VoxLens: Making Online Data Visualizations Accessible with an Interactive JavaScript Plug-In

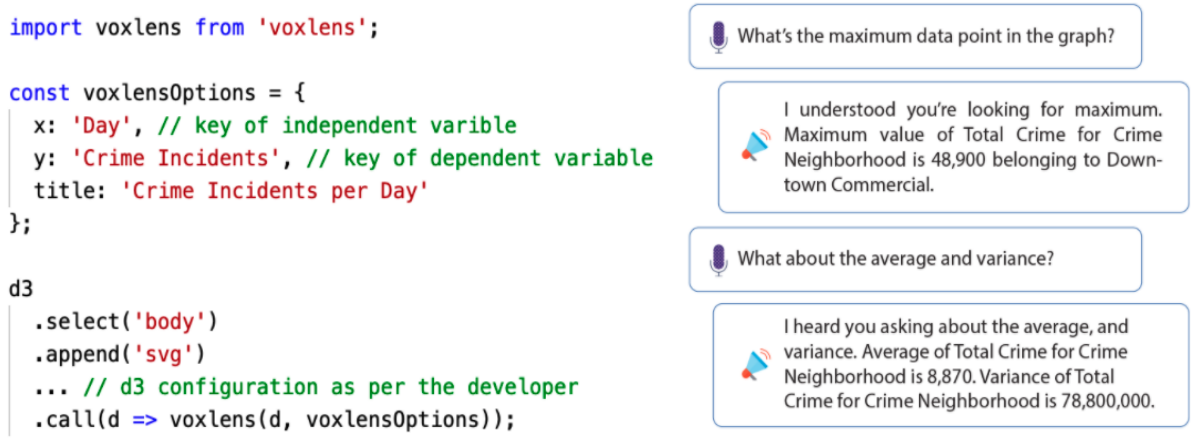
Ather Sharif
asharif@cs.washington.edu
Paul G. Allen School of Computer Science & Engineering | DUB Group,
University of Washington
Seattle, Washington, USA

Olivia H. Wang
wang4@cs.washington.edu
Paul G. Allen School of Computer Science & Engineering,
University of Washington
Seattle, Washington, USA

Alida T. Muongchan
alidatm@uw.edu
Human Centered Design and Engineering,
University of Washington
Seattle, Washington, USA

Katharina Reinecke
reinecke@cs.washington.edu
Paul G. Allen School of Computer Science & Engineering | DUB Group,
University of Washington
Seattle, Washington, USA

Jacob O. Wobbrock
wobbrock@uw.edu
The Information School | DUB Group,
University of Washington
Seattle, Washington, USA



```
import voxlens from 'voxlens';

const voxlensOptions = {
  x: 'Day', // key of independent varible
  y: 'Crime Incidents', // key of dependent variable
  title: 'Crime Incidents per Day'
};

d3
  .select('body')
  .append('svg')
  ... // d3 configuration as per the developer
  .call(d => voxlens(d, voxlensOptions));
```

What's the maximum data point in the graph?

I understood you're looking for maximum. Maximum value of Total Crime for Crime Neighborhood is 48,900 belonging to Downtown Commercial.

What about the average and variance?

I heard you asking about the average, and variance. Average of Total Crime for Crime Neighborhood is 8,870. Variance of Total Crime for Crime Neighborhood is 78,800,000.

Figure 1: VoxLens is an open-source JavaScript plug-in that improves the accessibility of online data visualizations using a multi-modal approach. The code at left shows that integration of VoxLens requires only a single line of code. At right, we portray an example interaction with VoxLens using voice-activated commands for screen-reader users.

ABSTRACT

JavaScript visualization libraries are widely used to create online data visualizations but provide limited access to their information for screen-reader users. Building on prior findings about the experiences of screen-reader users with online data visualizations, we present VoxLens, an open-source JavaScript plug-in that—with a single line of code—improves the accessibility of online data visualizations for screen-reader users using a multi-modal approach. Specifically, VoxLens enables screen-reader users to obtain a holistic summary of presented information, play sonified versions of the data, and interact with visualizations in a "drill-down" manner using voice-activated commands. Through task-based experiments with 21 screen-reader users, we show that VoxLens improves the accuracy of information extraction and interaction time by 122% and 36%, respectively, over existing conventional interaction with online data visualizations. Our interviews with screen-reader users suggest that VoxLens is a "game-changer" in making online data visualizations accessible to screen-reader users, saving them time and effort.

CHI '22, April 29-May 5, 2022, New Orleans, LA, USA
© 2022 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9157-3/22/04.
https://doi.org/10.1145/3491102.3517431

CCS CONCEPTS

• **Human-centered computing → Information visualization**; **Accessibility systems and tools**; • **Social and professional topics → People with disabilities**.

KEYWORDS

Visualizations, accessibility, screen readers, voice-based interaction, blind, low-vision.

ACM Reference Format:
Ather Sharif, Olivia H. Wang, Alida T. Muongchan, Katharina Reinecke, and Jacob O. Wobbrock. 2022. VoxLens: Making Online Data Visualizations Accessible with an Interactive JavaScript Plug-In. In *CHI Conference on Human Factors in Computing Systems (CHI '22), April 29-May 5, 2022, New Orleans, LA, USA.* ACM, New York, NY, USA, 19 pages. https://doi.org/10.1145/3491102.3517431

1 INTRODUCTION

Online data visualizations are present widely on the Web, allowing experts and non-experts alike to explore and analyze data both simple and complex. They assist people in extracting information effectively and efficiently, taking advantage of the ability of the human mind to recognize and interpret visual patterns [57].

However, the visual nature of data visualizations inherently disenfranchises screen-reader users, who may not be able to see or recognize visual patterns [52, 57]. We define "screen-reader users," following prior work [69], as people who utilize a screen reader (*e.g.*, JAWS [68], NVDA [2], or VoiceOver [44]) to read the contents of a computer screen. They might have conditions including complete or partial blindness, low vision, learning disabilities (such as alexia), motion sensitivity, or vestibular hypersensitivity.

Due to the inaccessibility of data visualizations, screen-reader users commonly cannot access them at all. Even when the data visualization includes basic accessibility functions (e.g., alternative text or a data table), screen-reader users still spend 211% more time interacting with online data visualizations and answer questions about the data in the visualizations 61% less accurately, compared to non-screen-reader users [69]. Screen-reader users rely on the creators of visualizations to provide adequate alternative text, which is often incomplete. Additionally, they have to remember and process more information mentally than is often humanly feasible [74], such as when seeking the maximum or minimum value in a chart. Prior work has studied the experiences of screen-reader users with online data visualizations and highlighted the challenges they face, the information they seek, and the techniques and strategies that could make online data visualizations more accessible [69]. Building on this work, it is our aim to realize a novel interactive solution to enable screen-reader users to efficiently interact with online data visualizations.

To this end, we created an open-source JavaScript plug-in called "VoxLens," following an iterative design process [1]. VoxLens provides screen-reader users with a multi-modal solution that supports three modes of interaction: (1) *Question-and-Answer mode*, where the user verbally interacts with the visualizations on their own; (2) *Summary mode*, where VoxLens describes the summary of the information contained in the visualization; and (3) *Sonification mode*, where VoxLens maps the data in the visualization to a musical scale, enabling listeners to interpret the data trend. (Existing sonification tools are either proprietary [39] or written in a programming language other than JavaScript [5], making them unintegratable with popular JavaScript visualization libraries; VoxLens' sonification feature is open-source, integratable with other libraries, and customizable.) Additionally, VoxLens reduces the burden on visualization creators in applying accessibility features to their data visualizations, requiring inserting only a single line of JavaScript code during visualization creation. Furthermore, VoxLens enables screen-reader users to explore the data as per their individual preferences, without relying on the visualization creators and without having to process data in their minds. VoxLens is the first system to: (1) enable screen-reader users to interact with online data visualizations using voice-activated commands; and (2) offer a multi-modal solution using three different modes of interaction.

To assess the performance of VoxLens, we conducted controlled task-based experiments with 21 screen-reader users. Specifically, we analyzed the accuracy of extracted information and interaction time with online data visualizations. Our results show that with VoxLens, compared to without it, screen-reader users improved their accuracy of extracting information by 122% and reduced their overall interaction time by 36%. Additionally, we conducted follow-up semi-structured interviews with six participants, finding that VoxLens is a positive step forward in making online data visualizations accessible, interactive dialogue is one of the 'top' features, sonification helps in 'visualizing' data, and data summary is a good starting point. Furthermore, we assessed the perceived workload of VoxLens using the NASA-TLX questionnaire [38], showing that VoxLens leaves users feeling successful in their performance and demands low physical effort.

The main contributions of our work are as follows:

(1) VoxLens, an interactive JavaScript plug-in that improves the accessibility of online data visualizations for screen-reader users. VoxLens offers a multi-modal solution, enabling screen-reader users to explore online data visualizations, both holistically and in a drilled-down manner, using voice-activated commands. We present its design and architecture, functionality, commands, and operations. Additionally, we open-source our implementation at https://github.com/athersharif/voxlens.
(2) Results from formative and summative studies with screen-reader users evaluating the performance of VoxLens. With VoxLens, screen-reader users significantly improved their interaction performance compared to their conventional interaction with online data visualizations. Specifically, VoxLens increased their accuracy of extracting information by 122% and decreased their interaction time by 36% compared to not using VoxLens.

2 RELATED WORK

We review the previous research on the experiences of screen-reader users with online data visualizations and the systems designed to improve the accessibility of data visualizations for screen-reader users. Additionally, we review existing JavaScript libraries used to create online visualizations, and tools that generate audio graphs.

2.1 Experiences of Screen-Reader Users with Online Data Visualizations

Understanding the experiences and needs of users is paramount in the development of tools and systems [8, 40]. Several prior research efforts have conducted interviews with blind and low-vision (BLV) users to understand their experiences with technology [7, 43, 67, 69, 82]. Most recently, Zhang *et al.* [82] conducted interviews with 12 BLV users, reporting four major challenges of current emoji entry methods: (1) the entry process is time-consuming; (2) the results from these methods are inconsistent with the expectations of users; (3) there is a lack of support for discovering new emojis; and (4) there is a lack of support for finding the right emojis. They utilized these findings to develop Voicemoji, a speech-based emoji entry system that enables BLV users to input emojis. Schaadhardt *et al.* [67] conducted contextual interviews with 12 blind users, identifying key accessibility problems with 2-D digital artboards, such as Microsoft PowerPoint and Adobe Illustrator. Similarly, Sharif *et al.* [69] conducted contextual interviews with 9 screen-reader users, highlighting the inequities screen-reader users face when interacting with online data visualizations. They reported the challenges screen-reader users face, the information they seek, and the techniques and strategies they prefer to make online data visualizations more accessible. We rely upon the findings from Sharif *et al.* [69] to design VoxLens, an interactive JavaScript plug-in that improves the accessibility of online data visualizations, deriving motivation from Marriott *et al.*'s [57] call-to-action for creating inclusive data visualizations for people with disabilities.

