This is an excerpt from Frank Elavsky's dissertation on *Tool-making as an Intervention on the Accessibility of Interactive Data Experiences*, which can be accessed in full at this archival link:

http://reports-archive.adm.cs.cmu.edu/anon/hcii/CMU-HCII-26-103.pdf

This document contains the following sections:

- Abstract
- Table of contents
- Chapter 1: Introduction
- Chapter 2: Background & Related Work
- Chapter 3: Overview of Contributions
- Chapter 4: *Chartability*: Heuristics as a Tool and Resource
- Chapter 6: *Skeleton*: Visual Authoring of Non-visual Data Experiences
- Chapter 9: Discussion & Future Work (Sections 9.1 and 9.3 only)
- References

This document does not contain the following chapters:

- Chapter 5: *Data Navigator*: Low-level Tooling for Creating Navigable Data Structures
- Chapter 7: *Cross-perception*: Rethinking Input Design Towards Blind Analytical Interaction
- Chapter 8: *Softerware*: Enabling Personalization of Interactive Data Representations for Users with Disabilities
- Chapter 10: Biographical Sketch

Contents

List of Figures

Abstract

In this dissertation, we contribute practical advancements in tool-making as an intervention on the accessibility of interactive data experiences. The thesis of this dissertation is as follows: *This dissertation argues that the tools practitioners use to build interactive data experiences are themselves sites where accessibility barriers are produced, prevented, or alleviated for both end users and authors. This work contributes five tools—Chartability, Data Navigator, Softerware, Cross-perception, and Skeleton—that collectively center accessibility work on the empowerment of disabled and non-disabled practitioners across the full arc of evaluation, data navigation, analytical interaction, and personalization.*

Rather than framing accessibility research solely around ideal experiences for end users with disabilities, this thesis investigates why accessibility work is so difficult for the practitioners who build interactive data experiences and what tool-making can reveal about those difficulties. We organize this investigation around four domains where practitioners face the most persistent challenges: evaluation, navigation, interaction, and personalization. *Chartability*, a heuristic framework contributed first and maps the full landscape of accessibility barriers and identified these three latter domains (navigation, interaction, and personalization) as the areas where the most severe gaps remain. *Data Navigator* and *Skeleton* then investigate navigation, finding that practitioners struggle because navigation structure has no visible, manipulable representation in their workflows. *Cross-perception* engages interaction, demonstrating that blind data analysis has been constrained by existing tools and that a new interaction design framework can reshape what analytical work is possible. *Softerware* addresses personalization, revealing that access needs genuinely conflict across users and that meaningful personalization requires system-level infrastructure that does not yet exist in practitioner tooling ecosytems.

Combined, these contributions provide empirical insights and practical advancements in the state of the art for tooling that bridges gaps in current accessibility practices in visualization and data science. Our work ultimately enables people with and without disabilities to better evaluate barriers in, analyze with, design for, develop, and personalize interactive data experiences. We demonstrate that tool-making is a productive intervention that both engages accessibility barriers and elucidates why those gaps exist in practitioner work.

Part I

Introduction

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. 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" [53]. 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 [22, 62, 186]. 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 [144] 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 [51]. 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 [190]. 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 [89]? 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 [127]: 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 [127, 153, 155, 184, 218]. 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 first responsible* for repair: the curb designers and implementers.

And knowing who is first 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 first 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

Then 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 [210] and are sometimes, for this reason, regulated or made proprietary and controlled by powerful entities [66, 206].

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 [211] and also rules and guardrails about what anyone downstream from that tool's design *should* do [66, 205]. 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 [59].

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 [164], others are lower level but much more expressive [17]. 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 [77].

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 gaps that arise in this space: Practitioners face immense challenges when crafting accessible data experiences. We first need to educate practitioners on what accessibility barriers actually are in interactive visualizations. Then, we must help them engage the hardest barriers in this work and create building blocks that help them to construct navigable data experiences, build design frameworks that can inform entirely new kinds of data interaction, and develop software systems for end-user personalization and agency. Our research seeks to advance the state of the art in tools that assist in accessible data interaction while also using tool-making as an intervention that helps us to better understand and characterize *why* and *how* data practitioners face barriers themselves in this work.

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 [88, 91]. 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 [90]. 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 [185]. 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 [70]. 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 [32, 99]. 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 [68]. 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 [104, 192, 211].

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 [67]. 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 [41, 42, 159, 189]. 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 [8, 11], 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 [111, 148, 161, 178, 178]. 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 [80, 82, 148]. 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 [81]. 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 [47].

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 [9, 76, 117, 201]. 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 [77, 117, 214].

Automated data processing and cleaning have revolutionized workflows by reducing the time spent on manual data wrangling [49]. 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 [77]. 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 [164]. New visualization types and techniques—ranging from dynamic dashboards, faceting, to immersive 3D visualizations—offer novel ways to explore and interpret data [214].

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) [128]. 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 [208].

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 [46, 51]. Marriott et al. also found that visual disability considerations are the primary focus of data visualization literature [128], 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 [120]. 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 [196]. There has been a growing interest in developing guidelines for practitioners [43, 46] and even applying guidelines as a method of validation alongside human studies evaluations and co-design [51, 118, 119, 221]. 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 [197]. 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?" [120] nearly all topics of study at the intersection of accessibility and data are focused on visualization and vision-related disabilities [208]. 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) [109]. Marriott et al. found that there is no research at all that engages motor accessibility [128]. 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 [216] and seizure risk [182, 183]). 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) [188].

Since the 1990s, the most prominent and active accessibility topic in data has been color vision deficiency in data visualization [26, 116, 130, 142, 145]. Research projects that explore tactile sensory substitutions to charts have been a topic in computational sciences dating back to the 1983 [168], with tactile sensory substitutions being used for maps and charts as far back as the 1830s [69]. Sonification used both in comparison to and alongside visualization and tactile methods for accessibility dates as far back as 1985 [20, 35, 55, 124, 132, 219]. Some more recent work has explored robust screen reader data interaction techniques [63, 181], screen reader user experiences with digital, 2-D spatial representations, including data visualizations [165, 175], dug deeper into the semantic layers of effective chart descriptions [118], and investigated how to better understand the role of sensory substitution [31]. 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 [103].

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 [31, 51, 118, 175]. Jung et al. offer guidance to consider the order of information in textual descriptions and during navigation [103]. Kim et al. collected screen reader users' questions when interacting with data visualizations, which could open the door for more natural language data interaction [108].

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 [72, 89]. 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 [102, 177]. 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 [166] while another stresses that redundant encoding strategies are necessary [46].

These difficulties point to a deeper problem: the tools practitioners use to build data experiences were not designed with accessibility in mind, and the resulting gaps are not evenly distributed. Even in an ideal state where guidelines agree, the hardest remaining challenges cluster in three domains: navigation, interaction, and personalization. In these, practitioners lack the structural, technical, and infrastructural means to reason about accessible design and act on those considerations.

Chapter 3
Overview of Contributions

In this dissertation, we contribute practical advancements in tool-making as an intervention on the accessibility of interactive data experiences. The thesis of this dissertation is as follows: *This dissertation argues that the tools practitioners use to build interactive data experiences are themselves sites where accessibility barriers are produced, prevented, or alleviated for both end users and authors. This work contributes five tools—Chartability, Data Navigator, Softerware, Cross-perception, and Skeleton—that collectively center accessibility work on the empowerment of disabled and non-disabled practitioners across the full arc of evaluation, data navigation, analytical interaction, and personalization.*

Rather than framing accessibility research solely around ideal experiences for end users with disabilities, this thesis investigates why accessibility work is so difficult for the practitioners who build interactive data experiences and what tool-making can reveal about those difficulties. We organize this investigation around four domains where practitioners face the most persistent challenges: evaluation, navigation, interaction, and personalization. *Chartability*, a heuristic framework contributed first and maps the full landscape of accessibility barriers and identified these three latter domains (navigation, interaction, and personalization) as the areas where the most severe gaps remain. *Data Navigator* and *Skeleton* then investigate navigation, finding that practitioners struggle because navigation structure has no visible, manipulable representation in their workflows. *Cross-perception* engages interaction, demonstrating that blind data analysis has been constrained by existing tools and that a new interaction design framework can reshape what analytical work is possible. *Softerware* addresses personalization, revealing that access needs genuinely conflict across users and that meaningful personalization requires system-level infrastructure that does not yet exist in practitioner tooling ecosytems.

We engage each domain below with the questions: “Why does this work matter?”, “Why is it hard?”, and “What has tool-making within this domain showed us?”

The 4 domains of work that I engage in this thesis start first with **evaluation**. In work contexts where someone is designing and developing interactive data experiences, the practitioner must have the knowledge, tools, and resources available to systematically identify how their interfaces produce barriers for people with disabilities. A significant portion of professional accessibility work (arguably most, if not all) is founded on auditing and evaluating barriers to access. This work is pre-dominantly done through a standards-based approach [196], although in more robust evaluation work, people with disabilities are actively involved in the process [149].

Evaluation is difficult work because much of it is contextually defined by the author themselves, and most tasks at this intersection require careful, non-automated processes and methods [39, 149]. To make matters more difficult, no comprehensive guidelines, tests, and tools exist in any singular location. Practitioners often must gather these resources themselves, which tend to be situated towards accessibility in general or are high-level and provide minimal usefulness in practice. Additionally, practitioners themselves often have little knowledge about the veracity or quality of any given bit of information they gather [102, 177], and often do the work themselves to synthesize this disparate space of information into a usable format they can apply to their own evaluation work.

To engage this, our first main chapter focuses on *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. In this project, we did the hard work for other practitioners and contributed our collection of synthesized accessibility resources in a single workbook. We had practitioners try out our resource in real environments, in-situ, in order to learn more about the challenges and barriers they themselves faced in evaluation work. With *Chartability*, practitioners, especially those with limited accessibility expertise, gained more confidence and clarity in assessing and improving their work. Additionally, *Chartability* has since become widely applied as a framework that isn't just used for evaluation but also as design guidance in many contexts, internationally, including policy organizations, governmental groups, and more than 100 companies and businesses.

Chartability then opened up a significant landscape of new projects and research directions. From a combination of my existing expertise as a visualization designer and engineer, in addition to continued application of *Chartability* in the wild, we began to identify the trickiest and most-difficult domains of work for practitioners. Chartability has 50 total heuristics, or tests, each organized under one of 7 principles. But 3 larger domains began to emerge as the areas where the most severe and dramatic accessibility barriers remained unaddressed: on data *navigation*, analytical *interaction*, and interface *personalization*.

So the next section of this thesis engages the first of these three: **navigation**. Navigation is a fundamental type of interaction that is leveraged by modern software-based assistive technologies. Screen readers, the primary tool used by people who are blind to interact with computers, navigate content. Additionally, many other assistive technologies, such as a sip and puff device (like the "POSSUM" from as far back as '63 [122]) also navigate. Navigational technologies are leveraged by people with a significant array of disabilities, yet tend to be entirely ignored by existing data visualization tools, which are pre-dominantly built to support direct input (using a computer mouse).

Empirical work has already demonstrated that structural navigation is actually good [221], even when regular alternative text (image descriptions) exist. This is both because people who are blind can gain both a high level understanding (from the description) as well as lower-level sense of the data's structure and arrangement, in addition to the fact that discrete, structural navigation exposes interactivity that may exist on any visualization elements (such as they can be hovered or clicked with a mouse in order to perform some action). So, if good empirical work exists: *why haven't practitioners put this research into action? What makes this work hard to do?*

We first built *Data Navigator* to provide the building blocks we needed in order to address the technical and conceptual gaps that were required to make any visualization or visualization tool provide a navigable, interactive structure. *Data Navigator* is a low-level toolkit which can be used to construct accessible navigation structures such as lists, trees, and diagrams from an underlying graph structure. We leveraged graph theory for an applied HCI problem: nodes and edges represent any relationships within the data as a structure, which then supports rich expressiveness of data navigation experiences. Users can navigate discrete marks in a visualization, clusters, groupings, and more. In addition to its structural scaffolding, *Data Navigator* also supports a wide array of input modalities leveraged by people with disabilities (screen readers, keyboards, speech, and gestures).

Data Navigator provided a substrate, but this contribution alone wasn't enough to engage the question *why is navigation so hard, in practice?*. So the chapter following *Data Navigator* introduces *Skeleton*, a data navigation authoring tool built on top of *Data Navigator*. Our novel approach in *Skeleton* involves visualizing and making manipulable the nodes, edges, and textual data that comprise non-visual end user experiences. *Skeleton* visualizes the building blocks that comprise *Data Navigator*. Additionally, *Skeleton* provides expressive, rapid scaffolding capabilities that leverage data visualization rendering engines. This scaffolding engine helps practitioners quickly create common configurations for non-visual data navigation structures that retain visual congruence to the underlying structure.

But most importantly, *Skeleton* serves as a framework that shapes designerly consideration. Our conjecture was that because sighted practitioners cannot *see* navigation building blocks, they will not treat those elements as iterable design materials. We conjectured: Navigation is hard in practice because sighted designers face barriers to iteration and understanding. We conducted an empirical study with sighted practitioners and found that making non-visual elements visual helped practitioners shift from treating accessibility as a compliance task to treating it as a design problem, re-iterating on the visual aspects of their design, and engaging in the complex and nuanced components that comprise data navigation experiences.

Now, **interaction** becomes the next area we wanted to engage. Existing accessible data interaction for people who are blind, including our previous work on navigation (which is a form of interaction), predominantly seeks to expose information. This is what we call *access-oriented interaction*. In terms of low-level analytical tasks, most are then made feasible through navigation, sonification, or summarization-based and question-answering approaches: retrieving values, filtering, computing derived values, sorting, determining ranges, clustering, and finding outliers. What remains are analytical tasks that, despite being "low level" (understood as *unable to be reduced into more fundamental tasks*), are cognitively highly complex: finding correlations and characterizing distributions [4]. These tasks require complex hypothesization and exploration, rather than a system that simply encourages surfacing what is known or what is already present in the data: it requires combining, remixing, restructuring, and dividing data.

But blind *analytical interaction* isn't just important to engage because it is understudied, it is important to engage because many of interactive information visualization's most impactful tools for data science enable it [55, 77, 138, 201]. In visualization, *cross-filtering* is one example of an interaction that enables a user to filter one visual space while seeing a coordinated change in another visual space simultaneously and near-instantaneously. The speed of input interaction and perception of output also matters: even a small bit of latency changes the quality of a user's data exploration activities [117]. We conjectured that a screen reader, the most-used tool leveraged by blind people when interacting with computers, may be insufficient for engaging this task.

To engage this, this thesis introduces *Cross-perception*, an approach for building analytical interactions that support perception in one space of input interaction with simultaneous, non-competing perception of output in another space of data representation. We first formalized a design framework for producing *cross-perception* experiences and then built a novel prototype device, the *cross-feelter*, that enables blind *cross-perception* of a cross-filtering data exploration interface. In an empirical study with blind users (with and without existing data expertise), we found *cross-perception* speeds up analytical exploration by 90% and helps blind users consider vastly more questions of their dataset (+188% computational queries, +54% spoken aloud) compared to a screen reader-driven interaction.

Beyond performance, we found that our input modality itself shaped the character of analytical engagement: participants didn't just work faster, they considered more dimensions of their data and asked qualitatively different questions. The *cross-feelter* also reduced anxiety and substantially increased enjoyment, particularly for participants without prior data expertise, suggesting that the barriers blind practitioners face in data work are not only functional but affective. Additionally, we had our blind practitioners imagine new interaction possibilities that *cross-perception* could enable including and beyond our *cross-feelter* device.

Our final domain of work is the most difficult for visualization practitioners to engage: **personalization**. While *navigation* demanded better software tooling and visual support and *interaction* required new hardware, *personalization* completely re-orients how software authoring takes place. Personalization matters because of *access friction*, which is a design challenge where one design or interface configuration that might be accessible for one person or group of people turns out to create barriers for someone else [72, 89]. In existing work on personalization and accessibility, studies have demonstrated that end-user control is great to have and can alleviate friction [100, 215], but little work has been done to explore what personalization looks like for an existing data visualization library and how practitioners should build and maintain their existing systems to support it.

Our final chapter introduces *Softerware* to address the tension between standardized accessible design and the diverse needs of real users with disabilities. In the wild, *access friction* exists in every design that reaches a public audience; it is inevitable. This tension ultimately means that some users have a worse experience, and may even face exclusion, with any particular design configuration. So for this work, we conducted our research in-situ with visualization software engineers and designers and worked to build a scalable, flexible software system dubbed "softerware" that enables end users to manipulate the appearance and functionality of the charts and graphs they encounter according to their own preferences. We conducted empirical research to inform our collaborators, as well as other visualization system authors, with guidelines and considerations for building *softerware* systems. In our study, no two participants chose the same preference configuration and participants with the same diagnosed condition sometimes needed opposite design treatments. Practitioners, meanwhile, immediately raised ethical concerns about whether personalization would let designers off the hook for poor defaults. These findings revealed that the real barriers to personalization are not at the level of any individual chart but at the level of system infrastructure: without persistence, cross-system interoperability, and shared standards, the effort required of end users exceeds the value they receive.

Combined, these contributions provide empirical insights and practical advancements in the state of the art for tooling that bridges gaps in current accessibility practices in visualization and data science. Our work ultimately enables people with and without disabilities to better evaluate barriers in, analyze with, design for, develop, and personalize interactive data experiences. We demonstrate that tool-making is a productive intervention that both engages accessibility barriers and elucidates why those gaps exist in practitioner work.

Part II

Evaluation: Helping Practitioners Identify Accessibility Barriers

Chapter 4
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.

4.1 Abstract

Novices and experts have struggled to evaluate the accessibility of data visualizations because there are no common shared guidelines across environments, platforms, and contexts in which data visualizations are authored. Between non-specific standards bodies like WCAG, emerging research, and guidelines from specific communities of practice, it is hard to organize knowledge on how to evaluate accessible data visualizations. We present Chartability, a set of heuristics synthesized from these various sources which enables designers, developers, researchers, and auditors to evaluate data-driven visualizations and interfaces for visual, motor, vestibular, neurological, and cognitive accessibility. In this paper, we outline our process of making a set of heuristics and accessibility principles for Chartability and highlight key features in the auditing process. Working with participants on real projects, we found that data practitioners with a novice level of accessibility skills were more confident and found auditing to be easier after using Chartability. Expert accessibility practitioners were eager to integrate Chartability into their own work. Reflecting on Chartability's development and the preliminary user evaluation, we discuss tradeoffs of open projects, working with high-risk evaluations like auditing projects in the wild, and challenge future research projects at the intersection of visualization and accessibility to consider the broad intersections of disabilities.

4.2 Overview

26% of people in the United States self-report living with at least one disability [144]. 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 [146]. 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 [39], 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 [149] 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) [199] 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.

4.3 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.

4.3.1 Research Advancements in Data Visualization and Accessibility

In parallel to Mack et al.'s "What do we mean by Accessibility Research?" [120] 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) [109]. Marriott et al. found that there is no research at all that engages motor accessibility [128]. 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 [216] and seizure risk [182, 183]). 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) [188].

