Reimagined-Vis: Project Part 2

Kathryn (Katy) Koenig **Lucy Li** **Belen Michel Torino**

PROJECT DESCRIPTION

While many technologies exist to assist people with visual impairments, e.g. text-to-speech technologies, most data visualizations lack accessibility for people with visual impairments. We seek to change this status quo by making the creation of accessible data visualizations easier: our system will standardize the process of making visualizations accessible as well as democratize accessible visualization, encouraging a culture in which accessible plots are par for the course.

REQUIREMENTS SUMMARY

Our system will allow people with visual impairments to interpret data visualizations which were originally created for sighted users. Our system will allow for users to glean more information from the system's output than would have been understood from the original plot created for sighted consumers. Moreover, our system will allow its users to understand at least as much information regarding the data as sighted consumers would understand a data visualization.

As all of the participants in our study referenced their dependency on others to interpret plots for them, our system will empower people with visual impairments to understand data visualizations without the plot being mediated through another sighted person. Therefore, it is important that our system allows the users to feel more self-reliant.

Additionally, as most sighted users consume data visualizations created by data visualization designers, our system will not place the burden of data designer onto the consumer. As noted in our project proposal, many current systems to address the issue of translating plots for people with visual impairments only translate visualizations into data tables [2, 9]. Instead, our system will allow people with visual impairments to have a similar experience as sighted people to data visualizations: both are consumers of product, the plot, which is crafted and curated for the consumer by data designers.

Regarding more technical criteria, as almost all of our study participants use speech-to-text technology [1, 3, 4, 5], our product must facilitate the easy integration of data visualizations into screen reading software, e.g. Freedom Scientific's JAWS software.

Furthermore, as many of our study participants expressed concern regarding learning to use a new technology [1, 4, 5] and because we want our consumers to have a similar experience to sighted consumers, our product must minimize, or ideally, remove any learning curve to actually be useful to people with visual impairments.

Additionally, as described in the User Research Summary below, the majority of the participants in our study describe their visual acuity as "low vision" and because the range, and therefore the needs, of visual impairments experienced by many people is so vast, our system will cater to people experiencing low vision. As described by the American Foundation for the Blind [4], low vision is described as vision low which is permanent and cannot be corrected with medical intervention. While all of our study participants with low vision still relied heavily on their current visual acuity, they also described experiencing the following issues:

- Differentiating between similar colors.
- Reading text with a small font size.
- Experiencing sensitivity to bright lights.
- Perceiving differences in depth between objects.
- Needing to use zoom features and enlarging font size for most visualizations.

Therefore, our system will seek to specifically address these issues as they relate to data visualization, e.g. ensuring color schemes that have dissimilar colors for easy interpretability by low vision users.

USER RESEARCH SUMMARY

We conducted semi-structured interviews with our users in which we met with users in person and via phone. We first asked them open demographic question regarding their occupation and level of vision. We then asked our participants to reflect on their experiences when encountering data visualizations. Finally, we described design options we are exploring and asked for their feedback regarding these designs. Below, we provide a table detailing the demographics of our participants:

Participants			
Participant No.	Profession	Age	Vision Level
1	Student	29	Low vision
2	Economist	30	Low vision
3	Psychologist	67	Low vision
4	Student	20	Blind
5	Student	32	Colorblind

As many of our participants have issues reading printed paper without use of additionally instruments, we provided digital consent forms prior to our interviews. At the onset of the interviews, we confirmed receipt of the consent form by the participant as well as confirmed consent to participate in our study. With the consent of our interview participants, we audio recorded. We then transcribed the audio recordings. As one of our participants' primary language is Spanish, we provided consent forms in Spanish as well as conducted the interview in Spanish. This interview was transcribed in the original language, and, as Spanish is the first language for one person in our research team, she translated the transcription.

We then went through a process of iterative coding. We initialized our codes while doing a first pass on the first interview conducted as group. As we conducted more interviews, additional codes were added as necessary. We then completed a second pass on all interviews, recoding phrases in which more recently added codes where more relevant that the initial codes. We then completed affinity diagramming of these codes.

Upon completing our affinity diagram, we noticed common threads throughout the responses from our participants. Specifically, all our our participants relied on sighted users for descriptions and evaluations of data visualizations to gain or confirm understandings of the participants interpretations, and one participant noted feeling like a "bother" to others when asking for assistance [2]. Furthermore, when asked about their interactions with data visualizations when they are alone, one participant stated "[I] just basically skip it and wish for the best... Like I think I lose a lot of the information" [1]. Another user noted that she arranges her course schedule to avoid any classes that would involve statistics due to the visual acuity needed to complete the coursework [4]. Both participants reflect a hopelessness in evaluating data visualizations on their own.

