COLLEGE OF ENGINEERING AND COMPUTER SCIENCE

VinUniversity

COMP4010 - Data Visualization

Project 2
Inclusive Data Visualization: A Comprehensive Guide for Accessibility in R with ggplot2 and Quarto

Professor Le Duy Dung
Teaching Assistants Hoang Tien Vu and Luong Ha Tri Nhan

Nguyen Trong Nhan SIS ID: V202000224

June 3rd, 2024

Table of contents

1. Introduction

1.1. Goals

The project "Inclusive Data Visualization: A Comprehensive Guide for Accessibility in R with ggplot2 and Quarto" aims to equip participants with the necessary knowledge and skills to create accessible and inclusive visualizations using the ggplot2 and Quarto packages within the R ecosystem. With a clear understanding of the diverse data visualization tools available in ggplot2 and Quarto, participants will learn how to design visualizations that prioritize accessibility for individuals with diverse needs, including those with disabilities. By mastering techniques for creating accessible visualizations, participants will contribute to making data-driven insights more widely accessible and usable for all audiences. Through practical exercises and guided instruction, the project seeks to empower participants to create visualizations that adhere to accessibility best practices and promote inclusivity in data visualization practices. The project also augments the reader's knowledge and related research in the field of accessible data visualization.

1.2. Motivation

The motivation behind the project stems from a desire to create visualizations that reach a wider audience, including individuals with disabilities, thereby enhancing the inclusivity and equity of the information presented. By focusing on accessible data visualization, we aim to increase the effectiveness of visualizations, ensuring they are clearer and more understandable for all users. This project not only provides participants with the necessary skill set to develop accessible visualizations but also allows them to showcase these skills in their professional portfolios, highlighting their commitment to inclusivity. By contributing to the creation of more accessible and equitable information, the project underscores the importance of making data-driven insights universally usable and comprehensible. Ultimately, this initiative seeks to bridge the gap in data accessibility, promoting a more inclusive approach to data visualization practices.

1.3. Course overview

This course offers a comprehensive introduction to accessible data visualization, structured into six lectures. Each lecture delves into the motivation, problems, definitions, methodologies, and limitations of accessible data visualization. The course includes hands-on coding exercises to equip participants with essential skills in R, Quarto, Plotly, Shiny, and more. By the end of the course, participants will gain practical experience in implementing accessible data visualizations.

This course is a comprehensive guide to data visualization in R, spanning from fundamental concepts to advanced techniques. Lesson 1 introduces key definitions, applications, methods, and challenges in data visualization. In Lesson 2, accessibility principles are discussed, followed by quizzes and R exercises to reinforce these concepts. Lesson 3 delves into accessibility methods, with practical exercises using ggplot2 and Quarto. Lesson 4 covers advanced data visualization, focusing on advanced visualization functions. Lesson 5 introduces interactive data visualization through Shiny, allowing students to create interactive dashboards. Lastly, Lesson 6 explores inclusive data visualization research, providing a holistic view of creating accessible and engaging visualizations.

Lecture	Content	Exercise
Lecture 1	Definitions, Applications, Methods, and Challenges	Common plots using ggplot2
Lecture 2	Accessibility Principles	Quizzes on accessibility principles and R exercises to enhance understanding of common principles
Lecture 3	Accessibility Methods	ggplot2 and Quarto exercises applying accessibility methods
Lecture 4	Advanced Data Visualization	Advanced visualization functions
Lecture 5	Interactive Data Visualization/Shiny	Interactive dashboards using Shiny
Lecture 6	Inclusive Data Visualization Research	---

Table 1: Weekly content of the course

2. Lectures

2.1. Lecture 1 - Introduction to accessible data visualization

The need of data visualization

Data visualization is crucial for transforming raw data into meaningful insights. It offers a powerful way to communicate complex ideas and findings in a clear and understandable manner. By visually representing data, trends, patterns, and outliers become more apparent, aiding in the discovery of hidden relationships and making it easier for stakeholders to grasp the significance of the information presented. Additionally, data visualization enables effective decision-making by providing a comprehensive view of the data, helping to identify areas that require attention or improvement. Overall, data visualization is essential for turning data into actionable insights that drive informed decision-making and facilitate understanding across various audiences.

Types of disabilities:

- Cognitive and learning disabilities: People with dyslexia may have difficulty reading labels and titles, especially with ornate fonts. They may also experience issues with memory, language, reading, and writing.

- Color blindness: Individuals with color blindness may struggle to differentiate between data points within a colored bar chart.

- Blindness: Blind individuals cannot interpret visuals, so they rely on assistive technology like screen readers. Data sonification technology can also be used to convert visual information into sound. Tactile graphics, such as 3D-printed models, help them understand the spatial information of data. However, these solutions can be costly. Another approach is using multimodal representation on touchscreen devices, providing audio and haptic feedback when the reader's fingers touch the graphic.

- Limited literacy: People with limited literacy may struggle with complex data presentations.

- Motor impairment: Individuals with motor impairments may struggle with interactive charts using a mouse or keyboard. They may have difficulty performing actions like pinch-to-zoom or using the lasso tool. Solutions include using single-button interfaces, foot-activated switches, joysticks, sip and puff controllers, eye tracking, and speech input.

- Situational impairments: Factors such as having a hands-on job or low contrast in bright sunlight can create situational impairments, making data difficult to access or read.

- Approximately 15% of the world's population lives with some form of disability. (Source)

Challenges in accessible visualization

Creating accessible visualizations poses several challenges, requiring expertise from various fields such as accessibility, visualization design, disability studies, and research. One of the primary challenges is ensuring that visualizations are accessible to individuals with diverse abilities. This includes making visualizations perceivable, operable, understandable, and robust for users with disabilities.

Another challenge is the cost associated with creating accessible visualizations. Designing and implementing accessible features can be resource-intensive, requiring specialized knowledge and tools. Additionally, maintaining accessibility standards over time can incur ongoing costs.

Furthermore, there is a need for increased collaboration and knowledge sharing among experts in different disciplines to address these challenges effectively. This includes researchers, designers, developers, and accessibility advocates working together to create more inclusive visualizations.

Leveraging the human visual system for accessible content

In designing accessible content, we can leverage the remarkable capabilities of the human visual system. This system possesses a high bandwidth, allowing it to process vast amounts of visual information in parallel. Additionally, the human visual system excels at quickly recognizing patterns and extracting meaningful insights from complex visual stimuli.

Benefits of accessible content

Creating accessible content offers numerous benefits. Firstly, it promotes inclusivity by ensuring that information is accessible to everyone, regardless of age, gender, culture, or disability. This inclusivity leads to a better user experience for all individuals interacting with the content.

Moreover, accessible content can significantly expand the audience reach. By accommodating diverse needs and preferences, accessible content can attract a broader audience and foster greater engagement.

Ensuring accessibility also contributes to the long-term sustainability of content. By following accessibility guidelines, content creators can future-proof their work, making it easier to maintain and update over time.

Additionally, accessibility can drive innovation and creativity in content creation. By challenging creators to think inclusively and design for diverse audiences, accessibility requirements can inspire new ideas and approaches.

Audience for accessible content

Accessible content is designed to cater to a wide range of audiences. This includes individuals of all ages, genders, cultures, and backgrounds. Accessible content is particularly beneficial for people with disabilities or cognitive differences, ensuring that they can access and engage with content on an equal basis with others.

Common tools for data visualization

ggplot2 is a powerful data visualization package in R, known for its declarative and intuitive syntax. It is based on the grammar of graphics, allowing users to create complex and customized plots with relatively simple code. ggplot2 offers a wide range of plot types, including scatter plots, bar charts, line graphs, and more, making it suitable for various data visualization needs. With its ability to handle large datasets and its extensibility through themes and extensions, ggplot2 is a popular choice among data scientists and analysts for creating publication-quality visualizations.

Quarto is a versatile tool for reproducible research and scientific computing. It provides a unified environment for creating documents that integrate code, text, and visualizations. Quarto supports multiple programming languages, including R, Python, and Julia, allowing users to leverage their preferred language for analysis and visualization. With its emphasis on reproducibility, Quarto enables users to easily share and collaborate on projects while ensuring that results are transparent and reproducible. Its flexible layout options and support for interactive elements make Quarto a valuable tool for creating interactive reports, data-driven presentations, and academic publications.

2.2. Lecture 2 - Accessibility principles

Why should we have principles for accessible data visualization?