Since the 1990s, the most prominent and active accessibility topic in visualization has been color vision deficiency [26, 116, 130, 142, 145]. Research projects that explore tactile sensory substitutions have been a topic in computational sciences dating back to the 1983 [168], with tactile sensory substitutions being used for maps and charts as far back as the 1830s [69]. Sonification used both in comparison to and alongside visualization and tactile methods for accessibility dates as far back as 1985 [20, 35, 55, 124, 133, 219]. Some more recent work has explored robust screen reader data interaction techniques [63, 181], screen reader user experiences with digital, 2-D spatial representations, including data visualizations [165, 175], dug deeper into the semantic layers of effective chart descriptions [118], and investigated how to better understand the role of sensory substitution [31]. 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 [103]. 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 [6, 27, 28, 29, 114, 143, 150, 174] and haptic graphs and tactile interfaces [1, 5, 16, 21, 23, 60, 97, 98, 115, 168, 169, 178]. 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 [196] or The American Printing House for the Blind and Braille Authority of North America [7]. All of these challenges are factors that limit the generalizability of these artifacts and knowledge for practitioner use [119, 137, 175]. 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.

4.3.2 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 [83] or Visa Chart Components (VCC) [195] and tools like the Graphics Accelerator in SAS [163] 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 [24, 33, 57, 58, 203]. 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 [85]. Tableau users more broadly must resort to voting systems to gather attention to accessibility issues [38]. Semiotic's accessibility features were added by community member Melanie Mazanec [131]. For Microsoft's PowerBI, students have organized resources for how to make visualizations built with it more accessible [112] while non-profits like the City of San Francisco's data team have had to build features like keyboard instructions from scratch [160]. Mapbox GL JS is an example of a popular mapping library (over 400,000 weekly downloads) [126] 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 [125].

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.

4.3.3 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 [7], 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 [14, 119, 175].

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) [196] influence accessible technology policy and law for more than 55% of the world's population [94]. WAI and WCAG outline 4 types of functional accessibility principles: Perceivable, Operable, Understandable, and Robust, abbreviated as POUR [199]. POUR is the foundation that organizes all 78 accessibility testing criteria in WCAG.

4.3.4 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" [193, 194].

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 [18, 30, 61, 101, 130, 140, 141, 147, 162, 180]. Heuristics are also not new in visualization [34, 56, 156, 170] even among topics related to accessibility (color vision deficiency, specifically) [145, 162].

4.4 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 [65].

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 4.1:

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 [19] 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 accessibility 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 4.1, 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 4.1: 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				

4.4.1 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.

4.4.2 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" [187] and what Doug Schepers meant by "data visualization is an assistive technology" [167]. 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.

4.4.3 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 [10, 123]. 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 [113, 212].

4.5 Using Chartability

All of Chartability's tests are performed using Chartability's workbook [44] 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.

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.

4.5.1 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) [202]. 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 4.1). 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

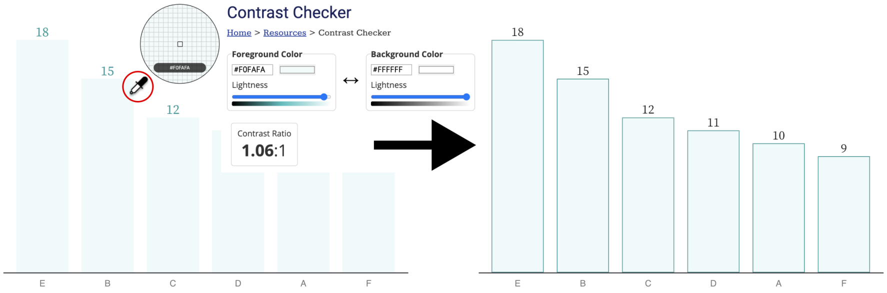


Figure 4.1: A low contrast chart (left) compared to a higher contrast version (right). A dropper tool is extracting the fill color of the bar and then a contrast ratio has been calculated. Note that the fill color is the same on both bars, but darker borders have been added to ensure the visualization passes contrast tests.

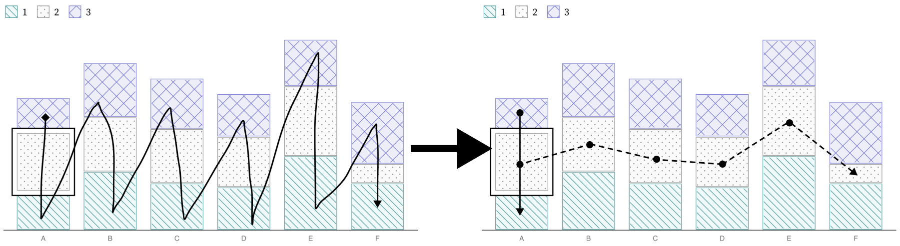


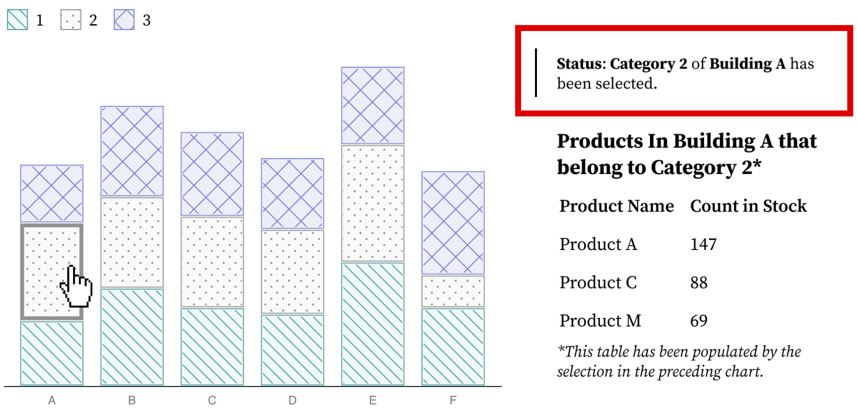
Figure 4.2: Keyboard navigation paths on a stacked bar chart. The left shows a serial navigation example, typically just a default of rendering order. The right shows both groups (the stack of bars) and categories (the color/texture shared among bars across stacks) as dimensions to explore laterally or vertically.

Figure 4.8). And another common, critical failure from Perceivable is text size. No text should be smaller than 12px/9pt in size.

4.5.2 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 4.2, Figure 4.3), 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 4.7). 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



Status: Category 2 of **Building A** has been selected.

Products In Building A that belong to Category 2*

Product Name	Count in Stock
Product A	147
Product C	88
Product M	69

**This table has been populated by the selection in the preceding chart.*

Figure 4.3: 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.

multiple inputs (Figure 4.3), whether the data structure that is rendered is navigable according to its structure (Figure 4.2), and whether keyboard navigability across all elements in a data interface is even necessary (Figure 4.7).

4.5.3 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 4.4) 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 whether that mark is interactive (Figure 4.5), whether status updates that reflect context change provide alerts (Figure 4.3), and whether summary textual information is provided about the whole chart (Figure 4.4) as well as statistically and visually important areas of that chart (Figure 4.7).

4.5.4 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 4.4). This alone is one of the most important things to check for. In complex cases where a chart has a visual feature

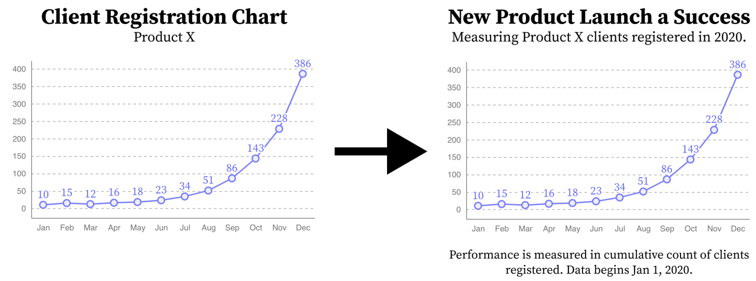
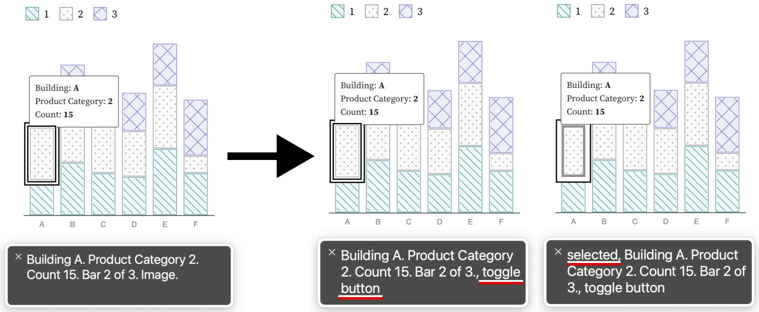


Figure 4.4: Charts must have a visually available textual explanation provided that summarizes the outcome. “Client Registration Chart” for “Product X” (left) is inaccessible while “New Product Launch a Success” (right) gives a clear takeaway.



Building A. Product Category 2. Count 15. Bar 2 of 3. Image.

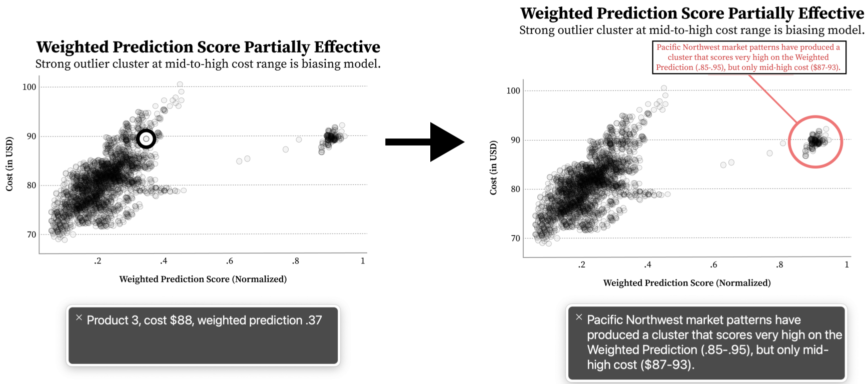
Building A. Product Category 2. Count 15. Bar 2 of 3., toggle button

selected, Building A. Product Category 2. Count 15. Bar 2 of 3., toggle button

Figure 4.5: 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.

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 4.6: 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.



Pacific Northwest market patterns have produced a cluster that scores very high on the Weighted Prediction (.85-.95), but only mid-high cost ($87-93).

× Product 3, cost $88, weighted prediction .37

× Pacific Northwest market patterns have produced a cluster that scores very high on the Weighted Prediction (.85-.95), but only mid-high cost ($87-93).

Figure 4.7: 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.

with an assumedly obvious takeaway, checking for annotations or textual callouts is important to help avoid interpretive issues [217] (Figure 4.7).

4.5.5 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 4.8), run automatic semantic evaluations as well as manually check for appropriate meaning (Figure 4.5), and check if dense or highly complex visuals have sonified, tactile, or textual summaries available (Figure 4.7). Auditors should also check whether system updates provide clear feedback textually (Figure 4.3) as well as checking if there are both high and low level representations of information available (Figure 4.6).

Auditors should be especially critical of static designs, such as those that either use textures by default or not (Figure 4.8), which are a high risk of compromising and assistive failure.

4.6 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

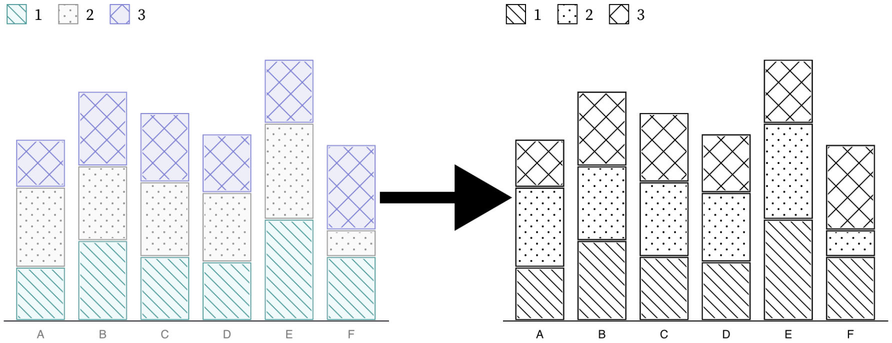


Figure 4.8: A bar chart with categories (left) shown not conforming to Windows High Contrast White Mode. High contrast mode on Windows requires limiting color palettes, using only black or white for most elements (shown on the right).

(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.

4.6.1 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).

4.6.2 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 4.1. 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.

4.6.3 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 [37]. 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 [3]. 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 features in mind [84] 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.

4.7 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?

---

[1] "Non-applicable:" any test in the auditing process 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

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.

4.7.1 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."

4.7.2 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."

4.7.3 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.”

4.7.4 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.”

4.7.5 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 [40]). 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.”

4.7.6 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.”

4.7.7 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."

4.8 Extended Results

Following calls to ensure accessibility work has practical benefits that exceed publications [90], 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 [200].

Projects like Turkopticon benefited from the discussion about how a community actually used their tool [95]. 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, [90] 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.

4.9 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.

4.10 Conclusion

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.

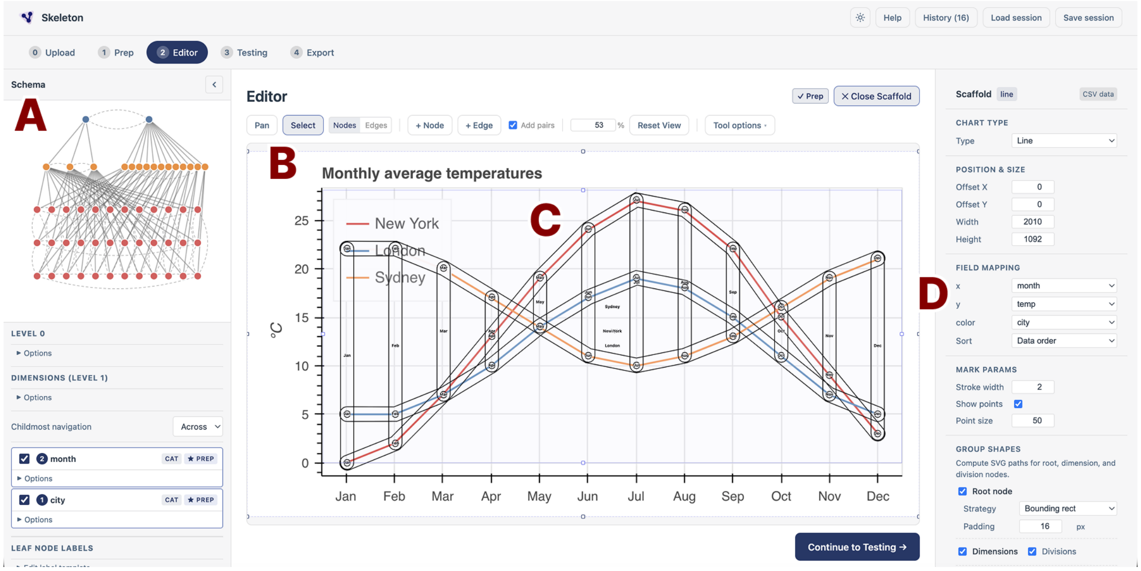
Part III

Navigation: Making Data Structures Traversable

Chapter 6
Skeleton: Visual Authoring of Non-visual Data Experiences

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

> F. Elavsky, C. Nnadozie, L. Nadolskis, P. Carrington, and D. Moritz, '*Skeleton*: Visual Authoring of Non-visual Data Experiences', *IEEE Transactions on Visualization and Computer Graphics*, 2026.



☑ Add pairs: 53 %

Type: Line

Offset X: 0

Offset Y: 0

Width: 2010

Height: 1092

x: month

y: temp

color: city

Sort: Data order

Stroke width: 2

Show points: ☑

Point size: 50

Compute SVG paths for root, dimension, and division nodes.

☑ Root node

Strategy: Bounding rect

Padding: 16 px

☑ Dimensions ☑ Divisions

Childmost navigation: Across

☑ 2 month CAT ★ PREP

☑ 1 city CAT ★ PREP

Figure 6.1: 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.

6.1 Abstract

When sighted practitioners author accessible data visualizations, they build navigation structures (the nodes, edges, and input bindings that govern how assistive technologies traverse an interface) entirely in code, with no visual representation. This invisibility makes navigation structures difficult to inspect, debug, and iterate on. To sighted practitioners, every other aspect of a visualization is iterated on because it is visible; navigation structure ships as a first draft, if at all,

because it is not. Without a representation to react to, practitioners cannot develop judgment about what makes navigation good or bad, and the quality ceiling of non-visual experiences is set by the absence of a feedback loop. We address this problem through longitudinal co-design with practitioners across cartography, design systems, and open-source visualization, and make three contributions. First, we introduce technical advancements for making the properties of accessible navigation structure visible and directly manipulable during authoring, grounded in two foundational pieces of infrastructure produced by our co-design work: an *Inspector* that renders navigation graphs as interactive node-link diagrams, and a *Dimensions API* that expresses navigation in terms of data dimensions rather than explicit graph construction. Second, building on these, we present *Skeleton*, a direct-manipulation authoring environment in which the properties of an accessible navigation structure are translated into visual representations authors can observe and manipulate. Key techniques include a dual-view editor that simultaneously shows the system's navigation model and the end user's spatial experience, a scaffolding engine that automates spatial node placement by repurposing a visualization rendering pipeline, a live label-template editor with real-time screen-reader-output preview, and a testing mode that makes traversal sequence visually trackable. Third, we evaluate *Skeleton* through an in-situ study with 8 practitioners across visualization design, engineering, and research. Making navigation structure visible changed how practitioners engaged with accessible design: they reconsidered the architecture of their own visualizations, attended to a broader range of input modalities, and shifted from treating accessibility as a compliance task to treating it as a design problem.

6.2 Overview

We start this work with a provocation: How might the discipline of visualization help the discipline of accessibility?

Visualization has spent decades developing techniques for a specific class of problem: representing and interacting with information visually. Grammars of visual encoding [134, 164, 204, 207, 222], direct-interaction interfaces [48, 92, 179], and iterative feedback loops between representation and understanding [54, 76, 117] are all methods that enable abstraction and manipulation of the information that underlies the visual representation. We argue these methods have a direct application within the discipline's own accessibility challenges, one that has not yet been explored.

Sighted practitioners who build accessible data visualizations face an unusual authoring problem. The non-visual navigation structures they construct (the nodes, edges, focus states, input bindings, and semantics that govern how assistive technologies traverse a chart) exist only as code. A practitioner can write a navigation hierarchy, but cannot see it, click on a node to inspect what will be announced, observe the spatial relationship between a navigation path and the chart it overlays, and then manipulate its properties through direct interaction. Every other aspect of a visualization has a visible, inspectable representation during authoring: the visual encodings are visible, the layout is visible, the interaction states are visible. And yet, navigation structure is not.

This invisibility has practical consequences. Without a way to see what they are building, sighted practitioners cannot easily catch structural errors, compare design alternatives, and iterate. Accessibility becomes downstream of every other design choice, not because practitioners choose to deprioritize it, but because the authoring conditions do not support anything else [102, 177]. The floor and ceiling of non-visual data experiences are constrained by what sighted authors can perceive of their own work.