Similarly, many participants noted that to gain an understanding of a plot, they must spend more time investigating the visualization [1, 3, 4, 5] while at the same time, participants were concerned regarding the amount of time needed to adapt to a new technology for translating data visualizations [1, 4, 5] or for a technology, like 3D printing, to translate a plot in a useful amount of time [2, 5].

The utilization of text-to-speech technologies, like iPhone's Voiceover and Freedom Scientific's JAWS software, were almost ubiquitous among our interviewees [1, 2, 3, 4].

DESIGN METHODOLOGY

To begin our prototype design, we first analyzed pre-existing systems and looked at their effectiveness with respect to their encodings of the data. From our interviews and our literature review, we developed three different prototypes that represent data more effectively for people with visual impairments. Specifically, we created prototypes of a data sonification software, an enhanced visualization application and an accessible visualization linter. To assist us in creating a better product, we used contextual inquiry. Because most of our interviews occurred via phone, we asked participants to describe which tools they used when encountering data visualizations and their likes and dislikes regarding these tools. This gave us insight into how our participants use (or in some cases did not use) available tools and how different levels of vision affected which users preferred tools. With this information, we could avoid parts of pre-existing systems that users found unhelpful and utilize parts which did assist in data comprehension for people with visual impairments. We also did participatory design with our participants: we concluded our interviews with descriptions of our prototype ideas and requested their feedback on these designs. By co-designing with our users, we found different perspectives on how our solutions can assist them as well as aspects which we could improve in our designs. We used this feedback to better tailor our designs for our users.

DESIGN SPACE

Difficult Requirements

In order to give low vision and blind users a similar experience to those who are sighted, there has to be a way for users to experience graphs in a whole-to-part way. Without seeing the visualization, it is difficult to understand the whole picture of a visualization. What we create also must be quick and easy, in order to minimize the time it takes to understand data using our prototypes. Since those with vision can quickly look at and understand different areas of a graph, it takes them less time to begin analyzing. This is different for low vision and blind users because of the additional time it takes to apply any tools or inspect different areas.

Another requirement we hope to address, but may not be able to, is to have a prototype that can be used by varying levels of visual impairments. For example, one user may be completely blind, while another may have color blindness. If possible, creating a product that could help both would allow more users to have access to visualizations, but would also require a prototype that has many different tools, which take a longer time to create. This would be especially difficult because of the time-constraint we have.

Data is also represented in multiple ways, such as bar graphs, line graphs, scatter plots, etc. so our prototype would ideally be easily adaptable to any of them in order to cover a wide range of data visualizations. However it is difficult to analyze different graphs since they have different defining characteristics, such as lines, dots, or bars. The prototype would have to identify how data is being represented, then translate it in a quick and efficient manner that is also accurate. This can be difficult since there may be many different edge cases when it comes to data visualizations.

Trade-offs Explored

Due to our time constraints and limited coding knowledge, we were unable to pursue a few different prototypes. Specifically, we considered creating a prototype that used haptic touch to represent data, however all group members did not have knowledge in creating such a device. Creating such a prototype would also require access to an Android phone, which none of us currently own, and Apple does not offer an easily accessible API as Android does. Given a longer time frame, we may have been able to learn about app development and creating an app with haptic feedback. We had similar issues with a physicalization prototype. Additionally, based off our literature review, we found data physicalization to be expensive, time consuming and difficult to parse for any user.

A trade-off we explore below is the creation of a separate consumable (and encoding of data) for people with visual impairments versus updating standard data visualizations to be more accessible for low vision users.

Another trade-off we observed was whether the prototypes we created should be for users to interact with directly, or for developers of data visualizations to use. For example, sonificiation would create a completely different representation of data through sound. Users can change settings in order to have the data represented to their liking. Wile the linter is designed for developers in order to make their visualizations more accessible. This means that it would be the responsibility of developers to use the prototypes, which would require a standardization of using these tools, which could take years to implement. The sonificiation would allow multiple data visualizations to be translated without relying on the developers to include the prototype. Finally, the enhanced visualization application would be a middle solution, since it would require that developers incorporate some extra data or descriptions to their graphs but, at the same time, some of it's feature would also work without relying on the developers.

