar time compared to all flights.

For our dataset built on local neighborhood data, we asked participants, “What is the relationship between filtering for number of fish fry locations and the rent price of a neighborhood?”

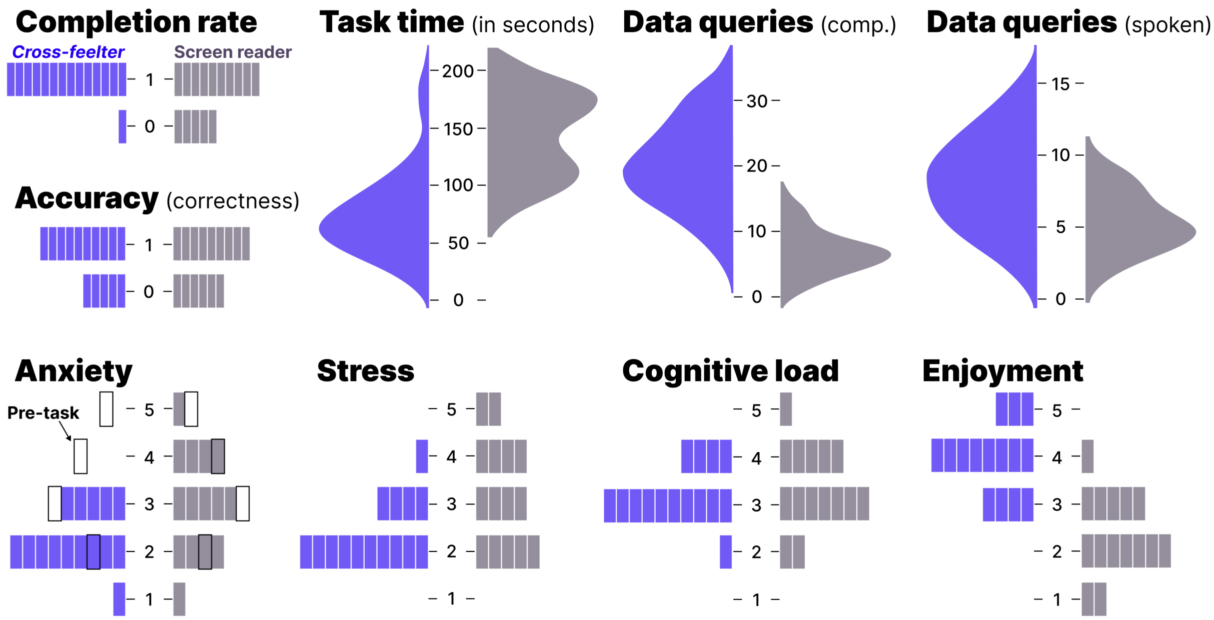


Figure 6.4: Results. Completion rate and accuracy are counts. Task time, data queries (computational), and data queries (spoken) are objective, observational measures. Anxiety, stress, cognitive load, and enjoyment are Likert-scale results from 1 ("very little") to 5 ("quite a lot").

The correct answer was "neighborhoods with more fish fry locations tend to have higher rent prices." Their options were, (**a**.) neighborhoods with more fish fry locations tend to have higher rent prices, (**b**.) neighborhoods with more fish fry locations tend to have lower rent prices, (**c**.) neighborhoods with more fish fry locations tend to have similar rent prices compared to all neighborhoods.

Our interface had the question and choices written out, with the choices listed in random order. We read the answers out loud in the order they were listed by our interface.

We measured task completion rate, task completion time, the precision of the spatial placement of their cross-filtering task, and the accuracy of their answer to the task's question.

6.5.5.2 Open-ended data exploration

After the participants performed their three minute task, we asked them to perform an open-ended, cross-filtering data exploration exercise where we encouraged them to look for relationships in the dataset and "ask out loud as many questions of the data as they could think of," while they worked, as a prompted, concurrent think-aloud method [4]. Before beginning, we explained that the purpose of the exercise, "is not to answer questions but to come up with questions. This is how many data scientists explore datasets they are unfamiliar with: by looking for possible relationships. Do low values here affect other values over there? And so on." We let participants know that we were taking notes on which questions they had already asked, in case they wanted to know which ones weren't yet asked.

In this second session, we measured the count of computational queries they made using the interface as well as the number of spoken queries they had of the data.

Our method for measuring a computational query was straightforward: if they manipulated the cross-filter at all and changed its values and then used the tactile display to feel what happened to one or both of the other charts, we counted it as a computational query. By comparison, changing the filter values without checking the output did not count.

For spoken queries, we did not expect any formal framing. We encouraged them to ask anything, even if they weren't sure if they would be able to answer it using cross-filtering. We recorded any query they spoke aloud that was related to the data.

6.5.6 Interview

The final portion of our study was a short, semi-structured interview where we asked them about their overall experience using our *cross-feelter* device, especially compared to using a screen reader alone. We also elicited feedback on the data interface and environment as a whole as well as ideas for future, additional applications where either our approach or *feelter* device would be useful.

6.6 Results and discussion

Our quantitative analysis compared the performance and subjective responses of participants using our *cross-feelter* device (**CF**) versus a traditional screen reader (**SR**) across a series of quantitative metrics. Our study employed a randomized, within-subjects design, ensuring that the order of conditions and task assignments was counterbalanced. We evaluated objective measures such as task completion, time to complete, accuracy on a multiple-choice question, filter precision, data queries during an open exploration task, and spoken queries. We also evaluated subjective measures including anxiety, cognitive load, enjoyment, and stress assessed via 5-point Likert scales.

We conducted our analysis in Python and have made our code and de-identified data available in our supplemental materials.

For our qualitative analysis of our final interview, we collated our notes and transcript materials, coded them thematically, and then used affinity diagramming to group themes [99]. Our final interview was largely informative for improving our prototype and contributing to our future use cases (see Section 6.7).

Our results indicate that our *cross-feelter* device not only enhances cross-filtering for performing specific data tasks but also for open data exploration compared to a screen reader experience. To highlight our strongest results: our approach significantly improves the speed of completing a cross-filtering data task (90% faster), the quantity of computational data queries performed during data exploration (188% increase), and the enjoyment of the analytical experience (with even stronger results observed with our non-data experts).

6.6.1 Analysis of objective results

6.6.1.1 Significantly improved task completion rate and time

For the timed data task, participants achieved a significantly higher completion rate with **CF** (mean = 0.93, SD = 0.26) compared to **SR** (mean = 0.60, SD = 0.51), with a paired t-test indicating significance (t = 2.646, p = 0.019). Similarly, the time taken to complete the task was significantly lower in the **CF** condition (mean = 75.53 s, SD = 36.14) than in the **SR** condition (mean = 143.93 s, SD = 34.65), with a highly significant difference (t = -6.876, p ¡ 0.001).

Note that for participants who did not complete the timed task, we recorded 180 seconds for their time, meaning an unbounded test may have yielded an even larger gap between our **CF** and **SR** conditions.

6.6.1.2 Similar test score accuracy

Accuracy and precision have a more nuanced set of results that we want to unpack. First, we observed the accuracy rates for correctly answering our multiple choice questions (out of 3 options). While our **CF** condition (mean = 0.67, SD = 0.49) outperformed **SR** (mean = 0.53, SD = 0.52), this difference was not statistically significant (t = 1.468, p = 0.164).

However, it is important to note that participants who were unable to complete the task in the given time automatically received a score of 0 (for failure). So, if we filter our data for **CF** and **SR** observations where the participant successfully completed the task, we find that the mean of our **SR** condition (mean = 0.78, SD = 0.44) actually outperformed our **CF** condition (mean = 0.71, SD = 0.47). While this is worth further investigation, these results are not statistically significant and we would require more data (t = 0.5547, p = 0.5943).

6.6.1.3 Similar filter precision

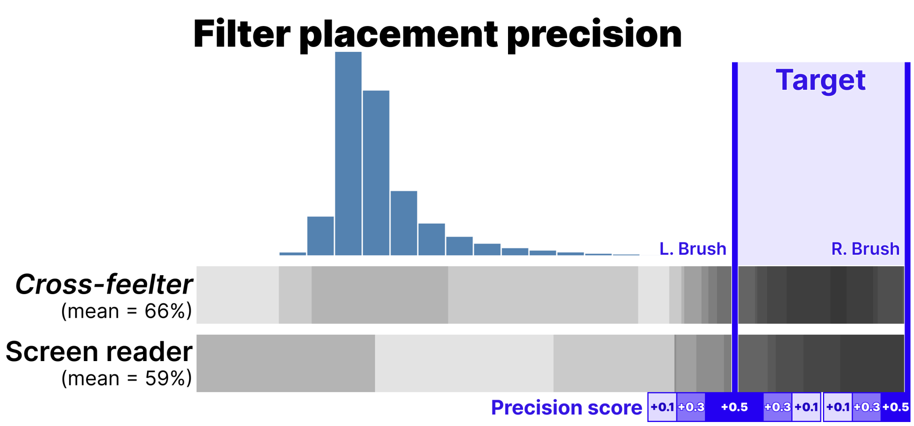


Figure 6.5: Precision heatmap of every filter placed by a participant before completing their task. An example histogram is shown from our data task. Precision was scored based on distance (in 4% increments) from the target left (75%) and right (100%) filter brush handle locations.

We also measured the precision of how our participants placed the filter right before they determined their answer. We visualize every filter position overlaid on an example histogram from our data environment between the **CF** and **SR** conditions (see Figure 6.5).

Our precision calculation was a scoring system, where the expected filter parameters for both tasks were to set the filter between 75% and 100% in order to successfully answer the question. We awarded 0.5 points if the left and right filters were within $\pm 4\%$ of 75% and 100%, respectively. 0.3 points were awarded if within an additional $\pm 4\%$ and 0.1 points were awarded if within yet another $\pm 4\%$ from there.

For the precision of the positioning of the filter handles, our **CF** condition's precision (mean = 0.66, SD = 0.37) was only marginally higher than **SR** (mean = 0.59, SD = 0.34). However, this difference was not statistically significant (t = 0.617, p = 0.547).

Observationally, we also note that the **CF** condition resulted in more naturally spread values, while the **SR** condition used almost exclusively whole multiples of 5 (note the second sub-figure of Figure 6.5). We were surprised that participants were able to attain similar (if not slightly better) precision when using our prototype device because text input with a screen reader is typed out explicitly.

6.6.1.4 Substantially increased quantity of data queries

Notably, the **CF** condition elicited a substantially greater number of computational data queries (mean = 20.13, SD = 6.45) compared to **SR** (mean = 7.00, SD = 3.18; t = 9.567, p ¡ 0.001), as well as significantly more spoken data questions (**CF**: mean = 8.20, SD = 3.14 vs. **SR**: mean = 5.33, SD = 1.99; t = 3.765, p = 0.002).

We believe that this result, coupled with our results for task completion time, are the best outcomes of our study because combined they strongly support our principled design goals of *fast, dynamic, reversible filtering* (**G5**) and *user-driven exploration* (**G6**).

6.6.2 Analysis of subjective results

As part of our analysis of subjective results, we note that the results varied between our subjects who had professional data expertise and those who did not. In each section below, we discuss our findings between the two conditions, as well as individually for each group if relevant.

The non-parametric, Likert-scale differences between our **CF** and **SR** conditions were evaluated using Wilcoxon Signed-Rank tests (with score marked below as W), and summary descriptive statistics (mean, median, standard deviation, minimum, and maximum) were computed for each measure. Our analysis files contain our full details, but we report the mean and median statistics below.

6.6.2.1 Reduced stress of task performance

In terms of subjective experience, stress ratings were notably lower with our **CF** condition. Descriptive statistics indicated that **CF** participants had a lower mean stress score (mean = 2.40, median = 2.0) compared to SR participants (mean = 3.20, median = 3.0). This reduction in stress was confirmed as statistically significant (W = 4.000, p = 0.013).

However, our data experts did not feel particularly stressed between either condition (**CF**: mean = 2.14, median = 2.0 vs. **SR**: mean = 2.57, W = 0.000, p = 0.083), while our non-data participants had much stronger results (**CF**: mean = 2.62, median = 2.5 vs. **SR**: mean = 3.75, median = 4.0; W = 2.500, p = 0.047).

6.6.2.2 Just-insignificant difference for cognitive load

For cognitive load, while there is an observed difference between means, the the **CF** (mean = 3.13, median = 3.0) and **SR** (mean = 3.47, median = 3.0) conditions were just barely statistically insignificant (W = 4.000, p = 0.059). Our interpretation is that both conditions experienced comparable levels of cognitive load during task performance.

6.6.2.3 Lower anxiety associated with future task performance

Anxiety was measured before we began, as an additional baseline of comparison. In the following, we compare 3 pairs of tests, each condition against the starting anxiety variable and then the two conditions against each other.

Participants in the **CF** condition reported significantly lower anxiety compared to those in the **SR** condition. Specifically, the **CF** condition had a lower mean anxiety score (mean = 2.27, median = 2.0) relative to the **SR** condition (mean = 3.00, median = 3.0). This difference was statistically significant (W = 10.000, p = 0.017).

Against starting anxiety, our **CF** condition was also significantly lower and statistically significant (W = 5.000, p = 0.003). However, the **SR** condition was not (W = 8.000, p = 0.132).

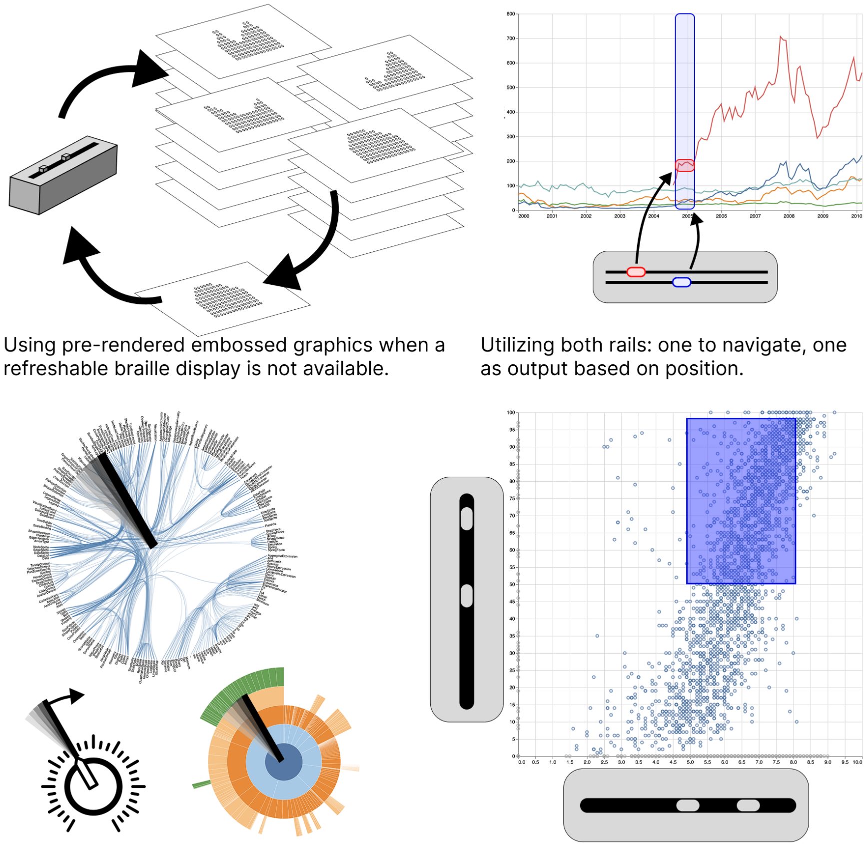
6.6.2.4 Greatly improved enjoyment

Finally, enjoyment ratings were substantially higher in the **CF** condition than in the **SR** condition. The **CF** condition yielded a higher mean enjoyment score (mean = 3.93, median = 4.0) compared to the **SR** condition (mean = 2.33, median = 2.0). This difference was highly statistically significant (W = 0.000, p = 0.001).