We engage this space with the following research questions:

R1 (Qualitative, Exploratory): What challenges do sighted practitioners face when designing and engineering navigation structures for accessible visualizations?
R2 (Qualitative, Exploratory): How do sighted authors reason about the non-visual experiences that accompany their visualizations?
R3 (System, Design): How can we make the properties of accessible navigation structure visible and directly manipulable during authoring?
R4 (Qualitative): In what ways does a directly manipulable visual representation of navigation structure change how practitioners find errors and improve upon their designs?

We address these questions through longitudinal co-design with practitioners across cartography, design systems, and open-source visualization, following an action research orientation [75] in which the research team was embedded in each community's active development work. This paper makes three contributions:

First, **we introduce technical advancements** for making the properties of accessible navigation structure visible and directly manipulable during authoring. These techniques are grounded in our co-design collaborations, which produced two foundational pieces of infrastructure: an *Inspector* that renders any navigation graph as an interactive node-link diagram, and a *Dimensions API* that formalizes a declarative grammar for expressing navigation in terms of data dimensions rather than explicit graph construction (**R1**, **R2**, **R3**).

Second, building on this infrastructure, **we present *Skeleton*, a direct-manipulation authoring environment** in which the topology, spatial mapping, semantics, and input logic of an accessible navigation structure are made visible and directly manipulable. *Skeleton* is built on Data Navigator [47], a code-based library for constructing interactive data navigation structures (**R3**). Our intention with *Skeleton* is to continue to develop it towards a usable, practical system beyond the scope of this research.

Third, **we contribute findings from an in-situ interview study** with 8 practitioners across visualization design, engineering, and research. We evaluate Skeleton as a design probe [71, 93] rather than a deployable system, focusing on qualitative shifts in practitioner engagement rather than task performance. We find that making navigation structure visible shifted how participants engaged with accessible design: they reconsidered the architecture of their own visualizations, attended to a broader range of input modalities, and shifted from treating accessibility as a compliance task to treating it as a design problem (**R1**, **R2**, **R4**).

6.3 Related Work

6.3.1 Non-visual Data Experiences

Blind people who rely on assistive technologies interact with data in fundamentally different ways than sighted users who use a direct pointer, like a mouse [109, 128, 175]. A substantial body of research has documented what these experiences look like across modalities, and what it takes to make them meaningful. In the context of "data experiences," this paper focuses specifically on interactive navigation structures, but we briefly survey adjacent modalities to situate our contribution.

Alternative text and natural language. A dominant strand of this work concerns the generation and evaluation of textual descriptions of visualizations [103, 106, 110, 118]. More recent LLM-driven systems and Q/A approaches can caption charts with varying degrees of semantic depth, some at a risk of producing bias [45, 52, 108]. While alt text makes a visualization's message available without sight, it is by nature static: a description conveys what a visualization says, but not how a user might explore or interact with it.

Sonification, haptics, and tactile rendering. Non-visual data experiences extend well beyond text. Sonification encodes data as sound [20, 79, 124], with declarative grammars emerging for authoring these experiences [107]. Haptic and tactile representations offer another channel through refreshable displays, 3D-printed graphics, and multimodal touchscreen interactions [23, 86, 129, 154]. Recent systems integrate multiple non-visual modalities around a single data representation [31, 87, 172].

Interactive navigation structures. The primary focus of our work centers on the state of research related to structured navigation: the traversal of data points, groupings, and interface elements through assistive technologies and keyboard input [47, 181, 198, 221]. Existing systems and interfaces have demonstrated that going beyond static descriptions to support hierarchical, traversable data structures meaningfully improves how blind users can explore and reason about charts [191, 221]. Giving users control over the textual tokens surfaced during navigation improves comprehension and agency [100]. And more recent work has found that perceptually congruent navigation structures for charts and diagrams can improve goal-driven exploration [135].

6.3.2 Authoring Non-visual Data Experiences

Accessible visualization has historically centered the experiences of disabled users, but a parallel and increasingly urgent body of work examines the experiences of the people who build these experiences: visualization designers, engineers, and researchers.

Practitioner challenges. Research consistently finds that sighted visualization practitioners struggle with accessibility [51, 102, 177]. Most visualizations in the wild are inaccessible and designers themselves report lacking guidance, especially for complex and interactive graphics [102, 177]. And screen reader users experience the downstream effects of these gaps: inconsistent structure, poor keyboard support, and information that is present visually but absent in the accessibility tree [51, 175]. The pattern across this work is consistent: the practitioners who build visualizations lack the tools and feedback mechanisms to make non-visual experiences effective, useful, and good.

Across practitioner-centered literature, a recurring finding is that non-visual experiences are treated as downstream of visual design choices, added after the visual representation is finished rather than designed in parallel [102, 119, 177, 223]. This sequencing has consequences: what is navigable and how it is structured is constrained by whatever visual decisions came first.

Authoring-oriented systems. A distinct line of work has focused on building authoring tools and libraries that give practitioners more tractable paths to accessible output. Few visualization tools support the kinds of interactive navigation structures that assistive technology users most benefit from [110]. Of those that do, most rely on code-based approaches [15, 47, 174, 176]. *Umwelt* [223] takes a different and notable approach: it is a structured editing environment where authors specify representations *across* modalities (sonification and visualization) in an integrated interface, where navigation is made available over the visualization using *Olli* [15]. The latest work in this space is *Arboretum*, a tool that provides automatic conversion of diagrams to a tabular structure, navigable structure, and tactile representation [209]. In *Arboretum*, the input visual is treated as the ground truth for the navigation structure, which is treated as downstream output.

Communicating visually, authoring invisibly. There is a revealing irony across this body of related work: research about navigation structures almost invariably communicates those structures visually. Papers such as "rich screen reader experiences" [221], *ChartReader* [191], *Benthic* [135], and *Data Navigator* [47] each use visual node-link diagrams and hierarchical schematics to explain navigation paths to their readers. The same pattern holds in adjacent domains that involve structuring data for navigation, such as PDF remediation [139].

And in authoring-oriented systems, none provide a visual interface through which practitioners can interactively inspect and manipulate navigation structures as a first-class design material. Structure output is either downstream of code or static visuals. Structure is always *derived* or *specified*; indirectly manipulated. Additionally, verification of the structure across all of these systems requires developers to manually navigate using a screen reader after the structure has been authored and rendered, before returning to the upstream visual design space or code.

Navigation structure is understood and communicated visually by sighted researchers and designers, yet built entirely without visual feedback by developers. We seek to address this gap.

6.4 Co-design Foundations

Our research follows an *action research* orientation [75], in which knowledge is generated by engaging with a community to solve a real problem in-situ, alongside them rather than studying it from the outside. These collaborations started from a shared motivation: practitioners needed to make their existing systems accessible to navigational assistive technologies, using Data Navigator [47] as a foundation. Across three projects, we worked with 12 individuals outside our research team. Three blind co-designers shaped the work throughout: CD1 and CD2 (anonymous) and a co-author on this paper, [REDACTED]. CD1's contributions are noted in Section 6.4.1, CD2's are noted in Section 6.5.3. [REDACTED] contributed to early ideation, problem formation, and framing for the project as a whole, helping define the authoring challenges that *Skeleton* addresses, in addition to feedback on study design.

The co-design literature on accessibility has centered people with disabilities as primary design partners [35, 36, 151, 191, 220, 221], and rightly so. Our co-design takes sighted practitioners as primary partners because the authoring challenges we address happen on the sighted side of the process: it is sighted authors who cannot see what they are building.

Two themes converged across all three engagements, and we present them here before describing the individual projects that surfaced them. First, **practitioners communicated and reasoned about accessible navigation using visual representations**: while designing for language, sound, and structure, our collaborators drew on paper, built wireframes of nodes and edges, and reflected on the design space using visual artifacts. Even when collaborating with blind co-designers, a visual medium was the first language of sighted authors. Second, **development that followed visual design work faced severe iteration barriers**: verifying a navigation experience required building a working code prototype and manually navigating it with a screen reader. The gap between visual design and code-based scaffolding with manual testing produced repeated mistakes, misinterpretations, and abandoned prototypes. Each project below contributed distinct evidence for these themes and surfaced specific requirements that shaped our infrastructure and tooling.

6.4.1 Geologic Map

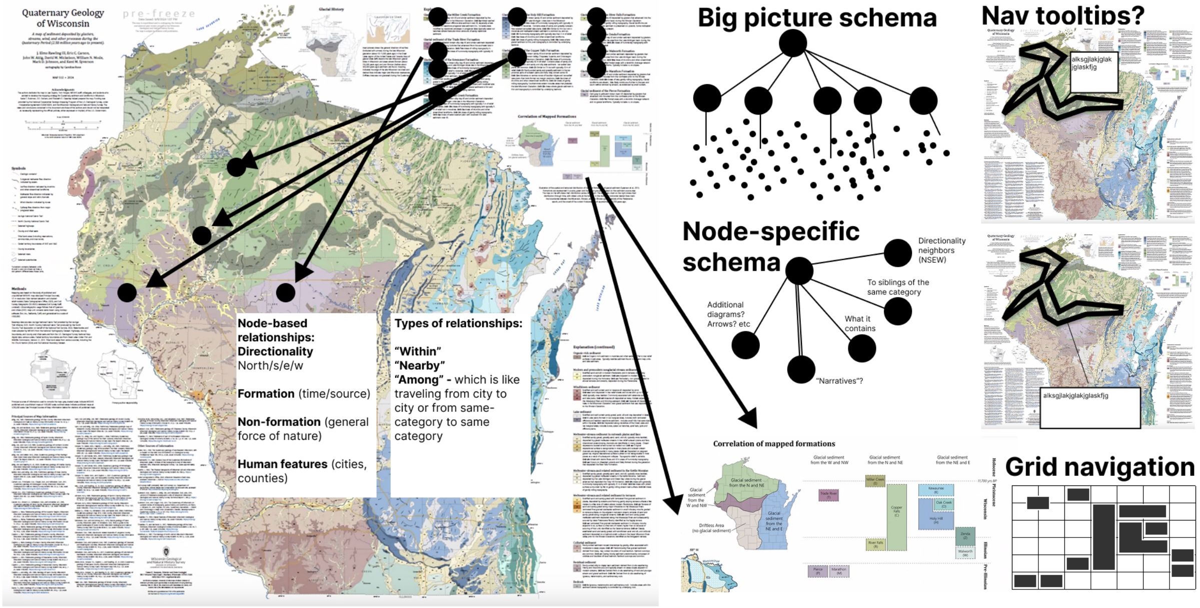


Figure 6.2: Our visual design work in Figma over a static geologic infographic map of Wisconsin. We use visual forms and illustrations over and beside the map to communicate flows, structure, navigation styles, and interaction patterns.

Our longest engagement spanned just over two years, with a cartographer, accessibility consultant, and blind co-designer (CD1) building an interactive version of a quaternary geologic map of Wisconsin [152]. The map combined dozens of irregular geographic regions organized categorically by a legend.

Early design sessions took place in Figma (Figure 6.2), where we laid out node-link diagrams of how a screen reader user might traverse the map, legend, and peripheral information. We annotated each node with text a screen reader would announce and connected them to relevant regions. Drawing the structure made it possible to discuss design choices, catch dead ends, and debate traversal strategies. We were informally doing what *Skeleton* later formalized.

The friction began when we moved from design to implementation. We had no way to verify that our designed structure would be good or ideal, and scaffolding the project into Data Navigator was arduous: the translation from even simple navigational designs to functional code was too complex to hand off or meaningfully iterate on. The collaboration also surfaced a design-space boundary: for within-map spatial navigation, an egocentric audio-game approach [13] fit better than a node-link graph, a distinction CD1 helped surface. Reaching this boundary was only possible because we could see the structure we were trying to build.

6.4.2 Design System Library

Our second engagement, spanning 7 months, was with 6 engineers and designers on Adobe's React Spectrum Charts library, an open-source chart component system. We worked in their codebase and in Miro on navigation design for bar charts, clustered and stacked bars, line charts, and related types. Because we needed generalized, reusable patterns, our design problems differed from the geologic map: we needed to account for use, re-use, and edge cases across chart types.

Our Miro sessions produced two kinds of artifacts: diagrams of navigation structure for specific chart *instances* and *schema* diagrams capturing generalized patterns in dimensional terms. The concept of a "dimension" emerged naturally: common transformations on Data Navigator's graph structure corresponded to properties within a dataset, such as a "categorical" dimension or a "numerical" dimension, where traversal took place within collections of grouped siblings. This dimensional thinking directly foreshadowed the Dimensions API (Section 6.4.4).

The dominant friction was iteration speed. We could sketch and converge on a schema in Miro in an hour, but verifying the design in a functional example required embedding Data Navigator into a large codebase, implementing changes, rebuilding, and manually testing with a screen reader. The collaboration also surfaced a limitation: mobile screen reader navigation uses swipe gestures rather than keyboard input, and our keyboard interaction model did not account for it. Together, these gaps made two requirements clear: a higher-level navigation abstraction and a visual tool for inspecting structures without a screen reader or a fully integrated build.

6.4.3 Open Source Visualization Library

Our third collaboration, spanning nearly two years, was with Quansight Labs and contributors to Bokeh, a Python-based open-source visualization library. We performed an accessibility audit [50] based on Chartability [46] and identified that Bokeh visualizations with interactive chart elements needed to be navigable by assistive technologies.

Unlike Adobe, Bokeh has no native chart types. Its API operates at the level of glyphs, renderers, and data sources, which users assemble freely. There was no standard unit around which to anchor a navigation pattern, and any grammar or tooling would need to accommodate an open-ended range of encoding combinations. The iteration gap from Adobe was present in a more severe form: making library-wide contributions to a fully open-source project required incremental tooling for the most common cases. For Bokeh, we needed to test functional, data-driven navigation abstractions without fully embedding Data Navigator into the library.

6.4.4 Infrastructure from Practice

The three collaborations converged on two concrete requirements: a way to visually render and inspect navigation structures, and a higher-level abstraction for specifying navigation without hand-wiring every node and edge. We built two pieces of infrastructure to address these requirements. Both were used in subsequent co-design work and became the foundation on which *Skeleton* was built.

6.4.4.1 An Inspector Gadget

We built an *Inspector* (`@data-navigator/inspector`) to render any Data Navigator structure as an interactive node-link graph using D3, with an accompanying console for debugging. Hierarchical structures are colored by level; edges are drawn as directed links; the entry point is visually marked. The *Inspector*'s graph can itself be navigated using Data Navigator, with visual focus tracking during navigation, allowing practitioners to manually verify structure and reachability.

This made structural verification immediate: a practitioner could generate a structure and check at a glance whether the hierarchy had the right levels, whether circular extents produced expected wrap-around edges, or whether a particular path was reachable. The interactive console logs API information and underlying data when nodes are activated, and hovering or focusing logged information highlights the corresponding node in the graph.

The *Inspector* remains a developer tool, however. It requires code familiarity to attach and renders structure as an abstract graph with no connection to the spatial layout of the underlying visualization. It shows topology but not instantiated geometry: a practitioner can see that two nodes are connected but not where their focus indicators will appear on-screen. This gap between navigable structure and its spatial instantiation over a rendered chart motivated *Skeleton*.

6.4.4.2 Alternative Dimensions

The original Data Navigator library requires explicit graph construction: practitioners specify nodes, edges, and navigation rules by hand. This is general but scales poorly.

The *Dimensions API* introduces a declarative abstraction one level above this. Rather than specifying the graph directly, a practitioner describes the *dimensions* of their data, the meaningful axes along which a user might want to navigate, and the API constructs the full node-link structure automatically. A dimension has a type (`categorical` or `numerical`), a data key, and behavioral properties governing traversal. Two properties are central: `extents` determines boundary behavior (`terminal` stops at edges; `circular` wraps around), and `childmostNavigation` determines whether leaf-level nodes are reachable laterally across a dimension's divisions without first returning to a parent.

The generated structure is a multi-level hierarchy: each dimension produces a root node, below which division nodes group the data, below which leaf nodes represent individual data points. Multiple dimensions over the same dataset share leaf nodes, so users navigating via different dimensions reach the same data through different paths. With the Dimensions API, bar chart navigation that would otherwise require constructing every node and edge by hand is expressed as a single dimension declaration:

```
dimensions: {
  values: [
    {
      dimensionKey: 'month',
      type: 'categorical',
      behavior: { extents: 'circular' }
    }
  ]
}
```

The abstraction is chart-type-agnostic: bar charts, scatter plots, line charts, and layered charts all use the same vocabulary, with different combinations producing different navigation topologies. This directly addressed Bokeh's problem: the API mirrors data fields and encoding choices as a set of dimensions.

6.5 *Skeleton*: System Design

With a visual structure renderer (the Inspector) and a declarative abstraction for producing navigation structures (the Dimensions API), the remaining problem was to bring these capabilities into an integrated, direct-manipulation authoring environment where practitioners could not only see structure but manipulate it interactively, test it with real input, and iterate on it with immediate feedback. *Skeleton* is that environment. It integrates the Inspector and Dimensions API into a unified workflow and extends both with a guided preparation phase, a spatial rendering canvas, and a live testing mode. Each of Skeleton's authoring techniques makes visible a specific property of non-visual interaction that sighted practitioners otherwise cannot see during authoring: the topology of what is navigable, the spatial mapping of where navigation lives over the chart, the semantics of what is announced, and the temporal sequence in which a user encounters nodes.

Where the Inspector requires a practitioner to write code first and visualize after, *Skeleton* reverses the direction: practitioners design a navigation structure visually and inspect the code representation as a consequence of that design.

6.5.1 Staging: Input and Preparation for Authoring

The authoring workflow proceeds through four stages: upload, prepare, edit, and test. Making a visualization accessible involves decisions about what is navigable, how navigation is triggered, and where focus indicators appear in space. These decisions are related but distinct, and collapsing them into a single undifferentiated interface, as code-only workflows effectively do, makes each one harder to reason about. The stage architecture surfaces them as separate concerns. It also preserves a direct correspondence between what practitioners see in the interface and the structure of the Data Navigator API [47]: each stage maps onto a distinct layer of the API, lowering the barrier to moving from visual authoring into code when production deployment requires it.

Upload. The upload phase is deliberately permissive. Practitioners can bring a dataset, a chart image, both, or neither. When a dataset is present, *Skeleton* parses its fields and infers dimension types automatically, producing a default, starting configuration for the *Prepare* step. When only an image is present, practitioners proceed directly to editing and construct nodes and edges manually over the image. This was motivated by our geologic map co-design work: many bespoke visualizations do not have a single underlying dataset, and any tool that requires structured data as a precondition for excludes the cases that need it most. *Skeleton* can be applied to any 2D image surface, not only to visualizations in the conventional sense.