2.2 Accessibility of Online Data Visualizations

Prior research efforts have explored several techniques to make data visualizations more accessible to BLV users, including automatically generating alternative text for visualization elements [48, 59, 70], sonification [3, 5, 16, 27, 39, 58, 83], haptic graphs [33, 76, 81], 3-D printing [15, 43, 71], and trend categorization [47]. For example, Sharif *et al.* [70] developed evoGraphs, a jQuery plug-in to create accessible graphs by automatically generating alternative text. Similarly, Kim *et al.* [47] created a framework that uses multimodal deep learning to generate summarization text from image-based line graphs. Zhao *et al.* [83] developed iSonic, which assists BLV users in exploring georeferenced data through non-textual sounds and speech output. They conducted in-depth studies with seven blind users, finding that iSonic enabled blind users to find facts and discover trends in georeferenced data. Yu *et al.* [81] developed a system to create haptic graphs, evaluating their system using an experiment employing both blind and sighted people, finding that haptic interfaces are useful in providing the information contained in a graph to blind computer users. Hurst *et al.* [43] worked with six individuals with low or limited vision and developed VizTouch, software that leverages affordable 3-D printing to rapidly and automatically generate tangible visualizations.

Although these approaches are plausible solutions for improving the accessibility of visualizations for screen-reader users, at least one of the following is true for all of them: (1) they require additional equipment or devices; (2) they are not practical for spontaneous everyday web browsing; (3) they do not offer a multi-modal solution; and (4) they do not explore the varying preferences of visualization interaction among screen-reader users. In contrast, VoxLens does not need any additional equipment, is designed for spontaneous everyday web browsing, and offers a multi-modal solution catering to the individual needs and abilities of screen-reader users.

2.3 Existing JavaScript Data Visualization Libraries

Several JavaScript data visualization libraries exist that enable visualization creators to make visualizations for the Web. We classified these visualization libraries into two categories based on accessibility features: (1) libraries that rely on developers to append appropriate alternative text (*e.g.*, D3 and ChartJS); and (2) libraries that automatically provide screen-reader users with built-in features for data access (*e.g.*, Google Charts).

Bostock *et al.* [11] developed D3—a powerful visualization library that uses web standards to generate graphs. D3 uses Scalable Vector Graphics (SVG) [22] to create such visualizations, relying on the developers to provide adequate alternative text for screen-reader users to comprehend the information contained in the visualizations.

Google Charts [23] is a visualization tool widely used to create graphs. An important underlying accessibility feature of Google Charts is the presence of a visually hidden tabular representation of data. While this approach allows screen-reader users to access the raw data, extracting information is a cumbersome task. Furthermore, tabular representations of data introduce excessive user workloads, as screen-reader users have to sequentially go through each data point. The workload is further exacerbated as data cardinality increases, forcing screen-reader users to memorize each data point to extract even the most fundamental information such as minimum or maximum values.

In contrast to these approaches, VoxLens introduces an alternate way for screen-reader users to obtain their desired information without relying on visualization creators, and without mentally computing complex information through memorization of data.

2.4 Audio Graphs

Prior work has developed sonification tools to enable screen-reader users to explore data trends and patterns in online data visualizations [3, 5, 39, 58, 83]. McGookin *et al.* [58] developed SoundBar, a system that allows blind users to gain a quick overview of bar graphs using musical tones. Highcharts [39], a proprietary commercial charting tool, offers data sonification as an add-on. Apple Audio Graphs [5] is an API for Apple application developers to construct an audible representation of the data in charts and graphs, giving BLV users access to valuable data insights. Similarly, Ahmetovic *et al.* [3] developed a web app that supports blind people in exploring graphs of mathematical functions using sonification.

At least one of the following is true for all of the aforementioned systems: (1) they are proprietary and cannot be used outside of their respective products [39]; (2) they are standalone hardware or software applications [3]; (3) they require installation of extra hardware or software [58]; or (4) they are incompatible with existing JavaScript libraries [5]. VoxLens provides sonification as a *separate* open-source library (independent from the VoxLens library) that is customizable and integratable with any JavaScript library or code.

3 DESIGN OF VOXLENS

We present the design and implementation of VoxLens, an open-source JavaScript plug-in that improves the accessibility of online data visualizations. We created VoxLens using a user-centered iterative design process, building on findings and recommendations from prior work [69]. Specifically, our goal was to provide screen-reader users with a comprehensive means of extracting information from online data visualizations, both holistically and in a drilled-down fashion. Holistic exploration involves overall trend, extremum, and labels and ranges for each axis, whereas drilled-down interaction involves examining individual data points [69]. We named our tool VoxLens, combining "vox," meaning "voice" in Latin, and "lens," since it provides a way for screen-reader users to explore, examine, and extract information from online data visualizations. Currently, VoxLens only supports two-dimensional single-series data.

3.1 Interaction Modes

Following the recommendations from prior work [69], our goal was to enable screen-reader users to gain a holistic overview of the data as well as to perform drilled-down explorations. Therefore, we explored three modes of interaction: (1) *Question-and-Answer* mode, where the user verbally interacts with the visualizations; (2) *Summary* mode, where VoxLens verbally offers a summary of the information contained in the visualization; and (3) *Sonification* mode, where VoxLens maps the data in the visualization to a musical scale, enabling listeners to interpret possible data trends or patterns. We iteratively built the features for these modes seeking feedback from screen-reader users through our Wizard-of-Oz studies. VoxLens channels all voice outputs through the user's local screen reader, providing screen-reader users with a familiar and comfortable experience. These three modes of interaction can be activated by pressing their respective keyboard shortcuts (Table 1).

3.1.1 Wizard-of-Oz Studies. Our goal was to gather feedback and identify areas of improvement for the VoxLens features. Therefore, we conducted a Wizard-of-Oz study [21, 35] with five screen-reader users (see Appendix B, Table 7). (For clarity, we prefix the codes for participants in our Wizard-of-Oz studies with "W.") We used the findings from the studies to inform design decisions when iteratively building VoxLens. In our studies, we, the "wizards," simulated the auditory responses from a hypothetical screen reader.

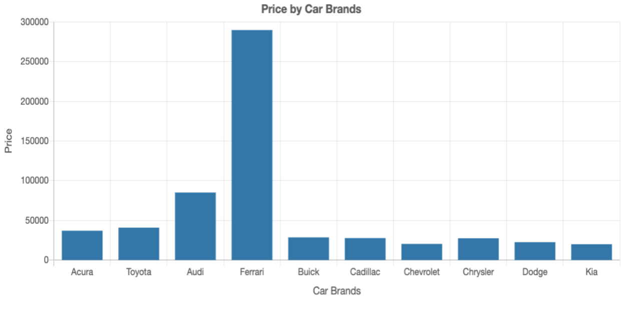


Figure 2: Sample visualization showing price by car brands.

Participants interacted with all of the aforementioned VoxLens modes and were briefly interviewed in a semi-structured manner with open prompts at the end of each of their interactions. Specifically, we asked them to identify the features that they liked and the areas of improvement for each mode. We qualitatively analyzed the data collected from the Wizard-of-Oz studies. We provide our findings from the Wizard-of-Oz studies for each VoxLens mode in its respective section, below.

3.1.2 Question-and-Answer Mode. In *Question-and-Answer* mode, screen-reader users can extract information from data visualizations by asking questions verbally using their microphone. We used the Web Speech API [60] and the P5 Speech library [24] for speech input, removing the need for any additional software or hardware installation by the user. Through manual testing, we found the P5 Speech library to perform quite well in recognizing speech from different accents, pronunciations, and background noise levels. After getting the text from the speech, we used an approximate string matching algorithm from Hall and Dowling [36] to recognize the commands. Additionally, we verified VoxLens' command recognition effectiveness through manual testing, using prior work's [73] data set on natural language utterances for visualizations.

Our Wizard-of-Oz studies revealed that participants liked clear instructions and responses, integration with the user's screen reader, and the ability to query by specific terminologies. They specified that having an interactive tutorial to become familiar with the tool, a help menu to determine which commands are supported, and the ability to include the user's query in the response as key areas of improvement. Therefore, after recognizing the commands and processing their respective responses, VoxLens delivers a single combined response to the user via their screen readers. This approach enables screen-reader users to get a response to multiple commands as one single response. Additionally, we also added each query as feedback in the response (Figure 1). For example, if the user said, "what is the maximum?", the response was, "I heard you ask about the maximum. The maximum is..." If a command was not recognized, the response was, "I heard you say [user input]. Command not recognized. Please try again."

Screen-reader users are also able to get a list of supported commands by asking for the commands list. For example, the user can ask, "What are the supported commands?" to hear all of the commands that VoxLens supports. The list of supported commands, along with their aliases, are presented in Table 2.

3.1.3 Summary Mode. Our Wizard-of-Oz studies, in line with the findings from prior work [69], revealed that participants liked the efficiency and the preliminary exploration of the data. They suggested the information be personalized based on the preferences of each user, but by default, it should only expose the minimum amount of information that a user would need to decide if they want to delve further into the data exploration. To delve further, they commonly seek the title, axis labels and ranges, maximum and minimum data points, and the average in online data visualizations. (The title and axis labels are required configuration options for VoxLens, discussed further in section 3.2.2 below. Axis ranges, maximum and minimum data points, and average are computed by VoxLens.) At the same time, screen-reader users preferred concisely stated information. Therefore, the goal for VoxLens's *Summary* mode was to generate the summary only as a means to providing the foundational holistic information about the visualization, and not as a replacement for the visualization itself. We used the "language of graphics" [10] through a pre-defined sentence template,

	Keyboard Shortcuts
Question-and-Answer Mode	`Modifier Keys + A` / `Modifier Keys + 1`
Summary Mode	`Modifier Keys + S` / `Modifier Keys + 2`
Sonification Mode	`Modifier Keys + M` / `Modifier Keys + 3`
Repeat Instructions	`Modifier Keys + I` / `Modifier Keys + 4`

Table 1: Keyboard shortcuts for VoxLens' interaction modes and preliminary commands. `Modifier Keys` for Windows and MacOS were `Control+Shift` and `Option`, respectively.

Information Type	**Command**	**Aliases**
Extremum	Maximum Minimum	Highest Lowest
Axis Labels and Ranges	Axis Labels Ranges	- -
Statistics	Mean Median Mode Variance Standard Deviation Sum	Average - - - - Total
Individual Data Point	[x-axis label] value	[x-axis label] data
Help	Commands -	Instructions Directions, Help

Table 2: Voice-activated commands for VoxLens' *Question-and-Answer* mode.

identified as *Level 1* by Lundgard *et al.* [55], to decide the sentence structure. Our sentence template was:

```
Graph with title: [title]. The X-axis is [x-axis title]. The Y-axis is [y-axis title] and ranges from [range minimum] to [range maximum]. The maximum data point is [maximum y-axis value] belonging to [corresponding x-axis value], and the minimum data point is [minimum y-axis value] belonging to [corresponding x-axis value]. The average is [average].
```

For example, here is a generated summary of a data visualization depicting the prices of various car brands (Figure 2):

```
Graph with title: Price by Car Brands. The X-axis is car brands. The Y-axis is price and ranges from $0 to $300,000. The maximum data point is $290,000 belonging to Ferrari, and the minimum data point is $20,000 belonging to Kia. The average is $60,000.
```

As noted in prior work [55, 69], the preference for information varies from one individual to another. Therefore, future work can explore personalization options to generate a summarized response that caters to the individual needs of screen-reader users.

Additionally, VoxLens, at present, does not provide information about the overall trend through the *Summary* mode. Such information may be useful for screen-reader users in navigating line graphs [47, 48]. Therefore, work is underway to incorporate trend information in the summarized response generated for line graphs, utilizing the findings from prior work [47, 48].

