Reimagined-Vis: Final Prototype & User Evaluation

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

PROJECT DESCRIPTION

Many data visualizations are not accessible to those with low vision. Because of this, many tools have been created to assist those who may not be able to see visualizations in their original state. However, these multitude of tools create a confusing and lengthy process for those who use them. These tools put the burden of translating plots on the consumer rather than the creator. Our goal is to create a standardization for making data visualizations accessible through our software, allowing those with and without low vision to have the same experience with data visualizations.

To employ this, we have created v_a11y_lint: a lint software that flags issues regarding accessibility in data visualizations. Our software is designed to be used by creator of data visualizations, in order to make it easier to have visualizations be more accessible.

REQUIREMENTS SUMMARY

Our main goal is allow people with low vision to gain greater insight into data from a visualization. This can be done by ensuring that each part of the graph is accessible and legible. By incorporating v_a11y_lint, we are able to ensure that graphs are achieve this, creating a more inclusive environment wherever data visualizations are used. We also want to make the process of understanding data visualizations for low vision users easier and more efficient, compared to having to use multiple tools at once.

We chose to move forward with our accessible data visualization linter as our final prototype as it allows people with low vision to have the same experience as viewers without visual impairments. Additionally, it was important to us that low vision viewers experience data visualizations in the same way as people without visual impairments. With v_a11y_lint, low vision viewers do not have to download an additional application or buy additional tools to understand charts created with v_a11y_lint. Instead, the data visualization designer has the responsibility of ensuring their visualizations are accessible, therefore avoiding any degradation of experience for low vision viewers.

Furthermore, we want v_a11y_lint to increase empowerment of low vision viewers. Many of the participants in our study reflected the desire more self-reliance in interpreting data visualizations. In order to achieve this, charts made with v_a11y_lint should not require mediation through a person without a visual impairment to describe the visualization, therefore, allowing low vision viewers to feel more empowered, independent, and self-reliant.

From the result of our user study, user testing, and user feedback, we address issues regarding data visualization accessibility through integrating the following best practices and guidelines:

- Increasing text font size of all text within the visualization.
- Increasing contrast between text color and background color.
- Adding descriptive titles for each plot.
- Increasing threshold for perceptible differences between colors in a color scheme.

Finally, as we want to ensure use of our software by programmers, it is important the v_a11y_lint is easy to use and incorporate into the workflow of those creating data visualizations. This is done through a simple process and readable result containing accessibility flags as well as recommendations for ensuring accessibility in visualizations created in Altair.

PROTOTYPE REVISIONS

Changes Made

Based off in-class prototype testing, we made the following changes to our software:

1.) Added a pretty print feature to output. A more readable output was the main critique noted during our prototype testing. We addressed this issue through adding a pretty print feature to our software. Our original prototype returned only Python's JSON-like object, a dictionary, of the flagged issues in the data visualization. In addition to this object, v_a11y_lint now prints a more readable version of this dictionary object. Programmers are still able to save the dictionary object and can refer to the object later while the pretty print feature allows programmers to much more easily view and understand the flagged accessibility issues.

2.) Added verbose option. The second most requested feature during prototype testing was including a recommendation system so that users can better understand why an issue was flagged and how to fix the issue to ensure accessibility in their chart. Therefore, we added a verbose option to our main function run_lint(). When users set the 'verbose=True' argument, they are now provided with information regarding the requirements to pass the accessibility test related to the flagged issue.

3.) Bug fixes. During prototype testing, one user did not want to set a custom theme but instead, wanted to use the Altair defaults to create a chart. Altair chart objects for which a theme has not been set by a user do not automatically display the default colors, fonts and other chart attributes. Therefore, v_a11y_lint in its original iteration was unable to flag accessibility issues in the default theme. Now, our software adds any default attributes that are missing in a given theme (whether there is no theme specified by a user or only certain aspects of the theme are defined by users). Thus, v_a11y_lint can now flag issues in the default settings whereas before it issued an error message.