Prepare. The prepare stage addresses a hard authoring bottleneck: not placing nodes, but deciding what structure to build at all. A practitioner who has never designed a navigation structure faces an open configuration space with no obvious entry point. The prep stage presents a four-chapter Q&A wizard that moves through authoring decisions sequentially: (1) whether the chart should have a root node and what it should announce, (2) which data fields should be navigable dimensions and how those dimensions should behave at their boundaries, (3) which keyboard interactions each dimension should be assigned to, and (4) what text labels each level of the hierarchy should produce. Each chapter is accompanied by an illustrative schematic diagram of which part of the heirarchy is being edited as well as a diagram showing examples of what these decisions look like when showing on a chart. The wizard's output populates a configuration in the editor that practitioners can then inspect, refine, and revise.

6.5.2 Edit: Interacting with Topology, Layout, and Semantics

Seeing the system, seeing the experience. The editor is *Skeleton*'s primary authoring environment (**??**) and presents two interlinked representations of the same navigation structure. A schema panel shows the structure as an abstract hierarchical tree layout that makes levels and parent-child relationships immediately readable. A graph canvas shows the same structure rendered as geometric elements positioned over the uploaded chart image, representing what an end user would encounter spatially. These two representations are bidirectionally linked: selecting a node in either view propagates the selection to the other, so practitioners can simultaneously hold in mind both the abstract topology of what is navigable and the spatial instantiation of where that navigation will live. The dual-view design makes visible a real conceptual divide between the system's model of navigation and a user's experience of it, one that practitioners recognize once they can see it, even without prior vocabulary for it.

Leveraging visualization as a scaffolding engine. Manually positioning leaf nodes over each data mark is the most mechanical step in the authoring workflow, and it scales poorly with dataset size. In our early pilot sessions, actually placing nodes in the canvas space was the slowest and most tedious part of the process. To address this, *Skeleton* includes a scaffold tool (Figure 6.3) that automates spatial placement by repurposing Vega [164] as a layout engine.

The scaffold renders a Vega chart specification to a hidden, off-screen container and extracts

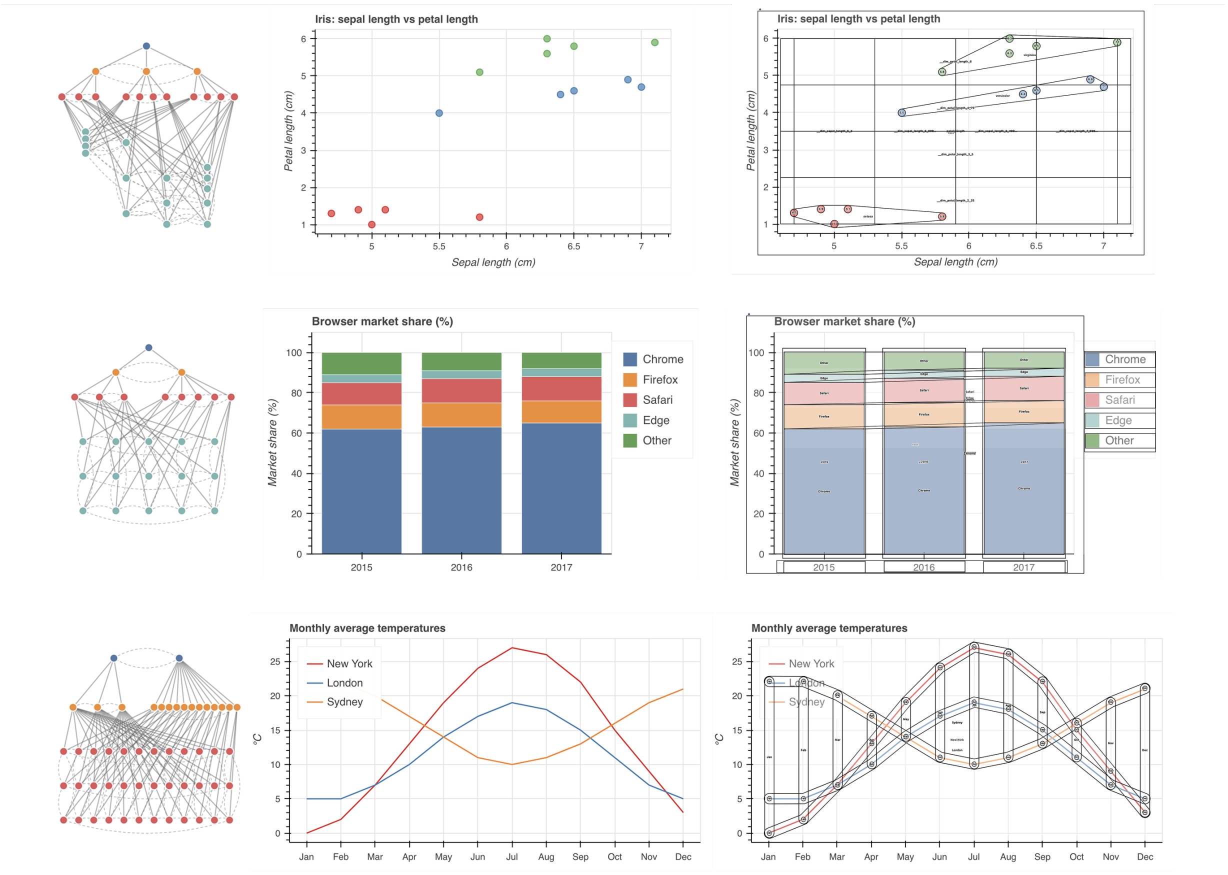


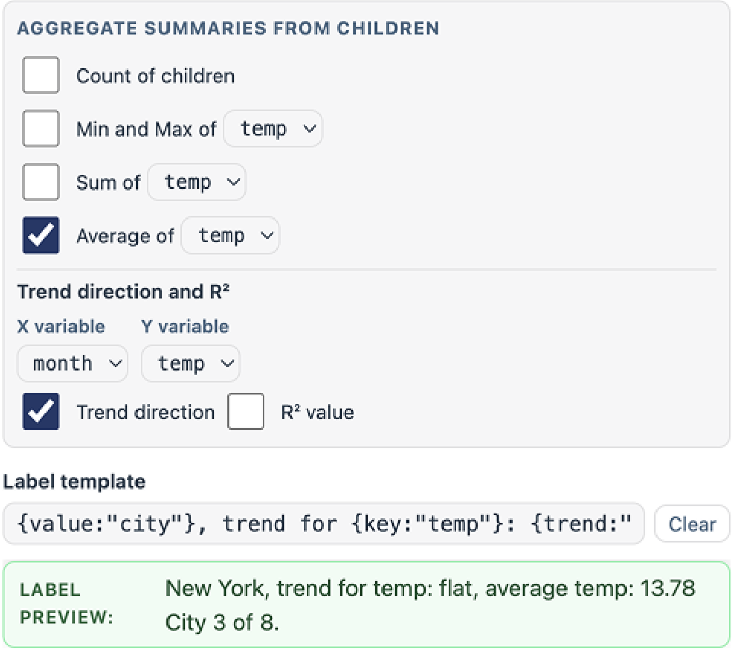
Figure 6.3: Input data transformed into a navigable structure using the *Dimensions API* and visualized with our *Inspector* gadget (left). The input chart (middle). The navigable structure is transformed and drawn over the chart using the *Scaffolding Engine* (right).

node positions via the library's internal scale and view APIs, using the rendering engine purely as a coordinate computation step: no actual chart is ever shown to the practitioner. Coincidentally, none of our co-designers were crafting visualizations using Vega or Vega-Lite (or derivatives), yet the Vega rendering engine could faithfully reconstruct every necessary mark position over the underlying, inaccessible data visualization provided to *Skeleton*.

Additionally, positions and outline strategies for category-level group nodes are computed from the leaf positions using geometric algorithms: a union of child node paths, a convex hull, a grid over numerical bounds, or a bounding rectangle. These designs were consistently produced as ideal treatments during co-design, so we chose them for our starting set of outline strategies.

The scaffold is optional and works by generating synthetic placeholder positions that practitioners can adjust manually. It dramatically improved authoring speed: in light pilot tests, a research team member completed a three-dimension structure without scaffolding in 8 minutes 22 seconds and with scaffolding in 56 seconds. A co-designer completed the task incorrectly (failing to account for one dimension's division nodes) in 13 minutes 7 seconds without scaffolding, and correctly in 2 minutes 44 seconds with it.

Specifying token patterns, editing instances. Selecting any node populates a properties panel with spatial properties (position, size, shape) and semantic properties (ARIA role, description, and a label template editor; Figure 6.4). The label template allows practitioners to assemble the text a screen reader will announce at that node from tokens drawn from data fields, including aggregate statistics at group-level nodes and precise data values at leaf nodes [100]. Editing labels can apply to all nodes of the same type or to a single instance.



AGGREGATE SUMMARIES FROM CHILDREN

☐ Count of children

☐ Min and Max of `temp`

☐ Sum of `temp`

☑ Average of `temp`

Trend direction and R^2

X variable: `month`

Y variable: `temp`

☑ Trend direction ☐ R^2 value

Label template

`{value:"city"}, trend for {key:"temp"}: {trend:"`

LABEL PREVIEW: New York, trend for temp: flat, average temp: 13.78 City 3 of 8.

Figure 6.4: Group label pattern builder, including an array of aggregate summary options, template formatter field, and preview.

At the bottom of the semantic section, a live preview displays the full assembled announcement string in the exact form a screen reader would produce: role, semantics, group membership, and label combined into a single rendered output that updates in real time. Prior to this preview, understanding what would be announced at a given node required running a screen reader and navigating to it sequentially. In deeply nested structures, this meant spending several seconds listening to labels and drilling in. This was consistently the main point where bugs were produced and missed during our co-design work.

The preview makes text announcements inspectable as visible, editable objects. This surfaces a class of highly specific, low-level problems that code-only workflows leave invisible: redundancies in announced text, missing contextual framing, and label ordering and punctuation that affects comprehension and reading speed. Of all the authoring decisions *Skeleton* exposes, label templates involve the most degrees of freedom and have the most direct bearing on the quality of the non-visual experience. Getting the structure right ensures navigability; getting the labels right determines whether navigation communicates anything meaningful.

6.5.3 Test: Debugging Interaction Interactively

The testing stage allows practitioners to navigate the structure they have built using the same keyboard input and navigation rules that assistive technologies would use, without leaving the tool. When a practitioner enters testing, the three Data Navigator modules are instantiated in sequence: the structure module rebuilds the navigation graph from the current configuration, the rendering module creates an HTML layer positioned over the chart image at each node's spatial coordinates, and the input module registers keyboard listeners for all navigation rules. The result is a live, keyboard-navigable structure. An event log records navigation events in order, letting practitioners verify that all nodes are reachable and that label sequences make sense when encountered serially rather than read simultaneously.

The abstract graph continues to display during testing. As the practitioner navigates, the focused node is highlighted simultaneously in the canvas (showing its spatial position over the chart image) and in the abstract graph (showing its structural position in the hierarchy). This parallel tracking makes the temporal traversal sequence visible, showing the order and path through which a user encounters nodes, so practitioners can verify at a glance both where focus is and what role it occupies. A text-chat mode is also available, in which practitioners navigate by typing natural language commands, motivated as a design by the Adobe collaboration (Section 6.4.2), to explore interaction alternatives for mobile screen reader users.

The testing stage also served, during development, as the primary debugging interface for *Skeleton*'s own data pipeline. Errors in position computation, label resolution, or structure generation that would propagate silently through code became visible the moment a node highlight appeared in the wrong location or a label read as undefined. A tool for making non-visual structure visible turned out to benefit from the same property during its own construction.

Practice-based validation. CD2, a subject matter expert who professionally evaluates interfaces for screen reader access and is familiar with other visualization navigation systems, used *Skeleton*'s testing stage to evaluate navigation output across several chart types and dimension configurations: line charts (3 configurations), bar charts (2), stacked bar charts (3), and scatterplots (4). This evaluation followed a manual, systematic approach combining standards-based criteria with expert screen reader testing. Scatterplots required the most iteration, surfacing bugs in Data Navigator's core library that were then fixed. CD2 also recommended that we rely on list-based navigation while in the editor (before the testing stage) in case users build themselves into a keyboard trap.

6.6 User Study

Our co-design work followed an action research orientation: the research team was embedded in practitioner communities, and our system work was motivated by the problems those communities faced. This process generated the techniques and infrastructure that comprise *Skeleton*, but we still needed to understand the impact our system had on visualization practitioners more broadly. Our co-designers were deeply familiar with the problem space, having spent months or years working on accessible navigation. We needed to understand what happens when practitioners who are *not* embedded in this process design, author, and debug navigation structure visually: whether the representations we built are legible to them, whether the techniques change how they reason about accessible design, and what new questions or problems emerge when navigation structure becomes visible. These are empirical questions that required a study.

To evaluate how *Skeleton* influences the way practitioners engage with accessible design, we conducted an in-situ interview study with 8 participants across visualization design, engineering, and research. The study was conducted remotely over video call, took approximately 45–60 minutes per session, and was approved by our IRB. Participants provided verbal consent at the start of each session. Video and audio were not recorded, however some participants consented to share the data/image they brought to the study as well as screenshots of their workflow; data collection was note-based throughout.

6.6.1 Participants

Participants were recruited through snowball sampling within the visualization and accessibility community, and through referrals from co-designers involved in our earlier collaborations. We asked each participant to self-report their primary work role (engineering, research, design, or student) and their existing level of accessibility expertise on a 1–5 Likert scale. We also asked whether the visualizations they build are ever bespoke, that is, custom rather than instances of a recognizable, standard chart type. This distinction mattered because bespoke visualizations represent an especially underserved case in accessible design tooling: no library pattern applies, and every navigation structure must be designed from scratch. Participants were not compensated.

6.6.2 Procedure

Each 45-minute session proceeded in four phases. Before the session, all participants were asked to prepare a chart image they were currently working on or had recently built, for use in the third phase.

Phase 1: Introduction and demographics (5 minutes). After obtaining verbal consent and recording a pseudonym, we collected self-reported role, accessibility expertise level, and whether the participant regularly builds bespoke visualizations.

Phase 2: Generic chart think-aloud [2] (10 minutes). Participants used *Skeleton* on a provided bar chart and dataset of fruit counts (Apples, Pears, Nectarines, Plums, Grapes), asked to design an accessible navigation experience for a screen reader user with no instructions on how the tool worked. The tool loaded with a default structure having both a categorical dimension (`fruit`) and a numerical dimension (`count`) active. This default was intentionally problematic for two reasons: numerical navigation sorts by count value and groups data into subdivided ranges, producing a traversal order different than the visual layout an additional, largely unhelpful level in the hierarchy for such a simple chart. We observed how participants reasoned about what they saw and whether and how they noticed this extra dimension.

Phase 3: Own-chart think-aloud (15 minutes). Participants loaded their own chart image and attempted the same task, except they were also asked to explain their graphic to the research team (purpose, role, data, and domain). This phase was open-ended: charts ranged from standard types to bespoke visualizations, and the goal was to observe how participants reasoned about navigation structure when the context was their own work.

Phase 4: Reflective interview (15 minutes). We conducted a semi-structured interview in which participants reflected on their decisions in Phase 2 versus Phase 3, their experience with the generic versus their own chart, and their assessment of the tool's capabilities and limitations. We asked what felt possible or impossible, what they wanted to do that they could not, and what they found themselves thinking about that they had not considered in Phase 2.

6.6.3 Analysis

Notes from each session were compiled into a shared document. Participant quotes reported in the results are reconstructed from these researcher notes, not verbatim transcripts. We analyzed the data using a combination of thematic analysis [19] and affinity diagramming [73], iterating across both methods to surface recurring patterns while preserving the specificity of individual participant experiences. Analysis attended particularly to differences in how participants engaged with accessible design before and after using the tool, the range of input modalities and user scenarios they considered, and moments when participants reconsidered or wished to redesign their own visualizations.

6.7 Results

We organize our findings into five themes that emerged from thematic analysis and affinity diagramming across all eight sessions. Each theme captures a qualitative pattern in how practitioners engaged with accessible navigation design when its structure was made visible and manipulable. We report these findings descriptively and ground them in specific participant moments; interpretation follows in the Discussion.

6.7.1 Seeing Navigation Made Structural Problems Legible as Design Problems

The generic bar chart in Phase 2 loaded with an intentionally problematic default: both a categorical dimension (`fruit`) and a numerical dimension (`count`) were active, producing overlapping navigation structures with different traversal orders over the same data. This configuration is a poor design choice, arguably a design failure, but one that would be difficult to detect in code alone (**R1**, **R4**).

Participants varied widely in how quickly they recognized the problem. P1 turned off the numerical dimension within seconds of seeing the editor, without commenting on it. Most participants, however, initially struggled to understand what they were seeing, remarking on the unfamiliar structure: "what is this? what are these?" when encountering the numerical divisions for the first time. P8 spent time trying to guess what the extra divisions represented but did not remove them during Phase 2, only realizing during Phase 3 that the additional dimension was "probably bad." P2 and P5 expressed suspicion early: P5 asked, "Is this too much data? This seems like way too much to just navigate through," and P2 noted, "I feel like a lot of data points would be bad, yeah? Like, too many at once is bad?" Both of these remarks were prompted by the visual density of the structure, not by navigating it.

The testing stage (Section 6.5.3) proved critical for resolution. P4, P5, and P7 each removed the extra dimension only after navigating the structure with keyboard input in testing mode, where the traversal sequence made the redundancy experientially apparent. In total, five of eight participants resolved the problem during the session: P1 and P2 during editing, and P4, P5, and P7 after testing.

Beyond the intentional default, participants identified other problems through visual inspection. P8 reacted to a generated node name: "Okay dim_fruit node…that is horrible, what is that?" During Phase 3, P7 looked at the edges of their multi-line chart and asked, "are all these bad? Is it bad that I don't even really know what the takeaway of this [structure] is?" P3, seeing a full hierarchy for their own simple six-item bar chart (during Phase 3), concluded "I should just skip the root and grouping and go straight to the data. This seems like too many steps." In each case, the visual representation of their navigation structure motivated judgment about potential negative design qualities.

6.7.2 Practitioners Developed a Designerly Interest in What Constitutes Good Navigation

The most pervasive pattern across sessions was that participants began asking design questions about navigation quality, unprompted by any instruction or guidance from the research team (**R2**). These questions went beyond identifying errors: participants wanted to know what *good* navigation would be for their charts.

P2, working through the bar chart in Phase 2, deliberated over boundary behavior: "Loop back or stop? I don't think there is a right way. I will just pick *fruit* for now and *loop* and see what this does." P6, who brought a bespoke flower visualization, wondered how to translate the affective quality of their chart: "I think my visualization should be more about the vibes, but I don't know how to make the alt text have good vibes. What is *fun*?" P4 asked fundamental questions about the interaction model itself: "Why do screen readers and keyboards have to work this way? Do people like that?…why do we navigate?" And later after testing, P4 concluded, "I bet we should make this *faster*" before cutting out the additional numerical dimension in Phase 2.

Several participants engaged with the concept of narrative and flow. P8 articulated this as a question about the goal of the visualization: "sometimes I want a big picture, not precision. I may want to drill down a little…" and observed that "we think too much in terms of components…sometimes accuracy isn't the actual goal, it's getting a general sense of something." P4, upon discovering that *Skeleton* supports text-based input, asked, "How do you make that good, though? Like chatGPT, or do people want to, like, interact with the chart [elements]?'