Tasks

Because we are working in the novel space of translating or altering data visualization for people with visual impairments, we must ensure accurate encoding of data to a form that may not be similar to the visual encoding of data in a plot. Specifically, for some of our prototypes as the dimensions into which we must encode the data may not be as vast as, for example, the entire color spectrum, we may have to discretize the data for ensure understanding of the encoding, which may prove difficult for specific plot types. For example, when sonifying a scatter plot, we cannot play each dot as a note or the cacophony of many sounds may render the sonification indiscernible to the user's ear.

Furthermore, as there are countless variations in the types of data visualizations, e.g. scatter plots, line plots, bar charts, violin plots, we recognize that making every type of data visualization accessible to people with visual impairments may not be possible

As a result of our design process, we decided to change one of our prototypes altogether. This was because we discovered that one of the prototypes we developed would be slightly more difficult to create without having bigger benefit. A participant [4] particularly voiced they would not be sure how effective such a representation would be. We also changed our requirements based off the response from our participants who expressed what they found to be the most important aspects of a design. This included having to include text-to-speech software as most of those we interviewed used programs that integrated this. Along with this, some users expressed concern for learning a new technology, so we further emphasized of goal of ours to minimize the learning curve for our prototypes. Another requirement we added was wanting to be able to have the prototype be accessible to a variety of visual impairments. Although this is a criteria that will be more difficult to accomplish, we still believed it to be important when creating our prototypes.

DESIGN SUMMARY

According to our literature review, there are a few research teams working towards making visualizations more accessible to visually impaired people, mainly in three directions. These are sonification, physicalization, a verbal description based on deep learning algorithms to detect data out of images and a toolkit similar to the Zhao et. al.'s "SeeingVR: A Set of Tools to Make Virtual Reality More Accessible to People with Low Vision"[11].

In our process to come up with three prototypes, with our study participants, we explored our prototypes as well as aspects of their ideal solutions. Specifically, in the final section of our interviews, we brainstormed first focusing on their ideas for a solution, then providing a short description of the ideas our team produced after completing our literature review and asked for participants' feedback.

Upon completion of our affinity diagram, we discovered that while most of our participants were excited by the idea of 3D printing to explore aspects of a visualization, they question whether it would be financially accessible and/or a timely solution [1, 2, 3, 5]. One of the participants [3] mentioned that a 3D representation of the brain could be a useful tool for her to give lectures and teach other people about this organ. However, she could not imagine herself learning from a 3D printing. Moreover, we asked them if they would rely on a tactile solution if the problem of costs and delays could be solved. For example, we asked them to imagine solutions that would involve different types of vibrations as we were considering a prototype that would function as an application on mobile phones, encoding data via haptic touch. Our users discarded this solution, some of them [3, 5] because they were not confident in being able to distinguish different vibrations, and others [1, 3, 4, 5] because they felt that the learning curve will be steep. In words of one participant "the problem with that is that you need to create like a common language between people with visual disability. Like you would need to learn a new Morse code and that like a barrier" [1].

Furthermore, some participants [1, 2, 3] mentioned that they already need significantly more time than sighted people to read a paper or a book, thus, they would prefer solutions that help them speed up the process of reading the visualizations. Some participants cited skipping visualizations altogether [1, 3] as it takes them too much time to parse the plots. This is why, when discussing the deep learning algorithm that detects images and extracts data, this solution was not very appreciated since it takes approximately seven minutes for the algorithm to translate the visualization into speech[2]. Moreover, one participant [3] mentioned that she does not rely on similar algorithmic of solutions because they often return inaccurate descriptions.

Finally, when exploring the idea of an augmentation toolkit, such as SeeingVR[11], we discovered that this solution would disable developers to tailor their design to give their audience the experience they wanted. Thus, we thought that an overcoming idea would be a linter that highlights improvements needed to be done at the moment of designing the visualization. This would allow the developer to make the changes needed in the way he better likes to create experience he wants to transmit.

Aligned with the ideas that the participants provided, we designed two other prototypes that we expect to be useful for them. The first one is the sonification, where graphs would be translated to music. After meeting with people with a variety of visual impairments, we wanted to include a system which could address all of their needs. The second is an application to enhance the experience when encountering visualizations. This would work as a layer on top of any app opened on the phone and will allow users to modify and interact with visualizations in an easier way. These last three prototypes mentioned will be described in detail in the next section.