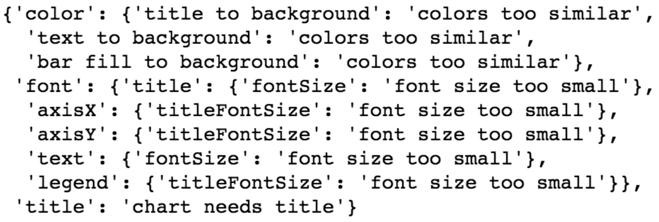
Design Rationale

We provide justification for these changes below in order of the changes mentioned in the section above:

1. Pretty print feature. We want to ensure that the output of our software was simple to use and easy to understand. During our interview process, most people expressed hesitation about using products with a considerable learning curve. While the users of v_a11y_lint are not the low vision users we interviewed, we wanted to ensure as small of a learning curve as possible as this is a universal desire. Furthermore, we understand that if the product is not easy to use, programmers are less inclined to use it, especially because they, as mostly people without visual impairments, do not directly benefit from accessible data visualizations. Therefore, it was important to ensure that programmers could easily understand the output by making the output more legible, thus lowering the barriers to use. We would like to note that we still include the original dictionary output object as we wanted an object which users to could save and to which users could later refer.

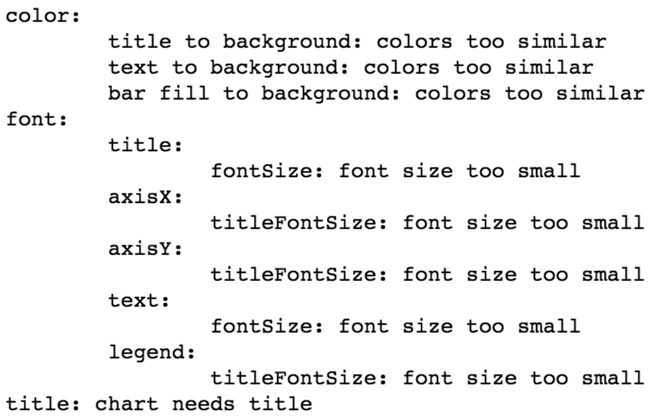
2. Verbose option. Because we want our users to understand the requirements in making a data visualization accessible, the verbose argument was added so that users can see the minimum requirement they need for accessibility. We want to ensure transparency in our product as accessibility guidelines change or are not readily available. The verbose option allows users to gain knowledge regarding accessibility that they can apply to other realms of their work, e.g. web development. Additionally, we make this argument optional as we do not want to force users to see minutiae of the requirements if it is not desired as this option may be overwhelming to new users, who may have many accessibility issues in their chart that need to be addressed.

3. Add default values for chart object. While we do not have statistics regarding Altair use, we hypothesize that most charts made in Altair use the Altair default theme values. Therefore, it was important to us to flag accessibility issues in the default values as well. To do so, we had to manually grab the default options from the Vega-Lite documentation (the visualization library from which Altair is automatically generated) and transform this information into a dictionary representation of a chart object, which is now publicly available in our GitHub code as well.



```
{'color': {'title to background': 'colors too similar',
  'text to background': 'colors too similar',
  'bar fill to background': 'colors too similar'},
 'font': {'title': {'fontSize': 'font size too small'},
  'axisX': {'titleFontSize': 'font size too small'},
  'axisY': {'titleFontSize': 'font size too small'},
  'text': {'fontSize': 'font size too small'},
  'legend': {'titleFontSize': 'font size too small'}},
 'title': 'chart needs title'}
```

Figure 1. Original Dictionary Output



```
color:
        title to background: colors too similar
        text to background: colors too similar
        bar fill to background: colors too similar
font:
        title:
                fontSize: font size too small
        axisX:
                titleFontSize: font size too small
        axisY:
                titleFontSize: font size too small
        text:
                fontSize: font size too small
        legend:
                titleFontSize: font size too small
title: chart needs title
```