For our participants without data experience the **CF** condition (mean = 3.875, median = 4.0) yielded an even higher difference between means compared to the **SR** condition (mean = 2.000, median = 2.0). This difference was statistically significant (W = 0.000, p = 0.015).

6.7 Future use cases

We intentionally focused the bulk of our qualitative interview on speculative futuring [58], eliciting ideas from our participants for additional data-centered applications of our prototype or general approach. Our participants with professional data expertise were especially motivated to come up with concrete ideas, some of which we have followed up on after the conclusion of our study. We have discussed and iterated on our future use cases, which we present below. All of which are roughly demonstrated in Figure 6.6.



Using pre-rendered embossed graphics when a refreshable braille display is not available.

Utilizing both rails: one to navigate, one as output based on position.

Using knobs, joysticks, and other tactile input that suit different data types.

Cross-feeltering, brushing, zooming, and panning in 2 or more dimensions.

Figure 6.6: Speculative future use cases for our approach and *feelter* device.

6.7.1 Hi-res tactile interaction cube

One idea which has come up in conversation with our larger tactile graphics community as well as our participants is using our *cross-feelter* in an environment without a refreshable braille display. Refreshable braille displays were fundamental to our research, especially related to our findings focused on speed, since the display can render on the magnitude of 800-1200ms. However, these devices can be expensive, ranging from $8k to $20k USD.

So in the absence of a refreshable braille display, it would be possible to use the *cross-feelter* with a braille embosser to print the output of a filter action. However, this *embosser-in-the-loop* approach would require waiting for 3 to 4 minutes per interaction.

Instead, we have discussed the possibility of "pre-rendering" interaction outputs based on a given dataset, where we print an "interaction cube" of a set of possible interactions in advance, in the same manner that *Mosaic* pre-computes a data cube for fast cross-filtering. We could either bin interaction ranges with the filter so that there are a controllable number of output possibilities or even just print generic histogram distributions and recommend a "best fit" given a particular input on a dataset. Either way, analysts could have a *booklet* of tactile graphs next to them and turn to a specified page after an interaction with the *cross-feelter*, potentially reducing the interaction loop from 3 to 4 minutes to 10 seconds or so.

6.7.2 *Feelter*-only 2-dimensional chart interaction

Another future idea we have discussed with our participants is using *only* a cross-feelter to navigate and interact with a chart.

Presently, serial navigation of large data can be slow and tedious [64]. But most charts and graphs, as well as datasets, have a finite number of observations in a given dimension. These finite bounds could be mapped to a position along the *feelter*'s slider rail, allowing someone to quickly slide through a set, list, array, or dimension of data. Additionally, the second rail could be encoded to the domain extents in a quantitative dimension such that during navigation, the second rail's motor would move the knob in a corresponding position according to the data value currently focused by the first rail's navigation.

6.7.3 Additional modalities and inputs

Sonification could also be used in conjunction with cross-filtering as output or by using rail navigation to brush the sonification. Navigating and "wayfinding" within sonifications is currently a difficult challenge, even on touch-enabled devices through brushing. This is largely due to the lack of *objects with permanence*, making it difficult for a user to query system state by touching a tactor's position.

Other input devices could also be leveraged, ones that have functional semantics that match more appropriately to different types of data. For example, a dial could be used to navigate circular data (like seasonal data). A controller joystick could navigate data with an infinite range (such as a function or model).

6.7.4 Multi-dimensional panning, zooming, and brushing

Lastly, our participants and team discussed using two *cross-feelter* devices to perform spatial panning, zooming, and brushing in combination with tactile maps or scatterplots and even in combination with non-tactile, digital maps (which are navigated using a screen reader or equivalent).

Presently, "pinch and zoom" strategies lack an easy way for users to determine the state of a zoom interaction (the same problem as mentioned in the previous section, due to lack of an *object with permanence*). But having one rail pair arranged as an "x axis" *feelter* with another rail pair as a "y axis" *feelter*, a wide variety of multi-dimensional interactions could be possible. It was even suggested that rails could be used as system output when a user has location-related questions, such as "where is this object?" on an image or map. The rails could position into x and y locations that correspond, to help someone iteratively build a sense of spatially understanding a layout.

6.8 Summary

Research in data interaction accessibility for people who are blind often focuses on the existing capabilities of screen readers or computers, through improved textual descriptions, language-based approaches, and sonification. Additionally, existing work focuses heavily on output methods and interaction as a means to understand computational output.

We contribute a quantitative study that demonstrates that our novel *cross-feelter* hardware prototype significantly enhances blind data interaction compared to a traditional screen reader. Participants using our cross-feelter achieved higher task completion rates and faster completion times while producing substantially more data queries during exploration. Subjective measures indicate that participants experienced lower anxiety and stress, along with greater enjoyment. These subjective results were especially strong for our participants who lacked professional data experience. Our subjective findings highlight the potential for our approach to improve how blind practitioners learn and become more comfortable with data analysis.

In this paper, we have presented a principled design approach for blind tactile data interaction, where interaction is framed not only to aid perception but ultimately to aid a user in accomplishing data analysis goals and outcomes. We argue that what computers and screen readers are currently capable of is not enough; we need more innovation in novel interaction hardware.

We call on both the visualization and accessibility research communities to further refine tactile interaction modalities and explore broader applications in accessible data analysis.

Chapter 7
(Proposed Work) *Skeleton*: Visual Tooling for Non-visual Data Experiences

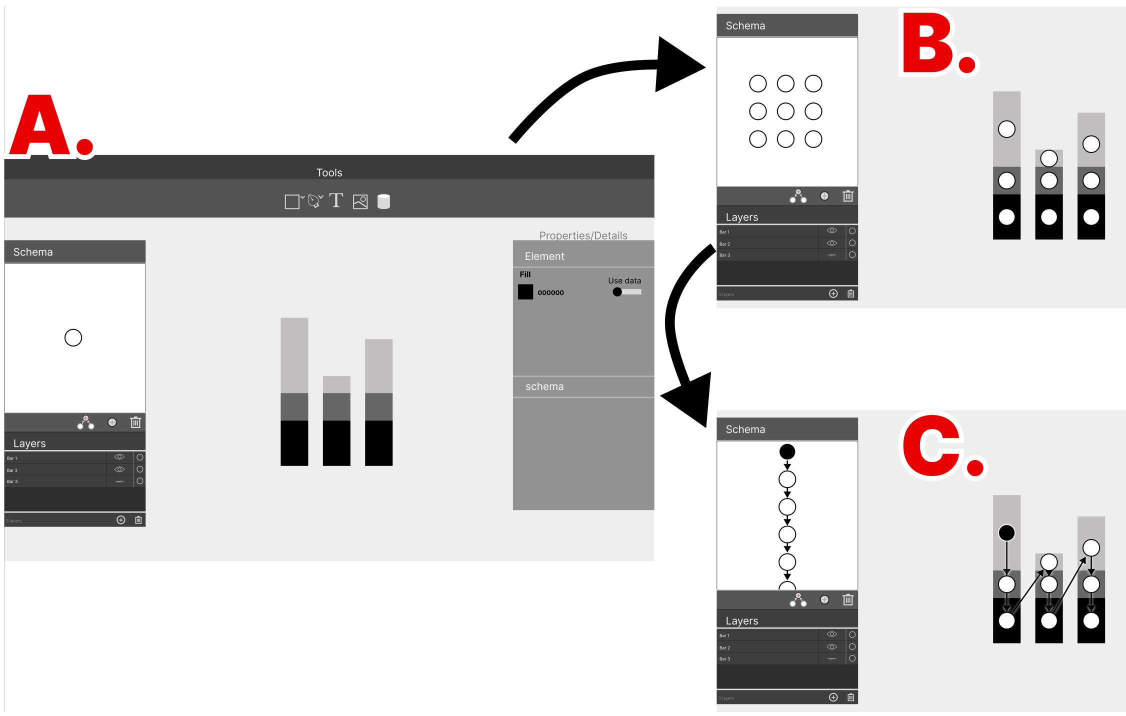


Figure 7.1: Low-fidelity design draft of *Skeleton*'s main user interface components and interactions. A. *Skeleton*, our graphical user interface for creating and debugging screen reader navigation experiences of data visualizations. B. Users can add nodes wherever they want over the chart, manually or automatically with algorithmic assistance. C. Users can then ”draw” edges between nodes, which signify navigation paths through the visualization.

7.1 Abstract

In our previous work we built Data Navigator, a tool for assembling non-visual data interfaces. Despite the existence of this tool, key gaps still persist: Ironically, making visualizations accessible in non-visual ways poses accessibility barriers for *sighted* practitioners, who may not be able to easily understand, build, and debug screen reader experiences for people who are blind.

This research proposes *Skeleton*, a tool that visualizes non-visual navigation paths, semantic structures, and screen reader interactions in data visualizations. By rendering these invisible experiences visually, Skeleton seeks to bridge the cognitive gap for sighted practitioners, enabling them to more easily inspect, debug, and author accessible experiences during development.

To evaluate our work, we will provide a sandbox prototyping experience to 12 sighted designers and observe how they work to make their own diagrams, maps, and bespoke visualizations accessible to navigational assistive technologies, in addition to performing a debugging task for a screen reader experience of our design. We will then conduct a qualitative interview with our practitioners, using our sandbox session as a probe to help us learn more about what ways a visual tool that constructs non-visual experiences could be improved.

7.2 Introduction

Data visualizations are a powerful way to gain insight from data, which is an increasingly important activity in the present context of our data-driven world. Decision-making in personal, industry, research, and governmental contexts relies on access to data and visualizations. Excluding people with disabilities from access to data creates gaps in job opportunities, civic engagement, and quality of life. However, making data visualizations accessible for users with disabilities remains a difficult task for designers, especially those without disabilities. Sighted visualization designers, as one example, must first gain an understanding of ways that their current designs produce barriers for people who are blind, then they must either learn to use new tools or learn to adapt the tools that they have in order to build more accessible artifacts. But even while building artifacts, most development tools focus on the authoring and debugging of visual experiences. The design, inspection, and development of non-visual experiences presently requires the practitioner to conduct non-visual tasks using audio-based tools, such as a screen reader. Our research aims are to study technological interventions on practitioner work in the domain of data visualization, towards more accessible outcomes for people with disabilities who use and interact with the data interfaces and artifacts that practitioners produced. We assert that practitioners who design and develop interactive data visualizations are the last people responsible for the exclusion of people with disabilities. We aim to examine how resources, techniques, and tools can shift the behavior of these practitioners.

However, some of the gaps in our knowledge center on what barriers practitioners themselves face. Do they lack knowledge about what is or is not a barrier that excludes people with disabilities from participating in the use of a data visualization? Would they be more successful with this knowledge? Do practitioners lack the correct building blocks and substrates for creating visualizations that are accessible? Would they be more successful with better tools and materials? Or perhaps do practitioners lack consideration for the ways that end users want to customize or fit their experiences? Would they be more successful if they had a better understanding of user personalization needs?

In prior work, we explored these gaps. Our initial project, Chartability, provided practitioners with a framework of heuristics for evaluating the accessibility of interactive data representations that they were authoring, so that they would have the knowledge to produce more accessible visualizations. We then developed a software tool, Data Navigator, which provides more robust building blocks for practitioners to assemble non-visual, navigational experiences for users of assistive technologies. Our *Softerware* project explored how system developers can build interactive data representations that end users are able to manipulate and customize to suit their accessibility needs. And our latest project, a novel physical input device called the *cross-feelter*, improved the speed of task completion and quantity of data queries of blind people when exploring and analyzing data.

However, gaps in our knowledge of practitioner needs still remain. In particular, we believe that practitioners without disabilities still face barriers when interpreting their own design and development decisions for users with disabilities.

In our proposed work, we will continue to intervene on practitioner work through tools. We conjecture that designing non-visual experiences remains a difficult task for sighted practitioners because practitioners lack visual representations and translations of non-visual experiences. We propose Skeleton, a tool which visualizes the navigation paths, descriptive text, and functional semantics leveraged by screen readers. We will apply Skeleton to the domain of data visualization, allowing practitioners to automatically generate, inspect, debug, and author the experiences that screen reader users will have of charts, maps, and diagrams.

Our research questions for this work are to explore the ways that practitioners use and make decisions with a tool that visually represents non-visual experiences: what conceptual barriers do practitioners face when visually authoring and debugging non-visual experiences? What ways can visual tools help facilitate practitioners to make more meaningful design decisions for non-visual computational interfaces? How might practitioners envision being able to find and fix accessibility barriers in the process of their authoring and development of interactive data visualizations?

To first test the effectiveness and usability of Skeleton, we will provide a sandbox prototyping experience to 12 sighted designers and observe how they work to make their own diagrams, maps, and bespoke visualizations accessible to navigational assistive technologies. We will also then provide our participants with a data visualization that intentionally includes several accessibility barriers and observe how practitioners identify, debug, and remediate those barriers. To understand tooling needs, we will then conduct a qualitative interview with our practitioners, using our sandbox session as a probe to help us learn more about what ways a visual tool that constructs non-visual experiences could be improved.

7.3 Approach

First our approach will be to investigate existing tooling that explores making non-visual experiences visual, and if any existing projects explore this with accessibility and software development goals in mind.

Second, we will develop a graphical user interface for *Skeleton* that builds on *Data Navigator*'s underlying graph-based substrate for scaffolding navigable, interactive data structures (see Figure 7.1).

Users will be able to edit nodes manually (see B. in Figure 7.1) by directly clicking on visual ”add nodes” interface elements and then directly clicking on their visualization where those nodes should exist. The exact location of nodes is shown in the main element inspector while the schema (abstract representation) of those nodes will be reflected in a schema inspector (shown on the left in Figure 7.1). Nodes that are selected in either view will be shown in high contrast (black) as selected in both views.