3.1.4 Sonification Mode.

For *Sonification* mode, our Wizard-of-Oz participants liked the ability to preliminarily explore the data trend. As improvements, participants suggested the ability to identify key information, such as the maximum and the minimum data points. Therefore, VoxLens's sonification mode presents screen-reader users with a sonified response (also known as an "audio graph"

[72]) mapping the data in the visualization to a musical scale. A sonified response enables the listeners to interpret the data trend or pattern and gain a big-picture perspective of the data that is not necessarily achievable otherwise [66]. To generate the sonified response, we utilized Tone.js [56], a JavaScript library that offers a wide variety of customizable options to produce musical notes. Our goal was to enable the listeners to directionally distinguish between data points and to interpret the overall data trend.

Varying tonal frequency is more effective at representing trends than varying amplitude [26, 42]. Therefore, we mapped each data point to a frequency within the range of 130 and 650 Hz based on its magnitude. For example, for the minimum data point the frequency was 130 Hz, for the maximum data point it was 650Hz, and the intermediate data points were assigned values linearly in-between, similar to prior work [19, 61]. Additionally, similar to design choices made by Ohshiro *et al.* [61], we used the sound of a sawtooth wave to indicate value changes along the x-axis. These approaches enabled us to distinctively differentiate between data values directionally, especially values that were only minimally different from each other. We chose this range based on the frequency range of the human voice [6, 58, 75], and by trying several combinations ourselves, finding a setting that was comfortable for human ears. We provide examples of sonified responses in our paper's supplementary materials. Our open-source sonification library is available at https://github.com/athersharif/sonifier.

In our work, we used the three common chart types (bar, scatter, and line) [65], following prior work [69]. All of these chart types use a traditional Cartesian coordinate system. Therefore, VoxLens's sonified response is best applicable to graphs represented using a Cartesian plane. Future work can study sonification responses for graphs that do not employ a Cartesian plane to represent data (*e.g.*, polar plots, pie charts, etc.).

3.2 Usage and Integration

3.2.1 Screen-Reader User. A pain point for screen-reader users when interacting with online data visualizations is that most visualization elements are undiscoverable and incomprehensible by screen readers. In building VoxLens, we ensured that the visualization elements were recognizable and describable by screen readers. Hence, as the very first step, when the screen reader encounters a visualization created with VoxLens, the following is read to users:

```
Bar graph with title: [title]. To listen
to instructions on how to interact with the
graph, press Control + Shift + I or Control +
Shift + 4. Key combinations must be pressed
all together and in order.
```

The modifier keys (`Control + Shift` on Windows, and `Option` on MacOS) and command keys were selected to not interfere with the dedicated key combinations of the screen reader, the Google Chrome browser, and the operating system. Each command was additionally assigned a numeric activation key, as per suggestions from our participants.

When a user presses the key combination to listen to the instructions, their screen reader announces the following:

```
To interact with the graph, press Control +
Shift + A or Control + Shift + 1 all together
and in order. You'll hear a beep sound, after
which you can ask a question such as, ''what
is the average?'' or ''what is the maximum
value in the graph?'' To hear the textual
summary of the graph, press Control + Shift +
S or Control + Shift + 2. To hear the sonified
version of the graph, press Control + Shift
+ M or Control + Shift + 3. To repeat these
instructions, press Control + Shift + I or
Control + Shift + 4. Key combinations must
be pressed all together and in order.
```

At this stage, screen-reader users can activate question-and-answer mode, listen to the textual summary, play the sonified version of the data contained in the visualization, or hear the instructions again. Activating the question-and-answer mode plays a beep sound, after which the user can ask a question in a free-form manner, without following any specific grammar or sentence structure. They are also able to ask for multiple pieces of information, in no particular order. For example, in a visualization containing prices of cars by car brands, a screen-reader user may ask:

```
Tell me the mean, maximum, and standard
deviation.
```

The response from VoxLens would be:

```
I heard you asking about the mean, maximum,
and standard deviation. The mean is $60,000.
The maximum value of price for car brands is
$290,000 belonging to Ferrari. The standard
deviation is 30,000.
```

Similarly, users may choose to hear the textual summary or play the sonified version, as discussed above.

3.2.2 Visualization Creators. Typically, the accessibility of online data visualizations relies upon visualization creators and their knowledge and practice of accessibility standards. When an alternative text description is not provided, the visualization is useless to screen-reader users. In cases where alternative text *is* provided, the quality and quantity of the text is also a developer's choice, which may or may not be adequate for screen-reader users. For example, a common unfortunate practice is to use the title of the visualization as its alternative text, which helps screen-reader users in understanding the topic of the visualization but does not help in understanding the content contained *within* the visualization. Therefore, VoxLens is designed to reduce the burden and dependency on developers to make accessible visualizations, keeping the interaction consistent, independent of the visualization library used. Additionally, VoxLens is engineered to require only a single line of code, minimizing any barriers to its adoption (Figure 1).

VoxLens supports the following configuration options: “x” (key name of the independent variable), “y” (key name of the dependent variable), “title” (title of the visualization), “xLabel” (label for x-axis), and “yLabel” (label for y-axis). “x,” “y,” and “title” are required parameters, whereas the “xLabel” and “yLabel” are optional and default to the key names of “x” and “y,” respectively. VoxLens allows visualization creators to set the values of these configuration options, as shown in Figure 1.

3.3 Channeling VoxLens' Output to Screen Readers

One of the challenges we faced was to channel the auditory response from VoxLens to the screen reader of the user. As noted by our participants during Wizard-of-Oz studies, screen-reader users have unique preferences for their screen readers, including the voice and speed of the speech output. Therefore, it was important for VoxLens to utilize these preferences, providing screen-reader users with a consistent, familiar, and comfortable experience. To relay the output from VoxLens to the screen reader, we created a temporary `div` element that was only visible to screen readers, positioning it off-screen, following WebAIM's recommendations [78].

Then, we added the appropriate Accessible Rich Internet Applications (ARIA) attributes [77] to the temporary element to ensure maximum accessibility. ARIA attributes are a set of attributes to make web content more accessible to people with disabilities. Notably, we added the "aria-live" attribute, allowing screen readers to immediately announce the query responses that VoxLens adds to the temporary element. For MacOS, we had to additionally include the "role" attribute, with its value set to "alert." This approach enabled VoxLens to promptly respond to screen-reader users' voice-activated commands using their screen readers. After the response from VoxLens is read by the screen reader, a callback function removes the temporary element from the HTML tree to avoid overloading the HTML Document Object Model (DOM).

3.4 Additional Implementation Details

At present, VoxLens only supports two-dimensional data, containing one independent and one dependent variable, as only the interactive experiences of screen-reader users with two-dimensional data visualizations are well-understood [69]. To support data dimensions greater than two, future work would need to investigate the interactive experiences of screen-reader users with *n*-dimensional data visualizations. VoxLens is customizable and engineered to support additional modifications in the future.

VoxLens relies on the Web Speech API [60], and is therefore only fully functional on browsers with established support for the API such as Google Chrome. JavaScript was naturally our choice of programming language for VoxLens, as VoxLens is a plug-in for JavaScript visualization libraries. Additionally, we used EcmaScript [37] to take advantage of modern JavaScript features such as destructured assignments, arrow functions, and the spread operator. We also built a testing tool to test VoxLens on data visualizations, using the React [45] framework as the user-interface framework and Node.js [28] as the back-end server—both of which also use JavaScript as their underlying programming language. Additionally, we used GraphQL [29] as the API layer for querying and connecting with our Postgres [34] database, which we used to store data and participants' interaction logs.

Creating a tool like VoxLens requires significant engineering effort. Our GitHub repository at https://github.com/athersharif/voxlens has a total of 188 commits and 101,104 lines of developed code, excluding comments. To support testing VoxLens on various operating systems and browsers with different screen readers, we collected 30 data sets of varying data points, created their visualizations using Google Charts, D3, and ChartJS, integrated VoxLens with each of them, and deployed a testing website on our server. The testing website was instrumental in ensuring the correct operation of VoxLens under various configurations, bypassing the challenges of setting up a development environment for testers.

3.5 Conflicts with Other Plug-ins

To the best of our knowledge, two kinds of conflicts are possible with VoxLens: *key combination* conflicts and *ARIA attribute* conflicts. As mentioned in section 4.2.1, we selected key combinations to avoid conflicts with the dedicated combinations of the screen reader, the Google Chrome browser, and the operating system. However, it is possible that some users might have external plug-ins using key combinations that would conflict with those from VoxLens. Future work could build a centralized configuration management system, enabling users to specify their own key combinations.

VoxLens modifies the "aria-label" attribute of the visualization container element to describe the interaction instructions for VoxLens, as mentioned in section 4.2.1. It is possible that another plug-in may intend to modify the "aria-label" attribute as well, in which case the execution order of the plug-ins will determine which plug-in achieves the final override. The execution order of the plug-ins depends on several external factors [63], and is, unfortunately, a common limitation for any browser plug-in. However, VoxLens does not affect the "aria-labelledby" attribute, allowing other systems to gracefully override the "aria-label" attribute set by VoxLens, as this attribute takes precedence over the "aria-label" attribute in the accessibility tree. Future iterations of VoxLens will attempt to ensure that VoxLens achieves the last execution order and that the ARIA labels set by other systems are additionally relayed to screen-reader users.

It is important to note that VoxLens's sonification library is supplied independently from the main VoxLens plug-in and does not follow the same limitations. Our testing did not reveal any conflicts of the sonification library with other plug-ins.

4 EVALUATION METHOD

We evaluated the performance of VoxLens using a mixed-methods approach. Specifically, we conducted an online mixed-factorial experiment with screen-reader users to assess VoxLens quantitatively. Additionally, we conducted follow-up interviews with our participants for a qualitative assessment of VoxLens.

4.1 Participants

Our participants (see Appendix A, Table 6) were 22 screen-reader users, recruited using word-of-mouth, snowball sampling, and email distribution lists for people with disabilities. Nine participants identified as women and 13 as men. Their average age was 45.3 years (*SD*=16.8). Twenty participants had complete blindness and two participants had partial blindness; nine participants were blind since birth, 12 lost vision gradually, and one became blind due to a brain tumor. The highest level of education attained or in pursuit was a doctoral degree for two participants, a Master's degree for seven participants, a Bachelor's degree for eight participants, and a high school diploma for the remaining five participants. Estimated computer usage was more than 5 hours per day for 12 participants, 2-5 hours per day for eight participants, and 1-2 hours per day for two participants. The average frequency of interacting with online data visualizations was over two visualizations per day, usually in the context of news articles, blog posts, and social media.

For the task-based experiment and questionnaire, participants were compensated with a $20 Amazon gift card for 30-45 minutes of their time. For the follow-up interview, they were compensated $10 for 30 minutes of their time. No participant was allowed to partake in the experiment more than once.

4.2 Apparatus

We conducted our task-based experiment online using a user study platform that we created with the JavaScript React framework [45]. We tested our platform with screen-reader users and ourselves, both with and without a screen reader, ensuring maximum and proper accessibility measures. We deployed the experiment platform as a website hosted on our own server.

We analyzed the performance of VoxLens comparing the data collected from our task-based experiments with that from prior work [69]. To enable a fair comparison to this prior work, we used the same visualization libraries, visualization data set, question categories, and complexity levels. The visualization libraries (Google Charts, ChartJS, and D3) were chosen based on the variation in their underlying implementations as well as their application of accessibility measures. Google Charts utilizes SVG elements to generate the visualization and appends a tabular representation of the data for screen-reader users, by default; D3 also makes use of SVG elements but does not provide a tabular representation; ChartJS uses HTML Canvas to render the visualization as an image and relies on the developers to add alternative text ("alt-text") and Accessible Rich Internet Applications ("ARIA") attributes [77]. Therefore, each of these visualization libraries provides a different experience for screen-reader users, as highlighted in prior work [69].