Figure 2. New Output with Pretty Print

Images of Changes

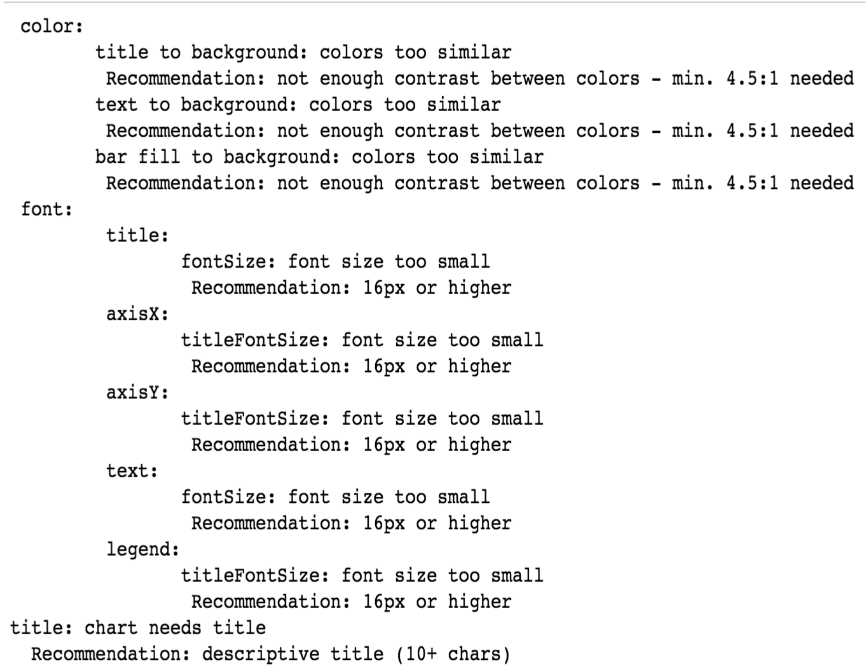
As noted above, originally, our lint software only returned a dictionary object of issues, as shown in Figure 1. We added the pretty print feature, which automatically created prints a more readable output as well as returning the dictionary object, as show in Figure 2. Additionally, we also added an optional verbose option to output. An example pretty printed version of this is shown in Figure 3

USER EVALUATIONS

In order to evaluate the prototype we designed, we now needed feedback from a new population. In this stage of the process, we already had the feedback and requirements from people with visual impairments and our tool was producing more accessible visualizations for them. Thus, the step of evaluating the usability of our solution required us to contact designers with experience using Altair. We developed a demo and a survey for testing the v_a11y_lint and we managed to reach five designers that conducted them.

Evaluation Techniques & Procedures

To evaluate our prototype, we designed a demo and surveys that were conducted among designers that had experience working with Altair. The users first completed a screener survey regarding their experience using relevant programming tools, e.g. Python, the Altair visualization library, to ensure that they were relevant to our target population. Participants then completed the demo using v_a11y_lint that guided them over the process of designing a visualization, testing it with the v_a11y_lint and being able to modify it and make it accessible. Upon completion of the demo, participants completed a final survey regarding their experiences using our product.



```
 color:
        title to background: colors too similar
         Recommendation: not enough contrast between colors - min. 4.5:1 needed
        text to background: colors too similar
         Recommendation: not enough contrast between colors - min. 4.5:1 needed
        bar fill to background: colors too similar
         Recommendation: not enough contrast between colors - min. 4.5:1 needed
 font:
         title:
                fontSize: font size too small
                 Recommendation: 16px or higher
         axisX:
                titleFontSize: font size too small
                 Recommendation: 16px or higher
         axisY:
                titleFontSize: font size too small
                 Recommendation: 16px or higher
         text:
                fontSize: font size too small
                 Recommendation: 16px or higher
         legend:
                titleFontSize: font size too small
                 Recommendation: 16px or higher
title: chart needs title
  Recommendation: descriptive title (10+ chars)
```

Figure 3. Printed Output with Verbose Option Utilized

The survey included both closed and open ended questions. The guiding principles measured in this survey were mostly related with the usability of the v_a11y_lint. We tested: a.) the effectiveness of the tool, i.e.. how well can the user complete the task. b.) the efficiency, i.e. what is the amount of effort the designer needs to apply to complete the task. c.) the satisfaction, i.e. how satisfied were the designers while completing the task. d.) the learning curve, i.e. how hard was to learn to use the tool.

We decided to use this type of evaluation because it was the most time efficient option to get the valuable feedback from the designers, and to be able to summarize and analyze the results efficiently and easily. The time the designers needed to perform the tasks included in our demo and fill both surveys was between 20 and 40 minutes depending on the dedication to their designs. We completed five user tests in total.

Before starting the evaluation, all participants accepted the consent form shown in the appendix and each survey was anonymous and not linked to any particular participant.

Evaluation Results & Feedback

The results and feedback from the evaluation where very rich both to confirm the decisions and modifications done in advance to the prototype and to learn which things need to be improved. In general, all the participants agreed that the tool is useful and easy to use. Moreover, they liked the detailed instructions received in order to improve their visualizations as shown in the newly added pretty print feature. However, they would have liked to see more resources or guidance in terms of accessible combinations of colors with sufficient contrast.