And users will be able to add edges using our edges drawing tool. This will enable users to

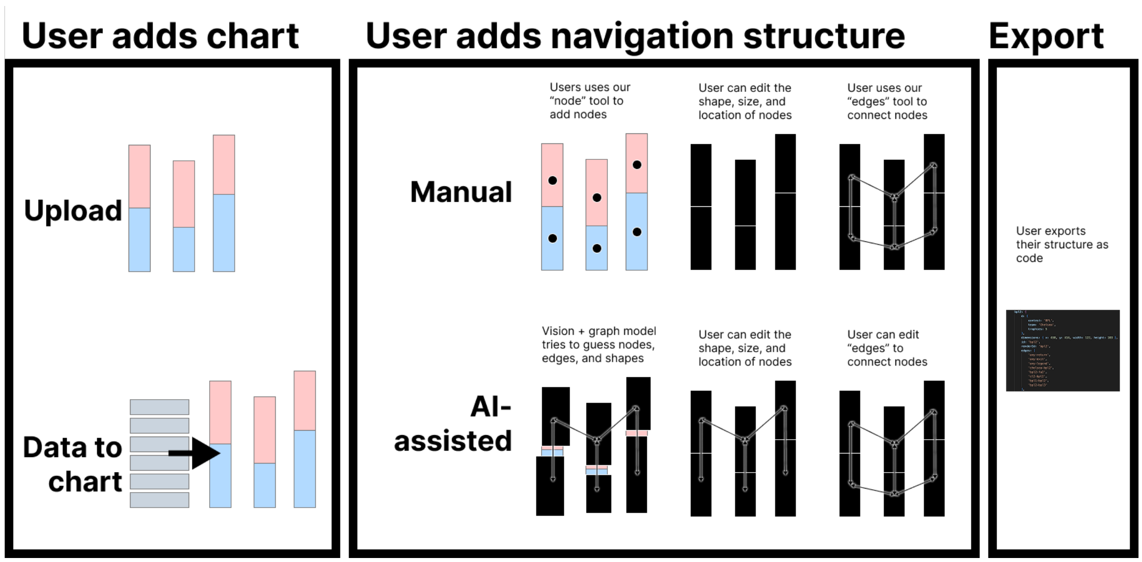


Figure 7.2: Design draft of user flow in *Skeleton*. Users can upload an image or data. If they choose data, they can generate a chart using *Vega-Lite*. From there, users can choose to leverage AI-assistance if they want, in order to generate nodes (with shape) and edges for navigation. Whether or not, they will be able to manually edit and tweak the nodes, edges, and navigational paths of their visualization as well as export code for use in their own environment.

drag a visible line to connect between nodes or add connections between all nodes in a selection. Edges can be drawn between nodes in both the main view or the schema inspector (see C. in Figure 7.1). Rules for navigation (direction, etc) will be automatically generated, up to a limit, but users will always be able to edit these features using the Details Inspector (shown on the right side of the interface).

Our interface tool will allow 2 different kinds of input: (**1**) data, in a standard JSON format (array of objects) or (**2**) as a rasterized image (png, jpg, etc). If users upload data, they can choose to render a simple visualization from it using vega-lite. Users will then be able to choose to either manually or "automatically" build navigation (with AI-assistance). Even when using assistance, users will always be able to tweak and override the model's decisions (see: Figure 7.2). Lastly, users can export the code generated by the system for use in their own environments.

7.4 Evaluation

Our research will address three questions: (**1**) What conceptual barriers arise when practitioners use visual tools to design non-visual experiences? (**2**) How can visual representations of non-visual interfaces improve practitioners' design decisions? (**3**) How might practitioners identify and resolve accessibility barriers during visualization authoring?

We have designed a study to be executed in three phases. First, 8-12 sighted designers will use *Skeleton* in a sandbox environment to create accessible diagrams and maps, with researchers observing their workflow and problem-solving strategies. Designers will be encouraged to bring their own visualizations to try out in *Skeleton*, however a visualization will be provided in case they do not have one ready. Second, participants will be given a task to diagnose and remediate accessibility barriers in a flawed visualization provided by the research team. Finally, a qualitative, semi-structured interview will explore practitioners' tooling needs, usability feedback, and perceived impact of *Skeleton* on their design processes.

7.5 Timeline

- 5-25 Create study design (1w), submit IRB (1w), finalize design draft (1w), and begin building *Skeleton* (1w)
- 6-25 Build web interface + scaffolding (2w), build data ingestion pipeline (1w), add *Vega-lite* chart picker (1w)
- 7-25 Build image ingestion pipeline (1w), integrate computer vision model (1w), build node-edge editing interface (1w), add exporting functionality (1w)
- 8-25 Begin recruitment, run pilot study, and make final changes (1w), run study and analyze results (2w), start writing paper (1w)
- 9-25 Finish writing paper and **submit to CHI 2026** (1w), take time off (2w), Prepare documentation for open source (1w)
- 10-25 Prepare for VIS (2w), polish interface and make final changes (1w),add additional use cases to site (pdfs? websites?) (1w)
- 11-25 IEEE VIS and travel (2w), respond to CHI submission results (1w), prepare for job search (1w)
- 12-25 Job search, time off
- 1-26 Job search
- 2-26 Job search
- 3-26 Job search
- 4-26 CHI 2026, Job search
- 5-26 Write thesis
- 6-26 Thesis defense

7.6 Expected Impact

This research will provide empirical insights into practitioner challenges in accessibility-focused design and the role of visual aids in mitigating them. It will refine *Skeleton* to better align with practitioners' workflows, enhancing their capacity to build inclusive visualizations. A theoretical framework will also emerge, emphasizing bidirectional translation between visual and non-visual experiences in authoring tools, which we hope will inspire other environments (such as within PDFs and web development) to follow suit.

By equipping practitioners with tools to visualize and address the design and development of non-visual experiences of data, this work aims improve design practices, foster better design collaboration between sighted and blind individuals, and inspire new technological developments in accessibility tooling. Our outcomes could transform how accessible data interfaces are created, fostering greater equity in data-driven decision-making for users with disabilities. Aligning with broader goals of digital inclusion, our research ensures that data's transformative potential is accessible everyone, including people with disabilities.

References

[1] W. Aigner, A. Rind, and S. Hoffmann. Comparative evaluation of an interactive time-series visualization that combines quantitative data with qualitative abstractions. *Computer Graphics Forum*, 31(3pt2):995–1004, June 2012. 5.5.3.2

[2] O. Aishwarya. Performing qualitative data analysis as a blind researcher: Challenges, workarounds and design recommendations. In *Proceedings of the 24th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '22, page 1–4. ACM, October 2022. doi: 10.1145/3517428.3551356. URL http://dx.doi.org/10.1145/3517428.3551356. 6.1

[3] Frances Aldrich. Talk to the Hand: An Agenda for Further Research on Tactile Graphics. In Gem Stapleton, John Howse, and John Lee, editors, *Diagrammatic Representation and Inference*, Lecture Notes in Computer Science, pages 344–346, Berlin, Heidelberg, 2008. Springer. ISBN 978-3-540-87730-1. doi: 10.1007/978-3-540-87730-1_31. 3.2.1

[4] Obead Alhadreti and Pam Mayhew. Rethinking thinking aloud: A comparison of three think-aloud protocols. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, page 1–12. ACM, April 2018. doi: 10.1145/3173574.3173618. URL http://dx.doi.org/10.1145/3173574.3173618. 6.5.5.2

[5] Ofunne Amaka and Amber Thomas. The Naked Truth, 2021. URL https://pudding.cool/2021/03/foundation-names. Accessed: 2021-08-20. 3.5.3

[6] ARIA Working Group. ARIA Data Visualisation module · Issue 991 · w3c/aria — github.com. https://github.com/w3c/aria/issues/991, 2019. [Accessed 01-04-2025]. 6.2.3

[7] E. Bailey. Accessibility preference settings, information architecture, and internalized ableism — ericwbailey.website. https://ericwbailey.website/published/accessibility-preference-settings-information-architecture-and-interna
2024. [Online]. 5.4.2

[8] Catherine M. Baker, Lauren R. Milne, Ryan Drapeau, Jeffrey Scofield, Cynthia L. Bennett, and Richard E. Ladner. Tactile graphics with a voice. *ACM Transactions on Accessible Computing*, 8(1):1–22, January 2016. ISSN 1936-7236. doi: 10.1145/2854005. URL http://dx.doi.org/10.1145/2854005. 6.2.4, 6.3.1

[9] Catherine M. Baker, Lauren R. Milne, Ryan Drapeau, Jeffrey Scofield, Cynthia L. Bennett, and Richard E. Ladner. Tactile graphics with a voice. *ACM Trans. Access. Comput.*, 8(1), jan 2016. ISSN 1936-7228. doi: 10.1145/2854005. 3.2.1

[10] Abhijit Balaji, Thuvaarakkesh Ramanathan, and Venkateshwarlu Sonathi. Chart-Text: A Fully Automated Chart Image Descriptor. *arXiv Preprint*, December 2018. URL `http://arxiv.org/abs/1812.10636`. Accessed: 2021-09-06. 3.2.1

[11] BANA. Guidelines and Standards for Tactile Graphics. Braille Standard, Braille Authority of North America, 2010. URL `http://www.brailleauthority.org/tg/`. [Accessed 01-04-2025]. 6.4

[12] BANA. Guidelines and Standards for Tactile Graphics. Braille Standard, Braille Authority of North America, 2010. URL `http://www.brailleauthority.org/tg/`. Accessed: 2021-09-03. 3.2.1, 3.2.3

[13] Andrea Batch, Biswaksen Patnaik, Moses Akazue, and Niklas Elmqvist. Scents and sensibility: Evaluating information olfactation. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, page 1–14. ACM, April 2020. doi: 10.1145/3313831.3376733. URL `http://dx.doi.org/10.1145/3313831.3376733`. 6.2.3

[14] Michel Beaudouin-Lafon, Susanne Bødker, and Wendy E. Mackay. Generative theories of interaction. *ACM Transactions on Computer-Human Interaction*, 28(6):1–54, November 2021. ISSN 1557-7325. doi: 10.1145/3468505. URL `http://dx.doi.org/10.1145/3468505`. 2.1.2

[15] Richard A. Becker and William S. Cleveland. Brushing scatterplots. *Technometrics*, 29 (2):127–142, May 1987. ISSN 1537-2723. doi: 10.1080/00401706.1987.10488204. URL `http://dx.doi.org/10.1080/00401706.1987.10488204`. 2.2.1, 6.3.2

[16] Cynthia L. Bennett, Erin Brady, and Stacy M. Branham. Interdependence as a Frame for Assistive Technology Research and Design. In *Proceedings of the 20th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '18, pages 161–173, New York, NY, USA, October 2018. Association for Computing Machinery. ISBN 978-1-4503-5650-3. doi: 10.1145/3234695.3236348. Accessed: 2021-09-03. 3.3.3

[17] Dan Bennett, Alan Dix, Parisa Eslambolchilar, Feng Feng, Tom Froese, Vassilis Kostakos, Sebastien Lerique, and Niels van Berkel. Emergent interaction: Complexity, dynamics, and enaction in hci. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, page 1–7. ACM, May 2021. doi: 10.1145/3411763.3441321. URL `http://dx.doi.org/10.1145/3411763.3441321`. 2.1.2

[18] Jeffrey P. Bigham, Chandrika Jayant, Hanjie Ji, Greg Little, Andrew Miller, Robert C. Miller, Robin Miller, Aubrey Tatarowicz, Brandyn White, Samual White, and Tom Yeh. VizWiz: nearly real-time answers to visual questions. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, pages 333–342. Association for Computing Machinery, New York, NY, USA, October 2010. ISBN 978-1-4503-0271-5. doi: 10.1145/1866029.1866080. Accessed: 2021-09-06. 3.2.3

[19] Matt Blanco, Jonathan Zong, and Arvind Satyanarayan. Olli: An extensible visualization library for screen reader accessibility. In *IEEE VIS Posters*, 2022. URL `http://vis.csail.mit.edu/pubs/olli`. 4.3.3.1, 4.3.3.2, 4.3.3.3

[20] Jens Bornschein, Denise Prescher, and Gerhard Weber. Collaborative Creation of Digital

Tactile Graphics. In *Proceedings of the 17th International ACM SIGACCESS Conference on Computers & Accessibility*, ASSETS '15, pages 117–126, New York, NY, USA, October 2015. Association for Computing Machinery. ISBN 978-1-4503-3400-6. doi: 10.1145/2700648.2809869. 3.2.1

[21] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. D3 data-driven documents. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2301–2309, Dec. 2011. ISSN 1077-2626. doi: 10.1109/TVCG.2011.185. 1.2, 4.4.1.1

[22] Denis Boudreau. Unlocking Accessibility for UX/UI Designers, December 2018. URL `https://www.24a11y.com/2018/unlocking-accessibility-for-ux-ui-designers/`. Accessed: 2021-09-07. 3.2.4

[23] L.H. Boyd, W.L. Boyd, and G.C. Vanderheiden. The graphical user interface: Crisis, danger, and opportunity. *Journal of Visual Impairment & Blindness*, 84(10):496–502, December 1990. doi: 10.1177/0145482x9008401002. 4.2.2.3

[24] Eric Brangier, Josefina Gil Urrutia, Véronique Senderowicz, and Laurent Cessat. Beyond "Usability and User Experience" , Towards an Integrative Heuristic Inspection: from Accessibility to Persuasiveness in the UX Evaluation A Case Study on an Insurance Prospecting Tablet Application. *Advances in Usability and User Experience*, June 2018. doi: 10.1007/978-3-319-60492-3_44. 3.2.4

[25] Virginia Braun and Victoria Clarke. Using thematic analysis in psychology. *Qualitative Research in Psychology*, 3(2):77–101, 2006. doi: 10.1191/1478088706qp063oa. 3

[26] S. Brewster. Visualization tools for blind people using multiple modalities. *Disability and Rehabilitation*, 24(11-12):613–621, August 2002. ISSN 0963-8288. doi: 10.1080/09638280110111388. 2.2.3, 3.2.1

[27] Craig Brown and Amy Hurst. VizTouch: automatically generated tactile visualizations of coordinate spaces. In *Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction*, TEI '12, pages 131–138, New York, NY, USA, February 2012. Association for Computing Machinery. ISBN 978-1-4503-1174-8. doi: 10.1145/2148131.2148160. Accessed: 2021-09-03. 3.2.1

[28] Trisha O'Connell Bryan Gould and Geoff Freed. Effective Practices for Description of Science Content within Digital Talking Books — wgbh.org. https://www.wgbh.org/foundation/ncam/guidelines/effective-practices-for-description-of-science-content-within-digital-talking-books, 2008. [Accessed 01-04-2025]. 6.2.2

[29] Jennifer L. Burke, Matthew S. Prewett, Ashley A. Gray, Liuquin Yang, Frederick R. B. Stilson, Michael D. Coovert, Linda R. Elliot, and Elizabeth Redden. Comparing the effects of visual-auditory and visual-tactile feedback on user performance: a meta-analysis. In *Proceedings of the 8th international conference on Multimodal interfaces*, ICMI06, page 108–117. ACM, November 2006. doi: 10.1145/1180995.1181017. URL `http://dx.doi.org/10.1145/1180995.1181017`. 1.1

[30] Matthew Butler, Leona M Holloway, Samuel Reinders, Cagatay Goncu, and Kim Marriott. Technology Developments in Touch-Based Accessible Graphics: A Systematic Review of

Research 2010-2020. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, pages 1–15, New York, NY, USA, May 2021. Association for Computing Machinery. ISBN 978-1-4503-8096-6. doi: 10.1145/3411764.3445207. Accessed: 2021-09-06. 3.2.1

[31] F. Cabitza and C. Simone. Malleability in the hands of end-users. In *New Perspectives in End-User Development*, pages 137–163. Springer International Publishing, 2017. 5.3.1.1