The motivation behind these principles lies in the fundamental belief that everyone, regardless of their abilities or background, should have equal access to information. By ensuring that data visualizations are accessible, we empower individuals with disabilities to engage with and understand the data. High contrast colors, readable fonts, and clear design not only benefit those with visual impairments but also enhance the overall user experience for everyone. Furthermore, by considering cultural sensitivities and collaborating with accessibility experts, we can create visualizations that are not only inclusive but also respectful and meaningful to diverse audiences. In essence, these principles are about promoting equity and ensuring that data is a tool for empowerment and understanding for all. The creation of these principles is thoroughly drafted in a structured order through my subjective online research, focusing on important notes when creating accessible data visualizations.

Principle 1: Accessibility first

Accessibility is key to ensuring that everyone, regardless of ability, can access and understand data visualizations. This principle encompasses several key aspects:

- **Color accessibility**: Ensure high contrast between background and data elements to aid readability. Consider color blindness and avoid relying solely on colors like red and green, which can be difficult for some individuals to distinguish. Use color and contrast intentionally to highlight important information. Avoid using too many or similar colors that may blend together. Use color schemes that are compatible with different types of color blindness and screen settings. Differentiation can also be achieved through the use of patterns, textures, or shapes.
- **Readable fonts**: Font type and size are crucial for readability. Choose fonts that are appealing and easy to read, especially for users with visual impairments.
- **Iconography**: Use universally recognized symbols or icons to enhance comprehension and make visualizations more memorable.
- **Graphic design**: Always include alternative text for each image to help people with visual impairments understand the figure.
- **Clear and simple language**: Avoid using jargon, acronyms, or complex language. Aim for clarity and simplicity to make the content accessible to non-experts and a broad spectrum of people.
- **Navigation and interaction**: Ensure keyboard, mouse, or neural-link accessibility. Design intuitive layouts with predictable behavior. A consistent layout helps users predict where to find and navigate information more effectively, which is particularly useful for people with cognitive disabilities by reducing cognitive load and improving the overall experience. The layout should also adapt to different devices for a seamless experience..

- **Alternative formats:** Provide options like audio descriptions or tactile representations for data. This could include text summaries or audio narrations to explain main points in a simple and engaging way. Offer downloadable or printable versions of visualizations to allow for offline usage. Incorporate interactive features like tooltips, filters, and sliders to enable users to explore the data at their own pace and level of interest.

By prioritizing accessibility in data visualization, we ensure that everyone can benefit from the insights and information presented, regardless of their abilities or limitations.

Principal 2: User-centered approach

The user-centered approach to creating accessible data visualizations is essential for ensuring that the visualizations effectively communicate information to a diverse audience. This approach involves several key practices: first, knowing your audience by conducting research and surveys to understand their needs and abilities, including the devices they use, their cultural backgrounds, and their demographic information. Second, conducting usability testing and obtaining feedback from people with disabilities to ensure that the visualizations are usable and understandable for everyone. Finally, continuously refining the visualizations based on user feedback through iterative improvement, ensuring that they remain accessible and useful to a wide range of users.

Principal 3: Visual clarity and simplicity

Visual clarity and simplicity are crucial aspects of creating effective and accessible data visualizations. To achieve this, it's important to follow several key guidelines. First, declutter the design by prioritizing key information and removing unnecessary elements that may distract or confuse users. Next, choose the right chart type that best suits the data and the audience's understanding, ensuring that the visualization is both informative and easy to interpret. Finally, pay attention to data labeling and annotations, explaining the data clearly and concisely to provide context and aid comprehension. By adhering to these principles, data visualizations can effectively convey information to a broad audience in a clear and accessible manner.

Principle 4: Beyond vision

Inclusive data visualization extends to considerations for individuals with visual impairments. Providing audio descriptions can make visualizations accessible to visually impaired users by offering narrated explanations of the data. Tactile representations, such as raised elements or textured maps, can further enhance accessibility by allowing users to explore the data through touch. Additionally, ensuring screen reader compatibility is essential, as it enables text elements within visualizations to be accessible to users relying on screen readers for information. These practices enhance the inclusivity of data visualizations, ensuring that they can be accessed and understood by a diverse audience.

Principle 5: Cultural sensitivity

Cultural sensitivity is paramount in creating inclusive data visualizations that resonate with diverse audiences. To achieve this, it's crucial to avoid stereotypes and biases by being mindful of cultural symbols and representations in visuals. Additionally, consider internationalization by providing language translations and cultural formatting of numbers and dates. These practices not only demonstrate respect for different cultures but also ensure that visualizations are meaningful and relevant to a global audience.

Principle 6: Collaboration and advocacy

To further enhance accessibility in data visualization, it is essential to collaborate with accessibility experts. By involving people with disabilities in the design process, you can gain valuable insights and ensure that the visualizations meet the needs of diverse users. Additionally, sharing resources and best practices can help raise awareness and educate others about the importance of inclusive design. Finally, advocating for change by pushing for standards and policies that promote accessibility in data visualization can lead to broader adoption of inclusive practices, ultimately benefiting a larger number of users.

2.3. Lecture 3 - Accessibility methods

Online tools

These tools can be incredibly useful for creating and designing data visualizations:

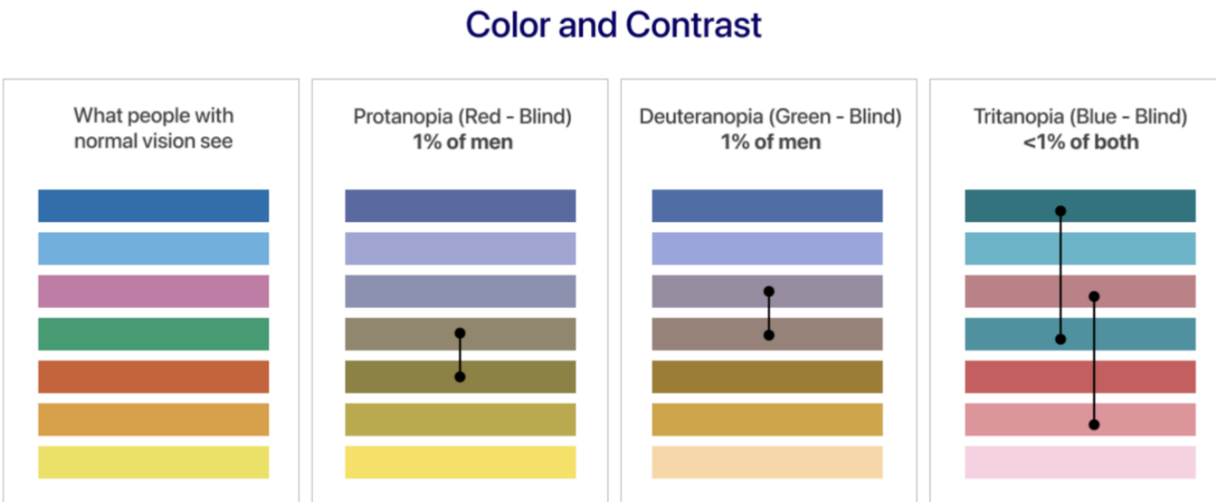
1. Mockup: Mockup is a tool used for creating wireframes of websites and applications. It allows you to quickly sketch out ideas and concepts, helping you visualize the layout and functionality of your design.

2. Moqups: Moqups is a web-based design tool that allows you to create wireframes, mockups, and prototypes. It offers a range of features such as drag-and-drop functionality, pre-designed templates, and collaboration tools.

3. Infogram: Infogram is a tool for creating interactive charts, infographics, and maps. It offers a wide range of templates and customization options, making it easy to create visually appealing data visualizations.

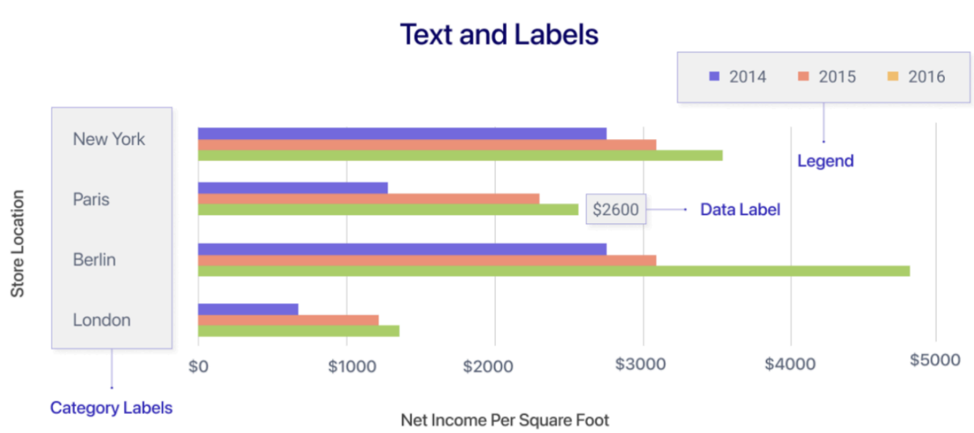
4. Figma: Figma is a collaborative interface design tool that allows multiple users to work on a design simultaneously. It offers features such as real-time collaboration, prototyping, and a variety of design tools to create high-fidelity prototypes and designs.