We provide all of the visualizations and data sets used in this work in this paper's supplementary materials. Readers can reproduce these visualizations using the supplementary materials in conjunction with the source code and examples presented in our open-source GitHub repository. We implemented the visualizations following the WCAG 2.0 guidelines [17] in combination with the official accessibility recommendations from the visualization libraries. For ChartJS, we added the "role" and "aria-label" attributes to the "canvas" element. The "role" attribute had the value of "img," and the "aria-label" was given the value of the visualization title, as per the official documentation from ChartJS developers [18]. We did not perform any accessibility scaffolding for Google Charts and D3 visualizations, as these visualizations rely on a combination of internal implementations and the features of SVG for accessibility. Our goal was to replicate an accurate representation of how these visualizations currently exist on the Web.

Recent prior work [46] has reported that the non-visual questions that users ask from graphs mainly comprise compositional questions, similar to the findings from Brehmer and Munzner's task topology [14]. Therefore, our question categories comprised one "Search" action (*lookup and locate*) and two "Query" actions (*identify and compare*), similar to prior work [13]. The categories, in ascending order of difficulty, were: (1) *Order Statistics* (extremum); (2) *Symmetry Comparison* (comparison of data points); and (3) *Chart Type-Specific Questions* (value retrieval for bar charts, trend summary for line charts, and correlation for scatter plots). As in prior work [69], all questions were multiple-choice questions with four choices: the correct answer, two incorrect answers, and the option for "Unable to extract information." The order of the four choices was randomized per trial.

4.3 Procedure

The study was conducted online by participants without direct supervision. The study comprised six stages. The first stage displayed the study purpose, eligibility criteria, and the statement of IRB approval. In the second stage, the participants were asked to fill out a pre-study questionnaire to record their demographic information, screen-reader software, vision-loss level, and diagnosis (see Appendix A, Table 6). Additionally, participants were asked about their education level, daily computer usage, and their frequency of interacting with visualizations.

In the third stage, participants were presented with a step-by-step interactive tutorial to train and familiarize themselves with the modes, features, and commands that VoxLens offers. Additionally, participants were asked questions at each step to validate their understanding. On average, the tutorial took 12.6 minutes (*SD*=6.8) to complete. Upon successful completion of the tutorial, participants were taken to the fourth stage, which displayed the instructions for completing the study tasks.

In the fifth stage, each participant was given a total of nine tasks. For each task, participants were shown three Web pages: Page 1 contained the question to explore, page 2 displayed the question and visualization, and page 3 presented the question with a set of four multiple-choice responses. Figure 3 shows the three pages of an example task. After the completion of the tasks, participants were asked to fill out the NASA-TLX [38] survey in the last stage. An entire study session ranged from 30-45 minutes in duration.

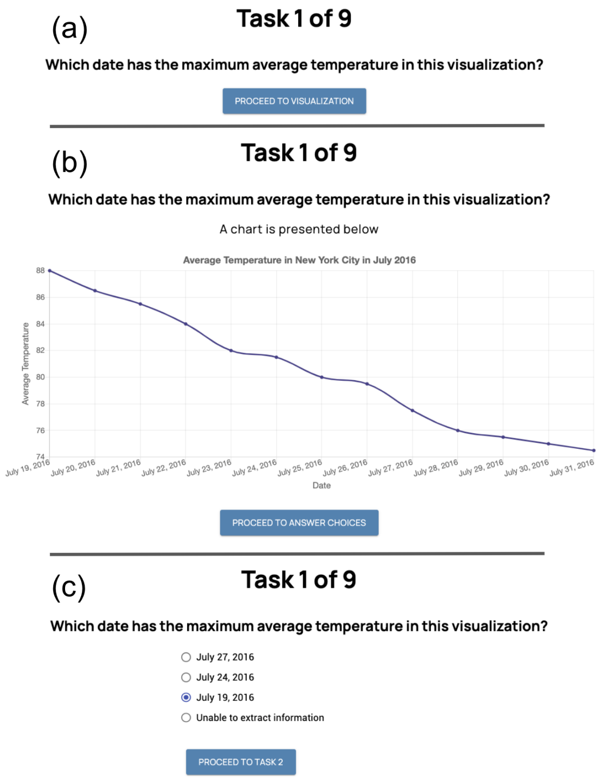
4.4 Design & Analysis

The experiment was a $2 \times 3 \times 3 \times 3$ mixed-factorial design with the following factors and levels:

- *VoxLens* (*VX*), between-Ss.: {yes, no}
- *Visualization Library* (*VL*), within-Ss.: {ChartJS, D3, Google Charts}
- *Data Complexity* (*CMP*), within-Ss.: {Low, Medium, High}
- *Question Difficulty* (*DF*), within-Ss.: {Low, Medium, High}

For the screen-reader users who did not use VoxLens (*VX=no*), we used prior work's data [69] (*N*=36) as a baseline for comparison.

Our two dependent variables were *Accuracy of Extracted Information* (AEI) and *Interaction Time* (IT). We used a dichotomous representation of *AEI* (i.e., "inaccurate" or 0 if the user was unable to answer the question correctly, and "accurate" or 1 otherwise) for our analysis. We used a mixed logistic regression model [32] with the above factors, interactions with VoxLens, and a covariate to control for *Age*. We also included $Subject_r$ as a random factor to account for repeated measures. The statistical model was therefore $AEI \leftarrow VX + VX{\times}VL + VX{\times}CMP + VX{\times}DF + Age + Subject_r$. We did not include factors for VL, CMP, or DF because our research



(a)

Task 1 of 9

Which date has the maximum average temperature in this visualization?

PROCEED TO VISUALIZATION

(b)

Task 1 of 9

Which date has the maximum average temperature in this visualization?

A chart is presented below

PROCEED TO ANSWER CHOICES

(c)

Task 1 of 9

Which date has the maximum average temperature in this visualization?

○ July 27, 2016
○ July 24, 2016
● July 19, 2016
○ Unable to extract information

PROCEED TO TASK 2

Figure 3: Participants were shown three pages for each task. (a) Page 1 presented the question to explore. (b) Page 2 displayed the same question and a visualization. (c) Page 3 showed the question again with a set of four multiple choice responses.

questions centered around VoxLens (VX) and our interest in these factors only extended to their possible interactions with VoxLens.

For *Interaction Time* (IT), we used a linear mixed model [30, 54] with the same model terms as for *AEI*. *IT* was calculated as the total time of the screen reader's focus on the visualization element. Participants were tested over three *Visualization Library* × *Complexity* ($VL \times CMP$) conditions, resulting in 3×3 = 9 trials per participant. With 21 participants, a total of 21×9 = 189 trials were produced and analyzed for this study. One participant, who was unable to complete the tutorial, was excluded from the analysis.

4.5 Qualitative and Subjective Evaluation

To qualitatively assess the performance of VoxLens, we conducted follow-up interviews with six screen-reader users, randomly selected from our pool of participants who completed the task-based experiment. Similar to prior work [80], we ceased recruitment of participants once we reached saturation of insights.

To analyze our interviews, we used thematic analysis [12] guided by a semantic approach [62]. We used two interviews to develop an initial set of codes, resulting in a total of 23 open codes. Each interview transcript was coded by three researchers independently, and disagreements were resolved through mutual discussions. As suggested by Lombard *et al.* [51], we calculated inter-rater reliability (IRR) using pairwise percentage agreement together with Krippendorff's α [49]. To calculate pairwise percentage agreement, we calculated the average pairwise agreement among the three rater pairs across observations. Our pairwise percentage agreement was 94.3%, showing a high agreement between raters. Krippendorff's α was calculated using ReCal [31] and found to be 0.81, indicating a high level of reliability [50].

In addition to conducting follow-up interviews, we administered the NASA-TLX survey [38] with all participants (N=21) to assess the perceived workload of VoxLens.

5 RESULTS

We present our experiment results using the *Accuracy of Extracted Information* (AEI) and *Interaction Time* (IT) for screen-reader users with and without VoxLens. We also present our interview results and the subjective ratings from the NASA-TLX questionnaire [38].

5.1 Accuracy of Extracted Information

Our results show a significant main effect of *VoxLens (VX)* on *AEI* ($\chi^2(1, N{=}57){=}38.16, p{<}.001$, Cramer's V=.14), with VoxLens users achieving 75% accuracy (SD = 18.0%) and non-VoxLens users achieving only 34% accuracy (SD = 20.1%). This difference constituted a 122% improvement due to VoxLens.

By analyzing the *VoxLens (VX)* × *Visualization Library (VL)* interaction, we investigated whether changes in *AEI* were proportional across visualization libraries for participants in each VoxLens group. The $VX \times VL$ interaction was indeed statistically significant ($\chi^2(4, N{=}57){=}82.82, p{<}.001$, Cramer's V=.20). This result indicates that *AEI* significantly differed among visualization libraries for participants in each VoxLens group. Figure 4 and Table 3 show *AEI* percentages for different visualization libraries for each VoxLens group. Additionally, we report our findings in Table 4.

Prior work [69] has reported a statistically significant difference between screen-reader users (SRU) and non-screen-reader users (non-SRU) in terms of *AEI*, attributing the difference to the inaccessibility of online data visualizations. We conducted a second analysis, investigating whether *AEI* was different between screen-reader users who used VoxLens and non-screen-reader users, to extract information from online data visualizations. Specifically, we investigated the effect of *SRU* on *AEI* but did not find a statistically significant effect ($p \approx .077$). This result itself does not provide evidence in support of VoxLens closing the access gap between the two user groups; further experimentation is necessary to confirm or refute this marginal result. In light of VoxLens's other benefits, however, this is an encouraging trend.

5.2 Interaction Time

Our preliminary analysis showed that the interaction times were conditionally non-normal, determined using Anderson-Darling [4] tests of normality. To achieve normality, we applied logarithmic transformation prior to analysis, as is common practice for time measures [9, 41, 53]. For ease of interpretation, plots of interaction times are shown using the original non-transformed values.

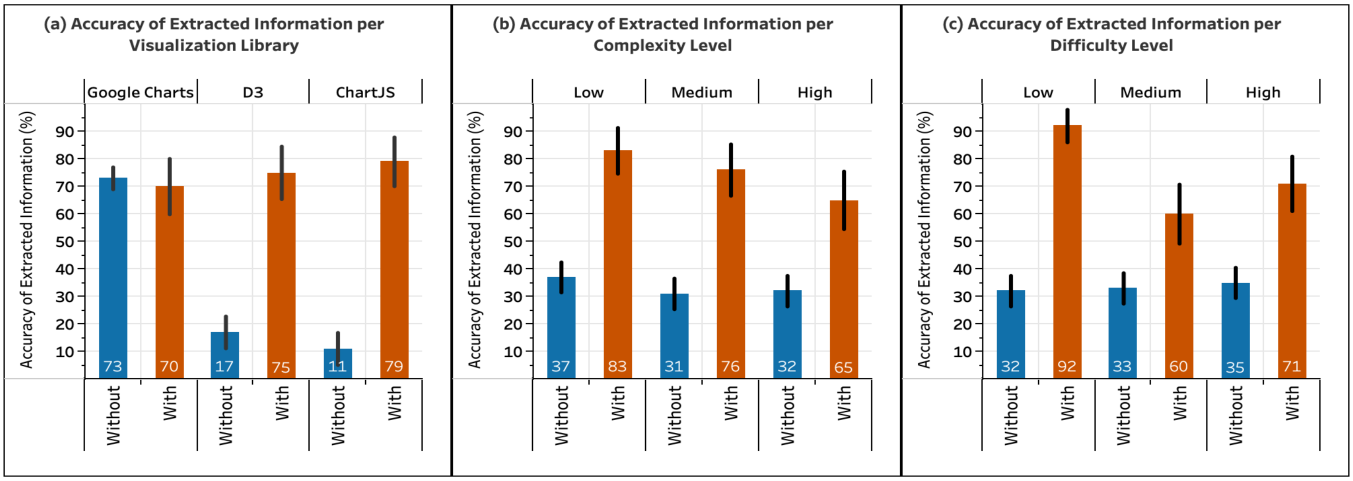


Figure 4: Accuracy of Extracted Information (*AEI*), as a percentage, for screen-reader users without (*N*=36) and with (*N*=21) VoxLens, by (a) Visualization Library, (b) Complexity Level, and (c) Difficulty Level. The percentage represents the "accurate" answers. Therefore, higher is better. Error bars represent mean ± standard deviation.