Among our participants there was a concern about their knowledge and possibility to make visualizations accessible. Almost all participants (80%) declared that they do not currently check for accessibility when creating visualizations. Indeed, 60% of participants express that do not feel they have the tools and knowledge to create accessible visualizations. However, after testing v_a11y_lint, all the participants agreed that using the software, they are able to make accessible plots more easily and that they would like to use it in the future to make their visualizations more accessible.

When asked about their favorite aspects of v_a11y_lint, four participants noted that the feedback from the software was clear and straightforward, and that it was easy to understand the aspects of their visualizations that needed to be improved. Moreover, one participant mentioned that the tool was really intuitive and easy to use, and another designer liked that the output received was "nice and clean to read." Finally, all the participants agree that the verbose option was useful since it helped them to easily understand what modification were needed to create an accessible visualization.

In terms of what the designers liked least about v_a11y_lint, there was a shared feeling that it would be useful to link to examples or resources that could guide them in ameliorating the accessibility issues in their visualizations, especially the first time using the tool, to better select colours with enough contrast. One of them mentioned:

> *"It was a little unclear how to modify the colours appropriately - not in the sense of there being bugs in the software, but in the sense that I don't really know the meaning of colour/contrast ratios in the first place (e.g. what does a ratio of 4.5 mean). More guidance on how to pick well-contrasting colours, even just by linking to external resources, might be good."*

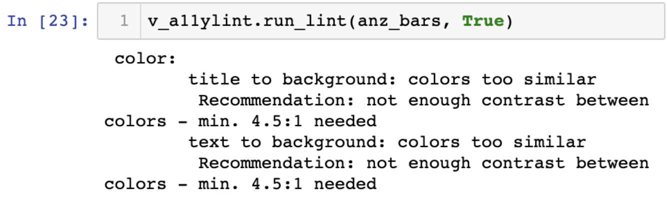
Another participant mentioned that he felt that the contrast recommendation was phrased in a way that sounded like part of the diagnosis. He suggested a modification to better distinguish between the diagnosis and the recommendations.

Proposed Changes in Prototype

To address the designers' feedback, we propose the following changes to implement in the next iteration of v_a11y_lint:

The most repetitive feedback we received was related to the designers lack of knowledge on how to improve their choice of colors to chose those with enough contrast. To address this need we would like to add another argument option to our main function run_lint(). This will probably be called 'examples' or 'resources'. When users set this argument to 'True', they will be provided with examples or external resources that could help them with the requirements to pass the accessibility test related to the flagged issue. For example, in the case of having an error with the level of contrast among the colours selected, the current tool will show a result similar to Figure 4. However, with this new argument set to 'True', they would receive an output like the one on Figure 5.

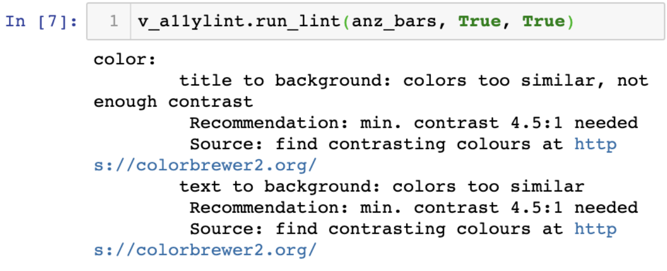
We decided that this additional information should be a different argument that they can chose to activate or not because we were concern the output could get to overwhelming for some designers if it printed paragraphs of issues and suggestions for improvement. We believe that having it as new argument will help developers that are new with this tool to easier improve their visualizations, while allowing more experienced accessible designers to have a cleaner and less verbose output if preferred.



```
In [23]:  v_a11ylint.run_lint(anz_bars, True)

           color:
                   title to background: colors too similar
                    Recommendation: not enough contrast between
          colors - min. 4.5:1 needed
                   text to background: colors too similar
                    Recommendation: not enough contrast between
          colors - min. 4.5:1 needed
```

Figure 4. Current Output



```
In [7]:  v_a11ylint.run_lint(anz_bars, True, True)

         color:
                 title to background: colors too similar, not
         enough contrast
                  Recommendation: min. contrast 4.5:1 needed
                  Source: find contrasting colours at http
         s://colorbrewer2.org/
                 text to background: colors too similar
                  Recommendation: min. contrast 4.5:1 needed
                  Source: find contrasting colours at http
         s://colorbrewer2.org/
```

Figure 5. Improved Output

Figure 4 and Figure 5 also show the second improvement we would like to implement. This one is related to the catch of one of the designers how felt that the diagnosis and recommendation were mixed in the case of a color issued. Thus, we will change the diagnosis to be "colors too similar, not enough contrast" and the recommendation to "min. contrast 4.5:1 needed."