OUR DESIGNS

Design 1: Sonification

Overview

The sonification prototype uses different instruments to represent different functions on a graph, and different pitches to represent how the functions change. For example, if there are two different functions on a graph, one could be represented by a trumpet, and the other a piano. Using two different timbres allows the user to be able to differentiate the two much easier than if they were both the same. Pitches offer a wide range of representation since they theoretically can continue forever, and cover areas between defined pitches. For example, between the notes C and D, there are actually an infinite amount of variations between them, which assists in graphs when they progress between points. This would most likely be implemented as a web browser extension, to make it easily accessible to more users, and to easily translate data present online. When looking at a data visualization, text-to-speech is utilized to give more details about the graph itself. When starting a playback, each function will be named as well as what instrument is representing it. If a user decides to pause the playback, the software will notify at which point on the x-axis and y-axis the playback is paused at.

Illustration

In Figure 1, we provide a sample sketch of this prototype. USA would be represented by a trumpet, and China would be represented by a piano. As you can see in the picture, the user can customize the instruments, volume, and which of these sounds are played.

Scenario

You are a student in a class with a required reading that includes various different data visualizations. As someone who is blind, you use JAWS in order to listen to the text-to-speak version of the reading. You come across a data representation of birth rate in different countries over time that is a line graph. You open up the extension which is the Reimagined-Vis sonification extension. You are able to change the settings to your liking, resulting in the trumpet representing the United States and the piano representing China. Once you press play, a voice will then say the following:
"You have begun your Reimagined-Vis data sonification. The graph's title is 'Birth Rate in Countries'. The data includes the United States and China. The United States is represented by

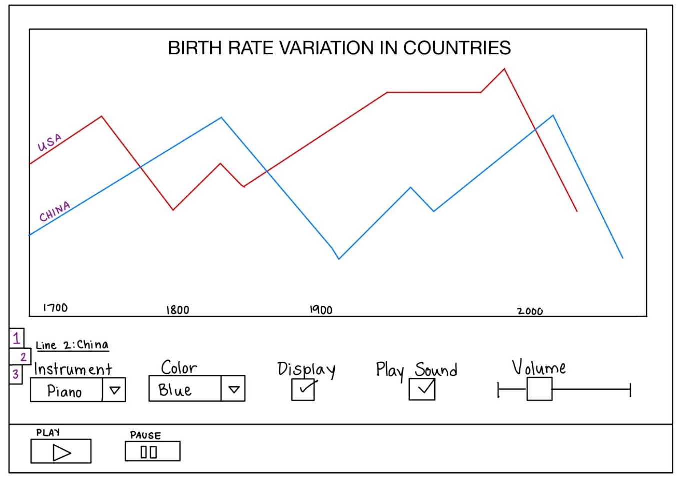


Figure 1: Sonification Prototype

a trumpet, and China by a piano. The United States will play first before a small pause, and then China will play. This will be followed by another small pause before all data is played at the same time."
You press play and listen to the United States and China representations. When listening to both play at the same time, you decide to pause the playback which results in a voice saying the following:
"You have paused your Reimagined-Vis playback. The current location of the playback is at 1981 on the x-axis. The United States is at 70 and China is at 40 on the y-axis
After the playback has finished, the voice will give a short summary: "The United States was represented by the trumpet and China was represented by the piano. The max point for the United States was at 90 on the y-axis at 1781 on the x-axis. The max point for China was 78 on the y-axis at 1999 on the y-axis. Both graphs ended at 2020 on the x-axis."

Assessment

The immediate disadvantage to this prototype is there is a small learning curve first coming across it, however it should be easier to approach since the settings are fairly simple. Another disadvantage is that some smaller details of the data might be lost. For example, when there is a small increase then decrease in a graph, that may be hard to keep track of when listening to the sonification of the data. One of the advantages to this prototype is it can be used by various users with different visual impairments. Although it is more tailored to those who are near-blind or blind, all users can use it to help their understanding of visualizations.

Based off of these disadvantages and advantages, we can see that this prototype meets most of our requirements. Although there is a slight learning curve, it is one we hope will not be too difficult to overcome. This also meets our requirement for being able to be used by various people with visual impairments. However, this requirement is somewhat met, since this prototype in particular is more towards those who are near-blind or blind. This results in limited advantage for those who have higher levels of sight. This prototype does also meet our requirements for being able to be used for various type of graphical data such as bar graphs, scatter plots, etc. since it is mainly focused on how a graph progresses rather than how a graph looks.