	Without VoxLens			*With VoxLens*		
	N	*AA*	*AA* (%)	*N*	*AA*	*AA* (%)
Overall	324	109	34%	189	141	75%
Visualization Library (VL)						
ChartJS	108	12	11%	63	50	79%
D3	108	18	17%	63	47	75%
Google Charts	108	79	73%	63	44	70%
Complexity Level (CMP)						
Low	108	40	37%	63	52	83%
Medium	108	34	31%	63	48	76%
High	108	35	32%	63	41	65%
Difficulty Level (DF)						
Low	108	35	32%	63	58	92%
Medium	108	36	33%	63	38	60%
High	108	38	35%	63	45	71%

Table 3: Numerical results for the $N = 513$ questions asked of screen-reader users with and without VoxLens for each level of *Visualization Library*, *Complexity Level*, and *Difficulty Level*. N is the total number of questions asked, *AA* is the number of "accurate answers," and *AA*(%) is the percentage of "accurate answers."

VoxLens (VX) had a significant main effect on *Interaction Time* (IT) (F(4,54)=12.66, p<.05, η_p^2=.19). Specifically, the average *IT* for non-VoxLens users was 84.6 seconds (*SD*=75.2). For VoxLens users, it was 54.1 seconds (*SD*=21.9), 36% lower (faster) than for participants without VoxLens.

The $VX \times VL$ and $VX \times DF$ interactions were both significant (F(4,444)=33.89, p<.001, η_p^2=.23 and F(444)=14.41, p<.001, η_p^2=.12, respectively). Figure 5 shows interaction times across different visualization libraries, difficulty levels, and complexity levels for VoxLens group. For VoxLens users, all three visualization libraries

	N	χ^2	p	Cramer's V
VX(VoxLens)	57	38.16	< .001	.14
$VX \times VL$	57	82.82	< .001	.20
$VX \times CMP$	57	8.90	.064	.07
$VX \times DF$	57	17.95	.001	.09
Age	57	3.58	.058	.04

Table 4: Summary results from 57 screen-reader users with (N=21) and without (N=36) VoxLens. "VL" is the visualization library, "CMP" is data complexity, and "DF" is question difficulty. Cramer's V is a measure of effect size [25]. All results are statistically significant ($p < .05$) or marginal ($.05 \leq p < .10$).

	df_n	df_d	F	p	η_p^2
VX (VoxLens)	4	54	12.66	.001	.19
$VX \times VL$	4	444	33.89	< .001	.23
$VX \times CMP$	4	444	1.85	.118	.02
$VX \times DF$	4	444	14.41	< .001	.12
Age	4	54	5.03	.029	.09

Table 5: Summary results from 57 screen-reader participants with (N=21) and without (N=36) VoxLens using a linear mixed model [30, 54]. "VL" is the visualization library, "CMP" is data complexity, and "DF" is question difficulty. Partial eta-squared (η_p^2) is a measure of effect size [20]. All results are statistically significant ($p < .05$) except $VX \times CMP$.

resulted in almost identical interaction times. Figure 5 portrays larger variations in interaction times for users who did not use VoxLens (data used from prior work [69]) compared to VoxLens users. We attribute these observed differences to the different underlying implementations of the visualization libraries.

We investigated the effects of *Age* on IT. *Age* had a significant effect on IT (F(1,54)=5.03, p<.05, η_p^2 =.09), indicating that IT differed significantly across the ages of our participants, with participants aged 50 or older showing higher interaction times by about 7% compared to participants under the age of 50. Table 8 (Appendix C) shows the average IT for each age range by VoxLens group. Additionally, we report our findings in Table 5.

Similar to our exploration of investigating the effect of screen-reader users (*SRU*) on *AEI*, we examined the main effect of *SRU* on IT. Our results show that *SRU* had a significant effect on IT (F(4,54)=48.84, p<.001, η_p^2=.48), with non-screen-reader users performing 99% faster than VoxLens users.

5.3 Interview Results

To assess VoxLens qualitatively, we investigated the overall experiences of our participants with VoxLens, the features they found helpful, the challenges they faced during the interaction, and the improvements and future features that could enhance the performance of VoxLens. We identified five main results from analyzing our participants' feedback about VoxLens: (1) a positive step forward in making online data visualizations accessible, (2) interactive dialogue is one of the "top" features, (3) sonification helps in "visualizing" data, (4) data summary is a good starting point, and (5) one-size-fits-all is not the optimal solution. We present each of these in turn.

5.3.1 A Positive Step Forward in Making Online Data Visualizations Accessible. All participants found VoxLens to be an overall helpful tool to interact with and quickly extract information from online data visualizations. For example, S1 and S3 expressed their excitement about VoxLens:

> I have never been able to really interact with graphs before online. So without the tool, I am not able to have that picture in my head about what the graph looks like. I mean, like, especially when looking up news articles or really any, sort of, like, social media, there's a lot of visual representations and graphs and pictographs that I don't have access to so I could see myself using [VoxLens] a lot. The tool is really great and definitely a positive step forward in creating accessible graphs and data. (S1)

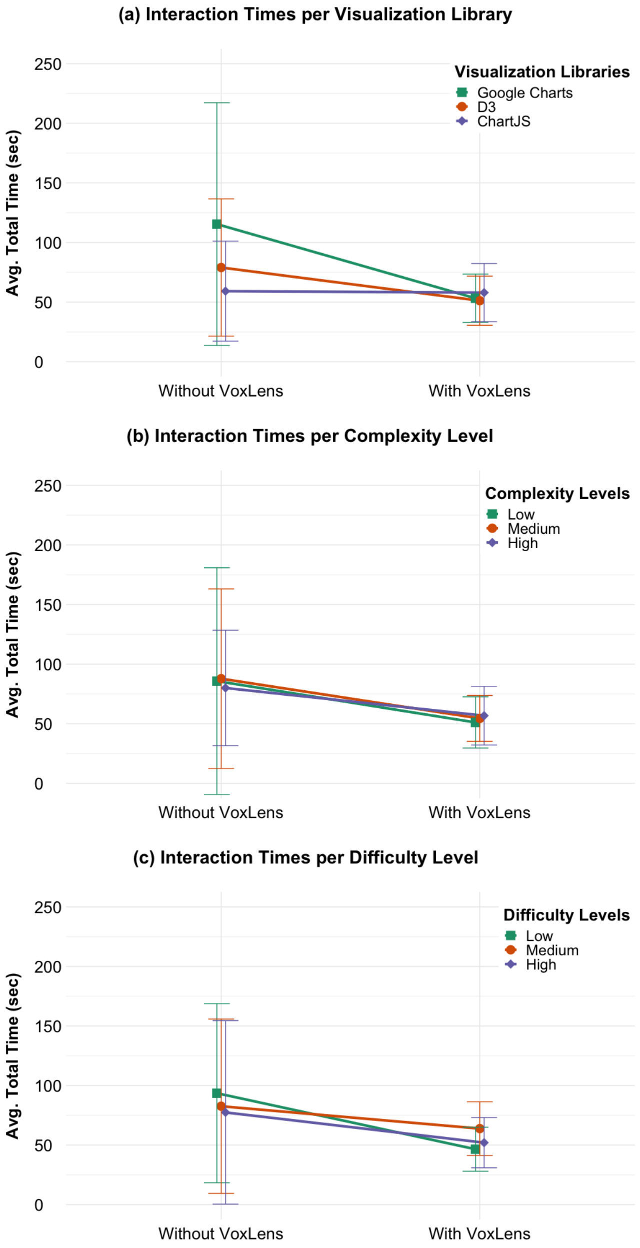


Figure 5: Interaction Times (*IT*), in seconds, for screen-reader users without (*N*=36) and with (*N*=21) VoxLens by (a) Visualization Library (*VL*), (b) Data Complexity Level (*CMP*), and (c) Question Difficulty Level (*DF*). Error bars represent mean ± standard deviation. Lower is better (faster).

> `Oh, [VoxLens] was outstanding. It's definitely a great way to visualize the graphs if you can't see them in the charts. I mean, it's just so cool that this is something that allows a blind person to access a graph and a chart and be able to parse data from it.` (S3)

Participants highlighted that VoxLens contributes to bridging the access gap between screen-reader- and non-screen-reader users. As S4 said:

> `So, as a sighted person looks at a graph and as they can tell where the peak is or which one has the most or whatever, we want to be able to do that quickly as well. And even if there is a text description under the graph, and I've not seen that very much, you have to read through everything to find a certain piece of information that you're looking for. [Using VoxLens], I can find out specific pieces of information without having to read an entire page of text.` (S4)

Additionally, participants identified that VoxLens enables them to quickly extract information from online data visualizations. S5 shared his experiences:

> `Again, you know, [VoxLens] helps you find data a little bit quicker than navigating with a screen reader, and it'll give you a brief idea of what the data is about before you start digging deeper into it.` (S5)

The findings from our first result show that VoxLens contributes to reducing the access gap for screen-reader users, and is a positive step forward, enabling screen-reader users to interact with and explore online data visualizations.

5.3.2 Interactive Dialogue is One of the "Top" Features. Similar to our first finding, all the participants found the question-and-answer mode of VoxLens a fast and efficient way to extract information from online data visualizations. S2 considered the question-and-answer mode as one of the key features of VoxLens:

> `So I believe that one of the really top features is, kind of, interactive dialogue.` (S2)

Similarly, S1 found the question-and-answer mode a fast and reliable way to extract information, requiring "a lot less brain power." She said:

> `I especially liked the part of the tool where you can ask it a question and it would give you the information back. I thought it was brilliant actually. I felt like being able to ask it a question made everything go a lot faster and it took a lot less brain power I think. I felt really confident about the answers that it was giving back to me.` (S1)

S3 noted the broader utility and applicability of the question-and-answer mode:

> `The voice activation was very, very neat. I'm sure it could come in handy for a variety of uses too. I definitely enjoyed that feature.` (S3)

S5 faced some challenges in activating the right command but was able to learn the usage of the question-and-answer mode in a few tries:

> You know, sometimes the word was wrong and I think it says something like, it didn't understand, but basically eventually I got it right. (S5)

Our second finding indicates that VoxLens' question-and-answer mode is a fast, efficient, and reliable way for screen-reader users to extract information. Additionally, the feedback from the question-and-answer mode assists screen-reader users to resolve the challenges by themselves within a few tries.