During the interview, several participants explicitly requested guidance. P1, P2, P3, and P6 wanted to see examples of well-designed navigation experiences. P1, P3, P4, P5, P6, and P7 wanted embedded guidelines within the tool. P2 and P4 actually used web search to look for "chart navigation for accessibility guidelines" (P2) and "accessible viz screen reader design" (P4). P3 was interested in automation and heuristics that could suggest reasonable defaults. These requests are consistent with the pattern (**R2**): practitioners who could see the design space wanted orientation within it.

6.7.3 Iteration Was Substantive, Self-directed, and Concentrated on Semantics

Every participant iterated on their navigation designs, and this iteration was neither perfunctory nor prompted by the research team (**R4**). Participants revisited decisions, revised configurations, and in some cases restructured their entire approach after encountering their design in a new stage of the tool.

The most sustained iteration concerned labels and text announcements. Every participant spent the largest share of their authoring time in the label template editor (Figure 6.4), editing the text that a screen reader would announce at each node. This editing was granular: participants wrote full sentences, rearranged the order of data tokens, debated whether to include field keys alongside values or values alone, experimented with how to name groups and individual elements, and considered the length and density of the resulting announcements. At the division and dimension levels, some participants added multiple aggregate statistics (count, sum, average, range, trend) and then returned to trim them. P2, for instance, edited data point labels, left them, and returned to revise them two additional times: “This label is way too complicated, I think.” These label iterations were small, fast, and frequent, and they reflected an intuitive grasp of the importance of the textual tokens that constitute a screen reader user’s primary interface to data.

A second, distinct pattern of iteration emerged around testing. The editor displays all navigation nodes simultaneously, showing the full structure at a glance. The testing stage (Section 6.5.3), by contrast, shows only the currently focused node, highlighting it in the canvas one at a time as the practitioner navigates. This difference consistently prompted participants to revise. Several returned to the editor after testing to adjust group node outlines, because outline strategies that looked distinct when displayed simultaneously (such as bounding rectangles vs. convex hulls) became harder to differentiate when encountered one at a time. Others adjusted their dimension configurations: P3 and P5 restructured their dimensions after testing, and P4, P6, and P8 experimented with different key bindings and navigation rules. P2 used the testing stage specifically to identify labels that needed revision at the dimension and division levels. In these cases, testing was not only treated as a bug-finding activity but also as a way to encounter, reason about, and then improve a design that had looked adequate in the editor but felt inadequate in sequential traversal.

The most extensive iteration came from P1, who returned to the preparation wizard after reaching the testing stage and re-took the entire wizard from scratch. Seeing the full structure in testing motivated them to re-evaluate their earliest decisions. They reduced their navigation from three dimensions to one, producing a navigation experience they described as deliberately minimal. In the reflective interview, P1 explained that they had been thinking about trimming excess, joking that they were “working on a data-to-word ratio.”

The scaffolding tool shaped the pace and character of these iterations. Participants who used the scaffold (Figure 6.3) generally required only minor manual adjustment of node positions, spending one to two minutes refining placements in-aggregate before moving on. Iteration on spatial layout was primarily about the appearance of group-level outlines rather than individual node positions: because the scaffold computed leaf positions from the chart’s encoding, the main remaining spatial decision was how division and dimension nodes should visually indicate their grouping. When participants returned to scaffolding, it was most often because they wanted to change the chart type used for coordinate computation or to try a different group outline strategy.

6.7.4 Seeing Navigation Prompted Practitioners to Reconsider the Architecture of Their Own Charts

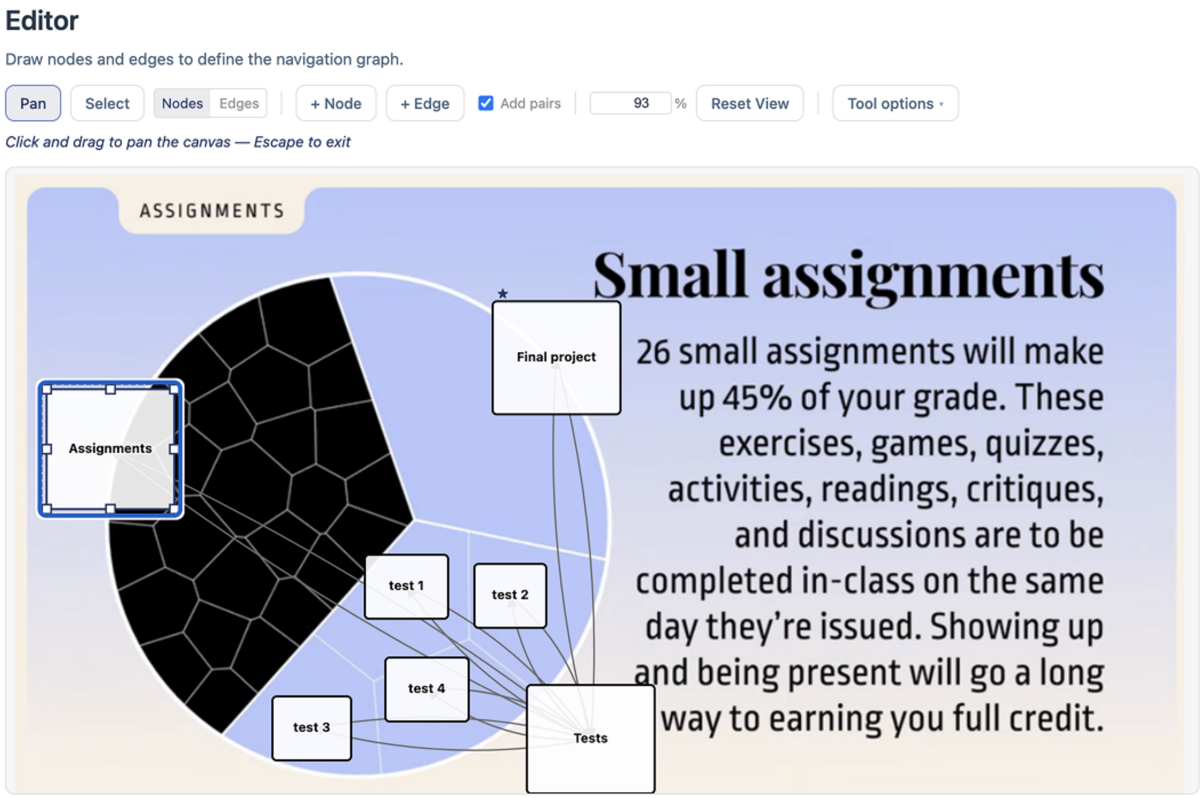
An unexpected finding was that working with *Skeleton* prompted several participants to question the design of the visualization they had brought, not just the navigation structure they were building over it (**R4**). Five participants (P4, P5, P6, P7, P8) expressed a desire to simplify their own charts after seeing what the corresponding navigation structure required. Three (P1, P5, P6) considered whether a different chart type might serve the same communicative goal with a simpler navigational architecture. And in 2 cases (P1, P6), participants questioned whether a chart was the right medium at all.

P5, who brought a scatter plot with over two thousand data points, initially tried to build a navigation structure over the full dataset, recognized that the result was unmanageable, and then asked: "do I even need visualization?" They went on to wonder whether the information could be communicated as "a few sentences or like, some data [users] can prompt" (referring to using a large-language model). P6, who brought a bespoke flower visualization in which each petal encoded a different variable, initially tried to express the chart's full complexity in navigation (grouping by flower and then navigating by petal) but then reversed course: "okay, what if we actually just treat this like a bar chart?" Using the scaffold tool, they produced a simple list-style navigation that worked well for their data, and reflected: "my visualization has a lot, but actually, this could be pretty easy to navigate, I think." P6 also wondered whether there was a non-chart way to "tell their story," and in further discussion described something closer to a scrollytelling article or guided walkthrough than a single interactive chart.

P8's case was the most striking. They brought a voronoi pie chart (Figure 6.5) used to communicate to students that 26 assignments make up 45% of their grade, the largest slice of the pie. Working through *Skeleton*, P8 first considered making all 26 voronoi cells navigable, then considered making only the three main slices of the pie navigable, and then questioned whether navigation was needed at all. The "point of this," as they described it, was to communicate a single ratio; students did not need to traverse individual cells to understand it. P8 concluded that well-written alternative text was sufficient for the screen reader experience and chose not to build a navigation structure. They also reflected on the design of the visualization itself, concluding that the voronoi treatment served a visual purpose (it is visually striking) that was separable from the informational purpose (communicating a grade breakdown). They kept the visual design and simplified the non-visual experience accordingly.

6.7.5 Experiencing Keyboard Navigation Surfaced a Broader Range of Users and Input Technologies

The testing stage (Section 6.5.3) provided what was, for most participants, their first experience of keyboard navigation through a data visualization. Only three participants (P1, P2, P8) reported prior experience using a screen reader, and only two (P1, P8) had previously navigated a visualization using keyboard input alone. Yet during the study, every participant navigated using



Draw nodes and edges to define the navigation graph.

Click and drag to pan the canvas — Escape to exit

Small assignments

26 small assignments will make up 45% of your grade. These exercises, games, quizzes, activities, readings, critiques, and discussions are to be completed in-class on the same day they're issued. Showing up and being present will go a long way to earning you full credit.

Figure 6.5: Re-creation of P8's moment of realization, placing nodes manually: not every element in their voronoi pie chart *should* be navigable.

a keyboard.

Several participants responded to this experience with genuine engagement. P4 reacted with enthusiasm: "Woah, this is incredible. Wait, what other visualizations can do this?" and later, "Maybe you could make a visualization game, where you can move around the data?" P7 said, "I love this. I want to make charts with this." P5, who had not previously encountered keyboard-navigable charts, said, "I didn't know you could do this." These reactions were not about the tool; they were about the interaction modality itself, about the experience of traversing data spatially using directional input.

The experience also expanded participants' sense of who these structures serve (**R2**, **R4**). P8 adjusted a node's spatial dimensions and said, "Let's make the hitbox as big as possible here. . . for my neighbor with Parkinson's to click these," treating the navigation structure as relevant to motor accessibility, not only screen reader access. P3, observing the visual focus indicators that appeared during scaffold-based placement, remarked that "this is for more people than [someone] blind," recognizing that sighted keyboard users and users with partial vision also rely on visible focus states. During the interview, P4 asked sustained questions about what kinds of technologies different people with disabilities use, and P4 and P7 both asked why focus indication was designed to be visible rather than invisible.

P4's curiosity extended to alternative input modalities. Upon learning that *Skeleton* supports text-based navigation and (due to leveraging Data Navigator) also supports a wide array of other input modalities, they asked, “Can you talk at it, too? Is that what some people do?” and followed up with, “How cool is that? How do you make that good, though?” P8 raised a concern about discoverability in the current interaction model: “I’ve never used j or w to drill out, always shift arrow or option arrow…” and worried that a user might not know how to exit a nested level. These observations reflect an broad mental model of the user, from an abstract “screen reader user” to a person with multiple capabilities, preferences, and interaction patterns (**R2**).

6.8 Discussion

6.8.1 Visibility as a Precondition for Iteration

The central finding of this work is not that *Skeleton* taught sighted practitioners how to design accessible navigation, but that it gave them something to react to. When navigation structure was invisible, our co-designers would only seek to verify whether navigation followed our design documentation. Once visible, questions shifted to whether it was good, why, how it could be improved, and what else it could do. Visibility encouraged iteration, and iteration is enabled continuous design improvements.

This mechanism is consistent with Schön’s account of reflective practice [171]: designers iterate by externalizing a representation, perceiving its properties, and responding to what they see. The representation talks back, and the designer adjusts. But this loop requires a representation. In our co-design work, we assumed that the gap was hand-off between design and development, and the slow process of manual verification. Instead, the gap is that development was not participating in, and encouraging, further design reflection. In a developer’s code-only workflow, navigation structure has no externalization that supports this kind of perceptual engagement. A practitioner can read the code that specifies a navigation graph, but they cannot see the graph, cannot perceive its topology at a glance, cannot notice that a label is redundant or that a hierarchy is too deep by looking at it. The reflective loop is disrupted and occluded at the first step.

Skeleton encourages this loop by rendering navigation structure in a form that supports the same kind of perceptual judgment that visualization practitioners already apply to every other aspect of their work. The results show what happened when this loop was available: every participant iterated, substantively and self-directedly (Section 6.7.3). They revised labels repeatedly, restructured dimensions after testing, adjusted group outlines when sequential traversal revealed problems that simultaneous display had hidden. P1 restarted the entire preparation wizard after seeing their structure in testing mode. These are not the behaviors of practitioners following a specification; they are the behaviors of practitioners negotiating with a design material.

The co-design work reported in Section 6.4 provides complementary evidence: in each collaboration, sighted practitioners naturally reached for visual representations when reasoning about navigation, through node-link diagrams in Figma, schema sketches in Miro, and annotated wireframes on paper. They were already thinking visually about non-visual structure; the development tooling simply had not caught up. This has a practical implication for the broader field: if sighted authors depend on visibility, then any authoring workflow that keeps navigation structure invisible limits design iteration. Making structure visible does not guarantee good design, but it is a precondition for the kind of sustained, judgment-driven refinement that good design requires.

6.8.2 From Compliance to Design

A consistent pattern across the accessibility literature is that practitioners frame accessibility as a compliance problem: a set of requirements to satisfy, a checklist to complete, a legal or institutional obligation to meet [46, 102, 177]. The framing matters because compliance and design orient practitioners toward fundamentally different activities. Compliance asks: "does this pass?" Design asks: "is this good?" Our results suggest that *Skeleton* produced a partial shift from the first orientation to the second. Participants' initial instincts clustered around compliance-oriented responses: provide alternative text, follow guidelines, ask an expert. But alongside that desire for guidance, they began doing something compliance framing does not typically produce: they encountered complexity and then *iterated*. They revised labels repeatedly, restructured dimensions after testing, debated boundary behavior and hierarchical depth. This sustained, self-directed refinement is the behavioral signature of design, not compliance. As P4 put it: "I'd love to have someone blind actually just with me while I make this, but I also understand that I should learn what makes a good experience too."

The shift extended beyond the non-visual experience itself. As reported in Section 6.7.4, five participants reconsidered the design of the visualization they had brought, not just the navigation structure overlaid on it. Making the accessibility consequences of visual design choices visible prompted practitioners to question whether a different chart type, a simpler encoding, or a non-chart medium might better serve their communicative goals. This interrelation between non-visual and visual design suggests that the widespread treatment of accessibility as a compliance activity may be partly a consequence of tooling that offers no legible, manipulable design surface. Auditing frameworks are valuable, but they are *evaluative* tools, not *authoring* tools. The field needs both.

6.8.3 Bespoke Visualizations as an Unaddressed Accessibility Research Problem

Diagrams, infographics, and data-driven illustrations are often one-off, custom designs with bespoke symbols, layouts, and visual languages. These representations are increasingly common in journalism, scientific communication, personal projects, art, and public-facing data work, and they represent the cases where accessibility-focused tools are needed most and available least. *Skeleton*'s image-based workflow (upload any 2D image, place nodes manually) provides a starting point, but the study made clear that bespoke visualizations need more than node placement. They need support for reasoning about what navigational structure (if any) is appropriate when no template applies, a research problem that remains largely unexplored.

6.8.4 What Visualization Owes Accessibility

Our approach, to make visual non-visual experiences, should not be limited to data visualization's own accessibility challenges. Navigation structure is a foundational component of accessible experience across domains: PDF and document reading order, web page structures, and software application layouts. In each of these areas, sighted practitioners author non-visual experiences without visual feedback, and in each, the same gap between design intent and verifiable outcome constrains quality. Testing is slow, error prone, and requires expertise in assistive technology use. Visual tooling for authoring, inspecting, and debugging non-visual structure (**R3**) is a tractable and high-value problem across application domains.

There is also a deeper question about what visibility can accomplish in principle. Work on multi-modal authoring environments has argued for de-centering visual representation and treating modalities as equal partners in the design process [223], an important ethical commitment. *Skeleton* does not do this: it re-centers visual representation as the medium through which sighted practitioners engage with non-visual structure. We believe this is justified pragmatically, because sighted authors need to articulate navigation design in their own perceptual language before they can reason about it at all, and this paper provides evidence that they do. But articulating a design in one's own language is not the same as understanding how it will be experienced in someone else's. The structures participants built during our study were never evaluated by blind users, and visibility alone cannot substitute for that evaluation. The risk we want to name is that making non-visual structure visible to sighted practitioners could be mistaken for making it *understood*, when in fact it makes it *designable*, a real but bounded gain. The fuller design process requires collaboration with disabled users, not as an occasional supplement but as a regular practice. *Skeleton* can make that collaboration more productive by giving both parties a shared representation or space of translation between representations, but it cannot replace it.

What visualization owes accessibility, then, is not simply better output but authoring tools that better stimulate reasoning, both individually and collaborative, about design.

6.9 Limitations and Future Work

Skeleton makes navigation structure visible to sighted authors, but it cannot reassure those authors whether the structure they have built is good for the people who will use it. The tool surfaces design questions; it does not answer them. Several participants asked what constitutes good navigation, and *Skeleton* had nothing authoritative to offer. Our study with sighted practitioners evaluated whether making navigation structure visible stimulated design consideration, not whether the designs sighted practitioners produced were actually good. These are related but distinct questions, and the second remains open. CD2's expert screen reader evaluation of *Skeleton*'s navigation output (Section 6.5.3) provided practice-based validation for several common chart types and surfaced concrete bugs, but this evaluation was neither comprehensive nor controlled: many configurations remain untested, and expert review is not a substitute for evaluation with a broader population of end users.

Additionally, we treated *Skeleton* as a design probe [71, 93] rather than comparing it to a controlled condition: the goal was to elicit qualitative insight about how the tool elicits engagement, not to measure performance differences.

Our approach using *Skeleton* as a design probe only with sighted participants is both a limitation and, we believe, the right sequencing: *Skeleton* improves the intentionality and iterability of what sighted practitioners produce, which is a necessary precondition for a subsequent evaluation or future collaborative design work between sighted and blind authors. Further research and evaluation should close this loop, ideally to engage how mixed-ability teams co-design multi-modal data experiences.

Skeleton is also a prototype with substantial work remaining. Not within the scope of the paper, but our participants provided ample feedback on the functionality of the prototype itself. The most urgent gap is export functionality: practitioners can design and inspect navigation structures in the tool but cannot yet produce deployable output.

6.10 Conclusion

Accessible navigation structure has long occupied an awkward position in visualization practice: known to matter, difficult to design, and invisible to the people responsible for building it. The invisibility was not incidental. Without a way to see what they were making, sighted practitioners could not catch errors, could not iterate, and could not develop the kind of considered judgment that good design requires. Accessibility remained downstream of every other decision not because of any single failure, but because the authoring conditions did not support anything else.