Design 2: Enhanced Visualization Application

Overview

This prototype is an accessibility layer for mobile phones that would enhance the experience of visually impaired people when encountering visualizations. It can be easily used: a menu will pop up when a visualization is encountered and the user will be able to select the features she wants to use. Those will reproduce a default action, but they can also be customized by the user to have a more tailored experience.

- The magnifying lenses will allow the user to select a determined area -no need to be precise- and the area will be zoomed in as many Xs as they define.

- The speaker will allow the user to select a determined area -no need to be precise- and the area will be traduce into speech.

- The color palette will allow the user to change the colors of the whole visualization and change the shape/pattern used to fill the drawn area/line (for example, a solid line on a 2D-line graph could be changed to a dashed line)

- The finger will traduce the pointed object (ex. the whole visualization) into a text or speech. This will be particularly useful for the users to have a general idea of the visualization and decide how much time they want to engage with it.

- The light bulb allows users to change the brightness and/or the contrast of the image.

Illustrations

In Figure 2, we provide examples interfaces of the application from the user's prospective. We can see that the user can select charts and hear a short description of the shown visualization.

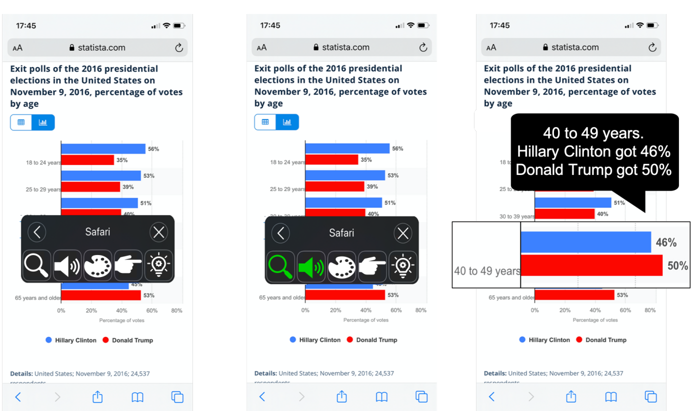


Figure 2: Enhanced Visualization Application Prototype

Scenario

Let's assume that the user is reading the news on his phone. She has activated this visualization enhancement app on her phone because she has a visual impairment that reduced her visual field, thus she has no peripheral vision and she needs high contrast to distinguish colors. Given these needs, she customized the settings of the app when she started using it. She knows that the tools that she likes the most are the magnifying lenses, the light bulb to control the contrast and the finger that helps her to decide which images are of her interest to take a closer look. In addition, she sometimes uses the speaker. She likes it when she is commuting or she does not have much time to dedicate to a detailed view. For these frequently used tools she has defined the following settings: the magnifying lenses are set to 2X, the light bulb changes the background to black and the letters to white, the finger shows a bright red line around the object she is pointing, and it uses text-to-speech to described it. Finally, the speaker is set to 1.5X speed of reading, since she usually uses this option when she wants to minimize the time dedicated to visualizations.

Since these settings were already defined by her, now, whenever she has the app activates and she encounters a visualization on her news, the menu pops up and she selects as needed. Let's say that the first graph she encounters is a pie showing the percentage of votes received by Hillary Clinton vs. the ones received by Trump on the last presidential elections. She clicks the finger, select the approximate area around the graph and the app reads out loud "This is a pie chart, showing that Hillary Clinton got 48% of votes and Donald Trump 46%". The menu pops-up again, however, she believes she does not need more information about this graph so she clicks the X to close the menu. She proceeds with the following text of this article. Later she encounters another graph and the menu pops-up. Again, she first selects the finger, and this time she hears "This is a bar chart. It shows Donald Trump's votes in red bars and Hillary Clinton's votes in blue bars. It distinguishes by voters age". This graph sound more interesting to the user, thus, she clicks the magnifying lenses and the light bulb. This allows her to select the different parts of the graph in stages and explore it in detail. With this tool, she is able to observe the different parts zoomed-in and with the adequate colors for her view.

Assessment

One of the advantages of this prototype is that there is no learning curve for users. Moreover, users can customize the features they want to use and the magnitude of the enhancement that they want to experience. This customization is an appropriate solution to involve the developers in the process of making visualizations more accessible, but without demanding them too much extra time on the design. They will only need to write a few general descriptions of the graph. However, we detected two disadvantages of this prototype. The first one is that it is more tailored towards the needs of low vision and colorblind users. Not all the needs of blind people might be addressed. The second is that it needs some descriptions from the developers to be able to use all the tools present on the menu. A workaround for this could be to attach the table with the data to the graph, to allow the app to explore the data to be translated to speech.