Color and Contrast



- Prioritize high contrast: Aim for a contrast ratio of at least 4.5:1 between text and background colors. This helps users with low vision or color blindness to read the text easily.
- Use color-blind friendly palettes: Avoid relying solely on color to convey information. Combine colors with patterns, labels, and text explanations to make your visualizations accessible to users with color blindness.
- Provide alternative color schemes: Offer options for users to customize color palettes based on their preferences or needs. This allows users to choose colors that are most distinguishable to them.
- Test color combinations: Use online color contrast checkers or tools like Adobe Color to test the legibility of your color combinations. This ensures that your visualizations are accessible to users with different visual abilities.

Text and Labels

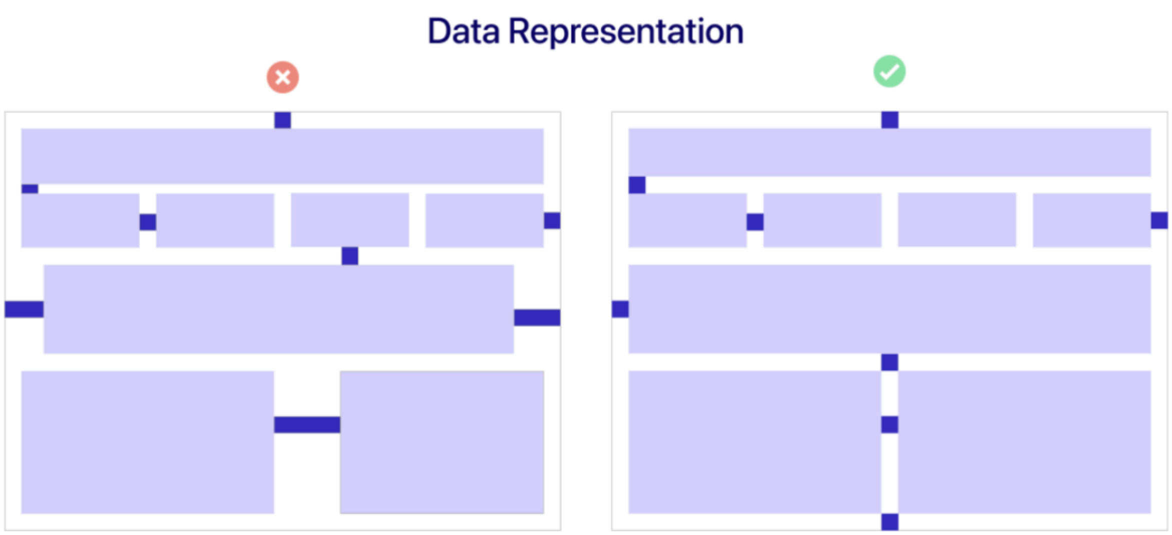


- Use clear, concise language: Avoid using jargon or overly technical terms that may confuse users. Instead, use simple language that is easy to understand.
- Choose legible fonts and sizes: Opt for sans-serif fonts and ensure a minimum size of 12pt for comfortable reading. This helps users with visual impairments to read the text more easily.
- Ensure adequate spacing: Avoid clutter and crowding by providing enough space between text elements. This improves readability and comprehension for all users.
- Provide alternative text: Include descriptive alternative text for images and charts that can be accessed by screen readers. This ensures that users with visual impairments can understand the content of the visualizations.

Layout and navigation

- Organize information logically: Use clear headings, subheadings, and visual cues to guide users through the content. This helps users understand the structure of the information and find what they're looking for more easily.
- Provide alternative navigation options: Consider including a table of contents, breadcrumbs, or search functionality to help users navigate the content. This provides alternative ways to explore the data and can improve accessibility for users with different needs.

Data Representation



- Choose appropriate chart types: Select chart types based on the complexity of the data and the audience's familiarity with different visualizations. Use simple charts like bar or line graphs for straightforward data, and more complex charts like heatmaps or treemaps for detailed data sets.

- Use annotations and labels clearly: Explain the data and any patterns or trends visually using annotations and labels. This helps users understand the information presented in the visualization.
- Provide interactive options: Allow users to filter, sort, or zoom in on data to explore it in more detail. This interactivity can enhance user engagement and comprehension of the data.

Alternative formats

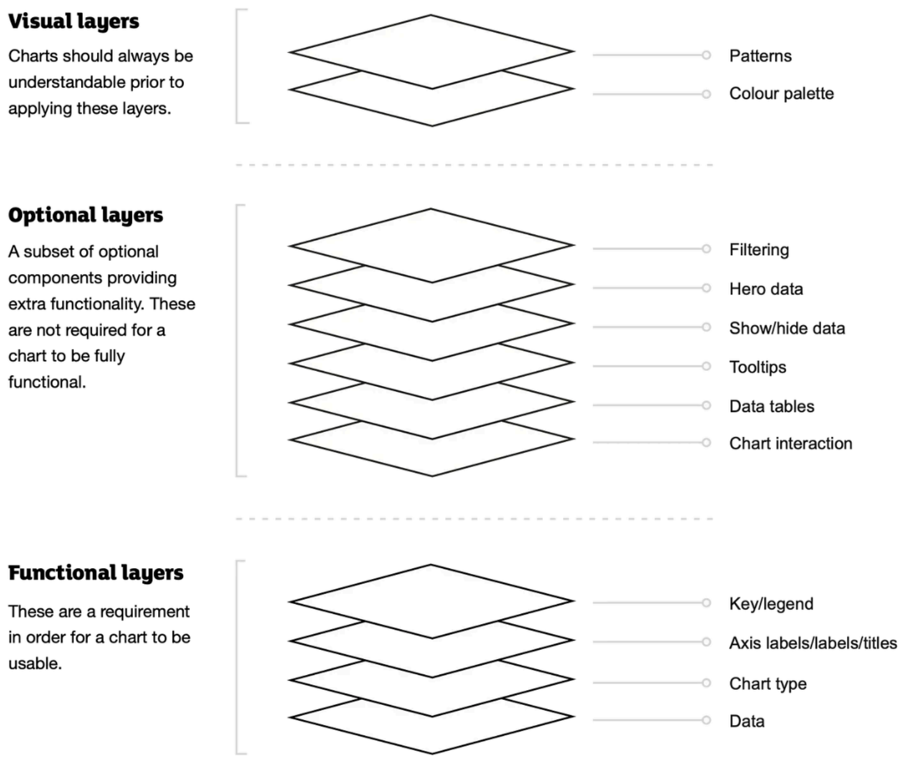
- Offer audio descriptions of visuals: Record narrations that describe the data and its meaning for users with visual impairments. This helps them understand the content of the visualizations without relying on visual cues.
- Consider tactile representations: For users who see differently, consider providing tactile charts like textured maps or 3D-printed graphs. These tactile representations allow users to explore and understand the data through touch.
- Provide transcripts or captions for videos: Make video content accessible to users with hearing impairments by providing transcripts or captions. This ensures that all users can access the information presented in the videos.

In the iterative process of creating data visualizations, it's crucial to consider feedback from both experts and users to continually improve the design and accessibility. By soliciting input from experts in accessibility and data visualization, you can identify potential issues and areas for improvement early in the design process. Additionally, gathering feedback from users, including those with disabilities, allows you to understand their needs and preferences. This ensures that data visualization is a norm, not an exception!

If one want to expand the knowledge on visualization, further readings on design principles can be found here: Designing Charts — Principles Every Designer Should Know | by Ryan Bales, Designing Charts: principles every designer should know (part 2) - Ryan Bales.