5.3.3 *Sonification Helps in "Visualizing" Data.*

Our third result reveals that our participants found sonification helpful in understanding general trends in the data. Specifically, participants were able to infer whether an overall trend was increasing or decreasing, obtaining holistic information about the data. S2 said:

> The idea of sonification of the graph could give a general understanding of the trends. The way that it could summarize the charts was really nice too. The sonification feature was amazing. (S2)

S1, who had never used sonification before, expressed her initial struggles interpreting a sonified response but was able to "visualize" the graph through sonification within a few tries. She said:

> The audio graph... I'd never used one before, so I kind of struggled with that a little bit because I wasn't sure if the higher pitch meant the bar was higher up in the graph or not. But being able to visualize the graph with this because of the sound was really helpful. (S1)

Overall, our third result shows that sonification is a helpful feature for screen-reader users to interact with data visualizations, providing them with holistic information about data trends.

5.3.4 *Data Summary is a Good Starting Point.*

In keeping with findings from prior work [69], our fourth finding indicates that screen-reader users first seek to obtain a holistic overview of the data, finding a data summary to be a good starting point for visualization exploration. The summary mode of VoxLens enabled our participants to quickly get a "general picture" of the data. S1 and S4 expressed the benefits of VoxLens' summary mode:

> I thought the summary feature was really great just to get, like, a general picture and then diving deeper with the other features to get a more detailed image in my head about what the graphs look like. (S1)

> So, um, the summary option was a good start point to know, okay, what is, kind of, on the graph. (S4)

Our fourth result indicates that VoxLens' summary mode assisted screen-reader users to holistically explore the online data visualizations, helping them in determining if they want to dig deeper into the data.

5.3.5 *One-Size-Fits-All Is Not the Optimal Solution.*

To enhance the usability of and interaction experience with VoxLens, our participants identified the need to cater to the individual preferences of the screen-reader users. For example, S3 recognized the need to have multiple options to "play" with the sonified response:

> So I was just thinking maybe, you know, that could be some sort of option or like an alternate way to sonify it. Perhaps having an option to do it as continuous cause I noticed, like, they were all discrete. 'Cause sometimes, you know, it's just preference or that could be something that could add some usability. It's just some little things to maybe play with or to maybe give an option or something. (S3)

Similarly, S4 was interested in VoxLens predicting what she was going to ask using artificial intelligence (A.I.). She said:

> You know, I think that [VoxLens] would need a lot more artificial intelligence. It could be a lot [more] intuitive when it comes to understanding what I'm going to ask. (S4)

Additionally, S2 suggested adding setting preferences for the summary and the auditory sonified output:

> [Summary mode] could eventually become a setting preference or something that can be disabled. And you, as a screen-reader user, could not control the speed of the [sonification] to you. To go faster or to go slower, even as a blind person, would be [helpful]. (S2)

Our findings indicate that a one-size-fits-all solution is not optimal and instead, a personalizable solution should be provided, a notion supported by recent work in ability-based design [79]. We are working to incorporate the feedback and suggestions from our participants into VoxLens.

5.4 Subjective Workload Ratings

We used the NASA Task Load Index (TLX) [38] workload questionnaire to collect subjective ratings for VoxLens. The NASA-TLX instrument asks participants to rate the workload of a task on six scales: mental demand, physical demand, temporal demand, performance, effort, and frustration. Each scale ranges from low (1) to high (20). We further classified the scale into four categories for a score x: *low* ($x < 6$), *somewhat low* ($6 \leq x < 11$), *somewhat high* ($11 \leq x < 16$), and *high* ($16 \leq x$). Our results indicate that VoxLens requires low physical- (M=3.4, SD=3.3) and temporal demand (M=5.7, SD=3.8), and has high perceived performance (M=5.6, SD=5.6). Mental demand (M=7.8, SD=4.4), effort (M=9.9, SD=6.1), and frustration (M=8.3, SD=6.6) were somewhat low.

Prior work [69], which is the source of our data for screen-reader users who did not use VoxLens, did not conduct a NASA-TLX survey with their participants. Therefore, a direct workload comparison is not possible. However, the subjective ratings from our study could serve as a control for comparisons in future work attempting to make online visualizations accessible for screen-reader users.

6 DISCUSSION

In this work, we created VoxLens, an interactive JavaScript plug-in to make online data visualizations more accessible to screen-reader users. This work has been guided by the recommendations and findings from prior work [69] that highlight the barriers screen-reader users face in accessing the information contained in online data visualizations. In creating VoxLens, we sought to improve the accessibility of online data visualizations by making them discoverable and comprehensible to screen readers, and by enabling screen-reader users to explore the data both holistically and in a drilled-down manner. To achieve this, we designed three modes of VoxLens: (1) *Question-and-Answer* mode; (2) *Summary* mode, and (3) *Sonification* mode. Our task-based experiments show that screen-reader users extracted information 122% more accurately and spent 36% less time when interacting with online data visualizations using VoxLens than without. Additionally, we observed that screen-reader users utilized VoxLens uniformly across all visualization libraries that were included in our experiments, irrespective of the underlying implementations and accessibility measures of the libraries, achieving a consistent interaction.

6.1 Simultaneous Improvement In Accuracy and Interaction Times

Prior work [69] has reported that due to the inaccessibility of online data visualizations, screen-reader users extract information 62% less accurately than non-screen-reader users. We found that VoxLens improved the accuracy of information extraction of screen-reader users by 122%, reducing the information extraction gap between the two user groups from 62% to 15%. However, in terms of interaction time, while VoxLens reduced the gap from 211% to 99%, the difference is still statistically significant between non-screen-reader and VoxLens users. Non-screen-reader users utilize their visual system's power to quickly recognize patterns and extrapolate information from graphs, such as overall trends and extrema [57]. In contrast, screen-reader users rely on alternative techniques, such as sonification, to understand data trends. However, hearing a sonified version of the data can be time-consuming, especially when the data cardinality is large, contributing to the difference in the interaction times between the two user groups. Additionally, issuing a voice command, pressing a key combination, and the duration of the auditory response can also contribute to the observed difference. However, it is worth emphasizing that screen-reader users who used VoxLens improved their interaction time by 36% while also increasing their accuracy of extracted information by 122%. In other words, VoxLens users became both faster and more accurate, a fortunate outcome often hard to realize in human performance studies due to speed-accuracy tradeoffs.

6.2 Role of Voice-Recognition Technology

For screen-reader users who used VoxLens, 75% (N=141) of the answers were correct and 11% (N=20) were incorrect. Our participants were unable to extract the answers from the remaining 15% (N=28) of the questions. Further exploration revealed that among the 25% (N=48) of the questions that were not answered correctly, 52% (N=25) involved symmetry comparison. Symmetry comparison requires value retrieval of multiple data points and relies on the effectiveness of voice-recognition technology. Additional investigation showed that to answer the questions in our experiment tasks, screen-reader users utilized the *Question-and-Answer* mode 71.9% of the time, compared to the *Summary* (22.5%) and *Sonification* (5.5%) modes. Out of the 71.9% *Question-and-Answer* mode usage, VoxLens accurately recognized and responded to commands 49.9% of the time; 34% of the time VoxLens was unable to accurately parse the speech input, and the remaining 16.1% of the time VoxLens received commands that were not supported (*e.g.,* "correlation coefficient"). VoxLens uses the Web Speech API [60] for recognizing voice commands. While the Web Speech API is a great leap forward in terms of speech-input and text-to-speech output features [60], it is still an experimental feature with limited performance of about 70% accuracy [64]. Therefore, future work could evaluate the performance of VoxLens with alternatives to the Web Speech API.

6.3 Qualitative Assessment of VoxLens

All six screen-reader users we interviewed expressed that VoxLens significantly improved their current experiences with online data visualizations. Participants showed their excitement about VoxLens assisting them in "visualizing" the data and in extracting information from important visualizations, such as the ones portraying COVID-19 statistics. Furthermore, some of our participants highlighted that VoxLens reduces the access gap between screen-reader- and non-screen-reader users. For example, S4 mentioned that with the help of VoxLens, she was able to "*find out specific pieces of information without having to read an entire page of text*," similar to how a "sighted person" would interact with the graph. Additionally, our participants found VoxLens *"pretty easy," "meaningful," "smooth,"* and *"intuitive,"* without requiring a high mental demand.

6.4 VoxLens is a Response to Call-to-Action for Inclusive Data Visualizations

Taking these findings together, VoxLens is a response to the call-to-action put forward by Marriott *et al.* [57] that asserts the need to improve accessibility for disabled people disenfranchised by existing data visualizations and tools. VoxLens is an addition to the tools and systems designed to make the Web an equitable place for screen-reader users, aiming to bring their experiences on a par with that of non-screen-reader users. Through effective advertisement, and by encouraging developers to integrate VoxLens into the codebase of visualization libraries, we hope to broadly expand the reach and impact of VoxLens. Additionally, through collecting anonymous usage logs (VoxLens modes used, commands issued, and responses issued) and feedback from users—a feature already implemented in VoxLens—we aspire to continue improving the usability and functionality of VoxLens for a diverse group of users.

7 LIMITATIONS & FUTURE WORK

At present, VoxLens is limited to two-dimensional data visualizations with a single series of data. Future work could study the experiences of screen-reader users with n-dimensional data visualizations and multiple series of data, and extend the functionality of VoxLens based on the findings. Additionally, VoxLens is currently only fully functional on Google Chrome, as the support for the Web Speech API's speech recognition is currently limited to Google

Chrome. Future work could consider alternatives to the Web Speech API that offer cross-browser support for speech recognition.

Our findings showed that some of our participants preferred to have the ability to control the speed, frequency, and waveform of the sonified response. Therefore, future work could extend the functionality of VoxLens by connecting it to a centralized configuration management system, enabling screen-reader users to specify their preferences. These preferences could then be used to generate appropriate responses, catering to the individual needs of screen-reader users.

8 CONCLUSION

We presented VoxLens, a JavaScript plug-in that improves the accessibility of online data visualizations, enabling screen-reader users to extract information using a multi-modal approach. In creating VoxLens, we sought to address the challenges screen-reader users face with online data visualizations by enabling them to extract information both holistically and in a drilled-down manner, using techniques and strategies that they prefer. Specifically, VoxLens provides three modes of interaction using speech and sonification: *Question-and-Answer* mode, *Summary* mode, and *Sonification* mode.

To assess the performance of VoxLens, we conducted task-based experiments and interviews with screen-reader users. VoxLens significantly improved the interaction experiences of screen-reader users with online data visualizations, both in terms of accuracy of extracted information and interaction time, compared to their conventional interaction with online data visualizations. Our results also show that screen-reader users considered VoxLens to be a "game-changer," providing them with "exciting new ways" to interact with online data visualizations and saving them time and effort. We hope that by open-sourcing our code for VoxLens and our sonification solution, our work will inspire developers and visualization creators to continually improve the accessibility of online data visualizations. We also hope that our work will motivate and guide future research in making data visualizations accessible.

ACKNOWLEDGMENTS

This work was supported in part by the Mani Charitable Foundation, the University of Washington Center for an Informed Public, and the University of Washington Center for Research and Education on Accessible Technology and Experiences (CREATE). We extend our gratitude to the AccessComputing staff for their support and assistance in recruiting participants. We would also like to thank the anonymous reviewers for their helpful comments and suggestions. Any opinions, findings, conclusions, or recommendations expressed in this work are those of the authors and do not necessarily reflect those of any supporter.

REFERENCES

[1] Chadia Abras, Diane Maloney-Krichmar, Jenny Preece, et al. 2004. User-centered design. *Bainbridge, W. Encyclopedia of Human-Computer Interaction. Thousand Oaks: SAGE Publications* 37, 4 (2004), 445–456.