This prototype would be ideally designed for iPhones since this is the technology that our five participants mentioned that they are currently using. However, we detected that IOS phones do not offer an easily accessible API to build something like this as Android phones do. In addition, none of the group members have experience in creating an accessibility app as such.

This prototype is considering 5 tools. These ones were carefully selected from the feedback that the participants gave us. The most innovative aspect of this solution is the possibility to use more than one of these tools at the same time. One of the participants mentioned that she "needs a million different devices/apps for different actions". According to the needs and desires of the participants the selected tools want to solve the following:

- The magnifying lenses were included to solve the distance and small font issues.
- The speaker was included because almost all the participants [1,2,3,4] mentioned that they like to use the voice-over functionality on their iPhones.
- The color palette is planned to solve the issue of distinguishing different lines and colors. For example, the colorblind participant mentioned that he likes using different scales of grays for his graphs to avoid confusions. And that he likes adding shapes, such as triangles, dots and squares to the lines to better distinguish them -and to better connect them to their respective label-
- The finger was introduced to give a general overview and save time to the user. This was seen as a util tool to maximize productivity [1,2], since many other activities may take them longer than to other people. A tool like this might help them to catch up faster.
- The light bulb allows users to change the brightness and/or the contrast of the image, something that it was mentioned as needed by three of the participants [1,2,3]

Design 3: Accessible Visualization Linter

Overview

Lint software that analyzes source code for bugs, errors and stylistic issues has been

used to foster better coding practices since the late 1970s [6]. Currently, there is minimal research investigating techniques for validating programs that create visualizations [7]. Furthermore, while there are standards for making webpages accessible, the Web Content Accessibility Guidelines (WCAG)[1], and subsequently, software to lint source code for websites to flag issues in not in accordance with WCAG, such as the ESLint plugin jsx-a11y [3], there does not exist a lint software to flag accessibility issues in code producing visualizations.

Our accessible visualization linter would function in a similar manner to an ordinary linter: a data visualization designer would write their visualization code in the editor of their choice. Then, run the linter on their code in their terminal. The output would state the aspects of the code which are not not accessible for low vision viewers and which lines correspond to that code. The programmer would then update their code to address these issues until the visualization code no longer has any flags for accessibility. The visualization is now interpretable for both sighted and low vision users.

Due to the constraints of the quarter and our research team's skills, our lint program would specifically lint visualizations created using the Altair library in Python. Also as noted previously, most of our study participants experience low vision. Therefore, our accessible visualization linter would flag accessibility issues for low vision users. These include, but are not limited to, the following issues: fonts choice, font size, color schemes and color contrast, thickness of boundary lines, labels and title locations, brightness, encoding variables by more than just color.

Illustrations

In Figure 3 below, we provide an illustration of the developer view of our accessible visualization linter. On the right hand side, we have our text editor with our Python file for our visualization. On the left hand side, we have our terminal in which the lint software is run on our Python file and an example output from running this command.