2.4. Lecture 4 - Advanced data visualization

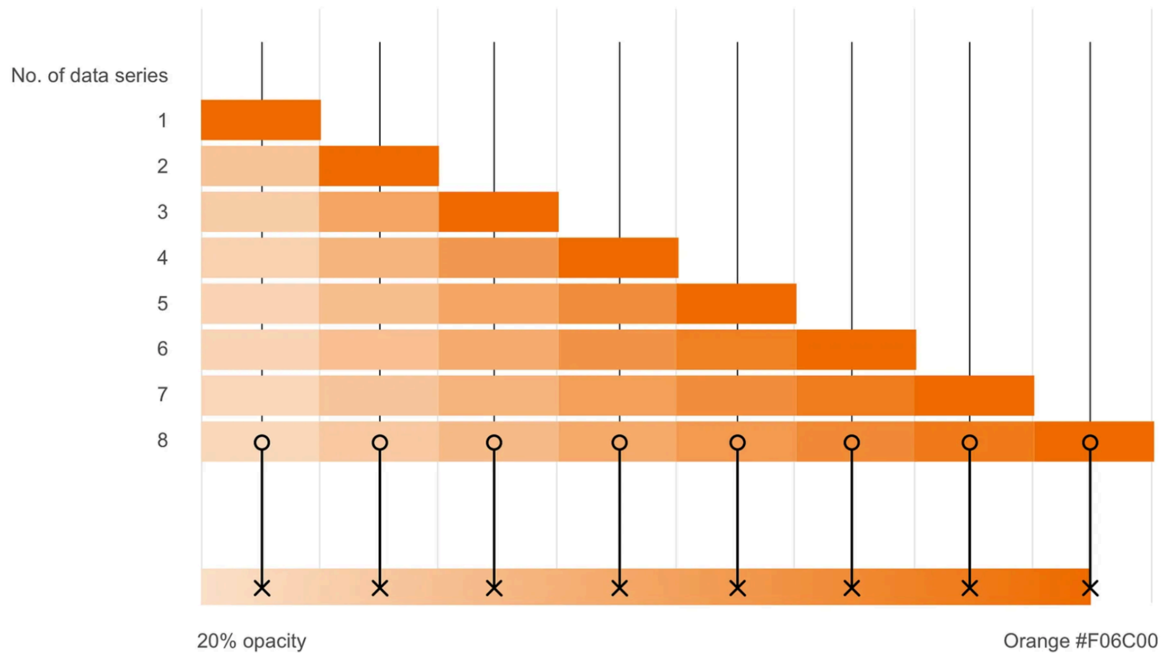
Layering rules



Layering charts according to three principles can help segregate chart elements, facilitating the extraction of data as needed. The three rules—functional, optional, and visual—guide this process. Starting with a functional layer that includes essential elements like data, axis labels, titles, and a key/legend establishes a clear foundation for the chart. Adding optional elements, such as tooltips, the ability to show/hide data, and incorporating visual elements like color and pattern, enhances the chart's accessibility and inclusivity. For this project, doughnut, bar, and line graphs were selected as the preferred chart types.

Multi-hue gradient

Using multi-hue gradients with reduced 20% opacity in accessible data visualization offers several benefits. First, it enhances visual appeal by introducing a smooth transition of colors, which can aid in capturing the viewer's attention and making the data more engaging. Second, it allows for better differentiation between data points or categories, making it easier for users to distinguish between them. This is particularly beneficial for individuals with color vision deficiencies, as the gradual shift in color tones provides additional cues for interpretation. Additionally, the reduced opacity helps prevent visual overload, ensuring that the chart remains clear and easy to read. Overall, these gradients can improve the accessibility and inclusivity of data visualizations by providing a visually pleasing and informative experience for all users.



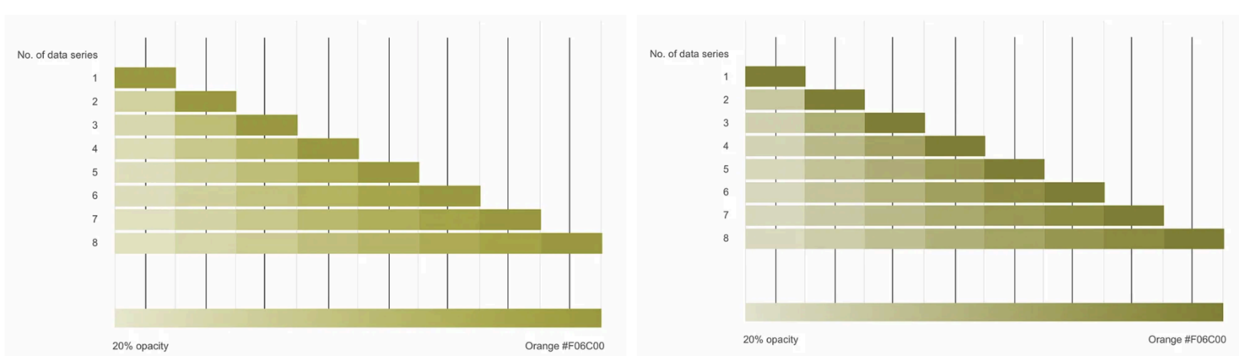
Colours are swatched from the gradient and this is resized on the grid to give greater contrast for the number of data series.

The palette was designed to maintain clear definition for individuals with different forms of color blindness:

- **Deuteranopia**: difficulty distinguishing between red and green
- **Protanopia**: difficulty distinguishing between blue and green, as well as between red and green
- **Tritanopia**: difficulty distinguishing between blue and yellow
- **Greyscale**: total color blindness, viewing in shades of grey (or black and white)

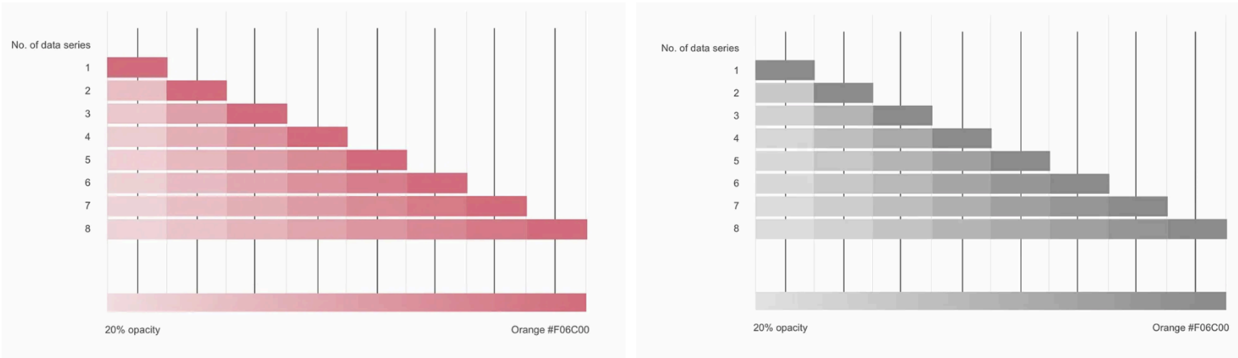
To ensure the effectiveness of the palette, the author of this post utilized a remarkable color blindness simulator, Colour Oracle.

The resulting view of the orange palette with the color simulator:



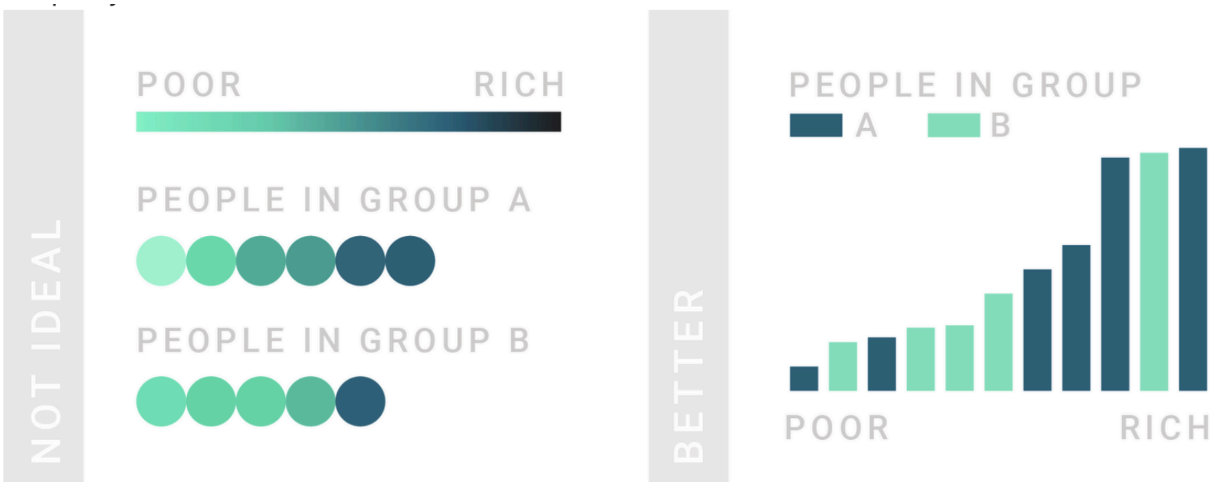
Deuteranopia and Protanopia

Deuteranopia and Protanopia

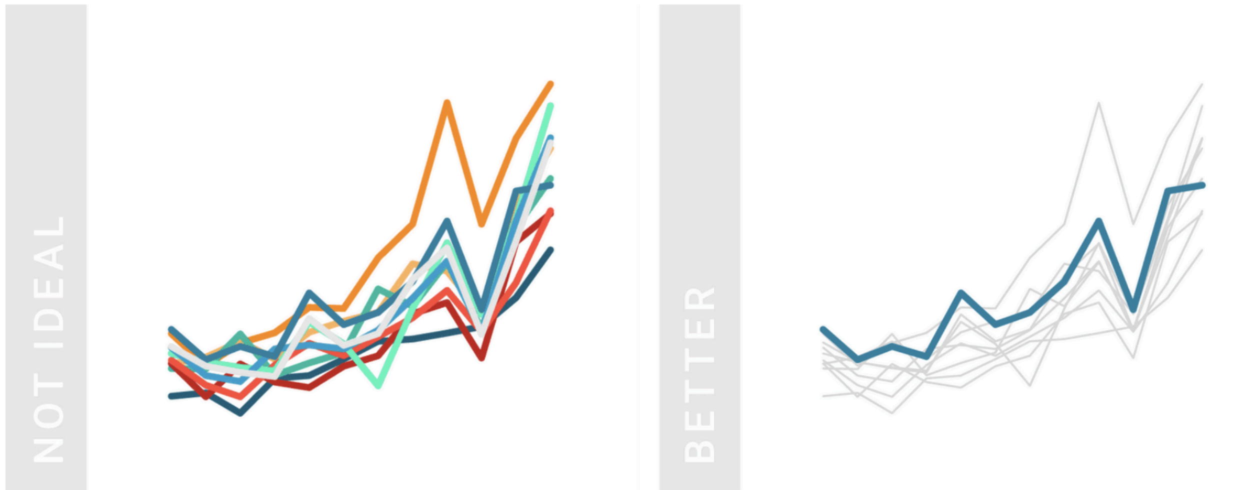


Tritanopia and Greyscale

Choosing the right color for your visualization



When encoding your most important values, explore alternatives to gradient colors. While gradients are effective for illustrating patterns, such as on a choropleth map, they can obscure the actual values and make it challenging to discern differences between them. Instead, consider representing your key values using bars, positional encoding (e.g., in a dot plot), or even areas. Reserve the use of colors for indicating categories. This approach enables readers to decipher values more quickly.



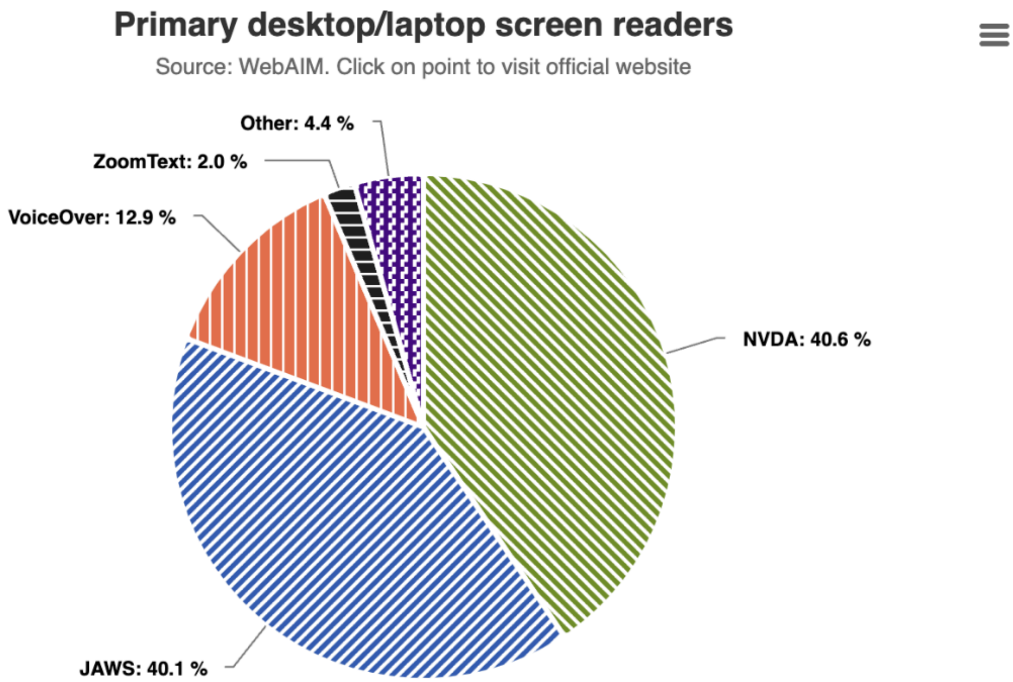
Grey can play a crucial role in data visualization. It can be used for less important elements in your chart, allowing highlight colors reserved for your most important data points to stand out even more. Grey is also useful for indicating general context data, less important annotations, or to show what's unselected by the user. Additionally, it can help in toning down the overall visual impression of your charts. Since grey can sometimes appear cold, consider adding a hint of warmth by using a warm grey (grey with a hint of yellow, orange, or red). Alternatively, you can use another very light color, such as a super light yellow, as an alternative to grey.

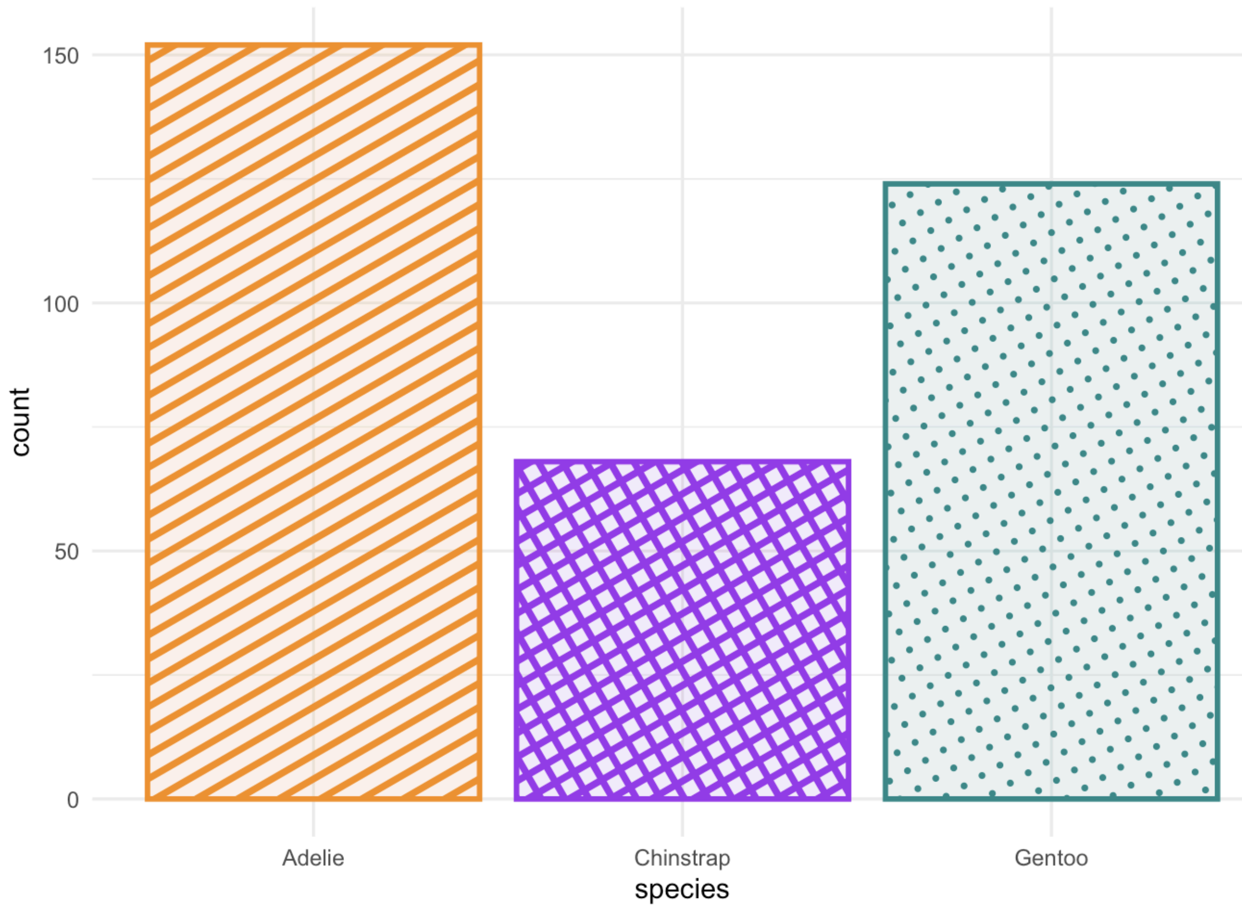
Select colors that are intuitive and meaningful to your audience's culture. When choosing a color palette, consider using colors that readers already associate with your data, such as party colors (e.g., Republican = red, Democrats = blue), natural colors (e.g., forest = green, lake = blue), or learned colors (e.g., red = attention/stop/bad, green = good/to go). When encoding gender data, consider avoiding the stereotypical pink-blue combination. Instead, use a cold color for men (e.g., blue or purple) and a warmer color for women (e.g., yellow, orange, or warm green) to prevent confusion among your readers.