[32] Silvia Canelón and Liz Hare. Revealing Room for Improvement in Accessibility within a Social Media Data Visualization Learning Community, 2021. URL `https://silvia.rbind.io/talk/2021-05-04-data-viz-accessibility/`. Accessed: 2021-09-06. 3.2.2

[33] C. R. Ceja, C. M. McColeman, C. Xiong, and S. L. Franconeri. Truth or square: Aspect ratio biases recall of position encodings. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):1054–1062, February 2021. 5.6.1.2

[34] Alex Chaparro and Maria Chaparro. Applications of Color in Design for Color-Deficient Users. *Ergonomics in Design*, 25(1):23–30, January 2017. ISSN 1064-8046. doi: 10.1177/1064804616635382. Accessed: 2021-09-06. 2.2.3, 3.2.1

[35] Charles Chen, Ruiyi Zhang, Sungchul Kim, Scott Cohen, Tong Yu, Ryan Rossi, and Razvan Bunescu. Neural caption generation over figures. In *Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers*, UbiComp/ISWC '19 Adjunct, pages 482–485, New York, NY, USA, September 2019. Association for Computing Machinery. ISBN 978-1-4503-6869-8. doi: 10.1145/3341162.3345601. Accessed: 2021-09-06. 3.2.1

[36] Charles Chen, Ruiyi Zhang, Eunyee Koh, Sungchul Kim, Scott Cohen, and Ryan Rossi. Figure Captioning with Relation Maps for Reasoning. In *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1526–1534, March 2020. doi: 10.1109/WACV45572.2020.9093592. ISSN: 2642-9381. 3.2.1

[37] Jinho Choi, Sanghun Jung, Deok Gun Park, Jaegul Choo, and Niklas Elmqvist. Visualizing for the non-visual: Enabling the visually impaired to use visualization. *Computer Graphics Forum*, 38(3):249–260, June 2019. ISSN 1467-8659. doi: 10.1111/cgf.13686. URL `http://dx.doi.org/10.1111/cgf.13686`. 6.2.2

[38] Jinho Choi, Sanghun Jung, Deok Gun Park, Jaegul Choo, and Niklas Elmqvist. Visualizing for the Non-Visual: Enabling the Visually Impaired to Use Visualization. *Computer Graphics Forum*, 38(3):249–260, 2019. ISSN 1467-8659. doi: 10.1111/cgf.13686. Accessed: 2021-09-03. 3.2.1

[39] Ngip Khean Chuan, Ashok Sivaji, and Wan Fatimah Wan Ahmad. Usability Heuristics for Heuristic Evaluation of Gestural Interaction in HCI. In Aaron Marcus, editor, *Design, User Experience, and Usability: Design Discourse*, Lecture Notes in Computer Science, pages 138–148, Cham, 2015. Springer International Publishing. ISBN 978-3-319-20886-2. doi: 10.1007/978-3-319-20886-2_14. 3.2.4

[40] Pramod Chundury, Biswaksen Patnaik, Yasmin Reyazuddin, Christine Tang, Jonathan

Lazar, and Niklas Elmqvist. Towards understanding sensory substitution for accessible visualization: An interview study. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):1084–1094, January 2022. ISSN 2160-9306. doi: 10.1109/tvcg.2021.3114829. URL http://dx.doi.org/10.1109/TVCG.2021.3114829. 6.2.3

[41] Pramod Chundury, Biswaksen Patnaik, Yasmin Reyazuddin, Christine Tang, Jonathan Lazar, and Niklas Elmqvist. Towards understanding sensory substitution for accessible visualization: An interview study. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):1084–1094, Jan. 2022. doi: 10.1109/tvcg.2021.3114829. 2.2.3, 4.2.1.2

[42] Pramod Chundury, Biswaksen Patnaik, Yasmin Reyazuddin, Christine Tang, Jonathan Lazar, and Niklas Elmqvist. Towards Understanding Sensory Substitution for Accessible Visualization: An Interview Study. *IEEE transactions on visualization and computer graphics*, 28(1):1084–1094, January 2022. ISSN 1941-0506. doi: 10.1109/TVCG.2021.3114829. 2.2.3, 3.2.1

[43] Pramod Chundury, Urja Thakkar, Yasmin Reyazuddin, J. Bern Jordan, Niklas Elmqvist, and Jonathan Lazar. Understanding the visualization and analytics needs of blind and low-vision professionals. In *The 26th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '24, page 1–5. ACM, October 2024. doi: 10.1145/3663548.3688496. URL http://dx.doi.org/10.1145/3663548.3688496. 2.1.1, 6.1

[44] RStudio Community. Making Shiny apps accessible for all humans, May 2018. URL https://community.rstudio.com/t/making-shiny-apps-accessible-for-all-humans/8458. Accessed: 2021-09-06. 3.2.2

[45] Maxime Cordeil, Benjamin Bach, Andrew Cunningham, Bastian Montoya, Ross T. Smith, Bruce H. Thomas, and Tim Dwyer. Embodied axes: Tangible, actuated interaction for 3d augmented reality data spaces. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, page 1–12. ACM, April 2020. doi: 10.1145/3313831.3376613. URL http://dx.doi.org/10.1145/3313831.3376613. 6.2.4, 6.4

[46] B. Craft and P. Cairns. Beyond guidelines: what can we learn from the visual information seeking mantra? In *Ninth International Conference on Information Visualisation (IV'05)*, pages 110–118, July 2005. doi: 10.1109/IV.2005.28. ISSN: 2375-0138. 3.2.4

[47] Clare Cullen and Oussama Metatla. Co-designing Inclusive Multisensory Story Mapping with Children with Mixed Visual Abilities. In *Proceedings of the 18th ACM International Conference on Interaction Design and Children*, IDC '19, pages 361–373, New York, NY, USA, June 2019. Association for Computing Machinery. ISBN 978-1-4503-6690-8. doi: 10.1145/3311927.3323146. Accessed: 2021-09-06. 2.2.3, 3.2.1

[48] d. akbaba, J. Wilburn, M. T. Nance, and M. Meyer. Manifesto for putting "chartjunk" in the trash 2021! arXiv, 2021. 5.6.4.3

[49] Lilian de Greef, Dominik Moritz, and Cynthia Bennett. Interdependent variables: Remotely designing tactile graphics for an accessible workflow. In *The 23rd International*

ACM SIGACCESS Conference on Computers and Accessibility. ACM, October 2021. doi: 10.1145/3441852.3476468. 4.4.3.1

[50] Chris DeMartini. Coin flip game., 2017. URL https://public.tableau.com/views/CoinFlipGame/CoinFlipGame. Accessed: 2021-08-20. 3.5.3

[51] Chris DeMartini. A Tableau Accessibility Journey - Part IV - Keyboard Accessibility, 2021. URL https://www.datablick.com/blog/2021/8/10/a-tableau-accessibility-journey-part-iv-keyboard-accessibility. Accessed: 2021-09-03. 3.2.2

[52] Deque. axe-core, August 2021. URL https://github.com/dequelabs/axe-core. Accessed: 2021-08-20. 3.6.5

[53] Deque. Automated Testing Identifies 57 Percent of Digital Accessibility Issues. Technical report, Deque, March 2021. URL https://www.deque.com/blog/automated-testing-study-identifies-57-percent-of-digital-accessibility Accessed: 2021-09-06. 3.1

[54] Alan Dix. Designing for appropriation. In *Electronic Workshops in Computing*. BCS Learning & Development, September 2007. doi: 10.14236/ewic/hci2007.53. URL http://dx.doi.org/10.14236/ewic/HCI2007.53. 2.1.2

[55] Dot Pad inc. Dot pad - the first tactile graphics display for the visually impaired. https://pad.dotincorp.com/, 2020. 4.4.3.1, 6.4

[56] Paul Dourish. The appropriation of interactive technologies: Some lessons from placeless documents. *Computer Supported Cooperative Work (CSCW)*, 12(4):465–490, 2003. doi: 10.1023/a:1026149119426. 4.5

[57] Sebastian Draxler, Gunnar Stevens, Martin Stein, Alexander Boden, and David Randall. Supporting the social context of technology appropriation: on a synthesis of sharing tools and tool knowledge. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, page 2835–2844. ACM, May 2012. doi: 10.1145/2207676.2208687. URL http://dx.doi.org/10.1145/2207676.2208687. 2.1.2

[58] Anthony Dunne and Fiona Raby. *Speculative everything*. The MIT Press. MIT Press, London, England, December 2013. 6.7

[59] Eloi Durant, Mathieu Rouard, Eric W. Ganko, Cedric Muller, Alan M. Cleary, Andrew D. Farmer, Matthieu Conte, and Francois Sabot. Ten simple rules for developing visualization tools in genomics. *PLOS Computational Biology*, 18(11), November 2022. doi: 10.1371/journal.pcbi.1010622. 2.2.2.2, 4.2.1.3

[60] Frank Elavsky. Method and system for accessible data visualization on a web platform. *Defensive Publication Series*, 4220, April 2021. 4.3.3.3

[61] Frank Elavsky. Chartability, 2021. URL https://chartability.github.io/POUR-CAF/. Accessed: 2021-09-07. 3.4

[62] Frank Elavsky. Picking the right tools for the job: Learning and building for data visualization and accessibility. https://www.urban.org/research/publication/do-no-harm-guide-centering-accessibility-data-visualization, 2022. [Accessed 01-04-2025]. 6.2.2

[63] Frank Elavsky, Cynthia Bennett, and Dominik Moritz. How accessible is my visualization? evaluating visualization accessibility with chartability. *Computer Graphics Forum*, 41(3):57–70, June 2022. doi: 10.1111/cgf.14522. 2.2.2.1, 2.2.2.2, 2.2.3, 4.2.1, 4.2.1.3, 4.4.1.1, 4.5, 5.2.1, 5.4.1, 6.2.1, 6.2.2

[64] Frank Elavsky, Lucas Nadolskis, and Dominik Moritz. Data navigator: An accessibility-centered data navigation toolkit. *IEEE Transactions on Visualization and Computer Graphics*, page 1–11, 2023. ISSN 2160-9306. doi: 10.1109/tvcg.2023.3327393. URL http://dx.doi.org/10.1109/TVCG.2023.3327393. 2.1.2, 6.2.1, 6.7.2

[65] A. Endert, L. Bradel, and C. North. Beyond control panels: Direct manipulation for visual analytics. *IEEE Computer Graphics and Applications*, 33(4):6–13, July 2013. ISSN 0272-1716. doi: 10.1109/mcg.2013.53. URL http://dx.doi.org/10.1109/MCG.2013.53. 6.3.2

[66] Christin Engel and Gerhard Weber. Analysis of tactile chart design. In *Proceedings of the 10th International Conference on PErvasive Technologies Related to Assistive Environments*, PETRA '17. ACM, June 2017. doi: 10.1145/3056540.3064955. URL http://dx.doi.org/10.1145/3056540.3064955. 6.2.4

[67] Will Epperson, Vaishnavi Gorantla, Dominik Moritz, and Adam Perer. Dead or alive: Continuous data profiling for interactive data science. *IEEE Transactions on Visualization and Computer Graphics*, page 1–11, 2023. ISSN 2160-9306. doi: 10.1109/tvcg.2023.3327367. URL http://dx.doi.org/10.1109/TVCG.2023.3327367. 2.2.1

[68] Olutayo Falase, Alexa F. Siu, and Sean Follmer. Tactile code skimmer: A tool to help blind programmers feel the structure of code. In *Proceedings of the 21st International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '19, page 536–538. ACM, October 2019. doi: 10.1145/3308561.3354616. URL http://dx.doi.org/10.1145/3308561.3354616. 6.2.4

[69] Danyang Fan, Alexa Fay Siu, Wing-Sum Adrienne Law, Raymond Ruihong Zhen, Sile O'Modhrain, and Sean Follmer. Slide-tone and tilt-tone: 1-dof haptic techniques for conveying shape characteristics of graphs to blind users. In *CHI Conference on Human Factors in Computing Systems*, CHI '22, page 1–19. ACM, April 2022. doi: 10.1145/3491102.3517790. URL http://dx.doi.org/10.1145/3491102.3517790. 6.2.4

[70] Danyang Fan, Alexa Fay Siu, Hrishikesh Rao, Gene Sung-Ho Kim, Xavier Vazquez, Lucy Greco, Sile O'Modhrain, and Sean Follmer. The accessibility of data visualizations on the web for screen reader users: Practices and experiences during covid-19. *ACM Transactions on Accessible Computing*, 16(1):1–29, March 2023. ISSN 1936-7236. doi: 10.1145/3557899. URL http://dx.doi.org/10.1145/3557899. 6.2.1

[71] Danyang Fan, Alexa Fay Siu, Hrishikesh Rao, Gene Sung-Ho Kim, Xavier Vazquez, Lucy Greco, Sile O'Modhrain, and Sean Follmer. The accessibility of data visualizations on the web for screen reader users: Practices and experiences during COVID-19. *ACM Transactions on Accessible Computing*, 16(1):1–29, March 2023. doi: 10.1145/3557899. 1.1, 2.2.2.1, 2.2.2.2, 2.2.3, 4.2.1, 4.2.1.2, 4.2.1.3

[72] Ali Mazraeh Farahani, Peyman Adibi, Mohammad Saeed Ehsani, Hans-Peter Hutter, and Alireza Darvishy. Automatic chart understanding: A review. *IEEE Access*, 11: 76202–76221, 2023. ISSN 2169-3536. doi: 10.1109/access.2023.3298050. URL http://dx.doi.org/10.1109/ACCESS.2023.3298050. 6.2.2

[73] Stephen Few. Visual Business Intelligence - Data Visualization and the Blind. https://www.perceptualedge.com/blog/?p=1756, 2013. [Accessed 05-04-2025]. 1.1

[74] C. Fleizach and J. P. Bigham. System-class accessibility: The architectural support for making a whole system usable by people with disabilities. *Queue*, 22(5):28–39, October 2024. 5.6.3

[75] John H. Flowers, Dion C. Buhman, and Kimberly D. Turnage. Cross-Modal Equivalence of Visual and Auditory Scatterplots for Exploring Bivariate Data Samples. *Human Factors*, 39(3):341–351, September 1997. ISSN 0018-7208. doi: 10.1518/001872097778827151. Accessed: 2021-09-03. 2.2.3, 3.2.1

[76] Camilla Forsell and Jimmy Johansson. An heuristic set for evaluation in information visualization. In *Proceedings of the International Conference on Advanced Visual Interfaces*, AVI '10, pages 199–206, New York, NY, USA, May 2010. Association for Computing Machinery. ISBN 978-1-4503-0076-6. doi: 10.1145/1842993.1843029. Accessed: 2022-03-06. 3.2.4

[77] Plotly Community Forum. Are plotly tables accessible?, February 2018. URL https://community.plotly.com/t/are-plotly-tables-accessible/8263. Accessed: 2021-09-06. 3.2.2

[78] Plotly Community Forum. Datatables and Accessibility, November 2019. URL https://community.plotly.com/t/solved-datatables-and-accessibility/31085. Accessed: 2021-09-06. 3.2.2