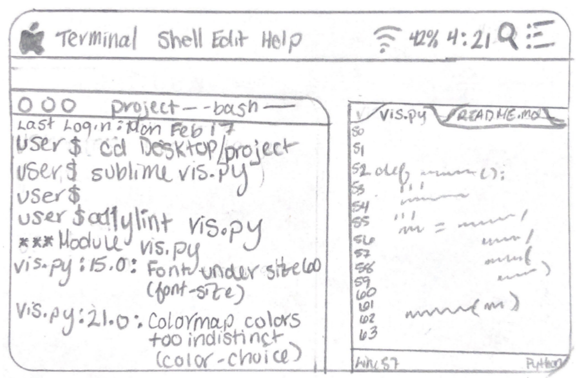


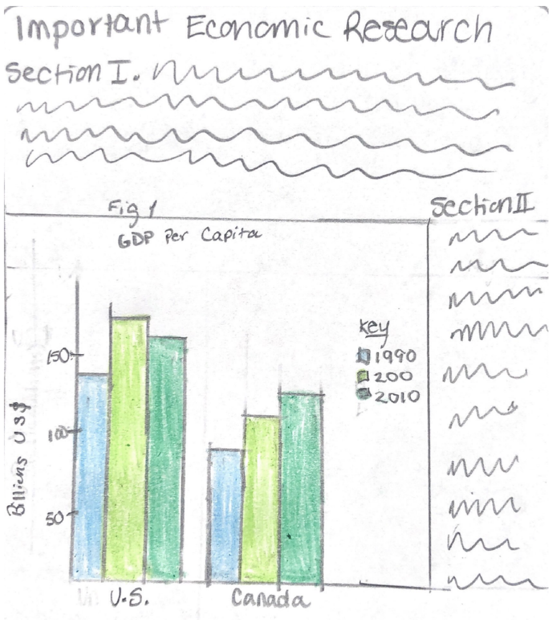
Figure 3: Developer View of Linter

On the following page, we also provide a user's view regarding a visualization made with and the same visualization made without using our accessible visualization lint program. As we can see, the visualization created with utilizing our visualization accessibility linter provides larger text, a more descriptive title, white space between each bar and encodes the year in pattern as well as color.

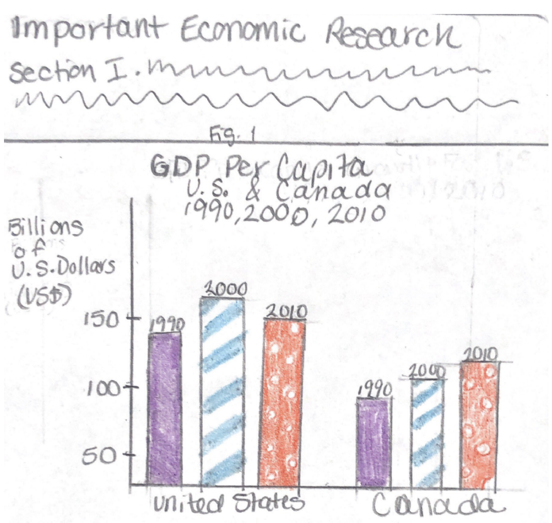
Scenario

Scenario I: Visualization Created without Accessibility Linter

You are a student who identifies as experiencing low vision. For your class, you are assigned an economics paper to read which will be discussed in class. You utilize the school's and other online resources to get a version of the paper which is compatible with JAWS, your choice of screen reading technology. You are in the midst of reading this paper when the conclusions are summarized using a data visualization. This visualization does not have an alternate text description which



(a) Visualization Created Using No Accessibility Linter



(b) Visualization Created Using Accessibility Linter

Figure 4: Consumer View

JAWS can read. You try to zoom in on the visualization as you do for most images. You can tell that it is a bar chart, although the text for each label is too small to read. You cannot tell what the x nor y axes are labelled no matter how much you magnify the text as the image was not saved with high enough resolution. You cannot tell where one bar ends and another begins as there is no white space dividing the bars and the color scheme is too similar for you to differentiate each bar. You search your syllabus, hoping there are TA office hours between today and the in-class discussion so that your teaching assistant can provide some insight regarding the basic information conveyed in the plot. You see there are no office hours scheduled and debate if you should email the teaching assistant requesting extra office hours for yourself despite it being the weekend or whether you should go to class and hopefully not lose too many points for being unable to contribute to the discussion regarding the study's conclusion.

Scenario II: Visualization Created with Accessibility Linter

You are a student who identifies as experiencing low vision. For your class, you are assigned an economics paper to read which will be discussed in class. You utilize the school's and other online resources to get a version of the paper which is compatible with JAWS, your choice of screen reading technology. You are in the midst of reading this paper when the conclusions are summarized using a data visualization. This visualization does not have an alternate text description which JAWS can read, but you are surprised to find that the title is large enough for you to read and descriptive enough to convey the gist of the plot. You magnify the axes labels easily. Because the data is encoded with patterns and colors and there is ample white space between each bar, you can understand the relationships between the data. You spend only a couple minutes looking at the plot and understand the data depicted in the visualizations. You continue reading the article and contribute useful insights in class regarding the paper.

Assessment

The accessible visualization linting prototype offers many advantages: Most importantly, there is no learning curve. Low vision consumers do not need to incorporate and understand any additional technology to understand the plots. Similarly, low vision consumers and sighted consumers view the same visualization, making accessible data visualizations the standard.