If you're looking for more tips on coloring, you can refer to this post.

Using color patterns

☑ Enable color patterns





Color patterns, when used alongside colors in data visualization, offer a range of benefits that enhance understanding and communication of complex information. Patterns can provide an additional layer of distinction, making it easier to differentiate between categories or groups, especially for individuals with color vision deficiencies. They also improve accessibility by offering alternatives to color-based distinctions. Furthermore, patterns can increase the aesthetic appeal of visualizations, making them more engaging and memorable. When combined effectively, colors and patterns create a visually rich experience that helps audiences interpret data more quickly and accurately.

Choosing the right font for your visualization

This can be done by following these tips:
- Choose a font style that suits your audience.
- Avoid using imposter letter shapes that closely resemble other letters.

Upper case ‘i’, lower case ‘L’ and 1 in Gill Sans

Upper case ‘i’, lower case ‘L’ and 1 in Verdana

- Minimize the use of mirroring letter shapes.

BBC Reith on the left has unique shapes whilst Public Sans on the right employs mirroring

- Ensure letters are easily distinguishable from one another.

Microsoft Sans on the left and Trebuchet on the right

- Humanist typefaces are more legible at smaller sizes compared to grotesque typefaces.

Focus on the spaces between the letters

- Ensure adequate letter spacing in the chosen typeface.

Helvetica Neue on the left and Lucida Sans on the right

- Maintain a visible difference between capital height and ascenders.

Microsoft Sans on the left, Public Sans on the right

- Test the suitability of any typeface or font in its intended context.

2.5. Lecture 5 - Interactive visualization

History of interactive visualization

Interactive data visualization has evolved to accommodate various user needs, including accessibility for impaired users. Early tools like Power BI Desktop introduced keyboard interactions to enable navigation within visualizations. However, these interactions became challenging as visual models grew in complexity.

For instance, scatter plots with many closely located points or network visualizations posed difficulties for users to explore effectively using only keyboard inputs.

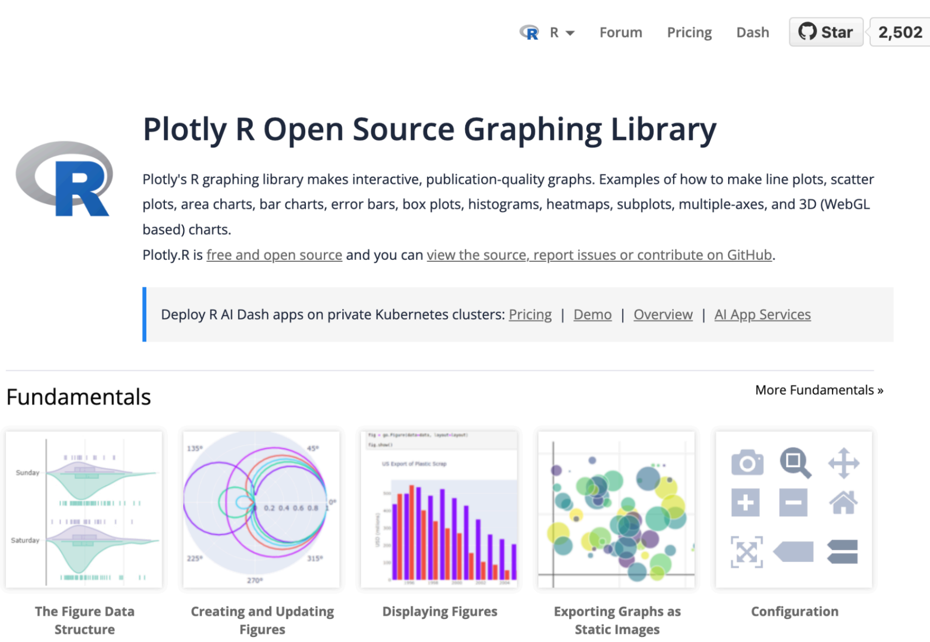
To address these challenges, the Olli open-source Javascript library was developed. Olli converts existing visualization specifications into keyboard-navigable hierarchical structures. This transformation supports self-guided data exploration, generating an accessible HTML tree containing textual descriptions.

Olli's hierarchical structure allows impaired users to explore a visualization from the overview down to the details. It offers different levels of information granularity, enhancing the accessibility of complex visualizations.

Motivation

The motivation behind building interactive visualizations stems from the desire to enhance data comprehension and decision-making processes. Interactive visualizations enable users to explore data from multiple perspectives, uncovering patterns, trends, and outliers that might otherwise go unnoticed in static representations. By allowing users to interact with data, such as filtering, zooming, and highlighting specific elements, interactive visualizations empower them to extract deeper insights and make more informed decisions. Furthermore, interactive visualizations facilitate communication of complex information to a broader audience, fostering engagement and understanding across diverse groups of users.

Available tools for interactive data visualization



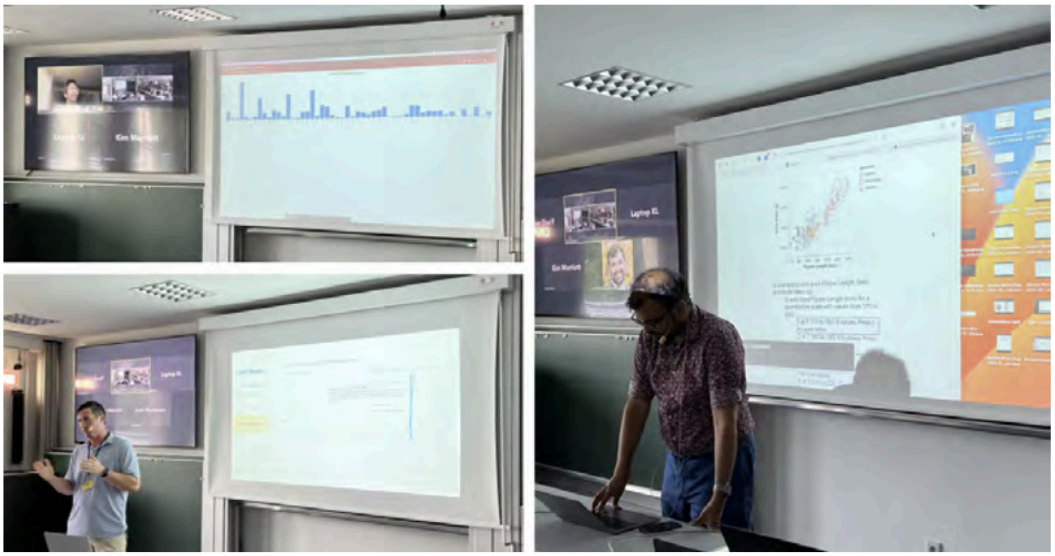
[Plotly](#) and [Shiny](#) are powerful tools used for creating interactive data visualizations in R. Plotly is a graphing library that allows users to create a variety of interactive plots and charts, including line charts, scatter plots, and bar charts. It offers a wide range of customization options, such as hover effects, zooming, and panning, enabling users to explore data dynamically. Shiny, on the other hand, is a web application framework for R that allows users to build interactive web applications directly from R. By integrating Plotly with Shiny, users can create highly interactive and responsive web applications that enable real-time data visualization and analysis. This integration makes it easy for users to develop interactive dashboards, data exploration tools, and interactive reports, making complex data more accessible and understandable.