Skeleton demonstrates that those conditions can be changed. Making navigation structure visible and manipulable, as an interactive graph rendered over the spatial layout of a real visualization with live label previews and testable traversal, shifted how practitioners engaged with and reasoned about accessible design. They began asking qualitatively different questions than someone seeking compliance: whether the features and design of their structure was good, despite not having readily available answers.

If any conclusive take-aways can be gleaned from this project: for researchers interested in engaging accessibility, this would mean future projects might explore translational spaces between visuals and non-visuals that help sighted partners engage blind designers. For practitioners who build, design, or audit this would mean that more visualization is needed in current tooling, to enhance and make multi-modal existing non-visual methods of authoring and evaluation.

What the paper leaves open is more than what it closes. We used *Skeleton* as a design probe with sighted practitioners; we did not evaluate the navigation structures they produced with the end users those structures are meant to serve. And the broader disciplinary conversation, about what visualization research owes accessibility and what methods might transfer between them, has more questions than answers. We offer *Skeleton* not as a solution to these problems but as evidence that engaging them directly, with the full weight of visualization's methodological tradition, is both possible and fruitful.

Part VI

Conclusion

Chapter 9
Discussion & Future Work

9.1 What is a "tool?" A reflection on the social and material identity of tools

In the introduction of this dissertation, I use the example of a hammer: a hammer can destroy and it can construct. So is the *use* of a technology what constitutes it? Do we understand the hammer as the *thing we swing, to destroy and to build*? Should we?

This thesis engages domains of tools and tool-making for accessibility: evaluation, navigation, interaction, and personalization. But these categories for work do not fully characterize the upstream conditions that our software systems and data interfaces inherit.

In my work specifically on accessibility, a larger social reality becomes apparent that shapes the question, "what is a tool?" far more than how an individual might use one, or the domains of work that our tool-making inhabits. My research journey has navigated multiple social and political thresholds, from changes to law in the European Union, to the enactment of Title II as part of the update to the Americans with Disabilities Act. These laws have motivated a significant interest in accessibility research, solutions, guidelines, and technologies. In the midst of this, we have seen the rise of overlays and generative AI solutionism [74] and subsequent lawsuits and grass-roots resistance.

For my work, this is mostly good news. Legal change produces motivation, and even with predatory technology attempting to address real problems, pushback is widespread and active. But this paints a picture of the reality that my work inherits: many tools cannot even be used, or cease to be used, if there is not a social, political, and material set of conditions in place motivating those tools, providing resources for their construction, regulating their use, and examining the outcomes of what they accomplish. Tools and technologies are often a response to social, cultural, political, and legal realities that we first negotiate.

I recently spoke on this at a keynote in Australia, on how a hammer isn't *just* a tool and that the idea that "the only thing that matters is how a tool is used" limits how we really understand tools. Instead, I spoke about how a standard, household hammer requires iron and wood. That alone leads to a whole universe of different questions. Western Australia's conservation efforts were disrupted when a significant amount of natural iron was discovered in a wildlife preserve. So laws were passed and now iron is mined there. That iron is largely exported. And Australia then, whether with Australian iron or not, mostly imports their small tools. Iron is sent out, and through a complex network of trade (likely indirectly related to the iron), hammers are brought in. A "hammer," to even exist at all, relies on multiple levels of human governance, international relations, and complex infrastructures of trade.

And while my metaphor is largely motivated to encourage younger practitioners to consider the "iron mines" in the technologies they use, such as modern generative AI, it is also an area that is not adequately explored and addressed in terms of accessibility research.

Research on accessibility is dependent on funding, which is often dependent on political priorities and action. Depending on the current social and political state of the world at large, accessibility research itself may never gain the opportunities required in order to innovate and produce new tools at all. And as the US's 2025 federal cuts to research demonstrated, millions of dollars devoted to accessibility research can be lost to political agendas. It is for this reason that engagement with policy recommendation and guidance is essential. Personal political activity and involvement is also essential. Researchers who genuinely believe in accessibility as a human right or as a dignity that all people deserve should work with policymakers to ensure that there are material and structural resources in place for this work to continue. We cannot naively believe that technology, divorced from the realm of social and political forces, is capable of solving accessibility barriers [184]. Without enforcement and threat of litigation, very little accessibility work has been accomplished in the past by technology companies alone.

Not featured in these chapters (as they were merely stapled in research papers from previous publications) is the policy and outreach work involved in seeing that work like *Chartability* and *Data Navigator* are used in real contexts, including by organizations that govern and influence the lives of many people. Immediate incentives to produce novelty may not be enough to sustain the larger socio-cultural and political ecosystems that our work participates in and is downstream from. We must also get involved.

9.3 Who is responsible for repair?

Lastly, I want to revisit one of my opening points, where I argue that the *tool-makers* are first responsible for repair. This is true. However, the most pressing issue I have faced in recent years is mostly unmentioned across these research projects: tool-makers might be responsible, but this is because they are the only ones who have the *power* to make things accessible. Does this always need to be the case? Can we imagine an artifact's authority over the user's ability to access being designed towards self-subversion [64] or de-centralized agency [25, 105, 136], instead? What might that look like, concretely?

In *Softerware*, we begin to engage this larger problem in terms of an idealized state where a user can repair or re-design their own experiences. But to me, this self-repair is like laying down train tracks for yourself as you move a locomotive, but then lifting up your own tracks behind you as you go. You're the only one helping yourself. This is not ideal, for you or others.

What we need are broad, lasting, infrastructural changes. On the web, this problem becomes quite difficult to solve. A personal computer or device? Again, someone can auto-design their interfaces into a better state. But back when I started *Chartability*, the WebAim Million's report showed more than 95% of the top one million website home pages contain at least one critical accessibility error. And now, more than 6 years later, that proportion is unchanged [202].

Some had imagined that generative AI would solve the massive infrastructural repair problems we face. But unfortunately, the latest WebAim Million report shows that since 2020, ARIA usage has increased and correlates to more errors, while use of tabindex on a page has increased nearly 300% and also correlates to more errors on a page. If anything, during the age of generative AI, we have seen existing bad patterns worsen in prevalence and complexity.

I firmly believe that a tools-based approach is not enough on its own. Tool-making cannot be the *only* intervention on inaccessibility. Tools and tool-making, as our thesis argues, have a powerful role to play. But we simply can't tool our way out of failed infrastructure and inadequate policy when someone else *owns* the tools and tool-making. Visiting a website is like going into someone else's home: arranged according to their effort, tastes, and so on. If you can't access their home, you essentially need to request that they let you in personally. Website repair always falls to the owner and maintainer of a website, and they largely don't take any meaningful action.

Sidewalks outside of homes are a good parallel to this problem. Sidewalk accessibility is a massive infrastructural problem [158], and yet localities treat sidewalk maintenance in different ways: some, like where I presently live in the south hills of Pittsburgh, put the onus on the homeowner whose house and property the sidewalk touches. In other places, sidewalks are considered a public path, similar to a roadway, and are maintained through public tax and resource management. To no surprise, privately-maintained, public-access sidewalks are worse for people in pretty much every way than publicly-maintained ones [213]. This is because private homeowners don't care about sidewalk maintenance unless the city manages to fine them or they get sued.

And the web is a collection of private spaces that you visit privately. There is no truly shared, universally democratic, public space on the web. Centralization is partly to blame: sharing space while scaling leads to consolidation.

So my future work will continue to wrestle with the same tensions of scale, repair, and anti-consolidation of power, motivated by the same WebAim Million report. But now I look to questions of *democratic* and *radical* access to accessibility repair. The barriers I hope to tackle in the future are political and infrastructural. Perhaps tool-making will participate in this work, but it seems clear now from my work that the upstream technical problems and socio-political conditions that tools inherit, will likely not be addressed by tools alone.

What does "democratic" and "radical" infrastructure work look like? It will probably be an extension of *Softerware*, to some degree. I imagine future research that explores public-first spaces, ones where access is socially negotiated and repair belongs to all of us. Is this an autonomous space, like an autonomous zone [12] separate from the web? Above it, looking down into it, like shared annotation tools but capable of sharing the manipulation of websites [157]? A space with ambient co-repair, modeled after projects that bring people together [173] or that allow community "fixing" of misinformation [96]? Perhaps feminist thought on the ethics of care can help us [78, 121]? Or maybe it will be something else entirely; I'm not yet sure. But what made the web fantastic years ago is long gone; most of it has been hedged into corporate spaces that are controlled, maintained, and repaired by corporate power. And these entities are notoriously bad at repair. What I imagine in the future involves reclaiming a sense that the web is *ours*, belongs to *us*, and that ultimately *we* are responsible for making it accessible.

References

[1] 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. doi: 10.1007/978-3-540-87730-1_31. 4.3.1

[2] Obead Alhadreti and Pam Mayhew. Rethinking thinking aloud. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pages 1–12. ACM, April 2018. doi: 10.1145/3173574.3173618. URL `https://doi.org/10.1145/3173574.3173618`. 6.6.2

[3] Ofunne Amaka and Amber Thomas. The Naked Truth. https://pudding.cool/2021/03/foundation-names, 2021. Accessed: 2021-08-20. 4.6.3

[4] R. Amar, J. Eagan, and J. Stasko. Low-level components of analytic activity in information visualization. In *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.*, pages 111–117. IEEE, November 2005. doi: 10.1109/infvis.2005.1532136. URL `https://doi.org/10.1109/infvis.2005.1532136`. 3

[5] 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), 1 2016. doi: 10.1145/2854005. 4.3.1

[6] Abhijit Balaji, Thuvaarakkesh Ramanathan, and Venkateshwarlu Sonathi. Chart-Text: A Fully Automated Chart Image Descriptor. *arXiv Preprint*, 12 2018. Accessed: 2021-09-06. 4.3.1

[7] BANA. Guidelines and Standards for Tactile Graphics. Braille standard, Braille Authority of North America, 2010. Accessed: 2021-09-03. 4.3.1, 4.3.3

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

[9] 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 `https://doi.org/10.1080/00401706.1987.10488204`. 2.2.1

[10] 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, 10 2018. Association for Computing Machinery. doi: 10.1145/3234695.3236348. Accessed: 2021-09-03. 4.4.3

[11] 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, pages 1–7. ACM, May 2021. doi: 10.1145/3411763.3441321. URL `https://doi.org/10.1145/3411763.3441321`. 2.1.2

[12] Hakim Bey. *TAZ: The temporary autonomous zone, ontological anarchy, poetic terrorism*. Autonomedia, 2003. 9.3

[13] Brandon Biggs, Christopher Toth, Tony Stockman, James M. Coughlan, and Bruce N. Walker. Evaluation of a non-visual auditory choropleth and travel map viewer. In *Proceedings of the 27th International Conference on Auditory Display (ICAD 2022)*, pages 82–90. International Community for Auditory Display, June 2022. doi: 10.21785/icad2022.027. URL `https://doi.org/10.21785/icad2022.027`. PMCID: PMC10010675. 6.4.1

[14] 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. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, pages 333–342, New York, NY, USA, October 2010. ACM. ISBN 978-1-4503-0271-5. doi: 10.1145/1866029.1866080. URL `https://doi.org/10.1145/1866029.1866080`. Accessed: 2021-09-06. 4.3.3

[15] Matt Blanco, Jonathan Zong, and Arvind Satyanarayan. Olli: An Extensible Visualization Library for Screen Reader Accessibility. In *IEEE VIS Posters*, 2022. URL `https://vis.csail.mit.edu/pubs/olli`. Accessed: 2026-03-30, `https://vis.csail.mit.edu/pubs/olli`. 6.3.2

[16] 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, 10 2015. Association for Computing Machinery. doi: 10.1145/2700648.2809869. 4.3.1

[17] M. Bostock, V. Ogievetsky, and J. Heer. D³ data-driven documents. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2301–2309, December 2011. ISSN 1077-2626. doi: 10.1109/tvcg.2011.185. URL `https://doi.org/10.1109/tvcg.2011.185`. 1.2

[18] Denis Boudreau. Unlocking Accessibility for UX/UI Designers. https://www.24a11y.com/2018/unlocking-accessibility-for-ux-ui-designers/, 12 2018. Accessed: 2021-09-07. 4.3.4

[19] Virginia Braun and Victoria Clarke. Using thematic analysis in psychology. *Qualitative Research in Psychology*, 3(2):77–101, January 2006. ISSN 1478-0895. doi: 10.1191/1478088706qp063oa. URL `https://doi.org/10.1191/1478088706qp063oa`. 3, 6.6.3

[20] S. Brewster. Visualization tools for blind people using multiple modalities.

Disability and Rehabilitation, 24(11-12):613–621, January 2002. ISSN 1464-5165. doi: 10.1080/09638280110111388. URL `https://doi.org/10.1080/09638280110111388`. 2.2.3, 4.3.1, 6.3.1

[21] Craig Brown and Amy Hurst. Viztouch. In *Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction*, TEI '12, pages 131–138, New York, NY, USA, February 2012. ACM. ISBN 978-1-4503-1174-8. doi: 10.1145/2148131.2148160. URL `https://doi.org/10.1145/2148131.2148160`. Accessed: 2021-09-03. 4.3.1

[22] 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. In *Proceedings of the 8th international conference on Multimodal interfaces*, ICMI06, pages 108–117. ACM, November 2006. doi: 10.1145/1180995.1181017. URL `https://doi.org/10.1145/1180995.1181017`. 1.1

[23] 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. ACM, May 2021. doi: 10.1145/3411764.3445207. URL `https://doi.org/10.1145/3411764.3445207`. 4.3.1, 6.3.1

[24] Silvia Canelón and Liz Hare. Revealing Room for Improvement in Accessibility within a Social Media Data Visualization Learning Community. https://silvia.rbind.io/talk/2021-05-04-data-viz-accessibility/, 2021. Accessed: 2021-09-06. 4.3.2

[25] Katherine Casey. Radical decentralization: Does community-driven development work? *Annual Review of Economics*, 10(1):139–163, August 2018. ISSN 1941-1391. doi: 10.1146/annurev-economics-080217-053339. URL `http://dx.doi.org/10.1146/annurev-economics-080217-053339`. 9.3

[26] Alex Chaparro and Maria Chaparro. Applications of Color in Design for Color-Deficient Users. *Ergonomics in Design*, 25(1):23–30, 1 2017. doi: 10.1177/1064804616635382. Accessed: 2021-09-06. 2.2.3, 4.3.1

[27] 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, 9 2019. Association for Computing Machinery. doi: 10.1145/3341162.3345601. Accessed: 2021-09-06. 4.3.1

[28] 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, 3 2020. doi: 10.1109/WACV45572.2020.9093592. ISSN: 2642-9381. 4.3.1

[29] 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. doi: 10.1111/cgf.13686. Accessed: 2021-09-03. 4.3.1

[30] 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. doi: 10.1007/978-3-319-20886-2_14. 4.3.4

[31] 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. doi: 10.1109/tvcg.2021.3114829. URL `https://doi.org/10.1109/tvcg.2021.3114829`. 2.2.3, 4.3.1, 6.3.1

[32] 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, pages 1–5. ACM, October 2024. doi: 10.1145/3663548.3688496. URL `https://doi.org/10.1145/3663548.3688496`. 2.1.1

[33] RStudio Community. Making Shiny apps accessible for all humans. https://community.rstudio.com/t/making-shiny-apps-accessible-for-all-humans/8458, 5 2018. Accessed: 2021-09-06. 4.3.2

[34] 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, 7 2005. doi: 10.1109/IV.2005.28. ISSN: 2375-0138. 4.3.4

[35] 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. ACM, June 2019. doi: 10.1145/3311927.3323146. URL `https://doi.org/10.1145/3311927.3323146`. 2.2.3, 4.3.1, 6.4

[36] Lilian de Greef, Dominik Moritz, and Cynthia Bennett. Interdependent variables: Remotely designing tactile graphics for an accessible workflow. In *Proceedings of the 23rd International ACM SIGACCESS Conference on Computers and Accessibility*, pages 1–6. ACM, October 2021. doi: 10.1145/3441852.3476468. URL `https://doi.org/10.1145/3441852.3476468`. 6.4

[37] Chris DeMartini. Coin flip game. https://public.tableau.com/views/CoinFlipGame/CoinFlipGame, 2019. Accessed: 2021-08-20. 4.6.3

[38] Chris DeMartini. A Tableau Accessibility Journey - Part IV - Keyboard Accessibility. https://www.datablick.com/blog/2021/8/10/a-tableau-accessibility-journey-part-iv-keyboard-accessibility, 2021. Accessed: 2021-09-03. 4.3.2

[39] Deque. Automated Testing Identifies 57 Percent of Digital Accessibility Issues. Technical report, Deque, 3 2021. Accessed: 2021-09-06. 3, 4.2

[40] Deque. Axe-core, 8 2021. Accessed: 2021-08-20. 4.7.5

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

[42] Sebastian Draxler, Gunnar Stevens, Martin Stein, Alexander Boden, and David Randall. Supporting the social context of technology appropriation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 2835–2844. ACM, May 2012. doi: 10.1145/2207676.2208687. URL `https://doi.org/10.1145/2207676.2208687`. 2.1.2

[43] 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):e1010622, November 2022. doi: 10.1371/journal.pcbi.1010622. URL `https://doi.org/10.1371/journal.pcbi.1010622`. 2.2.2.2

[44] Frank Elavsky. Chartability, 2021. URL `https://chartability.github.io/POUR-CAF/`. Accessed: 2021-09-07. 4.5

[45] Frank Elavsky and Cindy Xiong Bearfield. Playing telephone with generative models: "verification disability," "compelled reliance," and accessibility in data visualization. In *2025 IEEE Workshop on Accessible Data Visualization (AccessViz)*, pages 14–24. IEEE, November 2025. doi: 10.1109/accessviz68666.2025.00008. URL `https://doi.org/10.1109/accessviz68666.2025.00008`. 6.3.1

[46] 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. URL `https://doi.org/10.1111/cgf.14522`. 2.2.2.1, 2.2.2.2, 2.2.3, 6.4.3, 6.8.2

[47] Frank Elavsky, Lucas Nadolskis, and Dominik Moritz. Data navigator: An accessibility-centered data navigation toolkit. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–11, 2023. ISSN 2160-9306. doi: 10.1109/tvcg.2023.3327393. URL `https://doi.org/10.1109/tvcg.2023.3327393`. 2.1.2, 6.2, 6.3.1, 6.3.2, 6.4, 6.5.1

[48] 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 `https://doi.org/10.1109/mcg.2013.53`. 6.2

[49] 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*, pages 1–11, 2023. ISSN 2160-9306. doi: 10.1109/tvcg.2023.3327367. URL `https://doi.org/10.1109/tvcg.2023.3327367`. 2.2.1

[50] Pavithra Eswaramoorthy, Frank Elavsky, Tania Allard, and gabalafou. Quansight-Labs/bokeh-a11y-audit: v1.0.0, February 2025. URL `https://bokeh-a11y-audit.readthedocs.io/`. 6.4.3

[51] 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. URL `https://doi.org/10.1145/3557899`. 1.1, 2.2.2.1, 2.2.2.2, 2.2.3, 6.3.2

[52] 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 `https://doi.org/10.1109/access.2023.3298050`. 6.3.1