[2] NV Access. n.d.. NV Access | Download. https://www.nvaccess.org/download/. (Accessed on 08/08/2021).

[3] Dragan Ahmetovic, Cristian Bernareggi, João Guerreiro, Sergio Mascetti, and Anna Capietto. 2019. Audiofunctions. web: Multimodal exploration of mathematical function graphs. In *Proceedings of the 16th International Web for All Conference*. 1–10.

[4] Theodore W. Anderson and Donald A. Darling. 1954. A test of goodness of fit. *Journal of the American statistical association* 49, 268 (1954), 765–769.

[5] Apple. n.d.. Audio Graphs | Apple Developer Documentation. https://developer.apple.com/documentation/accessibility/audio_graphs. (Accessed on 08/01/2021).

[6] Ronald J. Baken and Robert F. Orlikoff. 2000. *Clinical Measurement of Speech and Voice.* Singular Thomson Learning, San Diego, California, USA. https://books.google.com/books?id=ElPyvaJbDiwC

[7] Nikola Banovic, Rachel L. Franz, Khai N. Truong, Jennifer Mankoff, and Anind K. Dey. 2013. Uncovering Information Needs for Independent Spatial Learning for Users Who Are Visually Impaired. In *Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility* (Bellevue, Washington) *(ASSETS '13)*. Association for Computing Machinery, New York, NY, USA, Article 24, 8 pages. https://doi.org/10.1145/2513383.2513445

[8] Katja Battarbee. 2004. *Co-experience: understanding user experiences in interaction.* Aalto University, Helsinki, Finland. 103 + app. 117 pages. http://urn.fi/URN:ISBN:951-558-161-3

[9] Donald A. Berry. 1987. Logarithmic Transformations in ANOVA. *Biometrics* 43, 2 (1987), 439–456. http://www.jstor.org/stable/2531826

[10] Jacques Bertin. 1983. *Semiology of graphics; diagrams networks maps.* Technical Report.

[11] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. 2011. D^3 data-driven documents. *IEEE transactions on visualization and computer graphics* 17, 12 (2011), 2301–2309.

[12] Virginia Braun and Victoria Clarke. 2006. Using thematic analysis in psychology. *Qualitative research in psychology* 3, 2 (2006), 77–101.

[13] Matthew Brehmer, Bongshin Lee, Petra Isenberg, and Eun Kyoung Choe. 2018. Visualizing ranges over time on mobile phones: a task-based crowdsourced evaluation. *IEEE transactions on visualization and computer graphics* 25, 1 (2018), 619–629.

[14] Matthew Brehmer and Tamara Munzner. 2013. A multi-level typology of abstract visualization tasks. *IEEE transactions on visualization and computer graphics* 19, 12 (2013), 2376–2385.

[15] Craig Brown and Amy Hurst. 2012. VizTouch: Automatically Generated Tactile Visualizations of Coordinate Spaces. In *Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction* (Kingston, Ontario, Canada) *(TEI '12)*. Association for Computing Machinery, New York, NY, USA, 131–138. https://doi.org/10.1145/2148131.2148160

[16] Lorna M. Brown, Stephen A. Brewster, Ramesh Ramloll, Mike Burton, and Beate Riedel. 2003. Design guidelines for audio presentation of graphs and tables. In *Proceedings of the 9th International Conference on Auditory Display.* Citeseer, Boston University, USA, 284–287.

[17] Ben Caldwell, Michael Cooper, Loretta Guarino Reid, Gregg Vanderheiden, Wendy Chisholm, John Slatin, and Jason White. 2008. Web content accessibility guidelines (WCAG) 2.0.

[18] ChartJS. [n. d.]. Accessibility | Chart.js. https://www.chartjs.org/docs/3.5.1/general/accessibility.html. (Accessed on 01/08/2022).

[19] Peter Ciuha, Bojan Klemenc, and Franc Solina. 2010. Visualization of Concurrent Tones in Music with Colours. In *Proceedings of the 18th ACM International Conference on Multimedia* (Firenze, Italy) *(MM '10)*. Association for Computing Machinery, New York, NY, USA, 1677–1680. https://doi.org/10.1145/1873951.1874320

[20] Jacob Cohen. 1973. Eta-squared and partial eta-squared in fixed factor ANOVA designs. *Educational and psychological measurement* 33, 1 (1973), 107–112.

[21] Nils Dahlbäck, Arne Jönsson, and Lars Ahrenberg. 1993. Wizard of Oz studies—why and how. *Knowledge-based systems* 6, 4 (1993), 258–266.

[22] Patrick Dengler, Anthony Grasso, Chris Lilley, Cameron McCormack, Doug Schepers, and Jonathan Watt. 2011. Scalable Vector Graphics (SVG) 1.1.

[23] Google Developers. 2014. Charts. https://developers.google.com/chart/

[24] Roger L. DuBois. 2017. Web Audio Speech Synthesis / Recognition for p5.js. https://github.com/IDMNYU/p5.js-speech

[25] Christopher J. Ferguson. 2016. An effect size primer: A guide for clinicians and researchers. In *Methodological issues and strategies in clinical research*, A.E. Kazdin (Ed.). American Psychological Association, Washington, DC, USA, 301––310.

[26] John H. Flowers. 2005. Thirteen years of reflection on auditory graphing: Promises, pitfalls, and potential new directions. In *Proceedings of the 11th Meeting of the International Conference on Auditory Display.* Citeseer, Limerick, Ireland, 406–409.

[27] John H. Flowers, Dion C. Buhman, and Kimberly D. Turnage. 1997. Cross-modal equivalence of visual and auditory scatterplots for exploring bivariate data samples. *Human Factors* 39, 3 (1997), 341–351.

[28] OpenJS Foundation. n.d.. Node.js. https://nodejs.org/en/. (Accessed on 08/08/2021).

[29] The GraphQL Foundation. n.d.. GraphQL | A query language for your API. https://graphql.org/. (Accessed on 08/08/2021).

[30] Brigitte N. Frederick. 1999. Fixed-, random-, and mixed-effects ANOVA models: A user-friendly guide for increasing the generalizability of ANOVA results. In *Advances in Social Science Methodology*, B. Thompson (Ed.). JAI Press, Stamford, Connecticut, 111–122.

[31] Deen G. Freelon. 2010. ReCal: Intercoder reliability calculation as a web service. *International Journal of Internet Science* 5, 1 (2010), 20–33.

[32] Arthur Gilmour, Robert D. Anderson, and Alexander L. Rae. 1985. The analysis of binomial data by a generalized linear mixed model. *Biometrika* 72, 3 (1985), 593–599.

[33] Nicholas A. Giudice, Hari Prasath Palani, Eric Brenner, and Kevin M. Kramer. 2012. Learning Non-Visual Graphical Information Using a Touch-Based Vibro-Audio Interface. In *Proceedings of the 14th International ACM SIGACCESS Conference on Computers and Accessibility* (Boulder, Colorado, USA) *(ASSETS '12)*. Association for Computing Machinery, New York, NY, USA, 103–110. https://doi.org/10.1145/2384916.2384935

[34] The PostgreSQL Global Development Group. n.d.. PostgreSQL: The world's most advanced open source database. https://www.postgresql.org/. (Accessed on 08/08/2021).

[35] Melita Hajdinjak and France Mihelic. 2004. Conducting the Wizard-of-Oz Experiment. *Informatica (Slovenia)* 28, 4 (2004), 425–429.

[36] Patrick A.V. Hall and Geoff R. Dowling. 1980. Approximate string matching. *ACM computing surveys (CSUR)* 12, 4 (1980), 381–402.

[37] Jordan Harband, Shu-yu Guo, Michael Ficarra, and Kevin Gibbons. 1999. Standard ecma-262.

[38] Sandra G. Hart and Lowell E. Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. In *Advances in psychology*. Vol. 52. Elsevier, North-Holland, Netherlands, 139–183.

[39] Highcharts. n.d.. Sonification | Highcharts. https://www.highcharts.com/docs/accessibility/sonification. (Accessed on 08/01/2021).

[40] Clare J. Hooper. 2011. Towards Designing More Effective Systems by Understanding User Experiences. *SIGWEB Newsl.* Autumn, 4, Article 4 (Sept. 2011), 3 pages. https://doi.org/10.1145/2020936.2020940

[41] Mike H. Hoyle. 1973. Transformations: An Introduction and a Bibliography. *International Statistical Review / Revue Internationale de Statistique* 41, 2 (1973), 203–223. http://www.jstor.org/stable/1402836

[42] Weijian Hu, Kaiwei Wang, Kailun Yang, Ruiqi Cheng, Yaozu Ye, Lei Sun, and Zhijie Xu. 2020. A comparative study in real-time scene sonification for visually impaired people. *Sensors* 20, 11 (2020), 3222.

[43] Amy Hurst and Shaun Kane. 2013. Making "Making" Accessible. In *Proceedings of the 12th International Conference on Interaction Design and Children* (New York, New York, USA) *(IDC '13)*. Association for Computing Machinery, New York, NY, USA, 635–638. https://doi.org/10.1145/2485760.2485883

[44] Apple Inc. n.d.. Accessibility - Vision - Apple. https://www.apple.com/accessibility/vision/. (Accessed on 08/08/2021).

[45] Facebook Inc. n.d.. React – A JavaScript library for building user interfaces. https://reactjs.org/. (Accessed on 08/08/2021).

[46] Dae Hyun Kim, Enamul Hoque, and Maneesh Agrawala. 2020. Answering Questions about Charts and Generating Visual Explanations. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3313831.3376467

[47] Edward Kim and Kathleen F McCoy. 2018. Multimodal deep learning using images and text for information graphic classification. In *Proceedings of the 20th International ACM SIGACCESS Conference on Computers and Accessibility*. 143–148.

[48] Edward Kim, Connor Onweller, and Kathleen F McCoy. 2021. Information Graphic Summarization using a Collection of Multimodal Deep Neural Networks. In *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 10188–10195.

[49] Klaus Krippendorff. 2011. Computing Krippendorff's Alpha-Reliability. https://repository.upenn.edu/asc_papers/43 Retrieved from.

[50] Klaus Krippendorff. 2018. *Content analysis: An introduction to its methodology.* SAGE Publications Inc., Pennsylvania, USA.

[51] Richard J. Landis and Gary G. Kock. 1977. The Measurement of Observer Agreement for Categorical Data. *Biometrics* 33, 1 (1977), 159–174. http://www.jstor.org/stable/2529310

[52] Bongshin Lee, Arjun Srinivasan, Petra Isenberg, John Stasko, et al. 2021. Post-WIMP Interaction for Information Visualization. *Foundations and Trends® in Human-Computer Interaction* 14, 1 (2021), 1–95.

[53] Eckhard Limpert, Werner A. Stahel, and Markus Abbt. 2001. Log-normal distributions across the sciences: keys and clues: on the charms of statistics, and how mechanical models resembling gambling machines offer a link to a handy way to characterize log-normal distributions, which can provide deeper insight into variability and probability—normal or log-normal: that is the question. *BioScience* 51, 5 (2001), 341–352.

[54] Ramon C. Littell, Henry P. Raymond, and Clarence B. Ammerman. 1998. Statistical analysis of repeated measures data using SAS procedures. *Journal of animal science* 76, 4 (1998), 1216–1231.

[55] Alan Lundgard and Arvind Satyanarayan. 2021. Accessible Visualization via Natural Language Descriptions: A Four-Level Model of Semantic Content. *IEEE transactions on visualization and computer graphics* (2021).