[79] Steven L. Franconeri, Lace M. Padilla, Priti Shah, Jeffrey M. Zacks, and Jessica Hullman. The science of visual data communication: What works. *Psychological Science in the Public Interest*, 22(3):110–161, December 2021. ISSN 1539-6053. doi: 10.1177/15291006211051956. URL http://dx.doi.org/10.1177/15291006211051956. 1.2, 6.2.2

[80] A. Gallace and C. Spence. To what extent do Gestalt grouping principles influence tactile perception? *Psychological bulletin*, 2011. doi: 10.1037/a0022335. 3.2.1

[81] Emden R. Gansner and Stephen C. North. An open graph visualization system and its applications to software engineering. *Software: Practice and Experience*, 30(11):1203–1233, 2000. doi: 10.1002/1097-024x(200009)30:11⟨1203::aid-spe338⟩3.0.co;2-n. 4.3.1.1

[82] F. A. Geldard, W. Schiff, and E. Foulke. *Tactual Perception: A Source Book*. Cambridge University Press, 1983. doi: 10.2307/1422824. 2.2.3, 3.2.1, 6.2.4

[83] Elena Gherri and Martin Eimer. Active listening impairs visual perception and selectivity: An erp study of auditory dual-task costs on visual attention. *Journal of Cognitive Neu-

roscience, 23(4):832–844, April 2011. ISSN 1530-8898. doi: 10.1162/jocn.2010.21468. URL http://dx.doi.org/10.1162/jocn.2010.21468. 6.1, 6.2.3

[84] Stéphanie Giraud and Christophe Jouffrais. Empowering low-vision rehabilitation professionals with "do-it-yourself" methods. In *Lecture Notes in Computer Science*, pages 61–68. Springer International Publishing, 2016. doi: 10.1007/978-3-319-41267-2_9. 4.2.3

[85] Michele E. Gloede and Melissa K. Gregg. The fidelity of visual and auditory memory. *Psychonomic Bulletin & Review*, 26(4):1325–1332, April 2019. ISSN 1531-5320. doi: 10.3758/s13423-019-01597-7. URL http://dx.doi.org/10.3758/s13423-019-01597-7. 1.1

[86] A. Jonathan R. Godfrey, Paul Murrell, and Volker Sorge. An accessible interaction model for data visualisation in statistics. In *Lecture Notes in Computer Science*, pages 590–597. Springer International Publishing, 2018. doi: 10.1007/978-3-319-94277-3_92. 4.1, 4.2.1.1, 4.2.2.1

[87] A. Jonathan R. Godfrey, Paul Murrell, and Volker Sorge. An Accessible Interaction Model for Data Visualisation in Statistics. In Klaus Miesenberger and Georgios Kouroupetroglou, editors, *Computers Helping People with Special Needs*, volume 10896, pages 590–597. Springer International Publishing, Cham, 2018. ISBN 978-3-319-94276-6 978-3-319-94277-3. doi: 10.1007/978-3-319-94277-3_92. Accessed: 2021-10-12. 2.2.3, 3.2.1

[88] Jenna L. Gorlewicz, Jennifer L. Tennison, P. Merlin Uesbeck, Margaret E. Richard, Hari P. Palani, Andreas Stefik, Derrick W. Smith, and Nicholas A. Giudice. Design guidelines and recommendations for multimodal, touchscreen-based graphics. *ACM Transactions on Accessible Computing*, 13(3):1–30, August 2020. ISSN 1936-7236. doi: 10.1145/3403933. URL http://dx.doi.org/10.1145/3403933. 6.3.1

[89] Colin M. Gray, Erik Stolterman, and Martin A. Siegel. Reprioritizing the relationship between HCI research and practice: bubble-up and trickle-down effects. In *Proceedings of the 2014 conference on Designing interactive systems*, DIS '14, pages 725–734, New York, NY, USA, June 2014. Association for Computing Machinery. ISBN 978-1-4503-2902-6. doi: 10.1145/2598510.2598595. 3.3

[90] David Gray Widder, Sarah West, and Meredith Whittaker. Open (for business): Big tech, concentrated power, and the political economy of open ai. *SSRN Electronic Journal*, 2023. ISSN 1556-5068. doi: 10.2139/ssrn.4543807. URL http://dx.doi.org/10.2139/ssrn.4543807. 1.2

[91] Catherine Grevet and Eric Gilbert. Piggyback prototyping: Using existing, large-scale social computing systems to prototype new ones. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, page 4047–4056. ACM, April 2015. doi: 10.1145/2702123.2702395. URL http://dx.doi.org/10.1145/2702123.2702395. 2.1.2

[92] Nathan Hahn, Joseph Chang, Ji Eun Kim, and Aniket Kittur. The knowledge accelerator: Big picture thinking in small pieces. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI'16, page 2258–2270. ACM, May 2016. doi: 10.1145/2858036.2858364. URL http://dx.doi.org/10.1145/2858036.

2858364. 2.1.1

[93] Jen Hale. Extensive digitization of tactile map collection, 2016. URL https://www.perkins.org/extensive-digitization-of-tactile-map-collection/. Perkins Archives Blog, Perkins School for the Blind. July 2016. Accessed: 2022-02-23. 2.2.3, 3.2.1

[94] Foad Hamidi, Patrick Mbullo, Deurence Onyango, Michaela Hynie, Susan McGrath, and Melanie Baljko. Participatory design of diy digital assistive technology in western kenya. In *Proceedings of the Second African Conference for Human Computer Interaction: Thriving Communities*, AfriCHI '18. ACM, December 2018. doi: 10.1145/3283458.3283478. URL http://dx.doi.org/10.1145/3283458.3283478. 2.1.1

[95] Kathryn Hamlyn, Katie Welldon, Benjamin Clark, Willem van Steenbrugge, and Sarosh Kapadia. The effect of increased cognitive demand on auditory processing assessment. *Journal of the American Academy of Audiology*, 29(9):788–801, October 2018. ISSN 1050-0545. doi: 10.3766/jaaa.16152. URL http://dx.doi.org/10.3766/jaaa.16152. 6.2.3

[96] N. Hammad et al. Exploring the affordances of game-aware streaming to support blind and low vision viewers: A design probe study. In *Proceedings of the 26th International ACM SIGACCESS Conference on Computers and Accessibility*, 2024. 5.5.2

[97] A. Hamraie. Making access critical: Disability, race, and gender in environmental design. https://belonging.berkeley.edu/aimi-hamraie-making-access-critical-disability-race-and-gender-enviror 2019. [Online]. 2.2.3, 5.2.1

[98] G. Harboe and E. M. Huang. Real-world affinity diagramming practices: Bridging the paper-digital gap. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, 2015. 5.6

[99] Gunnar Harboe and Elaine M. Huang. Real-world affinity diagramming practices: Bridging the paper-digital gap. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, page 95–104. ACM, April 2015. doi: 10.1145/2702123.2702561. URL http://dx.doi.org/10.1145/2702123.2702561. 6.6

[100] Jeffrey Heer and Maneesh Agrawala. Design considerations for collaborative visual analytics. In *2007 IEEE Symposium on Visual Analytics Science and Technology*, page 171–178. IEEE, October 2007. doi: 10.1109/vast.2007.4389011. URL http://dx.doi.org/10.1109/VAST.2007.4389011. 2.2.1, 6.3.2

[101] Jeffrey Heer and Dominik Moritz. Mosaic: An Architecture for Scalable & Interoperable Data Views. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–11, 2023. ISSN 1077-2626, 1941-0506, 2160-9306. doi: 10.1109/TVCG.2023.3327189. URL https://ieeexplore.ieee.org/document/10297587/. 1.2, 2.2.1, 6.3.2, 6.4

[102] C. Patrick Heidkamp and Jeffrey T. Slomba. Geospatial sculpture: Nonverbal com-

munication through the tactilization of geospatial data. *GeoHumanities*, 3(2):554–566, May 2017. ISSN 2373-5678. doi: 10.1080/2373566x.2017.1306425. URL http://dx.doi.org/10.1080/2373566X.2017.1306425. 6.2

[103] Thomas Hermann, Andy Hunt, and John G Neuhoff, editors. *The Sonification Handbook*. Logos Verlag Berlin, Berlin, Germany, December 2011. 6.2.3

[104] Jaylin Herskovitz, Yi Fei Cheng, Anhong Guo, Alanson P. Sample, and Michael Nebeling. Xspace: An augmented reality toolkit for enabling spatially-aware distributed collaboration. *Proceedings of the ACM on Human-Computer Interaction*, 6(ISS):277–302, November 2022. ISSN 2573-0142. doi: 10.1145/3567721. URL http://dx.doi.org/10.1145/3567721. 2.1.2

[105] Jaylin Herskovitz, Andi Xu, Rahaf Alharbi, and Anhong Guo. Hacking, switching, combining: Understanding and supporting diy assistive technology design by blind people. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, page 1–17. ACM, April 2023. doi: 10.1145/3544548.3581249. URL http://dx.doi.org/10.1145/3544548.3581249. 2.1.2

[106] Jaylin Herskovitz, Andi Xu, Rahaf Alharbi, and Anhong Guo. Programally: Creating custom visual access programs via multi-modal end-user programming. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, UIST '24, page 1–15. ACM, October 2024. doi: 10.1145/3654777.3676391. URL http://dx.doi.org/10.1145/3654777.3676391. 2.1.2

[107] L. Hickman and A. Hagerty. Standardised access: the tension between scale and fit. https://www.adalovelaceinstitute.org/blog/standardised-access-tension-scale-fit/, 2021. [Online]. 5.3.1.2

[108] Highsoft. Highcharts - interactive charting library for developers. https://www.highcharts.com/, 2009. [Online]. 5.1

[109] Highsoft. Highcharts accessibility module, 2016. URL https://highcharts.com/docs/accessibility/accessibility-module. Accessed: 2021-09-06. 3.2.2

[110] Highsoft. Stacked column demo. https://www.highcharts.com/demo/column-stacked, 2018. Accessed: 2022-12-31. 4.4.1

[111] Highsoft. Highcharts accessibility module: information and demos. https://highcharts.com/docs/accessibility/accessibility-module, 2018. Accessed: 2021-09-06. 4.2.2.1

[112] Highsoft. Fixed placement columns | Highcharts.com, 2018. URL https://www.highcharts.com/demo/column-placement. Accessed: 2021-08-20. 3.5.3

[113] Highsoft. Highcharts: Render millions of chart points with the boost module. https://www.highcharts.com/blog/tutorials/highcharts-high-performance-boost-module/, 2022. Accessed: 2022-11-20. 4.4.1

[114] Toan Hoang. The TableauMagic DataTables Extension, September 2018. URL https://tableau.toanhoang.com/

the-tableau-magic-datatables-extension-now-available/. Accessed: 2021-09-03. 3.2.2

[115] Jonathan Hook, Sanne Verbaan, Abigail Durrant, Patrick Olivier, and Peter Wright. A study of the challenges related to diy assistive technology in the context of children with disabilities. In *Proceedings of the 2014 conference on Designing interactive systems*, DIS '14, page 597–606. ACM, June 2014. doi: 10.1145/2598510.2598530. URL http://dx.doi.org/10.1145/2598510.2598530. 2.1.1

[116] E. Hoque, P. Kavehzadeh, and A. Masry. Chart question answering: State of the art and future directions. *Computer Graphics Forum*, 41(3):555–572, June 2022. ISSN 1467-8659. doi: 10.1111/cgf.14573. URL http://dx.doi.org/10.1111/cgf.14573. 6.2.2

[117] S. Hsueh, B. Vincenzi, A. Murdeshwar, and M. Ciolfi Felice. Cripping data visualizations: Crip technoscience as a critical lens for designing digital access. In *Proceedings of the 25th International ACM SIGACCESS Conference on Computers and Accessibility*, pages 1–16, 2023. 1.1, 2.2.3, 5.2.1

[118] Amy Hurst and Shaun Kane. Making "making" accessible. In *Proceedings of the 12th International Conference on Interaction Design and Children*, IDC '13, pages 635–638, New York, NY, USA, 6 2013. Association for Computing Machinery. Accessed: 2021-09-03. 4.2.3

[119] Amy Hurst and Shaun Kane. Making "making" accessible. In *Proceedings of the 12th International Conference on Interaction Design and Children*, IDC '13, pages 635–638, New York, NY, USA, June 2013. Association for Computing Machinery. ISBN 978-1-4503-1918-8. doi: 10.1145/2485760.2485883. Accessed: 2021-09-03. 2.1.1, 3.7

[120] Amy Hurst and Jasmine Tobias. Empowering individuals with do-it-yourself assistive technology. In *The proceedings of the 13th international ACM SIGACCESS conference on Computers and accessibility*, ASSETS '11, page 11–18. ACM, October 2011. doi: 10.1145/2049536.2049541. URL http://dx.doi.org/10.1145/2049536.2049541. 2.1.1

[121] Web Accessibility Initiative. Web Content Accessibility Guidelines (WCAG) 2.1, 2018. URL https://www.w3.org/TR/WCAG21/. Accessed: 2021-09-03. 3.2.1, 3.2.3

[122] Web Accessibility Initiative. Web Accessibility Laws & Policies, August 2021. URL https://www.w3.org/WAI/policies/. Accessed: 2021-08-19. 3.2.3

[123] Lilly C. Irani and M. Six Silberman. Turkopticon: Interrupting worker invisibility in amazon mechanical turk. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, page 611–620, New York, NY, USA, 2013. Association for Computing Machinery. ISBN 9781450318990. doi: 10.1145/2470654.2470742. 3.7

[124] Yvonne Jansen, Pierre Dragicevic, Petra Isenberg, Jason Alexander, Abhijit Karnik, Johan Kildal, Sriram Subramanian, and Kasper Hornbæk. Opportunities and Challenges for Data Physicalization. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 3227–3236. Association for Computing Machinery, New York, NY, USA, April 2015. ISBN 978-1-4503-3145-6. doi: 10.1145/2702123.2702180. Accessed: 2021-09-03. 3.2.1

[125] Chandrika Jayant, Matt Renzelmann, Dana Wen, Satria Krisnandi, Richard Ladner, and Dan Comden. Automated tactile graphics translation: in the field. In *Proceedings of the 9th international ACM SIGACCESS conference on Computers and accessibility*, Assets '07, pages 75–82, New York, NY, USA, October 2007. Association for Computing Machinery. ISBN 978-1-59593-573-1. doi: 10.1145/1296843.1296858. Accessed: 2021-09-03. 3.2.1

[126] Chutian Jiang, Wentao Lei, Emily Kuang, Teng Han, and Mingming Fan. Understanding strategies and challenges of conducting daily data analysis (dda) among blind and low-vision people. In *The 25th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '23, page 1–15. ACM, October 2023. doi: 10.1145/3597638.3608423. URL `http://dx.doi.org/10.1145/3597638.3608423`. 2.1.1, 6.1

[127] S. Jones, I. Pedraza Pineros, D. Hajas, J. Zong, and A. Satyanarayan. "customization is key": Reconfigurable textual tokens for accessible data visualizations. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, pages 1–14, 2024. 5.2.3