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

[54] Danyel Fisher, Igor Popov, Steven Drucker, and m.c. schraefel. Trust me, i'm partially right. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 1673–1682. ACM, May 2012. doi: 10.1145/2207676.2208294. URL `https://doi.org/10.1145/2207676.2208294`. 6.2

[55] 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, 9 1997. doi: 10.1518/001872097778827151. Accessed: 2021-09-03. 2.2.3, 3, 4.3.1

[56] 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, 5 2010. Association for Computing Machinery. doi: 10.1145/1842993.1843029. Accessed: 2022-03-06. 4.3.4

[57] Plotly Community Forum. Are plotly tables accessible? https://community.plotly.com/t/are-plotly-tables-accessible/8263, 2 2018. Accessed: 2021-09-06. 4.3.2

[58] Plotly Community Forum. Datatables and Accessibility. https://community.plotly.com/t/solved-datatables-and-accessibility/31085, 11 2019. Accessed: 2021-09-06. 4.3.2

[59] 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 `https://doi.org/10.1177/15291006211051956`. 1.2

[60] A. Gallace and C. Spence. To what extent do Gestalt grouping principles influence tactile perception? *Psychological bulletin*, 2011. doi: 10.1037/a0022335. 4.3.1

[61] Josefina Isabel Gil Urrutia, Eric Brangier, Véronique Senderowicz, and Laurent Cessat. Beyond "usability and user experience", towards an integrative heuristic inspection: From accessibility to persuasiveness in the ux evaluation. *Advances in Intelligent Systems and Computing*, pages 460–470, June 2018. doi: 10.1007/978-3-319-60492-3_44. URL

https://doi.org/10.1007/978-3-319-60492-3_44. 4.3.4

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

[63] 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. ISBN 9783319942766. doi: 10.1007/978-3-319-94277-3_92. URL https://doi.org/10.1007/978-3-319-94277-3_92. 2.2.3, 4.3.1

[64] David Graeber and Charlie Rose. Never cower to authority, 2006. URL https://www.youtube.com/watch?v=qt5op6wft-8. Accessed [2026]. 9.3

[65] 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, 6 2014. Association for Computing Machinery. doi: 10.1145/2598510.2598595. 4.4

[66] 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 https://doi.org/10.2139/ssrn.4543807. 1.2

[67] Catherine Grevet and Eric Gilbert. Piggyback prototyping. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 4047–4056. ACM, April 2015. doi: 10.1145/2702123.2702395. URL https://doi.org/10.1145/2702123.2702395. 2.1.2

[68] Nathan Hahn, Joseph Chang, Ji Eun Kim, and Aniket Kittur. The knowledge accelerator. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI'16, pages 2258–2270. ACM, May 2016. doi: 10.1145/2858036.2858364. URL https://doi.org/10.1145/2858036.2858364. 2.1.1

[69] Jen Hale. Extensive digitization of tactile map collection. https://www.perkins.org/extensive-digitization-of-tactile-map-collection/, 2016. Perkins Archives Blog, Perkins School for the Blind. July 2016. Accessed: 2022-02-23. 2.2.3, 4.3.1

[70] 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, pages 1–11. ACM, December 2018. doi: 10.1145/3283458.3283478. URL https://doi.org/10.1145/3283458.3283478. 2.1.1

[71] Noor Hammad, Frank Elavsky, Sanika Moharana, Jessie Chen, Seyoung Lee, Patrick Carrington, Dominik Moritz, Jessica Hammer, and Erik Harpstead. Exploring the affordances

of game-aware streaming to support blind and low vision viewers: A design probe study. In *The 26th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '24, pages 1–13. ACM, October 2024. doi: 10.1145/3663548.3675665. URL https://doi.org/10.1145/3663548.3675665. 6.2, 6.9

[72] 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-enviro 2019. [Online]. 2.2.3, 3

[73] Gunnar Harboe and Elaine M. Huang. Real-world affinity diagramming practices. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 95–104. ACM, April 2015. doi: 10.1145/2702123.2702561. URL https://doi.org/10.1145/2702123.2702561. 6.6.3

[74] Parker Hartman and Tim Gorichanaz. Evaluating ai-powered website accessibility overlays. In *Proceedings of the 27th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '25, page 1–4. ACM, October 2025. doi: 10.1145/3663547.3759761. URL http://dx.doi.org/10.1145/3663547.3759761. 9.1

[75] Gillian R. Hayes. The relationship of action research to human-computer interaction. *ACM Transactions on Computer-Human Interaction*, 18(3):1–20, July 2011. ISSN 1073-0516. doi: 10.1145/1993060.1993065. URL https://doi.org/10.1145/1993060.1993065. 6.2, 6.4

[76] Jeffrey Heer and Maneesh Agrawala. Design considerations for collaborative visual analytics. In *2007 IEEE Symposium on Visual Analytics Science and Technology*, pages 171–178. IEEE, October 2007. doi: 10.1109/vast.2007.4389011. URL https://doi.org/10.1109/vast.2007.4389011. 2.2.1, 6.2

[77] 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://doi.org/10.1109/tvcg.2023.3327189. 1.2, 2.2.1, 3

[78] Ana O Henriques, Anna R. L. Carter, Beatriz Severes, Reem Talhouk, Angelika Strohmayer, Ana Cristina Pires, Colin M. Gray, Kyle Montague, and Hugo Nicolau. A feminist care ethics toolkit for community-based design: Bridging theory and practice. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, CHI '25, page 1–26. ACM, April 2025. doi: 10.1145/3706598.3713950. URL http://dx.doi.org/10.1145/3706598.3713950. 9.3

[79] Thomas Hermann, Andy Hunt, and John G Neuhoff, editors. *The Sonification Handbook*. Logos Verlag Berlin, Berlin, Germany, December 2011. ISBN: 978-3-8325-2819-5. 6.3.1

[80] 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 https://doi.org/

10.1145/3567721. 2.1.2

[81] 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, pages 1–17. ACM, April 2023. doi: 10.1145/3544548.3581249. URL https://doi.org/10.1145/3544548.3581249. 2.1.2

[82] 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, pages 1–15. ACM, October 2024. doi: 10.1145/3654777.3676391. URL https://doi.org/10.1145/3654777.3676391. 2.1.2

[83] Highsoft. Highcharts - interactive charting library for developers. https://www.highcharts.com/, 2009. [Online]. 4.3.2

[84] Highsoft. Highcharts accessibility module: information and demos. https://highcharts.com/docs/accessibility/accessibility-module, 2018. Accessed: 2021-09-06. 4.6.3

[85] Toan Hoang. The TableauMagic DataTables Extension. https://tableau.toanhoang.com/the-tableau-magic-datatables-extension-now-available/, 9 2018. Accessed: 2021-09-03. 4.3.2

[86] Leona Holloway, Peter Cracknell, Kate Stephens, Melissa Fanshawe, Samuel Reinders, Kim Marriott, and Matthew Butler. Refreshable tactile displays for accessible data visualisation, 2024. URL https://arxiv.org/abs/2401.15836. 6.3.1

[87] Leona M Holloway, Cagatay Goncu, Alon Ilsar, Matthew Butler, and Kim Marriott. Infosonics: Accessible infographics for people who are blind using sonification and voice. In *CHI Conference on Human Factors in Computing Systems*, CHI '22, pages 1–13. ACM, April 2022. doi: 10.1145/3491102.3517465. URL https://doi.org/10.1145/3491102.3517465. 6.3.1

[88] 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, pages 597–606. ACM, June 2014. doi: 10.1145/2598510.2598530. URL https://doi.org/10.1145/2598510.2598530. 2.1.1

[89] Stacy Hsueh, Beatrice Vincenzi, Akshata Murdeshwar, and Marianela Ciolfi Felice. Cripping data visualizations: Crip technoscience as a critical lens for designing digital access. In *The 25th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '23, page 1–16. ACM, October 2023. doi: 10.1145/3597638.3608427. URL http://dx.doi.org/10.1145/3597638.3608427. 1.1, 2.2.3, 3

[90] 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. doi: 10.1145/2485760.2485883. Accessed: 2021-09-03. 2.1.1, 4.8

[91] 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, pages 11–18. ACM, October 2011. doi: 10.1145/2049536.2049541. URL https://doi.org/10.1145/2049536.2049541. 2.1.1

[92] Edwin L. Hutchins, James D. Hollan, and Donald A. Norman. Direct manipulation interfaces. *Human–Computer Interaction*, 1(4):311–338, December 1985. ISSN 0737-0024. doi: 10.1207/s15327051hci0104_2. URL https://doi.org/10.1207/s15327051hci0104_2. 6.2

[93] Hilary Hutchinson, Wendy Mackay, Bo Westerlund, Benjamin B. Bederson, Allison Druin, Catherine Plaisant, Michel Beaudouin-Lafon, Stéphane Conversy, Helen Evans, Heiko Hansen, Nicolas Roussel, and Björn Eiderbäck. Technology probes. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '03, pages 17–24. ACM, April 2003. doi: 10.1145/642611.642616. URL https://doi.org/10.1145/642611.642616. 6.2, 6.9

[94] Web Accessibility Initiative. Web Accessibility Laws & Policies. https://www.w3.org/WAI/policies/, 8 2021. Accessed: 2021-08-19. 4.3.3

[95] 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, pages 611–620, New York, NY, USA, 2013. Association for Computing Machinery. doi: 10.1145/2470654.2470742. 4.8

[96] Farnaz Jahanbakhsh, Amy X. Zhang, Karrie Karahalios, and David R. Karger. Our browser extension lets readers change the headlines on news articles, and you won't believe what they did! *Proceedings of the ACM on Human-Computer Interaction*, 6 (CSCW2):1–33, November 2022. ISSN 2573-0142. doi: 10.1145/3555643. URL http://dx.doi.org/10.1145/3555643. 9.3

[97] Yvonne Jansen, Pierre Dragicevic, Petra Isenberg, Jason Alexander, Abhijit Karnik, Johan Kildal, Sriram Subramanian, and Kasper Hornbæk. *Opportunities and Challenges for Data Physicalization*, pages 3227–3236. Association for Computing Machinery, New York, NY, USA, 4 2015. doi: 10.1145/2702123.2702180. Accessed: 2021-09-03. 4.3.1

[98] Chandrika Jayant, Matt Renzelmann, Dana Wen, Satria Krisnandi, Richard Ladner, and Dan Comden. Automated tactile graphics translation. In *Proceedings of the 9th international ACM SIGACCESS conference on Computers and accessibility*, Assets '07, pages 75–82, New York, NY, USA, October 2007. ACM. ISBN 978-1-59593-573-1. doi: 10.1145/1296843.1296858. URL https://doi.org/10.1145/1296843.1296858. Accessed: 2021-09-03. 4.3.1

[99] 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, pages 1–15. ACM, October 2023. doi: 10.1145/3597638.3608423. URL https://doi.org/10.1145/3597638.3608423. 2.1.1

[100] 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, pages 1–14. ACM, May 2024. doi: 10.1145/3613904.3641970. URL `https://doi.org/10.1145/3613904.3641970`. 3, 6.3.1, 6.5.2

[101] 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. doi: 10.1007/978-3-319-41935-0_8. 4.3.4

[102] Shakila Cherise S Joyner, Amalia Riegelhuth, Kathleen Garrity, Yea-Seul Kim, and Nam Wook Kim. Visualization accessibility in the wild: Challenges faced by visualization designers. In *CHI Conference on Human Factors in Computing Systems*, CHI '22, pages 1–19. ACM, April 2022. doi: 10.1145/3491102.3517630. URL `https://doi.org/10.1145/3491102.3517630`. 2.2.3, 3, 6.2, 6.3.2, 6.8.2

[103] 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. URL `https://doi.org/10.1109/tvcg.2021.3114846`. 2.2.3, 4.3.1, 6.3.1

[104] 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, December 2022. ISSN 1557-7325. doi: 10.1145/3530013. URL `https://doi.org/10.1145/3530013`. 2.1.1

[105] Os Keyes, Josephine Hoy, and Margaret Drouhard. Human-computer insurrection: Notes on an anarchist hci. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, page 1–13. ACM, May 2019. doi: 10.1145/3290605.3300569. URL `http://dx.doi.org/10.1145/3290605.3300569`. 9.3

[106] 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, pages 1–13. ACM, April 2023. doi: 10.1145/3544548.3581139. URL `https://doi.org/10.1145/3544548.3581139`. 6.3.1

[107] 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, pages 1–19. ACM, May 2024. doi: 10.1145/3613904.3642442. URL `https://doi.org/10.1145/3613904.3642442`. 6.3.1

[108] 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*, pages 1–15. ACM, April 2023. doi: 10.1145/3544548.3581532. URL `https://doi.org/10.1145/3544548.`

3581532. 2.2.3, 6.3.1

[109] 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. URL `https://doi.org/10.1111/cgf.14298`. 2.2.3, 4.3.1, 6.3.1

[110] N. W. Kim, G. Ataguba, S. C. Joyner, Chuangdian Zhao, and Hyejin Im. Beyond alternative text and tables: Comparative analysis of visualization tools and accessibility methods. *Computer Graphics Forum*, 42(3):323–335, June 2023. ISSN 1467-8659. doi: 10.1111/cgf.14833. URL `https://doi.org/10.1111/cgf.14833`. 6.3.1, 6.3.2

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

[112] Rebecca Klein. Power BI Accessibility Best Practices. https://rklein324.github.io/PowerBIAccessibility/, 2020. Accessed: 2021-09-06. 4.3.2

[113] Richard E. Ladner. Design for user empowerment. *Interactions*, 22(2):24–29, 2 2015. doi: 10.1145/2723869. Accessed: 2022-03-07. 4.4.3

[114] Chufan Lai, Zhixian Lin, Ruike Jiang, Yun Han, Can Liu, and Xiaoru Yuan. *Automatic Annotation Synchronizing with Textual Description for Visualization*, pages 1–13. Association for Computing Machinery, New York, NY, USA, 4 2020. doi: doi.org/10.1145/3313831.3376443. Accessed: 2021-09-06. 4.3.1

[115] 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. 4.3.1

[116] 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, 4.3.1

[117] 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 `https://doi.org/10.1109/tvcg.2014.2346452`. 2.2.1, 3, 6.2

[118] 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. URL `https://doi.org/10.1109/tvcg.2021.3114770`. 2.2.2.2, 2.2.3, 4.3.1, 6.3.1

[119] Alan Lundgard, Crystal Lee, and Arvind Satyanarayan. Sociotechnical considerations for accessible visualization design. In *2019 IEEE Visualization Conference (VIS)*, pages 16–20. IEEE, October 2019. doi: 10.1109/visual.2019.8933762. URL `https://doi.org/10.1109/visual.2019.8933762`. 2.2.2.2, 4.3.1, 4.3.3, 6.3.2

[120] Kelly Mack, Emma McDonnell, Dhruv Jain, Lucy Lu Wang, Jon E. Froehlich, and Leah

Findlater. What do we mean by “accessibility research”? In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI ’21, pages 1–18, New York, NY, USA, May 2021. ACM. ISBN 978-1-4503-8096-6. doi: 10.1145/3411764.3445412. URL `https://doi.org/10.1145/3411764.3445412`. Accessed: 2022-02-22. 2.2.2.1, 2.2.3, 4.3.1

[121] Els Maeckelberghe. Feminist ethic of care: A third alternative approach. *Health Care Analysis*, 12(4):317–327, December 2004. ISSN 1573-3394. doi: 10.1007/s10728-004-6639-6. URL `http://dx.doi.org/10.1007/s10728-004-6639-6`. 9.3

[122] R G Maling and D C Clarkson. Electronic controls for the tetraplegic (possum) (patient operated selector mechanisms—p.o. s.m.). *Spinal Cord*, 1(3):161–174, November 1963. ISSN 1476-5624. doi: 10.1038/sc.1963.15. URL `http://dx.doi.org/10.1038/sc.1963.15`. 3

[123] 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, 10 2010. Association for Computing Machinery. doi: 10.1145/1878803.1878807. Accessed: 2021-09-06. 4.4.3

[124] 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. URL `https://doi.org/10.1007/bf00996201`. 2.2.3, 4.3.1, 6.3.1

[125] Mapbox. Mapbox gl accessibility. https://github.com/mapbox/mapbox-gl-accessibility, 8 2021. Accessed: 2021-09-06. 4.3.2

[126] Mapbox. Mapbox-gl js. https://www.npmjs.com/package/mapbox-gl, 2022. Accessed: 2021-09-03. 4.3.2

[127] Deborah Marks. Models of disability. *Disability and Rehabilitation*, 19(3):85–91, January 1997. ISSN 1464-5165. doi: 10.3109/09638289709166831. URL `https://doi.org/10.3109/09638289709166831`. 1.1

[128] 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, May 2021. doi: 10.1145/3457875. URL `https://doi.org/10.1145/3457875`. 2.2.1, 2.2.2.1, 2.2.3, 4.3.1, 6.3.1

[129] Kim Marriott, Matthew Butler, Leona Holloway, William Jolley, Bongshin Lee, Bruce Maguire, and Danielle Albers Szafir. From vision to touch: Bridging visual and tactile principles for accessible data representation. *IEEE Transactions on Visualization and Computer Graphics*, 32(1):659–669, January 2026. ISSN 2160-9306. doi: 10.1109/tvcg.2025.3634254. URL `https://doi.org/10.1109/tvcg.2025.3634254`. 6.3.1

[130] 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, February 2021. URL `https://doi.org/10.`

21203/rs.3.rs-156959/v1. 2.2.3, 4.3.1, 4.3.4

[131] Melanie Mazanec. Semiotic. https://semiotic.nteract.io/guides/accessibility, 2019. Accessed: 2021-09-03. 4.3.2

[132] 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, 10 2006. Association for Computing Machinery. doi: 10.1145/1182475.1182491. Accessed: 2021-09-03. 2.2.3

[133] David K. McGookin and Stephen A. Brewster. Soundbar. In *Proceedings of the 4th Nordic conference on Human-computer interaction: changing roles*, NordiCHI '06, pages 145–154, New York, NY, USA, October 2006. ACM. ISBN 978-1-59593-325-6. doi: 10.1145/1182475.1182491. URL https://doi.org/10.1145/1182475.1182491. Accessed: 2021-09-03. 4.3.1

[134] Andrew M. McNutt. No grammar to rule them all: A survey of json-style dsls for visualization. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–11, 2022. ISSN 2160-9306. doi: 10.1109/tvcg.2022.3209460. URL https://doi.org/10.1109/tvcg.2022.3209460. 6.2

[135] Catherine Mei∗, Josh Pollock∗, Daniel Hajas, Jonathan Zong, and Arvind Satyanarayan. Benthic: Perceptually congruent structures for accessible charts and diagrams. In *Proceedings of the 27th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '25, pages 1–17. ACM, October 2025. doi: 10.1145/3663547.3746342. URL https://doi.org/10.1145/3663547.3746342. 6.3.1, 6.3.2

[136] Giles Mohan and Kristian Stokke. Participatory development and empowerment: The dangers of localism. *Third World Quarterly*, 21(2):247–268, April 2000. ISSN 1360-2241. doi: 10.1080/01436590050004346. URL http://dx.doi.org/10.1080/01436590050004346. 9.3