Furthermore, as the changes needed to make the visualization more accessible are left up to the developer, low vision users are receiving the same handcrafted experience as sighted users instead of a degraded translation of a visualization. This coincides with the desires of developers: as stated in Zhao et. al.'s "SeeingVR: A Set of Tools to Make Virtual Reality More Accessible to People with Low Vision"[11], developers prefer to control and customize their designs for accessibility themselves as opposed to having their products edited post hoc for accessibility. Additionally, the accessibility is minimally intrusive for data visualization designers as standard linting is a widely used practice.

Our choice of linting Python's Altair library is advantageous in the following regards: 1.) As a team, we are most familiar with Python and therefore, the visualization accessibility linter is within our current coding skills; 2.) Python is a widely used programming language and it's popularity only continues to rise due to this low barriers to entry in learning the language. Therefore, our linter could be utilized by many; and 3.) Altair, unlike Python's other common visualization library Matplotlib, follows the grammar of graphics as originally described in Leland Wilkinson's book by the same name [10]. The grammar of graphics serves as a basis for many other visualization libraries, including R's ggplot and JavaScript's Vega and Vega-Lite libraries. Therefore, an accessibility visualization for Altair could serve as the foundation for visualization accessibility lint programs in other programming languages.

Conversely, we must recognize the disadvantages if we pursue this prototype: Most obviously, the lint program requires that the data designer incorporate the changes. It is not a guarantee that any data visualization will become more accessible as we cannot guarantee that any visualization programmer (aside from ourselves) will use the linter to create a better visualization.

Additionally, the lint software will only address consumers with low vision and people with any type of visual impairment. While we interviewed people with a variety of visual impairments to complete this study, this prototype would be unable to address all of our participants' needs regarding visual impairments.

Finally, there are no current guidelines regarding best practices for creating data visualizations for people with low vision. While there are useful tips and tricks available on the internet[8, 5], there is no widely accepted guide for what constitutes accessible data visualizations for people with low vision. To address this issue, we plan to conduct a survey with low vision users to further understand features which allow visualizations to be more easily understood.

REFERENCES

[1] Ben Caldwell, Michael Cooper, Loretta Guarino Reid, and Gregg Vanderheiden. 2008. Web content accessibility guidelines (WCAG) 2.0. *WWW Consortium (W3C)* (2008).

[2] Jinho Choi, Sanghun Jung, Deok Gun Park, Jaegul Choo, and Niklas Elmqvist. 2019. Visualizing for the Non-Visual: Enabling the Visually Impaired to Use Visualization. *Computer Graphics Forum* (2019). DOI: `http://dx.doi.org/10.1111/cgf.13686`

[3] Facebook. 2020. React Accessibility. `https://reactjs.org/docs/accessibility.html`. (2020). Accessed: 2020-02-16.

[4] American Foundation for the Blind. 2020. AFB Low Vision and Legal Blindness Terms and Descriptions. `https://www.afb.org/blindness-and-low-vision/eye-conditions/low-vision-and-legal-blindness-terms-and-descriptions`. (2020). Accessed: 2020-02-16.

[5] Guidea. 2019. Enterprise Data Viz: Rules and Best Practices. `http://www.storytellingwithdata.com/blog/2018/6/26/accessible-data-viz-is-better-data-viz`. (2019). Accessed: 2020-02-16.

[6] SC Johnson. 1977. inc Bell Telephone Laboratories. *Lint, a C program checker. Citeseer* (1977).

[7] Andrew McNutt and Gordon Kindlmann. 2018. Linting for Visualization: Towards a Practical Automated Visualization Guidance System. In *VisGuides: 2nd Workshop on the Creation, Curation, Critique and Conditioning of Principles and Guidelines in Visualization*.

[8] Cole Nussbaumer Knaflic. 2018. Accessible Data Viz is Better Data Viz. `http://www.storytellingwithdata.com/blog/2018/6/26/accessible-data-viz-is-better-data-viz`. (2018). Accessed: 2020-02-16.

[9] SAS. 2019. SAS Products SAS Graphics Accelerator. `http://support.sas.com/software/products/graphics-accelerator/index.html`. (2019). Accessed: 2020-01-27.

[10] Leland Wilkinson. 2012. The grammar of graphics. In *Handbook of Computational Statistics*. Springer, 375–414.

[11] Yuhang Zhao, Edward Cutrell, Christian Holz, Meredith Ringel Morris, Eyal Ofek, and Andrew D Wilson. 2019. SeeingVR: A set of tools to make virtual reality more accessible to people with low vision. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–14.