[128] Shuli Jones, Isabella Pedraza Pineros, Daniel Hajas, Jonathan Zong, and Arvind Satyanarayan. "customization is key": Reconfigurable textual tokens for accessible data visualizations. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, CHI '24, page 1–14. ACM, May 2024. doi: 10.1145/3613904.3641970. URL `http://dx.doi.org/10.1145/3613904.3641970`. 6.2.2

[129] Ger Joyce, Mariana Lilley, Trevor Barker, and Amanda Jefferies. Mobile Application Usability: Heuristic Evaluation and Evaluation of Heuristics. In Ben Amaba, editor, *Advances in Human Factors, Software, and Systems Engineering*, Advances in Intelligent Systems and Computing, pages 77–86, Cham, 2016. Springer International Publishing. ISBN 978-3-319-41935-0. doi: 10.1007/978-3-319-41935-0_8. 3.2.4

[130] S. C. S. Joyner, A. Riegelhuth, K. Garrity, Y.-S. Kim, and N. W. Kim. Visualization accessibility in the wild: Challenges faced by visualization designers. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, pages 1–19, 2022. 2.2.3, 5.2.1

[131] Crescentia Jung, Shubham Mehta, Atharva Kulkarni, Yuhang Zhao, and Yea-Seul Kim. Communicating visualizations without visuals: Investigation of visualization alternative text for people with visual impairments. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):1095–1105, January 2022. ISSN 2160-9306. doi: 10.1109/tvcg.2021.3114846. URL `http://dx.doi.org/10.1109/TVCG.2021.3114846`. 6.2.1, 6.2.2

[132] Crescentia Jung, Shubham Mehta, Atharva Kulkarni, Yuhang Zhao, and Yea-Seul Kim. Communicating visualizations without visuals: Investigation of visualization alternative text for people with visual impairments. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):1095–1105, January 2022. doi: 10.1109/tvcg.2021.3114846. 2.2.3, 4.1, 4.2.1.2

[133] Crescentia Jung, Shubham Mehta, Atharva Kulkarni, Yuhang Zhao, and Yea-Seul Kim. Communicating Visualizations without Visuals: Investigation of Visualization Alterna-

tive Text for People with Visual Impairments. *IEEE Transactions on Visualization and Computer Graphics*, 28(01):1095–1105, January 2022. ISSN 1077-2626. doi: 10.1109/TVCG.2021.3114846. Accessed: 2022-02-23. 2.2.3, 3.2.1

[134] Hyeonsu B. Kang, Xin Qian, Tom Hope, Dafna Shahaf, Joel Chan, and Aniket Kittur. Augmenting scientific creativity with an analogical search engine. *ACM Transactions on Computer-Human Interaction*, 29(6):1–36, November 2022. ISSN 1557-7325. doi: 10.1145/3530013. URL http://dx.doi.org/10.1145/3530013. 2.1.1

[135] Gyeongri Kim, Jiho Kim, and Yea-Seul Kim. "explain what a treemap is": Exploratory investigation of strategies for explaining unfamiliar chart to blind and low vision users. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, page 1–13. ACM, April 2023. doi: 10.1145/3544548.3581139. URL http://dx.doi.org/10.1145/3544548.3581139. 6.2.2

[136] Hyeok Kim, Yea-Seul Kim, and Jessica Hullman. Erie: A declarative grammar for data sonification. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, CHI '24, page 1–19. ACM, May 2024. doi: 10.1145/3613904.3642442. URL http://dx.doi.org/10.1145/3613904.3642442. 6.2.3

[137] Jiho Kim, Arjun Srinivasan, Nam Wook Kim, and Yea-Seul Kim. Exploring chart question answering for blind and low vision users. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. ACM, April 2023. doi: 10.1145/3544548.3581532. 2.2.3, 4.2.1.2, 6.2.2

[138] N. W. Kim, S. C. Joyner, A. Riegelhuth, and Y. Kim. Accessible visualization: Design space, opportunities, and challenges. *Computer Graphics Forum*, 40(3):173–188, June 2021. ISSN 1467-8659. doi: 10.1111/cgf.14298. URL http://dx.doi.org/10.1111/cgf.14298. 6.2.1, 6.2.2

[139] N. W. Kim, S. C. Joyner, A. Riegelhuth, and Y. Kim. Accessible visualization: Design space, opportunities, and challenges. *Computer Graphics Forum*, 40(3):173–188, June 2021. doi: 10.1111/cgf.14298. 4.5

[140] N. W. Kim, S. C. Joyner, A. Riegelhuth, and Y. Kim. Accessible Visualization: Design Space, Opportunities, and Challenges. *Computer Graphics Forum*, 40(3):173–188, 2021. ISSN 1467-8659. doi: 10.1111/cgf.14298. 2.2.3, 3.2.1

[141] N. W. Kim, G. Ataguba, S. C. Joyner, C. Zhao, and H. Im. Beyond alternative text and tables: Comparative analysis of visualization tools and accessibility methods. *Computer Graphics Forum*, 42(3):323–335, June 2023. 5.4.1

[142] Kitware, Inc. *The Visualization Toolkit user's guide*, January 2003. URL http://www.kitware.com/publications/item/view/1269. 2.1.2

[143] Rebecca Klein. Power BI Accessibility Best Practices, 2021. URL https://rklein324.github.io/PowerBIAccessibility/. Accessed: 2021-09-06. 3.2.2

[144] A. J. Ko et al. The state of the art in end-user software engineering. *ACM Computing Surveys*, 43(3):1–44, April 2011. 5.3.1.1

[145] Richard E. Ladner. Design for user empowerment. *Interactions*, 22(2):24–29, February 2015. ISSN 1072-5520. doi: 10.1145/2723869. Accessed: 2022-03-07. 3.3.3

[146] Chufan Lai, Zhixian Lin, Ruike Jiang, Yun Han, Can Liu, and Xiaoru Yuan. Automatic Annotation Synchronizing with Textual Description for Visualization. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–13. Association for Computing Machinery, New York, NY, USA, April 2020. ISBN 978-1-4503-6708-0. doi: doi.org/10.1145/3313831.3376443. Accessed: 2021-09-06. 3.2.1

[147] S. Lederman, G. Thorne, and B. Jones. Perception of texture by vision and touch: multidimensionality and intersensory integration. *Journal of experimental psychology. Human perception and performance*, 1986. doi: 10.1037//0096-1523.12.2.169. 3.2.1

[148] Bongshin Lee, Eun Kyoung Choe, Petra Isenberg, Kim Marriott, and John Stasko. Reaching broader audiences with data visualization. *IEEE Computer Graphics and Applications*, 40(2):82–90, 2020. doi: 10.1109/MCG.2020.2968244. 2.2.3, 3.2.1

[149] Chien-Yu Lin and Yu-Ming Chang. Increase in physical activities in kindergarten children with cerebral palsy by employing MaKey–MaKey-based task systems. *Research in Developmental Disabilities*, 35(9):1963–1969, September 2014. doi: 10.1016/j.ridd.2014.04.028. URL `https://doi.org/10.1016/j.ridd.2014.04.028`. 4.2.3

[150] Zhicheng Liu and Jeffrey Heer. The effects of interactive latency on exploratory visual analysis. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2122–2131, December 2014. ISSN 1077-2626. doi: 10.1109/tvcg.2014.2346452. URL `http://dx.doi.org/10.1109/TVCG.2014.2346452`. 2.2.1, 6.3.2

[151] L. Y. Lo, A. Gupta, K. Shigyo, A. Wu, E. Bertini, and H. Qu. Misinformed by visualization: What do we learn from misinformative visualizations? *Computer Graphics Forum*, 41(3):515–525, June 2022. 5.6.1.2

[152] Panagiotis Louridas. Design as bricolage: anthropology meets design thinking. *Design Studies*, 20(6):517–535, November 1999. doi: 10.1016/s0142-694x(98)00044-1. 4.5

[153] Alan Lundgard and Arvind Satyanarayan. Accessible visualization via natural language descriptions: A four-level model of semantic content. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):1073–1083, January 2022. doi: 10.1109/tvcg.2021.3114770. 2.2.2.2, 2.2.3, 4.2.1.2, 4.2.1.3, 4.4.1.1, 4.4.3, 4.5

[154] Alan Lundgard and Arvind Satyanarayan. Accessible visualization via natural language descriptions: A four-level model of semantic content. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):1073–1083, 2022. doi: 10.1109/TVCG.2021.3114770. 2.2.3, 3.2.1

[155] Alan Lundgard, Crystal Lee, and Arvind Satyanarayan. Sociotechnical considerations for accessible visualization design. In *2019 IEEE Visualization Conference (VIS)*, page 16–20. IEEE, October 2019. doi: 10.1109/visual.2019.8933762. URL `http://dx.doi.org/10.1109/VISUAL.2019.8933762`. 6.2.4

[156] Alan Lundgard, Crystal Lee, and Arvind Satyanarayan. Sociotechnical considerations for accessible visualization design. In *2019 IEEE Visualization Conference (VIS)*. IEEE,

October 2019. doi: 10.1109/visual.2019.8933762. 2.2.2.2, 4.2.1.3, 4.4.1.1, 4.4.3, 4.5

[157] Alan Lundgard, Crystal Lee, and Arvind Satyanarayan. Sociotechnical Considerations for Accessible Visualization Design. *2019 IEEE Visualization Conference (VIS)*, September 2019. doi: 10.1109/VISUAL.2019.8933762. 3.2.1, 3.2.3

[158] Léonie Watson. Notes on synthetic speech. https://tink.uk/notes-on-synthetic-speech/, 2021. [Accessed 01-04-2025]. 6.1

[159] Dor Ma'ayan, Wode Ni, Katherine Ye, Chinmay Kulkarni, and Joshua Sunshine. How domain experts create conceptual diagrams and implications for tool design. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, page 1–14, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450367080. doi: 10.1145/3313831.3376253. 4.3, 4.5

[160] M. Maceli and M. E. Atwood. "human crafters" once again: Supporting users as designers in continuous co-design. In *End-User Development*, pages 9–24. Springer Berlin Heidelberg, 2013. 5.2.3, 5.6.1.3

[161] Kelly Mack, Emma McDonnell, Dhruv Jain, Lucy Lu Wang, Jon E. Froehlich, and Leah Findlater. What Do We Mean by "Accessibility Research"? A Literature Survey of Accessibility Papers in CHI and ASSETS from 1994 to 2019. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, pages 1–18, New York, NY, USA, 5 2021. Association for Computing Machinery. Accessed: 2022-02-22. 2.2.2.1

[162] Kelly Mack, Emma McDonnell, Dhruv Jain, Lucy Lu Wang, Jon E. Froehlich, and Leah Findlater. What Do We Mean by “Accessibility Research”? A Literature Survey of Accessibility Papers in CHI and ASSETS from 1994 to 2019. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, pages 1–18, New York, NY, USA, May 2021. Association for Computing Machinery. ISBN 978-1-4503-8096-6. doi: 10.1145/3411764.3445412. Accessed: 2022-02-22. 2.2.3, 3.2.1

[163] Jennifer Mankoff, Gillian R. Hayes, and Devva Kasnitz. Disability studies as a source of critical inquiry for the field of assistive technology. In *Proceedings of the 12th international ACM SIGACCESS conference on Computers and accessibility*, ASSETS '10, pages 3–10, New York, NY, USA, October 2010. Association for Computing Machinery. ISBN 978-1-60558-881-0. doi: 10.1145/1878803.1878807. Accessed: 2021-09-06. 3.3.3

[164] Douglass L. Mansur, Merra M. Blattner, and Kenneth I. Joy. Sound graphs: A numerical data analysis method for the blind. *Journal of Medical Systems*, 9(3):163–174, June 1985. ISSN 1573-689X. doi: 10.1007/BF00996201. Accessed: 2021-09-03. 2.2.3, 3.2.1

[165] Mapbox. mapbox-gl js, 2019. URL https://www.npmjs.com/package/mapbox-gl. Accessed: 2021-09-03. 3.2.2

[166] Mapbox. Mapbox gl accessibility, August 2021. URL https://github.com/mapbox/mapbox-gl-accessibility. Accessed: 2021-09-06. 3.2.2

[167] Deborah Marks. Models of disability. *Disability and Rehabilitation*, 19(3):85–91, January 1997. ISSN 1464-5165. doi: 10.3109/09638289709166831. URL http://dx.doi.

org/10.3109/09638289709166831. 1.1

[168] K. Marriott et al. Inclusive data visualization for people with disabilities: a call to action. *Interactions*, 28(3):47–51, April 2021. 5.2.1

[169] Kim Marriott, Bongshin Lee, Matthew Butler, Ed Cutrell, Kirsten Ellis, Cagatay Goncu, Marti Hearst, Kathleen McCoy, and Danielle Albers Szafir. Inclusive data visualization for people with disabilities. *Interactions*, 28(3):47–51, April 2021. doi: 10.1145/3457875. 2.2.1, 2.2.2.1, 4.2.1, 4.2.3, 4.5

[170] Kim Marriott, Bongshin Lee, Matthew Butler, Ed Cutrell, Kirsten Ellis, Cagatay Goncu, Marti Hearst, Kathleen McCoy, and Danielle Albers Szafir. Inclusive data visualization for people with disabilities: a call to action. *Interactions*, 28(3):47–51, April 2021. ISSN 1072-5520. doi: 10.1145/3457875. 2.2.3, 3.2.1

[171] Rubén Alcaraz Martínez, Mireia Ribera Turró, and Toni Granollers Saltiveri. Methodology for heuristic evaluation of the accessibility of statistical charts for people with low vision and color vision deficiency. *Universal Access in the Information Society*, December 2021. doi: 10.21203/rs.3.rs-156959/v1. 2.2.3, 3.2.1, 3.2.4

[172] Melanie Mazanec. Semiotic, 2019. URL https://semiotic.nteract.io/guides/accessibility. Accessed: 2021-09-03. 3.2.2

[173] David K. McGookin and Stephen A. Brewster. SoundBar: exploiting multiple views in multimodal graph browsing. In *Proceedings of the 4th Nordic conference on Human-computer interaction: changing roles*, NordiCHI '06, pages 145–154, New York, NY, USA, October 2006. Association for Computing Machinery. ISBN 978-1-59593-325-6. doi: 10.1145/1182475.1182491. Accessed: 2021-09-03. 2.2.3, 3.2.1

[174] Priscilla Moraes, Gabriel Sina, Kathleen McCoy, and Sandra Carberry. Evaluating the accessibility of line graphs through textual summaries for visually impaired users. In *Proceedings of the 16th international ACM SIGACCESS conference on Computers & accessibility*, ASSETS '14, pages 83–90, New York, NY, USA, October 2014. Association for Computing Machinery. ISBN 978-1-4503-2720-6. doi: 10.1145/2661334.2661368. 3.2.1

[175] Jakob Nielsen. 10 Usability Heuristics for User Interface Design, 1994. URL https://www.nngroup.com/articles/ten-usability-heuristics/. Accessed: 2021-09-07. 3.2.4

[176] Jakob Nielsen. Heuristic evaluation. In *Usability inspection methods*, pages 25–62. John Wiley & Sons, Inc., USA, June 1994. ISBN 978-0-471-01877-3. Accessed: 2021-09-06. 3.2.4

[177] Jamie R. Nuñez, Christopher R. Anderton, and Ryan S. Renslow. Optimizing colormaps with consideration for color vision deficiency to enable accurate interpretation of scientific data. *PLOS ONE*, 13(7), August 2018. ISSN 1932-6203. doi: 10.1371/journal.pone.0199239. 2.2.3, 3.2.1

[178] Jason Obeid and Enamul Hoque. Chart-to-Text: Generating Natural Language Descriptions for Charts by Adapting the Transformer Model. *arXiv Preprint*, November 2020.

