

Visualizing the Nonvisual: Screen Readers & Data Visualization

Flannery Currin ♦ fhcurrin15@earlham.edu

Department of Computer Science ♦ Earlham College ♦ Richmond, IN

Introduction & Background

- Data visualizations help most people process large amounts of data and find patterns.
- Many visually impaired people rely on screen readers to navigate information displayed on a computer.
- User testing is the best way to identify social & functional usability concerns, but is not always a realistic step to expect developers to take [1].
- Developers who design data visualizations should be aware of potential accessibility & usability concerns.
- Most developers do not have experience using screen readers and learning to use one for testing purposes may seem prohibitive.
- Testing products for accessibility should be made easier.
- Some tools have attempted to do this in other domains [2].

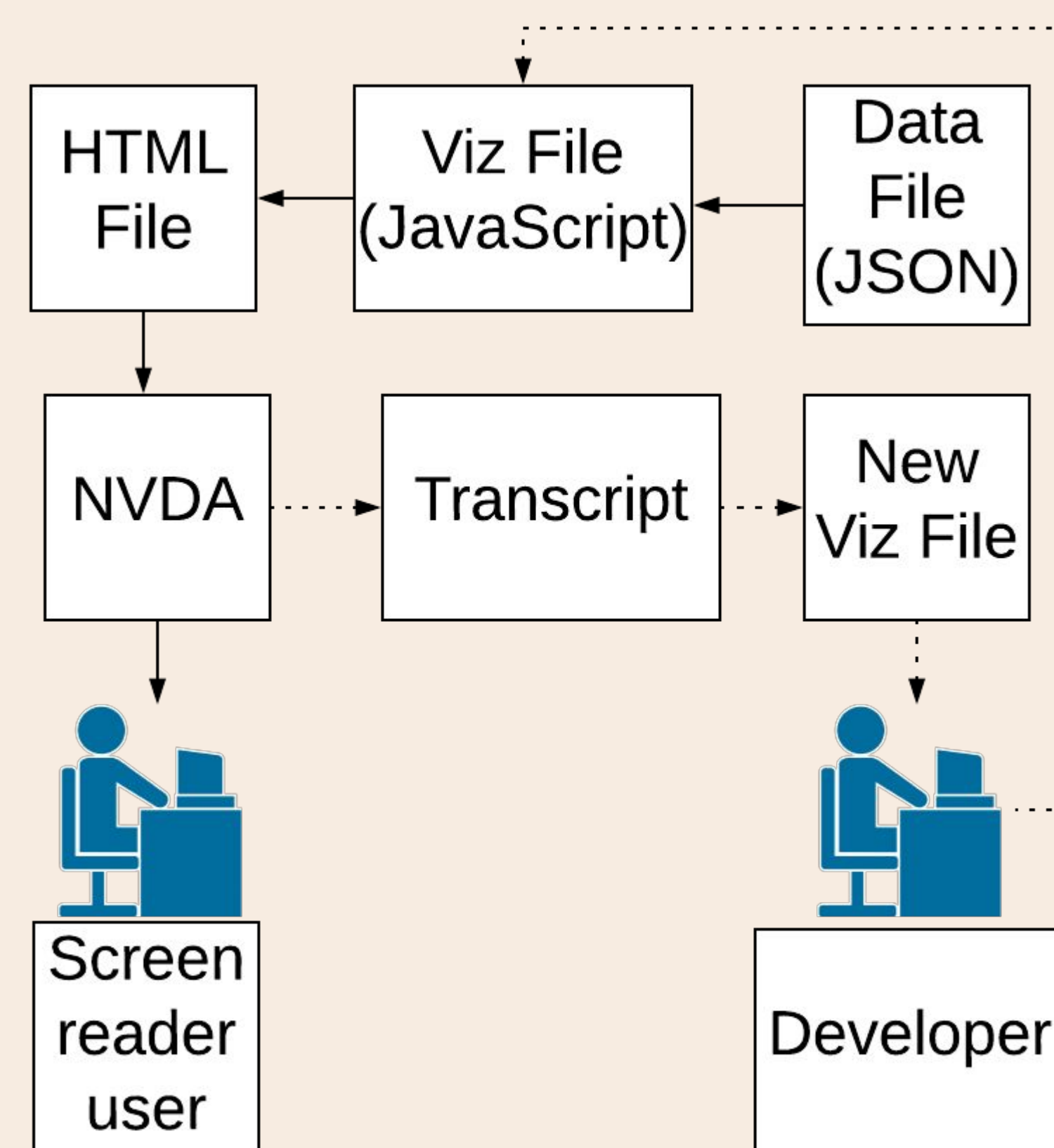


Figure 1. An illustration of the framework for this project. Pre-existing links are represented by solid lines. New links are represented by dashed lines.