[137] 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, 10 2014. Association for Computing Machinery. doi: 10.1145/2661334.2661368. 4.3.1

[138] Dominik Moritz, Bill Howe, and Jeffrey Heer. Falcon. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, pages 1–11. ACM, May 2019. doi: 10.1145/3290605.3300924. URL https://doi.org/10.1145/3290605.3300924. 3

[139] Peya Mowar, Aaron Steinfeld, and Jeffrey P. Bigham. iTagPDF: Towards finally automating PDF accessibility. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems*, CHI '26. ACM, 2026. doi: 10.1145/3772318.3790289. URL https://peyajm29.github.io/itagpdf.html. Best Paper Award. 6.3.2

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

09-07. 4.3.4

[141] Jakob Nielsen. *Heuristic evaluation*, pages 25–62. John Wiley & Sons, Inc., USA, 6 1994. Accessed: 2021-09-06. 4.3.4

[142] 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):e0199239, August 2018. ISSN 1932-6203. doi: 10.1371/journal.pone.0199239. URL `https://doi.org/10.1371/journal.pone.0199239`. 2.2.3, 4.3.1

[143] Jason Obeid and Enamul Hoque. Chart-to-Text: Generating Natural Language Descriptions for Charts by Adapting the Transformer Model. *arXiv Preprint*, 11 2020. Accessed: 2021-09-06. 4.3.1

[144] 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, 4.2

[145] Manuel M. Oliveira. Towards More Accessible Visualizations for Color-Vision-Deficient Individuals. *Computing in Science Engineering*, 15(5):80–87, 9 2013. doi: 10.1109/MCSE.2013.113. Conference Name: Computing in Science Engineering. 2.2.3, 4.3.1, 4.3.4

[146] World Health Organization. *World report on vision*. World Health Organization, 2019. ISBN 978-92-4-151657-0. Accessed: 2021-09-03. 4.2

[147] 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, 9 2017. Association for Computing Machinery. doi: 10.1145/3123818.3133832. 4.3.4

[148] Fabio Pittarello and Manuel Semenzato. Towards a data physicalization toolkit for non-sighted users. In *2024 IEEE 21st Consumer Communications & Networking Conference (CCNC)*, pages 1–6. IEEE, January 2024. doi: 10.1109/ccnc51664.2024.10454861. URL `https://doi.org/10.1109/ccnc51664.2024.10454861`. 2.1.2

[149] 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, pages 433–442, New York, NY, USA, 2012. Association for Computing Machinery. doi: 10.1145/2207676.2207736. 3, 4.2

[150] 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*, pages 2792–2804. Association for Computing Machinery, New York, NY, USA, 4 2021. doi: 10.1145/3442381.3449923. Accessed: 2021-09-06. 4.3.1

[151] 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, pages 1–8. ACM, October 2023. doi: 10.1145/3597638.3614546. URL `https://doi.org/10.1145/3597638.3614546`. 6.4

[152] J.E. Rawling, E.C. Carson, J.W. Attig, D.M. Mickelson, W.N. Mode, M.D. Johnson, and K.M. Syverson. Quaternary geology of wisconsin. Technical report, Wisconsin Geological and Natural History Survey, 2025. URL `https://doi.org/10.54915/xqpw9883`. 6.4.1

[153] Blake Ellis Reid. The curb-cut effect and the perils of accessibility without disability. *SSRN Electronic Journal*, 2022. doi: 10.2139/ssrn.4262991. URL `https://doi.org/10.2139/ssrn.4262991`. 1.1

[154] 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 `https://doi.org/10.1109/tvcg.2024.3456358`. 6.3.1

[155] 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

[156] Maurício Rossi de 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*, pages 225–232. INSTICC, SCITEPRESS - Science and Technology Publications, 2017. ISBN 978-989-758-228-8. doi: 10.5220/0006133202250232. URL `https://doi.org/10.5220/0006133202250232`. 4.3.4

[157] Heather Ruland Staines. Digital open annotation with hypothesis: Supplying the missing capability of the web. *Journal of Scholarly Publishing*, 49(3):345–365, April 2018. ISSN 1710-1166. doi: 10.3138/jsp.49.3.04. URL `http://dx.doi.org/10.3138/jsp.49.3.04`. 9.3

[158] Manaswi Saha, Michael Saugstad, Hanuma Teja Maddali, Aileen Zeng, Ryan Holland, Steven Bower, Aditya Dash, Sage Chen, Anthony Li, Kotaro Hara, and Jon Froehlich. Project sidewalk: A web-based crowdsourcing tool for collecting sidewalk accessibility data at scale. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, page 1–14. ACM, May 2019. doi: 10.1145/3290605.3300292. URL `http://dx.doi.org/10.1145/3290605.3300292`. 9.3

[159] 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 `https://doi.org/10.17011/ht/urn.200811065856`.

2.1.2

[160] City San Francisco. Covid-19 data and reports San Francisco. https://sf.gov/resource/2021/covid-19-data-and-reports, 2021. Accessed: 2021-09-03. 4.3.2

[161] Elizabeth B.-N. Sanders and Pieter Jan Stappers. Probes, toolkits and prototypes: three approaches to making in codesigning. *CoDesign*, 10(1):5–14, January 2014. doi: 10.1080/15710882.2014.888183. URL `https://doi.org/10.1080/15710882.2014.888183`. 2.1.2

[162] 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, 10 2018. doi: 10.1109/BELIV.2018.8634108. 4.3.4

[163] SAS. Sas Graphics Accelerator. https://support.sas.com/software/products/graphics-accelerator/, 2022. Accessed: 2021-09-08. 4.3.2

[164] 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. URL `https://doi.org/10.1109/tvcg.2016.2599030`. 1.2, 2.2.1, 6.2, 6.5.2

[165] 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, 5 2021. Association for Computing Machinery. Accessed: 2021-09-06. 2.2.3, 4.3.1

[166] D. Schepers and F. Elavsky. "accessible data viz". *The Investigative Reporters and Editors Journal*, 4:8–9, December 2024. 2.2.3

[167] Doug Schepers. Why Accessibility Is at the Heart of Data Visualization by Doug Schepers Nightingale Medium. https://medium.com/nightingale/accessibility-is-at-the-heart-of-data-visualization-64a38d6c505b, 2020. Accessed: 2021-09-06. 4.4.2

[168] William Schiff and Emerson Foulke, editors. *Tactual Perception*. Cambridge University Press, Cambridge, England, March 1982. 2.2.3, 4.3.1

[169] Jochen Schneider. Constructing the Yellow Brick Road: Route Bricks on Virtual Tactile Maps, 2000. 4.3.1

[170] Jean Scholtz. Developing guidelines for assessing visual analytics environments. *Information Visualization*, 10(3):212–231, 7 2011. doi: 10.1177/1473871611407399. Accessed: 2022-03-06. 4.3.4

[171] Donald Schön and John Bennett. Reflective conversation with materials, April 1996. URL `https://doi.org/10.1145/229868.230044`. 6.8.1

[172] 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, pages 1–22. ACM, May 2024. doi: 10.1145/3613904.3642730. URL `https://doi.org/10.`

1145/3613904.3642730. 6.3.1

[173] Hanieh Shakeri, Ye Yuan, Benett Axtell, Denise Y. Geiskkovitch, and Carman Neustaedter. Designing smart home technology for passive co-presence over distance. In *Designing Interactive Systems Conference*, DIS '24, page 3389–3406. ACM, July 2024. doi: 10.1145/3643834.3661508. URL http://dx.doi.org/10.1145/3643834.3661508. 9.3

[174] 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. IEEE, January 2018. doi: 10.1109/ccnc.2018.8319239. URL https://doi.org/10.1109/ccnc.2018.8319239. 4.3.1, 6.3.2

[175] Ather Sharif, Sanjana Shivani Chintalapati, Jacob O. Wobbrock, and Katharina Reinecke. Understanding screen-reader users' experiences with online data visualizations. In *Proceedings of the 23rd International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '21, pages 1–16. ACM, October 2021. doi: 10.1145/3441852.3471202. URL https://doi.org/10.1145/3441852.3471202. 2.2.3, 4.3.1, 4.3.3, 6.3.1, 6.3.2

[176] 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*, pages 1–19. ACM, April 2022. doi: 10.1145/3491102.3517431. URL https://doi.org/10.1145/3491102.3517431. 6.3.2

[177] Ather Sharif, Joo Gyeong Kim, Jessie Zijia Xu, and Jacob O. Wobbrock. Understanding and reducing the challenges faced by creators of accessible online data visualizations. In *The 26th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '24, pages 1–20. ACM, October 2024. doi: 10.1145/3663548.3675625. URL https://doi.org/10.1145/3663548.3675625. 2.2.3, 3, 6.2, 6.3.2, 6.8.2

[178] Lei Shi, Idan Zelzer, Catherine Feng, and Shiri Azenkot. Tickers and talker. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 4896–4907, New York, NY, USA, May 2016. ACM. ISBN 978-1-4503-3362-7. doi: 10.1145/2858036.2858507. URL https://doi.org/10.1145/2858036.2858507. Accessed: 2021-09-03. 2.1.2, 4.3.1

[179] Ben Shneiderman, Christopher Williamson, and Christopher Ahlberg. Dynamic queries. In *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '92*, CHI '92, pages 669–670. ACM Press, 1992. doi: 10.1145/142750.143082. URL https://doi.org/10.1145/142750.143082. 6.2

[180] 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, 5 1999. Association for Computing Machinery. doi: 10.1145/632716.632902. 4.3.4

[181] Volker Sorge. Polyfilling accessible chemistry diagrams. In *Lecture Notes in Computer Science*, pages 43–50. Springer International Publishing, 2016. ISBN

9783319412634. doi: 10.1007/978-3-319-41264-1_6. URL https://doi.org/10.1007/978-3-319-41264-1_6. 2.2.3, 4.3.1, 6.3.1

[182] Laura South and Michelle Borkin. Generating Seizure-Inducing Sequences with Interactive Visualizations. In *Proc. Of the IEEE VIS 2020 posters*, 10 2020. doi: 10.31219/osf.io/85gwy. 2.2.3, 4.3.1

[183] Laura South, David Saffo, and Michelle Borkin. Detecting and Defending Against Seizure-Inducing GIFs in Social Media. Technical report, OSF Preprints, 1 2021. Accessed: 2021-08-20. 2.2.3, 4.3.1

[184] Katta Spiel, Kathrin Gerling, Cynthia L. Bennett, Emeline Brulé, Rua M. Williams, Jennifer Rode, and Jennifer Mankoff. Nothing about us without us. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, pages 1–8. ACM, April 2020. doi: 10.1145/3334480.3375150. URL https://doi.org/10.1145/3334480.3375150. 1.1, 9.1

[185] Clay Spinuzzi. The methodology of participatory design. *Technical communication*, 52 (2):163–174, 2005. 2.1.1

[186] 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 https://doi.org/10.3758/s13428-024-02496-z. 1.1

[187] Henny Swan, Ian Pouncey, Heydon Pickering, and Léonie Watson. Inclusive Design Principles. https://inclusivedesignprinciples.org/, 2019. Accessed: 2021-09-07. 4.4.2

[188] Sarit Felicia Anais Szpiro, Shafeka Hashash, Yuhang Zhao, and Shiri Azenkot. How people with low vision access computing devices. In *Proceedings of the 18th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '16, pages 171–180, New York, NY, USA, October 2016. ACM. ISBN 9781450341240. doi: 10.1145/2982142.2982168. URL https://doi.org/10.1145/2982142.2982168. 2.2.3, 4.3.1

[189] Pierre Tchounikine. Designing for appropriation: A theoretical account. *Human–Computer Interaction*, 32(4):155–195, July 2017. ISSN 1532-7051. doi: 10.1080/07370024.2016.1203263. URL https://doi.org/10.1080/07370024.2016.1203263. 2.1.2

[190] Yuanyang (YY) Teng and Darren Gergle. Access is not enough: Toward developmental flourishing. In *Proceedings of the Extended Abstracts of the 2026 CHI Conference on Human Factors in Computing Systems*, CHI EA '26, page 1–6. ACM, April 2026. doi: 10.1145/3772363.3798552. URL http://dx.doi.org/10.1145/3772363.3798552. 1.1

[191] 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*, pages 1–18. ACM, April 2023. doi: 10.1145/3544548.3581186. URL https://doi.org/10.1145/3544548.3581186. 6.3.1, 6.3.2, 6.4

[192] 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, pages 917–933. ACM, July 2023. doi: 10.1145/3563657.3596057. URL `https://doi.org/10.1145/3563657.3596057`. 2.1.1

[193] US Government, 10x. Improving Accessibility in Data Visualizations (formerly USWDS Visualization Tool) on 10x. https://trello.com/c/OOOres7a/63-improving-accessibility-in-data-visualizations-formerly-uswds-visualization-tool, 2020. Accessed: 2022-03-04. 4.3.4

[194] US Government, General Services Administration. Data visualizations U.S. Web Design System (USWDS). https://designsystem.digital.gov/components/data-visualizations/, 2021. Accessed: 2022-03-04. 4.3.4

[195] Visa. Visa Chart Components. `https://github.com/visa/visa-chart-components`, 2022. Accessed: 2022-12-01. 4.3.2

[196] W3C. Web content accessibility guidelines (wcag) 2.1, 2018. 2.2.2.2, 3, 4.3.1, 4.3.3

[197] WAI. Understanding success criterion 4.1.2: name, role, value. WCAG standard, W3C, 2016. Accessed: 2022-03-04. 2.2.2.2

[198] WAI. Understanding success criterion 2.1.1: keyboard. WCAG standard, W3C, 2017. URL `https://www.w3.org/WAI/WCAG21/Understanding/keyboard.html`. Accessed: 2026-03-20, `https://www.w3.org/WAI/WCAG21/Understanding/keyboard.html`. 6.3.1

[199] WAI. Accessibility principles. WCAG Standard, W3C, 2019. URL `https://www.w3.org/WAI/fundamentals/accessibility-principles/`. Accessed: 2021-09-06. 4.2, 4.3.3

[200] WAI. W3C Accessibility Guidelines (WCAG) 3.0. WCAG standard, W3C, 2021. Accessed: 2022-03-04. 4.8

[201] Chris Weaver. Multidimensional visual analysis using cross-filtered views. In *2008 IEEE Symposium on Visual Analytics Science and Technology*, pages 163–170. IEEE, October 2008. doi: 10.1109/vast.2008.4677370. URL `https://doi.org/10.1109/vast.2008.4677370`. 2.2.1, 3

[202] WebAIM. Webaim: The WebAIM Million - An annual accessibility analysis of the top 1,000,000 home pages. https://webaim.org/projects/million/#wcag, 2021. Accessed: 2021-09-03. 4.5.1, 9.3

[203] Simon Wheatcroft. Making Graphs And Plots Accessible For The Blind. https://andadapt.com/making-graphs-and-plots-accessible-for-the-blind/, 6 2020. Accessed: 2021-09-06. 4.3.2

[204] Hadley Wickham. A layered grammar of graphics. *Journal of Computational and Graphical Statistics*, 19(1):3–28, January 2010. ISSN 1537-2715. doi: 10.1198/jcgs.2009.07098. URL `https://doi.org/10.1198/jcgs.2009.07098`. 6.2

[205] David Gray Widder and Richmond Wong. Thinking upstream: Ethics and policy oppor-

tunities in ai supply chains, 2023. URL `https://arxiv.org/abs/2303.07529`. 1.2

[206] 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 `https://doi.org/10.1038/s41586-024-08141-1`. 1.2

[207] Leland Wilkinson. The grammar of graphics. In *Handbook of Computational Statistics*, pages 375–414. Springer Berlin Heidelberg, December 2012. ISBN 9783642215506. doi: 10.1007/978-3-642-21551-3_13. URL `https://doi.org/10.1007/978-3-642-21551-3_13`. 6.2

[208] 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

[209] Brianna L. Wimer, Ritesh Kanchi, Kaija Frierson, Venkatesh Potluri, Ronald Metoyer, Jennifer Mankoff, Miya Natsuhara, and Matt X. Wang. Nonvisual support for understanding and reasoning about data structures, 2026. URL `https://arxiv.org/abs/2601.19168`. 6.3.2

[210] Langdon Winner. Do artifacts have politics? In *Computer Ethics*, pages 177–192. Routledge, May 2017. ISBN 9781315259697. doi: 10.4324/9781315259697-21. URL `https://doi.org/10.4324/9781315259697-21`. 1.2

[211] 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. doi: 10.1145/1952383.1952384. 1.2, 2.1.1

[212] 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. URL `https://doi.org/10.1145/1952383.1952384`. 4.4.3

[213] Mintesnot Woldeamanuel and Andrew Kent. Measuring walk access to transit in terms of sidewalk availability, quality, and connectivity. *Journal of Urban Planning and Development*, 142(2), June 2016. ISSN 1943-5444. doi: 10.1061/(asce)up.1943-5444.0000296. URL `http://dx.doi.org/10.1061/(ASCE)UP.1943-5444.0000296`. 9.3

[214] 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 `https://doi.org/10.1109/tvcg.2015.2467191`. 2.2.1

[215] 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. 3

[216] 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, 5 2021. Association for Computing Machinery. doi: 10.1145/3411764.3445743. Accessed: 2021-09-03. 2.2.3, 4.3.1

[217] 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, 10 2020. doi: 10.1109/TVCG.2019.2917689. Conference Name: IEEE Transactions on Visualization and Computer Graphics. 4.5.4

[218] 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 `https://doi.org/10.1016/s0305-750x(02)00218-8`. 1.1

[219] Haixia Zhao, Catherine Plaisant, Ben Shneiderman, and Jonathan Lazar. Data sonification for users with visual impairment. *ACM Transactions on Computer-Human Interaction*, 15 (1):1–28, May 2008. ISSN 1073-0516. doi: 10.1145/1352782.1352786. URL `https://doi.org/10.1145/1352782.1352786`. 2.2.3, 4.3.1

[220] Jonathan Zong. Using real names of disabled participant-contributors to practice citational justice in accessibility. In *2025 IEEE Workshop on Accessible Data Visualization (AccessViz)*, pages 30–33. IEEE, November 2025. doi: 10.1109/accessviz68666.2025.00011. URL `https://doi.org/10.1109/accessviz68666.2025.00011`. 6.4

[221] 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. URL `https://doi.org/10.1111/cgf.14519`. 2.2.2.2, 3, 6.3.1, 6.3.2, 6.4

[222] Jonathan Zong, Josh Pollock, Dylan Wootton, and Arvind Satyanarayan. Animated vega-lite: Unifying animation with a grammar of interactive graphics. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):149–159, January 2023. ISSN 2160-9306. doi: 10.1109/tvcg.2022.3209369. URL `https://doi.org/10.1109/tvcg.2022.3209369`. 6.2

[223] 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, pages 1–20. ACM, May 2024. doi: 10.1145/3613904.3641996. URL `https://doi.org/10.1145/3613904.3641996`. 6.3.2, 6.8.4