URL http://arxiv.org/abs/2010.09142. Accessed: 2021-09-06. 3.2.1

[179] City of San Francisco. COVID-19 data and reports | San Francisco, 2021. URL https://sf.gov/resource/2021/covid-19-data-and-reports. Accessed: 2021-09-03. 3.2.2

[180] CA Okoro, ND Hollis, AC Cyrus, and S. Griffin-Blake. Prevalence of disabilities and health care access by disability status and type among adults — united states, 2016. *Centers for Disease Control and Prevention MMWR*, 67:882–887, 2018. doi: 10.15585/mmwr.mm6732a3. 1.1, 3.1, 6.1

[181] Manuel M. Oliveira. Towards More Accessible Visualizations for Color-Vision-Deficient Individuals. *Computing in Science Engineering*, 15(5):80–87, September 2013. ISSN 1558-366X. doi: 10.1109/MCSE.2013.113. Conference Name: Computing in Science Engineering. 2.2.3, 3.2.1, 3.2.4

[182] Maurício Oliveira and Celmar Guimarães da Silva. Adapting Heuristic Evaluation to Information Visualization - A Method for Defining a Heuristic Set by Heuristic Grouping. In *Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 3: IVAPP, (VISIGRAPP 2017)*, pages 225–232. INSTICC, SciTePress, 2017. ISBN 978-989-758-228-8. doi: 10.5220/0006133202250232. 3.2.4

[183] World Health Organization. World report on vision, 2019. URL https://www.who.int/publications-detail-redirect/9789241516570. Accessed: 2021-09-03. 3.1

[184] Daniela Quiñones Otey. A methodology to develop usability / user experience heuristics. In *Proceedings of the XVIII International Conference on Human Computer Interaction*, Interacción '17, pages 1–2, New York, NY, USA, September 2017. Association for Computing Machinery. ISBN 978-1-4503-5229-1. doi: 10.1145/3123818.3133832. 3.2.4

[185] G. Pallapa, S. K. Das, M. Di Francesco, and T. Aura. Adaptive and context-aware privacy preservation exploiting user interactions in smart environments. *Pervasive and Mobile Computing*, 12:232–243, June 2014. 5.2.2

[186] Fabio Pittarello and Manuel Semenzato. Towards a data physicalization toolkit for non-sighted users. In *2024 IEEE 21st Consumer Communications & Networking Conference (CCNC)*, page 1–6. IEEE, January 2024. doi: 10.1109/ccnc51664.2024.10454861. URL http://dx.doi.org/10.1109/CCNC51664.2024.10454861. 2.1.2

[187] Venkatesh Potluri, Sudheesh Singanamalla, Nussara Tieanklin, and Jennifer Mankoff. Notably inaccessible — data driven understanding of data science notebook (in)accessibility. In *The 25th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '23, page 1–19. ACM, October 2023. doi: 10.1145/3597638.3608417. URL http://dx.doi.org/10.1145/3597638.3608417. 6.1

[188] C. Power, A. Freire, H. Petrie, and D. Swallow. Guidelines are only half of the story: accessibility problems encountered by blind users on the web. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2012. 5.5.2

[189] Christopher Power, André Freire, Helen Petrie, and David Swallow. Guidelines are only half of the story: Accessibility problems encountered by blind users on the web. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, page 433–442, New York, NY, USA, 2012. Association for Computing Machinery. ISBN 9781450310154. doi: 10.1145/2207676.2207736. 3.1

[190] Xin Qian, Eunyee Koh, Fan Du, Sungchul Kim, Joel Chan, Ryan A. Rossi, Sana Malik, and Tak Yeon Lee. Generating Accurate Caption Units for Figure Captioning. In *Proceedings of the Web Conference 2021*, pages 2792–2804. Association for Computing Machinery, New York, NY, USA, April 2021. ISBN 978-1-4503-8312-7. doi: 10.1145/3442381.3449923. Accessed: 2021-09-06. 3.2.1

[191] Lauren Race, Chancey Fleet, Danielle Montour, Lindsay Yazzolino, Marco Salsiccia, Claire Ferrari, Sheri Wells-Jensen, and Amy Hurst. Designing while blind: Nonvisual tools and inclusive workflows for tactile graphic creation. In *The 25th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '23, page 1–8. ACM, October 2023. doi: 10.1145/3597638.3614546. URL `http://dx.doi.org/10.1145/3597638.3614546`. 6.2.4

[192] Blake Ellis Reid. The curb-cut effect and the perils of accessibility without disability. *SSRN Electronic Journal*, 2022. doi: 10.2139/ssrn.4262991. 1.1, 4.5

[193] Samuel Reinders, Matthew Butler, Ingrid Zukerman, Bongshin Lee, Lizhen Qu, and Kim Marriott. When refreshable tactile displays meet conversational agents: Investigating accessible data presentation and analysis with touch and speech. *IEEE Transactions on Visualization and Computer Graphics*, 31(1):864–874, January 2025. ISSN 2160-9306. doi: 10.1109/tvcg.2024.3456358. URL `http://dx.doi.org/10.1109/TVCG.2024.3456358`. 6.2.4, 6.3.1

[194] Yvonne Rogers, Jeni Paay, Margot Brereton, Kate L. Vaisutis, Gary Marsden, and Frank Vetere. Never too old. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, April 2014. doi: 10.1145/2556288.2557184. URL `https://doi.org/10.1145/2556288.2557184`. 4.2.3, 4.3.2.1

[195] Rod D Roscoe, Erin K Chiou, and Abigail R Wooldridge, editors. *Advancing diversity, inclusion, and social justice through human systems engineering*. CRC Press, London, England, December 2019. 1.1

[196] Antti Salovaara. Inventing new uses for tools: A cognitive foundation for studies on appropriation. *Human Technology: An Interdisciplinary Journal on Humans in ICT Environments*, 4(2):209–228, November 2008. ISSN 1795-6889. doi: 10.17011/ht/urn.200811065856. URL `http://dx.doi.org/10.17011/ht/urn.200811065856`. 2.1.2

[197] Elizabeth B.-N. Sanders and Pieter Jan Stappers. Probes, toolkits and prototypes: three approaches to making in codesigning. *CoDesign*, 2014. doi: 10.1080/15710882.2014.888183. 2.1.2

[198] Beatriz Sousa Santos, Samuel Silva, and Paulo Dias. Heuristic Evaluation in Visualization: An Empirical Study : Position paper. In *2018 IEEE Evaluation and Beyond -*

Methodological Approaches for Visualization (BELIV), pages 78–85, October 2018. doi: 10.1109/BELIV.2018.8634108. 3.2.4

[199] Zhanna Sarsenbayeva, Niels Van Berkel, Eduardo Velloso, Jorge Goncalves, and Vassilis Kostakos. Methodological standards in accessibility research on motor impairments: A survey. *ACM Computing Surveys*, 55(7):1–35, December 2022. doi: 10.1145/3543509. URL https://doi.org/10.1145/3543509. 4.2.3

[200] SAS. SAS Graphics Accelerator, 2016. URL https://support.sas.com/software/products/graphics-accelerator/. Accessed: 2021-09-08. 3.2.2

[201] Arvind Satyanarayan, Dominik Moritz, Kanit Wongsuphasawat, and Jeffrey Heer. Vega-lite: A grammar of interactive graphics. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):341–350, January 2017. doi: 10.1109/tvcg.2016.2599030. 1.2, 2.2.1, 4.2.2.2, 4.2.3, 4.3.3.1, 4.4.2.1

[202] Anastasia Schaadhardt, Alexis Hiniker, and Jacob O. Wobbrock. Understanding Blind Screen-Reader Users’ Experiences of Digital Artboards. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, New York, NY, USA, May 2021. Association for Computing Machinery. ISBN 978-1-4503-8096-6. doi: 10.1145/3411764.3445242. Accessed: 2021-09-06. 2.2.3, 3.2.1

[203] D. Schepers and F. Elavsky. "accessible data viz". *The Investigative Reporters and Editors Journal*, 4:8–9, December 2024. 2.2.3, 5.2.1

[204] Doug Schepers. Why Accessibility Is at the Heart of Data Visualization | by Doug Schepers | Nightingale | Medium, 2020. URL https://medium.com/nightingale/accessibility-is-at-the-heart-of-data-visualization-64a38d6c505b. Accessed: 2021-09-06. 3.3.2

[205] Jochen Schneider. Constructing the Yellow Brick Road: Route Bricks on Virtual Tactile Maps, 2000. 3.2.1

[206] Jean Scholtz. Developing guidelines for assessing visual analytics environments. *Information Visualization*, 10(3):212–231, July 2011. ISSN 1473-8716. doi: 10.1177/1473871611407399. Accessed: 2022-03-06. 3.2.4

[207] JooYoung Seo, Yilin Xia, Bongshin Lee, Sean Mccurry, and Yu Jun Yam. Maidr: Making statistical visualizations accessible with multimodal data representation. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, CHI '24, page 1–22. ACM, May 2024. doi: 10.1145/3613904.3642730. URL http://dx.doi.org/10.1145/3613904.3642730. 6.2.4

[208] A. Sharif, J. G. Kim, J. Z. Xu, and J. O. Wobbrock. Understanding and reducing the challenges faced by creators of accessible online data visualizations. In *Proceedings of the 26th International ACM SIGACCESS Conference on Computers and Accessibility*, volume 5, pages 1–20, 2024. 2.2.3, 5.2.1

[209] Ather Sharif and Babak Forouraghi. evoGraphs — a jQuery plugin to create web accessible graphs. In *2018 15th IEEE Annual Consumer Communications Networking Conference CCNC*. IEEE, January 2018. doi: 10.1109/ccnc.2018.8319239. URL

https://doi.org/10.1109/ccnc.2018.8319239. 4.2.1.1

[210] Ather Sharif and Babak Forouraghi. evoGraphs — A jQuery plugin to create web accessible graphs. In *2018 15th IEEE Annual Consumer Communications Networking Conference (CCNC)*, pages 1–4, January 2018. doi: 10.1109/CCNC.2018.8319239. ISSN: 2331-9860. 3.2.1

[211] Ather Sharif, Sanjana Shivani Chintalapati, Jacob O. Wobbrock, and Katharina Reinecke. Understanding screen-reader users' experiences with online data visualizations. In *The 23rd International ACM SIGACCESS Conference on Computers and Accessibility*. ACM, October 2021. doi: 10.1145/3441852.3471202. 2.2.3, 4.1, 4.2.1.2

[212] Ather Sharif, Sanjana Shivani Chintalapati, Jacob O. Wobbrock, and Katharina Reinecke. Understanding screen-reader users' experiences with online data visualizations. In *The 23rd International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '21, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383066. doi: 10.1145/3441852.3471202. URL https://doi.org/10.1145/3441852.3471202. 2.2.3, 3.2.1, 3.2.3

[213] Ather Sharif, Olivia H. Wang, Alida T. Muongchan, Katharina Reinecke, and Jacob O. Wobbrock. VoxLens: Making online data visualizations accessible with an interactive JavaScript plug-in. In *CHI Conference on Human Factors in Computing Systems*. ACM, April 2022. doi: 10.1145/3491102.3517431. 4.2.1.1, 6.2.1

[214] Ather Sharif, Venkatesh Potluri, Jazz Rui Xia Ang, Jacob O. Wobbrock, and Jennifer Mankoff. Touchpad mapper: Examining information consumption from 2d digital content using touchpads by screen-reader users. In *The 26th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '24, page 1–4. ACM, October 2024. doi: 10.1145/3663548.3688505. URL http://dx.doi.org/10.1145/3663548.3688505. 6.2.4

[215] Lei Shi, Idan Zelzer, Catherine Feng, and Shiri Azenkot. *Tickers and Talker: An Accessible Labeling Toolkit for 3D Printed Models*, pages 4896–4907. Association for Computing Machinery, New York, NY, USA, 5 2016. Accessed: 2021-09-03. 2.1.2

[216] Lei Shi, Idan Zelzer, Catherine Feng, and Shiri Azenkot. Tickers and Talker: An Accessible Labeling Toolkit for 3D Printed Models. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 4896–4907. Association for Computing Machinery, New York, NY, USA, May 2016. ISBN 978-1-4503-3362-7. doi: 10.1145/2858036.2858507. Accessed: 2021-09-03. 2.1.2, 3.2.1

[217] Ben Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *The Craft of Information Visualization*, pages 364–371. Elsevier, 2003. doi: 10.1016/b978-155860915-0/50046-9. 4.4.2

[218] Ben Shneiderman, Christopher Williamson, and Christopher Ahlberg. Dynamic queries: database searching by direct manipulation. In *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '92*, CHI '92, page 669–670. ACM Press, 1992. doi: 10.1145/142750.143082. URL http://dx.doi.org/10.1145/142750.143082. 6.3.2

[219] Alexandru-Ionut Siean and Radu-Daniel Vatavu. Wearable interactions for users with motor impairments: Systematic review, inventory, and research implications. In *The 23rd International ACM SIGACCESS Conference on Computers and Accessibility*. ACM, October 2021. doi: 10.1145/3441852.3471212. 4.2.3

[220] Alexa Siu, Gene S-H Kim, Sile O'Modhrain, and Sean Follmer. Supporting accessible data visualization through audio data narratives. In *CHI Conference on Human Factors in Computing Systems*, CHI '22. ACM, April 2022. doi: 10.1145/3491102.3517678. URL `http://dx.doi.org/10.1145/3491102.3517678`. 6.2.4

[221] V. Sivaraman, F. Elavsky, D. Moritz, and A. Perer. Counterpoint: Orchestrating large-scale custom animated visualizations. In *2024 IEEE Visualization and Visual Analytics (VIS)*, pages 16–20, 2024. 5.2.1

[222] Aleksandra Slavkovic and Karen Cross. Novice heuristic evaluations of a complex interface. In *CHI '99 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '99, pages 304–305, New York, NY, USA, May 1999. Association for Computing Machinery. ISBN 978-1-58113-158-1. doi: 10.1145/632716.632902. 3.2.4

[223] Jim Smiley, Benjamin Lee, Siddhant Tandon, Maxime Cordeil, Lonni Besançon, Jarrod Knibbe, Bernhard Jenny, and Tim Dwyer. The made-axis: A modular actuated device to embody the axis of a data dimension. *Proceedings of the ACM on Human-Computer Interaction*, 5(ISS):1–23, November 2021. ISSN 2573-0142. doi: 10.1145/3488546. URL `http://dx.doi.org/10.1145/3488546`. 6.2.4, 6.4

[224] Volker Sorge. Polyfilling accessible chemistry diagrams. In *Lecture Notes in Computer Science*, pages 43–50. Springer International Publishing, 2016. doi: 10.1007/978-3-319-41264-1_6. 4.1, 4.2.1.1, 4.2.2.1