2.6. Lecture 6 - Accessible data visualization research

Research groups

The Dagstuhl Seminar on Inclusive Data Visualization convened a diverse group of researchers and practitioners from visualization, accessibility, human-computer interaction, and health informatics fields. This gathering aimed to advance the understanding and practice of inclusive data visualization, emphasizing the importance of making visualizations accessible to all users, including those with disabilities. More details on the gathering can be found in the [Inclusive Data Visualization document.]()

Typical latest work on accessible data visualization



- **Olli: An extensible visualization library for screen reader accessibility**: *Olli, an open source library that converts visualizations into a keyboard-navigable structure accessible to screen readers. Using an extensible adapter design pattern, Olli is agnostic to the specific toolkit used to author the visualization. Olli renders a chart as an accessible tree view following the HTML Accessible Rich Internet Applications (ARIA) standard. The fields participating in the visualization serve as branches of the tree, and levels of the tree correspond to different granularities of data (e.g., major axis regions, minor axis regions, individual data values). Users can navigate up and down the tree using the up/down arrow keys, or move between sibling nodes using the left/right arrow keys. Users can also jump to specific positions in the tree via a series of drop down menus, or press the "T" key to invoke a data table view for more traditional row-by-row, column-by-column navigation.*

- **Chart Reader: Accessible visualization experiences designed with screen reader users:** *We demonstrate Chart Reader, an accessibility engine that renders web visualizations optimized for screen reader access. By designing and developing Chart Reader during a five-month iterative co-design study with 10 blind or low vision people, we aim to improve accessible visualization experiences. Our approach, realized through three sequentially designed and developed prototypes, allows users to interrogate visualizations using keyboard interactions, resulting in multimodal audio (announcements and sonification) of the chart. The web-based accessibility engine generates bar charts, stacked bar charts, and single-/multiseries line charts.*

- **VoxLens: An interactive JavaScript library to make online data visualizations accessible to screen-reader users:** *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, in this demonstration, 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 multimodal 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-based information querying.*

Current tools for evaluation

Chartability, a tool highlighted in the discussion, provides a set of heuristics for evaluating the accessibility of data visualizations, systems, and interfaces, offering practical guidelines for ensuring inclusivity in visual communication.

3. Final product - Assignments for each homework

My final products would be assignments for each lecture. The purpose is to provide an opportunity for students to learn, practice and demonstrate what they have learnt in each lecture. Besides, it will serve as the evidence for me to ensure that the students have achieved my learning objectives.

Assignments for each lecture can be found at this Github link. The file "HW1.rmd" is the assignment for lecture 1, and same will go to the rest. For Lecture 6, there is no homework since it is a research lecture

Note: From this point to the end of Part 3, all information will serve as an instruction for learners to do their homeworks.

3.1. Homework 1:

Explanation of components:

This homework aims to teach participants how to create common data visualizations using the ggplot2 library in R, using the Iris dataset as an example. The key components of it include:

1. **Scatter plot**: Visualizes the relationship between two continuous variables (Sepal.Length vs. Sepal.Width).
2. **Scatter plot with color by species**: Adds a categorical variable (Species) to the scatter plot to differentiate between the species.
3. **Histogram**: Shows the distribution of a single continuous variable (Sepal.Length).
4. **Box plot**: Compares the distribution of a continuous variable across different levels of a categorical variable (Sepal.Length by Species).

5. **Violin plot**: Combines aspects of a box plot and a kernel density plot to compare distributions (Sepal.Length by Species).
6. **Pair plot**: Visualizes pairwise relationships between multiple variables using the GGally package.

Explanation of features:

The features of this homework are designed to enhance the understanding and accessibility of data visualization:

1. **Interactive and customizable visuals**: The use of ggplot2 allows for highly customizable and interactive visualizations. Users can modify aesthetics, themes, and scales to suit their needs.
2. **Comprehensive coverage of common plots**: The inclusion of scatter plots, histograms, box plots, violin plots, and pair plots covers a wide range of common visualization techniques, providing a robust foundation for further exploration.
3. **Accessible code and explanations**: Each section includes the R code needed to create the visualizations, accompanied by explanations that make it easier for users to understand and replicate the plots.

User manual:

The following guide helps users access, edit, and utilize the visualizations created in this homework:

1. **Accessing the visualizations**:
    - Ensure R and RStudio are installed on your computer.
    - Install the required libraries by running install.packages(c("ggplot2", "GGally", "datasets")) in your R console.
    - Open the R Markdown file (HW1.Rmd) in RStudio.
2. **Editing the visualizations**:
    - The R code for each plot is provided within code chunks in the R Markdown file.
    - To modify a plot, edit the code within its respective code chunk. For example, to change the color of points in the scatter plot, modify the aes(color = Species) parameter.
    - Re-knit the R Markdown file by clicking the "Knit" button in RStudio to generate the updated HTML document.
3. **Using the visualizations**:
    - The HTML output generated by knitting the R Markdown file will display the visualizations.
    - Users can interact with the plots, such as zooming in on specific areas or highlighting data points, depending on the visualization type.
    - The visualizations can be saved as images or incorporated into reports and presentations by exporting them from the HTML output.
4. **Maintaining the visualizations**:
    - Regularly update R and the libraries used (ggplot2, GGally) to ensure compatibility and access to the latest features.

- Keep the R Markdown file organized by commenting on the code and maintaining a clear structure for ease of future edits.

3.2. Homework 2:

Explanation of components

This homework includes quizzes, examples, and practical exercises aimed at teaching the principles of accessible data visualization. The key components of it include:

1. **Quizzes**: Designed to test understanding of key accessibility principles, including color accessibility, user-centered design, visual clarity and simplicity, considerations beyond vision, and cultural sensitivity.
2. **Examples with code**: Demonstrations of how to implement accessible data visualizations using the ggplot2 and ggpattern libraries in R. The examples include high contrast bar plots, pie charts with clear labels, and bar plots with patterns.
3. **Explanations**: Detailed explanations of why each principle is important and how to apply it in practice.

Explanation of features

The features of this homework are designed to enhance the understanding and accessibility of data visualization:

1. **Interactive quizzes**: Quizzes help reinforce the principles covered in the lectures, making the learning process interactive and engaging.
2. **Practical examples**: Real-world examples demonstrate how to implement accessible data visualizations using R, providing hands-on experience.
3. **Clear explanations**: Each example is accompanied by an explanation of the principle being applied, making it easier for users to understand and replicate the techniques.

User manual

The following guide helps users access, edit, and utilize the content created in this homework:

1. **Accessing the content**:
   - Ensure R and RStudio are installed on your computer.
   - Install the required libraries by running install.packages(c("ggplot2", "ggpattern")) in your R console.
   - Open the R Markdown file (HW2.Rmd) in RStudio.
2. **Editing the content**:
   - The R code for each example is provided within code chunks in the R Markdown file.
   - To modify an example, edit the code within its respective code chunk. For example, to change the colors used in the bar plot, modify the scale_fill_manual parameter.

- Re-knit the R Markdown file by clicking the "Knit" button in RStudio to generate the updated HTML document.

3. **Using the content**:
   - The HTML output generated by knitting the R Markdown file will display the quizzes and visualizations.
   - Users can interact with the quizzes to test their understanding of the principles.
   - The visualizations can be saved as images or incorporated into reports and presentations by exporting them from the HTML output.
4. **Maintaining the content**:
   - Regularly update R and the libraries used (ggplot2, ggpattern) to ensure compatibility and access to the latest features.
   - Keep the R Markdown file organized by commenting on the code and maintaining a clear structure for ease of future edits.