[56] Yotam Mann. n.d.. Tone.js. https://tonejs.github.io/. (Accessed on 08/02/2021).

[57] Kim Marriott, Bongshin Lee, Matthew Butler, Ed Cutrell, Kirsten Ellis, Cagatay Goncu, Marti Hearst, Kathleen McCoy, and Danielle Albers Szafir. 2021. Inclusive data visualization for people with disabilities: a call to action. *Interactions* 28, 3 (2021), 47–51.

[58] David K. McGookin and Stephen A. Brewster. 2006. SoundBar: Exploiting Multiple Views in Multimodal Graph Browsing. In *Proceedings of the 4th Nordic Conference on Human-Computer Interaction: Changing Roles* (Oslo, Norway) *(NordiCHI '06)*. Association for Computing Machinery, New York, NY, USA, 145–154. https://doi.org/10.1145/1182475.1182491

[59] Silvia Mirri, Silvio Peroni, Paola Salomoni, Fabio Vitali, and Vincenzo Rubano. 2017. Towards accessible graphs in HTML-based scientific articles. In *2017 14th IEEE Annual Consumer Communications Networking Conference (CCNC)*. IEEE, Las Vegas, NV, USA, 1067–1072. https://doi.org/10.1109/CCNC.2017.7983287

[60] André Natal, Glen Shires, and Philip Jägenstedt. n.d.. Web Speech API. https://wicg.github.io/speech-api/. (Accessed on 08/07/2021).

[61] Keita Ohshiro, Amy Hurst, and Luke DuBois. 2021. Making Math Graphs More Accessible in Remote Learning: Using Sonification to Introduce Discontinuity in Calculus. In *The 23rd International ACM SIGACCESS Conference on Computers and Accessibility*. 1–4.

[62] Michael Q. Patton. 1990. *Qualitative evaluation and research methods*. SAGE Publications Inc., Saint Paul, MN, USA.

[63] Pablo Picazo-Sanchez, Juan Tapiador, and Gerardo Schneider. 2020. After you, please: browser extensions order attacks and countermeasures. *International Journal of Information Security* 19, 6 (2020), 623–638.

[64] Ricardo Sousa Rocha, Pedro Ferreira, Inês Dutra, Ricardo Correia, Rogerio Salvini, and Elizabeth Burnside. 2016. A Speech-to-Text Interface for MammoClass. In *2016 IEEE 29th International Symposium on Computer-Based Medical Systems (CBMS)*. IEEE, Dublin, Ireland and Belfast, Northern Ireland, 1–6. https://doi.org/10.1109/CBMS.2016.25

[65] Bahador Saket, Alex Endert, and Çağatay Demiralp. 2018. Task-based effectiveness of basic visualizations. *IEEE transactions on visualization and computer graphics* 25, 7 (2018), 2505–2512.

[66] Nik Sawe, Chris Chafe, and Jeffrey Treviño. 2020. Using Data Sonification to Overcome Science Literacy, Numeracy, and Visualization Barriers in Science Communication. *Frontiers in Communication* 5 (2020), 46.

[67] Anastasia Schaadhardt, Alexis Hiniker, and Jacob O. Wobbrock. 2021. Understanding Blind Screen-Reader Users' Experiences of Digital Artboards. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, Article 270, 19 pages. https://doi.org/10.1145/3411764.3445242

[68] Freedom Scientific. n.d.. JAWS® – Freedom Scientific. https://www.freedomscientific.com/products/software/jaws/. (Accessed on 08/08/2021).

[69] Ather Sharif, Sanjana Chintalapati, Jacob O. Wobbrock, and Katharina Reinecke. 2021. Understanding Screen-Reader Users' Experiences with Online Data Visualizations. In *The 23rd International ACM SIGACCESS Conference on Computers and Accessibility* (Virtual Event) *(ASSETS '21)*. Association for Computing Machinery, New York, NY, USA, To Appear.

[70] Ather Sharif and Babak Forouraghi. 2018. evoGraphs — A jQuery plugin to create web accessible graphs. In *2018 15th IEEE Annual Consumer Communications Networking Conference (CCNC)*. IEEE, Las Vegas, NV, USA, 1–4. https://doi.org/10.1109/CCNC.2018.8319239

[71] Lei Shi, Idan Zelzer, Catherine Feng, and Shiri Azenkot. 2016. Tickers and Talker: An Accessible Labeling Toolkit for 3D Printed Models. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 4896–4907. https://doi.org/10.1145/2858036.2858507

[72] Boris Smus. 2013. *Web Audio API: Advanced Sound for Games and Interactive Apps*. O'Reilly Media, California, USA. https://books.google.com/books?id=eSPyRuL8b7UC

[73] Arjun Srinivasan, Nikhila Nyapathy, Bongshin Lee, Steven M. Drucker, and John Stasko. 2021. Collecting and Characterizing Natural Language Utterances for Specifying Data Visualizations. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, Article 464, 10 pages. https://doi.org/10.1145/3411764.3445400

[74] Jonathan E. Thiele, Michael S. Pratte, and Jeffrey N. Rouder. 2011. On perfect working-memory performance with large numbers of items. *Psychonomic Bulletin & Review* 18, 5 (2011), 958–963.

[75] Ingo R. Titze and Daniel W. Martin. 1998. Principles of voice production.

[76] Frances Van Scoy, Don McLaughlin, and Angela Fullmer. 2005. Auditory augmentation of haptic graphs: Developing a graphic tool for teaching precalculus skill to blind students. In *Proceedings of the 11th Meeting of the International Conference on Auditory Display*, Vol. 5. Citeseer, Limerick, Ireland, 5 pages.

[77] W3C. n.d.. WAI-ARIA Overview | Web Accessibility Initiative (WAI) | W3C. https://www.w3.org/WAI/standards-guidelines/aria/. (Accessed on 04/11/2021).

[78] WebAIM. n.d.. WebAIM: CSS in Action - Invisible Content Just for Screen Reader Users. https://webaim.org/techniques/css/invisiblecontent/. (Accessed on 09/01/2021).

[79] Jacob O. Wobbrock, Krzysztof Z. Gajos, Shaun K. Kane, and Gregg C. Vanderheiden. 2018. Ability-based design. *Commun. ACM* 61, 6 (2018), 62–71.

[80] Susan P. Wyche and Rebecca E. Grinter. 2009. Extraordinary Computing: Religion as a Lens for Reconsidering the Home. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 749–758. https://doi.org/10.1145/1518701.1518817

[81] Wai Yu, Ramesh Ramloll, and Stephen Brewster. 2001. Haptic graphs for blind computer users. In *Haptic Human-Computer Interaction*, Stephen Brewster and Roderick Murray-Smith (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 41–51.

[82] Mingrui Ray Zhang, Ruolin Wang, Xuhai Xu, Qisheng Li, Ather Sharif, and Jacob O. Wobbrock. 2021. Voicemoji: Emoji Entry Using Voice for Visually Impaired People. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, Article 37, 18 pages. https://doi.org/10.1145/3411764.3445338

[83] Haixia Zhao, Catherine Plaisant, Ben Shneiderman, and Jonathan Lazar. 2008. Data sonification for users with visual impairment: a case study with georeferenced data. *ACM Transactions on Computer-Human Interaction (TOCHI)* 15, 1 (2008), 1–28.

A PARTICIPANT DEMOGRAPHICS

	Gender	Age	Screen Reader	Vision-Loss Level	Diagnosis
P1	M	28	NVDA	Blind since birth, Complete blindness	Optic Nerve Hypoplasia
P2	M	61	JAWS	Complete blindness, Lost vision gradually	Optic Neuropathy
P3	M	48	JAWS	Complete blindness, Lost vision gradually	Leber Congenital Amaurosis
P4	F	29	NVDA	Blind since birth, Complete blindness	Optic Nerve Hypoplasia and Glaucoma
P5	F	37	JAWS	Blind since birth, Complete blindness	Leber Congenital Amaurosis
P6	F	51	JAWS	Blind since birth, Complete blindness	Retinopathy of Prematurity
P7	M	58	JAWS	Complete blindness, Lost vision gradually	Glaucoma
P8	M	30	NVDA	Blind since birth, Complete blindness	Leber Congenital Amaurosis
P9	F	64	JAWS	Complete blindness, Lost vision gradually	Retinitis Pigmentosa
P10	F	68	Fusion	Lost vision gradually, Partial blindness	Stargaart's Maculopathy
P11	F	73	JAWS	Complete blindness, Lost vision gradually	Retinitis Pigmentosa
P12	F	64	JAWS	Complete blindness, Lost vision gradually	Cataracts
P13	M	18	NVDA	Complete blindness	Brain Tumor
P14	M	36	JAWS	Blind since birth, Complete blindness	Leber Congenital Amaurosis
P15	M	25	NVDA	Lost vision gradually, Partial vision	Retinopathy of Prematurity and Subsequent Cataracts
P16	M	42	JAWS	Blind since birth, Complete blindness	Microphthalmia
P17	M	68	JAWS	Complete blindness, Lost vision gradually	Detached Retinas
P18	F	31	NVDA	Blind since birth, Complete blindness	Retinopathy of Prematurity
P19	F	47	JAWS	Complete blindness, Lost vision gradually	Optic Neuropathy
P20	M	48	NVDA	Complete blindness, Lost vision gradually	Retinitis Pigmentosa
P21	M	43	NVDA	Complete blindness, Lost vision gradually	Retinitis Pigmentosa
P22	M	19	NVDA	Blind since birth, Complete blindness	Retinopathy of Prematurity

Table 6: Screen-reader participants, their gender identification, age, screen reader, vision-loss level, and diagnosis. Under the *Gender* column, *M* = *Male*, *F* = *Female*, and *NB* = *Non-binary*.

B WIZARD-OF-OZ PARTICIPANT DEMOGRAPHICS

	Gender	Age	Screen Reader	Vision-Loss Level	Diagnosis
W1	M	25	VoiceOver	Partial vision	Extremely low vision
W2	M	28	NVDA	Blind since birth	Optic Nerve Hypoplaxia
W3	M	23	VoiceOver	Blind since birth	Septo-optic Dysplasia
W4	F	26	JAWS	Blind since birth	Leber Congenital Amaurosis
W5	M	31	JAWS	Blind since birth	Retinopathy of Prematurity

Table 7: Screen-reader participants for the Wizard-of-Oz experiment, their gender identification, age, screen reader, vision-loss level, and diagnosis. Under the *Gender* column, M = *Male*, F = *Female*, and NB = *Non-binary*.

C INTERACTION TIME PER AGE RANGE

	Both Groups			*Without VoxLens*			*With VoxLens*		
Age Range	*N*	*Mean*	*SD*	*N*	*Mean*	*SD*	*N*	*Mean*	*SD*
18-19	3	62.5	30.5	1	96.8	-	2	45.3	9.4
20-29	9	40.6	28.0	6	63.7	20.3	3	50.8	7.9
30-39	10	44.2	23.1	7	60.6	16.5	3	43.9	4.5
40-49	15	67.5	78.7	10	106.7	93.1	5	59.8	14.9
50-59	7	47.6	27.3	5	79.1	19.2	2	63.3	3.0
60-69	11	64.8	33.6	6	76.8	33.7	5	55.2	10.8
> 70	2	127.0	127.4	1	217.1	-	1	60.1	-

Table 8: Summary results from 57 screen-reader participants with (N=21) and without (N=36) VoxLens showing the numerical results for *Interaction Time* (IT), for each age range. N is the total number of participants for the given age range, *Mean* is the average IT in seconds, and SD represents the standard deviation.