Earlham
COLLEGE

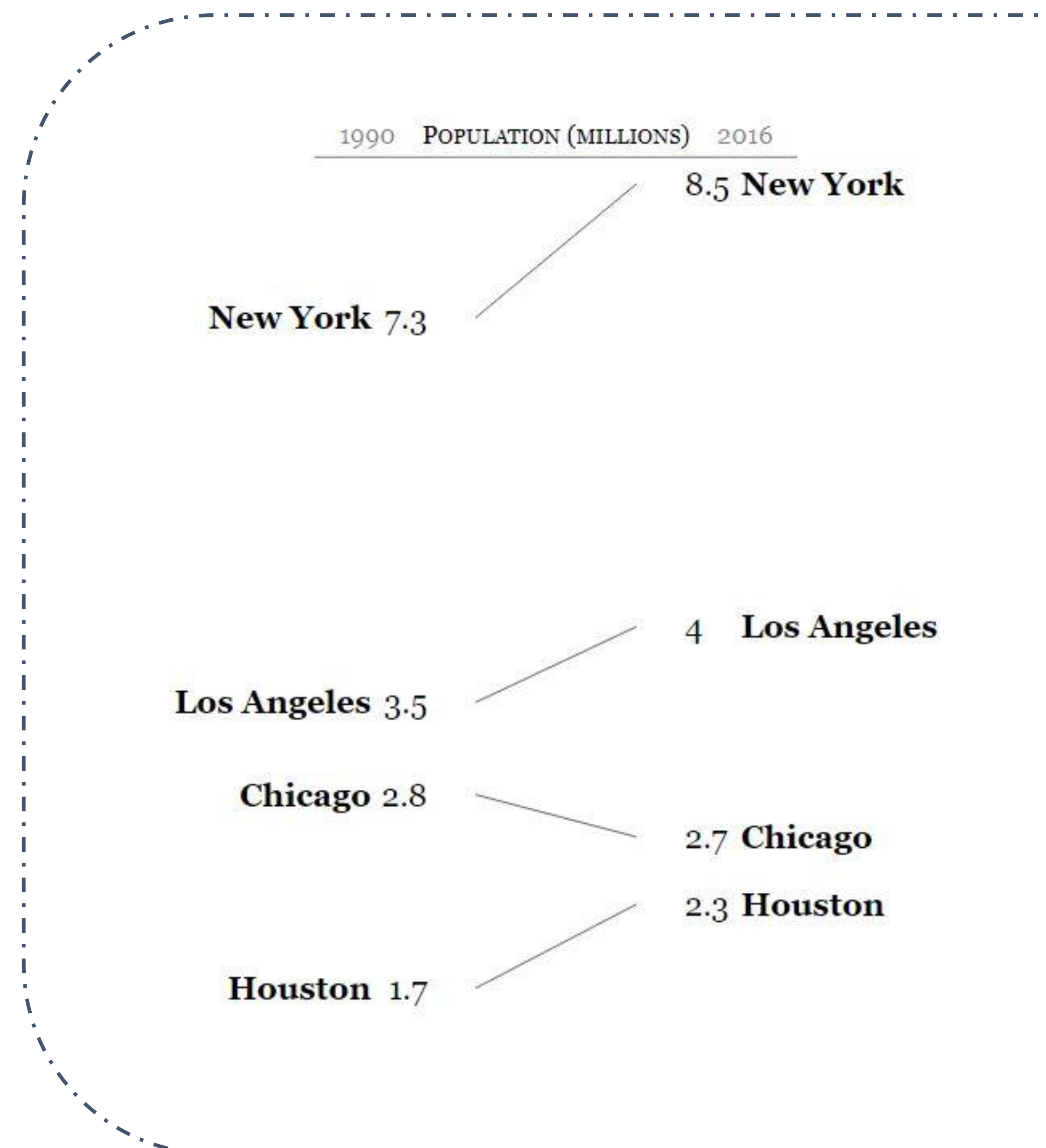


Figure 2. A simple slopegraph visualization showing the population of four major U.S. cities in 1990 and 2016.

Design

- Goal: demonstrate how an open-source screen reader (NVDA) extracts and presents information from data visualizations in a way that is easy for sighted individuals to understand.
- Testing was done with the sample data visualization shown in Figure 2 as well as two other simple visualizations.
- Process:
 - Developer creates a web-based visualization
 - Developer runs NVDA with the new plugin and opens their visualization
 - Plugin transcribes only the relevant information
 - Developer exits NVDA and opens the accessibility visualization creation tool and uploads the transcript
 - Developer can then navigate through the visualization as presented by NVDA

Implementation

- Starting visualization (shown in Figure 2) was created using the D3 JavaScript library.
- The app module for NVDA runs when Firefox is the active program but can be modified to use a non-default program.
- The module creates transcript (a file called log.txt) based on the page content.
- The web tool allows .txt files to be uploaded, generates visualization like the one shown by the series of images in Figure 3.
- Developers can see what content is and is not included, along with the order in which content is presented.