Additionally, in future iterations of v_a11y_lint, we hope to add checks for chart-specific features, e.g. check to ensure there is white space between bars in bar chart or redundantly encode data. These additional features require research regarding best practices and noticeable differences and encodings for low vision viewers [1].

Proposed Changes in Design & Evaluation Technique

Based on this initial feedback we received we would like to make two broader improvements to the design of our tool:

1.) As all of our participants created visualizations in multiple libraries, we would like to expand this lint to make it available in other programming languages. As v_a11y_lint currently lints Altair, a grammar of graphics visualization library, we believe that it could be easily expanded to other grammar of graphics libraries, like Vega-Lite and R's ggplot. With much more additional effort, the idea of linting for visualization accessible should be expanded to other non-grammar of graphics libraries like JavaScript's D3 and Python's Matplotlib.

2.) Ultimately, linting for data visualization accessibility should be automatic, like spellcheck. Instead of our participants had to mostly guess-and-check for accessibility, it would be beneficial if v_a11y_lint became a plugin for Jupyter Notebook, Excel and/or Tableau. The software would then flag accessibility issues as they arise. We believe that the tool we crated is easy to use. However, converting it to a plugin will help more developers to remember to use it and create their visualizations in a more accessible way.

3.) We would like to make our evaluation more robust by surveying more developers and interviewing people with visual impairments in order to improve the feedback this tool displays. Our research shows the extreme lack of standards and dearth of information surrounding accessible visualizations. Therefore, much future research is needed regarding color, spatial and other visual perception in data visualizations for low vision users. For v_a11y_lint to be universally effective, we need information to develop these standards.

[1] We filed an issue with Vega-Lite to add pattern/texture encodings (which would automatically trickle down to Altair) but the maintainers closed the issue as it is not easily done in Vega currently, and there is no way to easily redundantly encode with colors and patterns.

Appendix 1

University of Chicago Online Consent Form for Research Participation

Study Title: Reimagined-Vis: v_a11y_lint

Researcher(s): Kathryn (Katy) Koenig, Lucy Li, Belen Michel Torino

Description: We are researchers at the University of Chicago doing a research study about making data visualizations more accessible to viewers with visual impairments. This research allows us to understand how to make data visualizations more accessible to low vision viewers as well as better incorporate tools for making data visualizations more accessible. This procedure will involve surveys regarding readability of data visualizations, creation of accessible data visualizations and evaluation of a tool to make data visualizations more accessible. The length of this study is estimated as 20 minutes per participant.

Risks and Benefits: Your participation in this study does not involve any risk to you beyond that of everyday life. Taking part in this research study may not benefit you personally, but we may learn new things that could help others.

Rights as a participant: Your participation in this interview is voluntary. If there are any questions in the survey that you do not wish to answer, you may skip them. You may stop participating at any time.

Confidentiality: The researchers will take all possible steps to keep information about you confidential and anonymous, and to protect any information collected as part of this project from unauthorized disclosure, tampering, or damage.

- Your name will not be viewed by anyone except the three researchers in this team.
- All names will be deleted from our data files before saving and preparing the data for analysis.
- All data will be stored in password-protected files on password-protected computers or on secure cloud storage only accessible by members of this research.
- After the project is completed, any data that is shared for the purpose of scientific inquiry will remain completely anonymous and all identifying information will have been removed. When results from this study are shared, your name or the names of the people you mention during the interviews will never be mentioned.
- Any audio recordings and data collected as part of this research will be stored de-identified. Please note that if you decide to withdraw from the study and de-identified materials or data have already been submitted to an archive, it is possible that your data will not be able to be removed. When possible, however, your data will be withdrawn upon your decision to leave the study.

Contacts & Questions: If you have questions or concerns about the study, you can contact the researchers at:

Belen Michel Torino

belenmichel@uchicago.edu

+1(312)493-7726

Consent: Participation is voluntary. Refusal to participate or withdrawing from the research will involve no penalty or loss of benefits to which you might otherwise be entitled.

By choosing “Agree” below, you confirm that you have read the consent form, are at least 18 years old, and agree to participate in the research. Please print or save a copy of this page for your records.

□ **I agree to participate in the research**

□ **I do NOT agree to participate in the research. In this case we thank you for your consideration and we will not conduct the interview.**