[225] Volker Sorge. Polyfilling Accessible Chemistry Diagrams. In Klaus Miesenberger, Christian Bühler, and Petr Penaz, editors, *Computers Helping People with Special Needs*, Lecture Notes in Computer Science, pages 43–50, Cham, 2016. Springer International Publishing. ISBN 978-3-319-41264-1. doi: 10.1007/978-3-319-41264-1_6. 2.2.3, 3.2.1

[226] L. South and M. A. Borkin. Photosensitive accessibility for interactive data visualizations. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–11, 2022. 5.2.1

[227] Laura South and Michelle Borkin. Generating Seizure-Inducing Sequences with Interactive Visualizations. In *Proc. of the IEEE VIS 2020 Posters*, October 2020. doi: 10.31219/osf.io/85gwy. 2.2.3, 3.2.1

[228] Laura South, David Saffo, and Michelle Borkin. Detecting and Defending Against Seizure-Inducing GIFs in Social Media. Technical report, OSF Preprints, January 2021. Accessed: 2021-08-20. 2.2.3, 3.2.1

[229] Katta Spiel, Kathrin Gerling, Cynthia L. Bennett, Emeline Brulé, Rua M. Williams, Jennifer Rode, and Jennifer Mankoff. Nothing about us without us: Investigating the role of critical disability studies in hci. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, page 1–8. ACM, April 2020. doi: 10.1145/3334480.3375150. URL `http://dx.doi.org/10.1145/3334480.3375150`. 1.1

[230] Clay Spinuzzi. The methodology of participatory design. *Technical communication*, 52 (2):163–174, 2005. 2.1.1

[231] Arjun Srinivasan, Tim Harshbarger, Darrell Hilliker, and Jennifer Mankoff. Azimuth: Designing accessible dashboards for screen reader users. In *The 25th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '23, page 1–16. ACM, October 2023. doi: 10.1145/3597638.3608405. URL http://dx.doi.org/10.1145/3597638.3608405. 6.2.1

[232] Sebastian Paul Suggate. Beyond self-report: Measuring visual, auditory, and tactile mental imagery using a mental comparison task. *Behavior Research Methods*, 56(8):8658–8676, September 2024. ISSN 1554-3528. doi: 10.3758/s13428-024-02496-z. URL http://dx.doi.org/10.3758/s13428-024-02496-z. 1.1

[233] Henny Swan, Ian Pouncey, Heydon Pickering, and Léonie Watson. Inclusive Design Principles, 2020. URL https://inclusivedesignprinciples.org/. Accessed: 2021-09-07. 3.3.2

[234] Sarit Felicia Anais Szpiro, Shafeka Hashash, Yuhang Zhao, and Shiri Azenkot. How people with low vision access computing devices: Understanding challenges and opportunities. In *Proceedings of the 18th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '16, page 171–180, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450341240. doi: 10.1145/2982142.2982168. 2.2.3, 3.2.1

[235] Pierre Tchounikine. Designing for appropriation: A theoretical account. *Human–Computer Interaction*, 32(4):155–195, July 2016. ISSN 1532-7051. doi: 10.1080/07370024.2016.1203263. URL http://dx.doi.org/10.1080/07370024.2016.1203263. 2.1.2

[236] John R Thompson, Jesse J Martinez, Alper Sarikaya, Edward Cutrell, and Bongshin Lee. Chart reader: Accessible visualization experiences designed with screen reader users. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. ACM, April 2023. doi: 10.1145/3544548.3581186. 4.1, 4.2.1.1, 4.3.1.1, 4.4.3, 6.2.1, 6.2.3

[237] Alexandra To, Angela D. R. Smith, Dilruba Showkat, Adinawa Adjagbodjou, and Christina Harrington. Flourishing in the everyday: Moving beyond damage-centered design in hci for bipoc communities. In *Proceedings of the 2023 ACM Designing Interactive Systems Conference*, DIS '23, page 917–933. ACM, July 2023. doi: 10.1145/3563657.3596057. URL http://dx.doi.org/10.1145/3563657.3596057. 2.1.1

[238] T. Tran, H.-N. Lee, and J. H. Park. Discovering accessible data visualizations for people with adhd. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, pages 1–19, 2024. 5.2.1

[239] US Government, 10x. Improving Accessibility in Data Visualizations (formerly USWDS Visualization Tool) on 10x, 2018. URL https://trello.com/c/OOOres7a/63-improving-accessibility-in-data-visualizations-formerly-uswds-visua Accessed: 2022-03-04. 3.2.4

[240] US Government, General Services Administration. Data visualizations | U.S. Web De-

sign System (USWDS), 2018. URL https://designsystem.digital.gov/components/data-visualizations/. Accessed: 2022-03-04. 3.2.4

[241] Visa. Visa Chart Components, 2019. URL https://github.com/visa/visa-chart-components. Accessed: 2021-09-06. 3.2.2

[242] Visa. Visa Chart Components. https://github.com/visa/visa-chart-components, 2022. Accessed: 2022-12-01. 4.2.2.1

[243] W3C. Web content accessibility guidelines (wcag) 2.1, 2018. 2.2.2.2, 5.4.1

[244] W3C. Webaim: Screen reader user survey #10 results. https://webaim.org/projects/screenreadersurvey10/, 2024. [Online]. 5.6.2.1

[245] WAI. Understanding Success Criterion 1.3.1 — Understanding WCAG 2.0. https://www.w3.org/TR/UNDERSTANDING-WCAG20/content-structure-separation-programmatic.html, 2008. [Accessed 01-04-2025]. 6.3.1

[246] WAI. Understanding success criterion 2.4.7: focus-visible. WCAG standard, W3C, 2016. Accessed: 2022-12-11. 4.3.3.2

[247] WAI. Understanding success criterion 4.1.2: name, role, value. WCAG standard, W3C, 2016. Accessed: 2022-03-04. 2.2.2.2, 4.2.1.3

[248] WAI. Accessible rich internet applications (WAI-ARIA 1.1). Technical report, W3C, 2017. Accessed: 2022-12-11. 3

[249] WAI. Understanding success criterion 2.1.1: keyboard. WCAG standard, W3C, 2017. Accessed: 2022-12-11. 4.3.2.2

[250] WAI. Accessibility principles. WCAG Standard, W3C, 2019. URL https://www.w3.org/WAI/fundamentals/accessibility-principles/. Accessed: 2021-09-06. 3.1, 3.2.3

[251] WAI. W3C Accessibility Guidelines (WCAG) 3.0. WCAG Standard, W3C, 2021. URL https://www.w3.org/TR/wcag-3.0/. Accessed: 2022-03-04. 3.7

[252] Chris Weaver. Multidimensional visual analysis using cross-filtered views. In *2008 IEEE Symposium on Visual Analytics Science and Technology*, page 163–170. IEEE, October 2008. doi: 10.1109/vast.2008.4677370. URL http://dx.doi.org/10.1109/VAST.2008.4677370. 2.2.1, 6.3.2

[253] WebAIM. WAVE, the web accessibility evaluation tool. https://wave.webaim.org/, 2001. Accessed: 2023-01-10. 4.4.1.1

[254] WebAIM. WebAIM: The WebAIM Million - An annual accessibility analysis of the top 1,000,000 home pages, 2019. URL https://webaim.org/projects/million/#wcag. Accessed: 2021-09-03. 3.4.1

[255] Simon Wheatcroft. Making Graphs And Plots Accessible For The Blind, June 2020. URL https://andadapt.com/making-graphs-and-plots-accessible-for-the-blind/. Accessed: 2021-09-06. 3.2.2

[256] David Gray Widder and Richmond Wong. Thinking upstream: Ethics and policy opportunities in ai supply chains, 2023. URL https://arxiv.org/abs/2303.07529. 1.2

[257] David Gray Widder, Meredith Whittaker, and Sarah Myers West. Why 'open' ai systems are actually closed, and why this matters. *Nature*, 635(8040):827–833, November 2024. ISSN 1476-4687. doi: 10.1038/s41586-024-08141-1. URL http://dx.doi.org/10.1038/s41586-024-08141-1. 1.2

[258] B. L. Wimer, L. South, K. Wu, D. A. Szafir, M. A. Borkin, and R. A. Metoyer. Beyond vision impairments: Redefining the scope of accessible data representations. *IEEE Transactions on Visualization and Computer Graphics*, 30(12):7619–7636, December 2024. 2.2.1, 2.2.3, 5.2.1

[259] Langdon Winner. *Do Artifacts Have Politics?*, page 177–192. Routledge, May 2017. ISBN 9781315259697. doi: 10.4324/9781315259697-21. URL http://dx.doi.org/10.4324/9781315259697-21. 1.2

[260] J. O. Wobbrock, S. K. Kane, K. Z. Gajos, S. Harada, and J. Froehlich. Ability-based design: Concept, principles and examples. *ACM Transactions on Accessible Computing*, 3(3):1–27, April 2011. 1.2, 5.2.2

[261] Jacob O. Wobbrock, Shaun K. Kane, Krzysztof Z. Gajos, Susumu Harada, and Jon Froehlich. Ability-based design. *ACM Transactions on Accessible Computing*, 3(3):1–27, April 2011. doi: 10.1145/1952383.1952384. 4.5

[262] Jacob O. Wobbrock, Shaun K. Kane, Krzysztof Z. Gajos, Susumu Harada, and Jon Froehlich. Ability-Based Design: Concept, Principles and Examples. *ACM Transactions on Accessible Computing*, 3(3):9:1–9:27, April 2011. ISSN 1936-7228. doi: 10.1145/1952383.1952384. Accessed: 2021-09-07. 2.1.1, 3.3.3

[263] Kanit Wongsuphasawat, Dominik Moritz, Anushka Anand, Jock Mackinlay, Bill Howe, and Jeffrey Heer. Voyager: Exploratory analysis via faceted browsing of visualization recommendations. *IEEE Transactions on Visualization and Computer Graphics*, 22(1): 649–658, January 2016. ISSN 1077-2626. doi: 10.1109/tvcg.2015.2467191. URL http://dx.doi.org/10.1109/TVCG.2015.2467191. 2.2.1

[264] R. Wood, J. H. Feng, and J. Lazar. Health data visualization literacy skills of young adults with down syndrome and the barriers to inference-making. *ACM Transactions on Accessible Computing*, 17(1):1–1, March 2024. 5.2.1

[265] R. Wood et al. "creatures of habit": influential factors to the adoption of computer personalization and accessibility settings. *Universal Access in the Information Society*, 23(2): 927–953, March 2023. 5.2.3, 5.3.1.3, 5.6.4.3

[266] K. Wu, E. Petersen, T. Ahmad, D. Burlinson, S. Tanis, and D. A. Szafir. Understanding data accessibility for people with intellectual and developmental disabilities. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–16, 2021. 5.2.1

[267] Keke Wu, Emma Petersen, Tahmina Ahmad, David Burlinson, Shea Tanis, and

Danielle Albers Szafir. Understanding Data Accessibility for People with Intellectual and Developmental Disabilities. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, pages 1–16, New York, NY, USA, May 2021. Association for Computing Machinery. ISBN 978-1-4503-8096-6. doi: 10.1145/3411764.3445743. Accessed: 2021-09-03. 2.2.3, 3.2.1

[268] Yifan Wu, Lutao Yan, Leixian Shen, Yunhai Wang, Nan Tang, and Yuyu Luo. Chartinsights: Evaluating multimodal large language models for low-level chart question answering, 2024. URL https://arxiv.org/abs/2405.07001. 6.2.2

[269] Cindy Xiong, Lisanne Van Weelden, and Steven Franconeri. The Curse of Knowledge in Visual Data Communication. *IEEE Transactions on Visualization and Computer Graphics*, 26(10):3051–3062, October 2020. ISSN 1941-0506. doi: 10.1109/TVCG.2019.2917689. Conference Name: IEEE Transactions on Visualization and Computer Graphics. 3.4.4

[270] F. Yanez, C. Conati, A. Ottley, and C. Nobre. The state of the art in user-adaptive visualizations. *Computer Graphics Forum*, December 2024. 5.2.2

[271] Katherine Ye, Wode Ni, Max Krieger, Dor Ma'ayan, Jenna Wise, Jonathan Aldrich, Joshua Sunshine, and Keenan Crane. Penrose. *ACM Transactions on Graphics*, 39(4), August 2020. doi: 10.1145/3386569.3392375. 4.4.3.1

[272] Rebecca Yeo and Karen Moore. Including disabled people in poverty reduction work: "nothing about us, without us". *World Development*, 31(3):571–590, March 2003. ISSN 0305-750X. doi: 10.1016/s0305-750x(02)00218-8. URL http://dx.doi.org/10.1016/S0305-750X(02)00218-8. 1.1

[273] Ashley Zelinskie, Elaine Stewart, Maggie Masetti, and Kan Yang. Equitable and Inclusive Public Outreach with the James Webb Space Telescope: Combining Art, Science, and Technology. https://ntrs.nasa.gov/citations/20240009020, 2024. [Accessed 01-04-2025]. 6.2.2

[274] Xiaoyi Zhang, Tracy Tran, Yuqian Sun, Ian Culhane, Shobhit Jain, James Fogarty, and Jennifer Mankoff. Interactiles: 3d printed tactile interfaces to enhance mobile touchscreen accessibility. In *Proceedings of the 20th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '18, page 131–142. ACM, October 2018. doi: 10.1145/3234695.3236349. URL http://dx.doi.org/10.1145/3234695.3236349. 6.2.4

[275] Haixia Zhao, Catherine Plaisant, Ben Shneiderman, and Jonathan Lazar. Data Sonification for Users with Visual Impairment: A Case Study with Georeferenced Data. *ACM Transactions on Computer-Human Interaction*, 15(1), May 2008. ISSN 1073-0516. doi: 10.1145/1352782.1352786. 2.2.3, 3.2.1

[276] Jonathan Zong, Crystal Lee, Alan Lundgard, JiWoong Jang, Daniel Hajas, and Arvind Satyanarayan. Rich screen reader experiences for accessible data visualization. *Computer Graphics Forum*, 41(3):15–27, June 2022. ISSN 1467-8659. doi: 10.1111/cgf.14519. URL http://dx.doi.org/10.1111/cgf.14519. 6.2, 6.2.1

[277] Jonathan Zong, Crystal Lee, Alan Lundgard, JiWoong Jang, Daniel Hajas, and Arvind

Satyanarayan. Rich screen reader experiences for accessible data visualization. *Computer Graphics Forum*, 41(3):15–27, June 2022. doi: 10.1111/cgf.14519. 2.2.2.2, 4.1, 4.2.1.1, 4.2.1.3, 4.2.2.1, 4.3.1.1, 4.3.3.3, 4.4.1.1, 4.4.2, 4.4.3, 4.5

[278] Jonathan Zong, Isabella Pedraza Pineros, Mengzhu (Katie) Chen, Daniel Hajas, and Arvind Satyanarayan. Umwelt: Accessible structured editing of multi-modal data representations. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, CHI '24, page 1–20. ACM, May 2024. doi: 10.1145/3613904.3641996. URL `http://dx.doi.org/10.1145/3613904.3641996`. 6.2.3