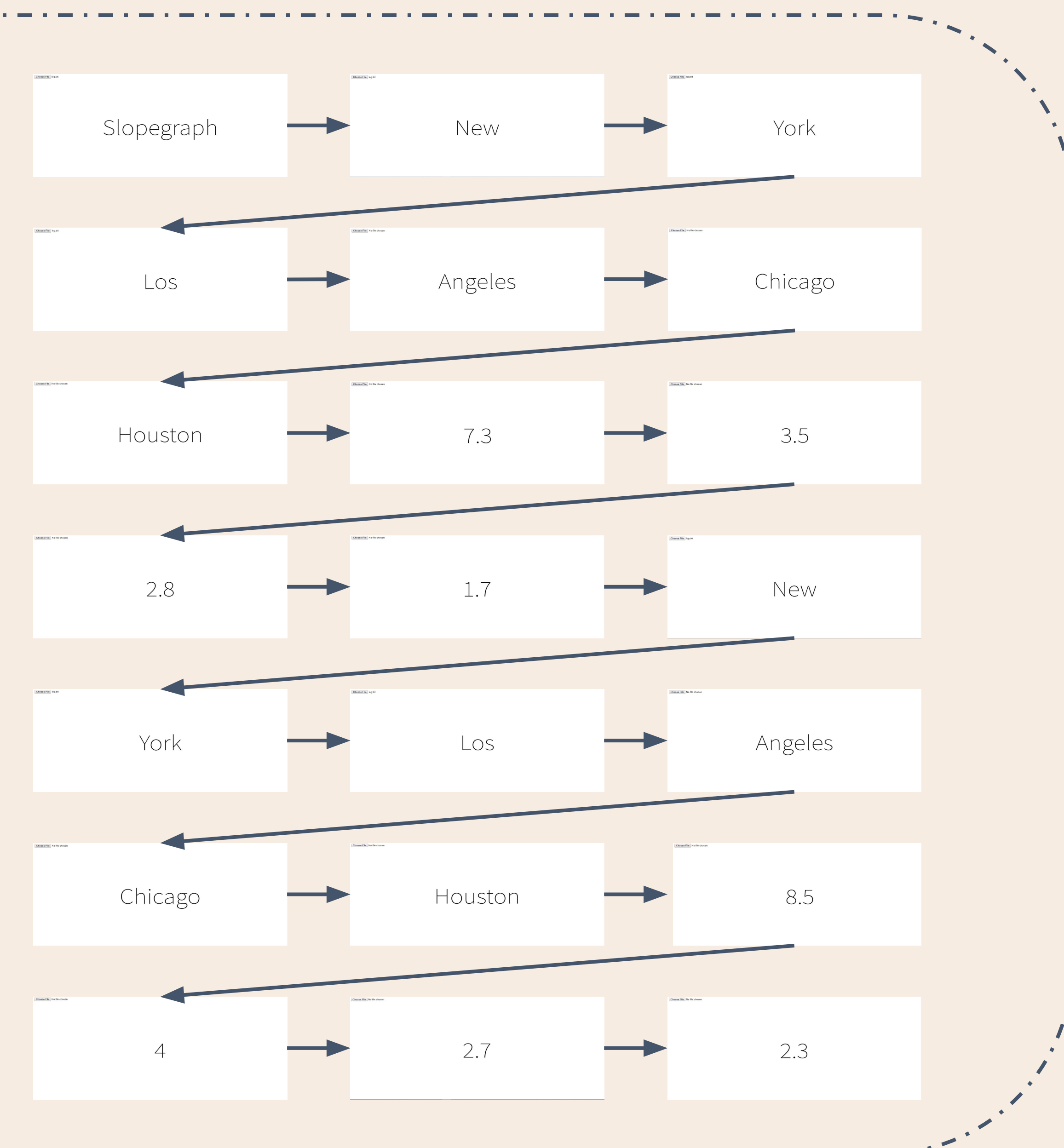


Figure 3. The first several slides of the visualization representing NVDA's parsing process of the chart in Figure 2.

Design Implications

- NVDA picks up and records this information in a confusing order: the cities are not actually linked to the numbers representing their populations.
- The chart and column titles are not read until the end, so users do not know what the information means as it is being presented.
- Visual information like the physical distance between points is not conveyed via NVDA.
- These concerns could be addressed by adding alternative text or re-configuring the order in which the items are stored in HTML.

Discussion & Future Steps

- This tool could make testing for accessibility of smaller-scale web projects (like data visualizations) easier.
- This extends work done by aDesigner [2], so its principles can be applied to a broader range of web content.
- Future work could attempt to streamline this process further.
- Initial user testing will be done in the spring.
- Looking at how this tool performs with a broader range of content could show the extent to which it might actually impact design practices.

References:

- [1] Kristen Shinohara, Cynthia L. Bennett, Wanda Pratt, and Jacob O. Wobbrock. 2018. Tenets for Social Accessibility: Towards Humanizing Disabled People in Design. *ACM Trans. Access. Comput.* 11, 1, Article 6 (March 2018), 31 pages.
- [2] Hironobu Takagi, Chieko Asakawa, Kentarou Fukuda, and Junji Maeda. 2004. Accessibility Designer: Visualizing Usability for the Blind. *In ACM SIGACCESS Accessibility and Computing*. ACM, 177–184.

Acknowledgements: I would like to thank Charlie Peck for help in the project proposal process, Xunfei Jiang for leading the capstone course, David Barbella for serving as my project mentor, and Becky Thomas for providing additional guidance and support. I owe the inspiration for this project in part to the Verification and Validation team of the Analysis in Motion group at Pacific Northwest National Laboratory.