3.3. Homework 3:

Components of the Homework 3

a. Libraries used:

1. **ggplot2:** For creating static and aesthetically pleasing visualizations.
2. **ggthemes:** For additional theming options to enhance the visual appeal and readability of the plots.
3. **plotly:** For adding interactivity to the visualizations, allowing users to explore the data in more detail.

b. Visualization techniques:

1. **Bar charts:** Utilized for representing categorical data with high contrast and color-blind friendly palettes.
2. **Line charts:** Used to depict trends over time, with annotations for emphasizing key data points.
3. **Interactive elements:** Implemented using plotly to provide dynamic features like tooltips and interactive annotations.

Features of the product

a. High contrast and color-blind friendly palettes:

- **Value:** These palettes enhance the accessibility of the visualizations, ensuring that they are readable by a broader audience, including those with visual impairments.
- **Feature:** Two distinct color palettes are provided. One is specifically tailored for color-blind viewers, using colors that are easily distinguishable by individuals with different types of color vision deficiencies. The second palette offers an alternative scheme.

b. Interactive visualizations:

- **Feature:** Interactive plots created using plotly, which allow users to hover over data points to see detailed information and interact with the data.

- **Value:** These interactive elements make the data more engaging and help users to understand and explore the data more deeply. The ability to zoom, filter, and highlight data points adds to the user's ability to extract insights.

c. **Annotations and labels:**

- **Feature:** Key data points are annotated to provide additional context and highlight important trends or observations.
- **Value:** Annotations help in storytelling by drawing attention to significant aspects of the data, making it easier for users to interpret the visualizations.

d. **Theme customization:**

- **Feature:** Use of ggthemes to apply minimalistic and high-contrast themes that improve the readability and visual appeal of the plots.
- **Value:** Clear and uncluttered themes help users focus on the data itself, enhancing the overall user experience.

User manual

a. **Accessing the product:**

Installation: Users need to have R and RStudio installed. The necessary libraries can be installed using the following commands in R:

```
install.packages("ggplot2")

install.packages("ggthemes")

install.packages("plotly")
```

b. **Using the product:**

- **Running the script:** Users can run the provided QMD file in RStudio to generate the visualizations. The file contains both static and interactive plots.
- **Modifying the product:** Users can modify the data and the appearance of the plots by changing the relevant sections in the script. For instance, to change the color palette, users can update the `color_blind_palette` or `alternative_palette` variables.
- **Exploring interactive plots:** Interactive plots created with plotly can be explored by hovering over data points to see tooltips, zooming in on specific sections, and interacting with annotations.

c. **Maintaining the product:**

- **Updating libraries:** Ensure that the ggplot2, ggthemes, and plotly libraries are up-to-date to avoid compatibility issues.
- **Customizing further:** Users can extend the functionality by integrating additional libraries or adding more interactive features as needed. They can refer to the documentation for ggplot2 and plotly for advanced customization options.

3.4. Homework 4:

Components of the Homework 4

a. **Libraries used:**
1. **ggplot2:** For creating static and aesthetically pleasing visualizations.

b. Visualization techniques:

1. **Bar charts:** Utilized for representing categorical data with high contrast and color-blind friendly palettes.
2. **Line charts:** Used to depict trends over time, with annotations for emphasizing key data points.

Features of the product

a. **Accessible Multi-Hue Gradient for Each Level of Granularity in the Data Series**
- **Value:** These palettes enhance the accessibility of the visualizations, ensuring that they are readable by a broader audience, including those with visual impairments.
- **Feature:** This script generates a visually accessible multi-hue gradient based on a specified brand color and applies it to different levels of data granularity. The gradient is created by varying the opacity of the base color, and this gradient is then applied to a grid plot. Additionally, the script includes a function to adjust the gradient based on the number of data series.

b. **Using Color to Emphasize the Most Important Data**
- **Feature:** It plots three data series, with two in grey for context and one in blue to emphasize its significance.
- **Value:** This script demonstrates how to use color to highlight the most important data in a line chart.

c. **Intuitive Color for Pie Chart:**
- **Feature:** Each pie chart represents different datasets with colors that are easy to interpret (e.g., green for “Good”, red for “Bad”).
- **Value:** This script creates pie charts with intuitive color schemes for different categories.

d. **Changing Font:**
- **Feature:** The example uses the “Roboto” font for all text elements in the plot.
- **Value:** This script shows how to change the font in a ggplot2 plot by using Google Fonts.

e. **.Color Patterns**
- **Value**: This script creates a pie chart with different colors and patterns for each category.
- **Feature**: Patterns add an extra layer of differentiation for better visual accessibility.

User manual

a. **Accessing the product:**

Installation: Users need to have R and RStudio installed. The necessary libraries can be installed using the following commands in R:

```
install.packages("ggplot2")
```

b. Using the product:

- **Running the script:** Users can run the provided QMD file in RStudio to generate the visualizations. The file contains both static and interactive plots.
- **Modifying the product:** Users can modify the data and the appearance of the plots by changing the relevant sections in the script. For instance, to change the color patterns, users can update the `scale_pattern_manual` variables.

c. Maintaining the product:

- **Updating libraries:** Ensure that the ggplot2 libraries are up-to-date to avoid compatibility issues.
- **Customizing further:** Users can extend the functionality by integrating additional libraries or adding more interactive features as needed. They can refer to the documentation for ggplot2 for advanced customization options.

3.5. Homework 5:

Components of the Homework 5

a. Libraries used:

1. **ggplot2:** For creating static and aesthetically pleasing visualizations.
2. **shiny**: Shiny is an open source R package that provides an elegant and powerful web framework for building web applications using R. Shiny helps you turn your analyses into interactive web applications without requiring HTML, CSS, or JavaScript knowledge.

b. Visualization techniques:

1. **Scatter plot:** A scatter plot is useful for visualizing the relationship between two continuous variables.

Features of the product

a. Interactive click to change variables

- **Value:** Study variables quickly without running the code block and render the file again
- **Feature:** User can change the target variable within an ease of a click

b. Slider for a short-ranged variable

- **Value:** Helps quickly filter out the necessary data points.
- **Feature:** For this MTCars dataset, users can quickly show the scatter plots of three types of variables that satisfy the number of cylinders by sliding the min and max values

User manual

a. Accessing the product:

Installation: Users need to have R and RStudio installed. The necessary libraries can be installed using the following commands in R:

```
install.packages("ggplot2")
```

```
install.packages("shiny")
```

b. **Using the product:**

- **Running the script:** Users can run the provided QMD file in RStudio to generate the visualizations. The file contains both static and interactive plots.
- **Modifying the product:** Users can modify the data and the appearance of the plots by changing the relevant sections in the script. For instance, to change the variables, users can update the `x and y and color` variables.
- **Exploring interactive plots:** Interactive plots created with shiny can be explored by click-to-change action

c. **Maintaining the product:**

- **Updating libraries:** Ensure that the ggplot2 libraries are up-to-date to avoid compatibility issues.
- **Customizing further:** Users can extend the functionality by integrating additional libraries or adding more interactive features as needed. They can refer to the documentation for ggplot2 for advanced customization options.

4. Limitations

This lesson plan is designed to be a springboard for your journey into accessible data visualization. We will focus on building a strong foundation, covering the key principles, common design tips, and practical exercises to get you started. This hands-on approach ensures you grasp the fundamentals and can confidently create clear, accessible visualizations.

While this plan caters primarily to beginners, we understand some of you might be data visualization veterans. For those hungry for more, I would recommend them to head to Lecture 6, which dives into cutting-edge research pushing the boundaries of accessibility. Given the limited resources, exciting new frontiers like sonification (transforming data into sound) and tactile graphics have not been covered. In addition to this, R is also the only coding language supported, which might cause some unfamiliarities for some people.

5. Future Directions

Neuralink, a company founded by Elon Musk and a team of seven scientists, introduced the world to a pioneering brain interface designed to augment ordinary human ability. Recent studies even show that it can enable a paralyzed person to play chess on a computer without using their limbs to move the cursor. Many ideas have been proposed to leverage the powerful application of the technology. I think it is plausible to conduct research into this sector to connect the Neuralink interface with data visualization. One possible idea can be using Neuralink to help augment the visualization or interact with the visualization dashboard for disabled people.

Building on this strong foundation, future iterations of this lecture series could delve deeper into cutting-edge accessibility research. Lecture could explore how advancements like sonification (transforming data into sound) and haptic feedback (using touch) can be incorporated for a truly inclusive data experience. Additionally, researchers working on Brain-Computer Interfaces (BCIs) like Neuralink could showcase how data visualization might be directly linked to neural pathways for individuals with physical limitations. This forward-thinking approach would solidify the series' position at the forefront of accessible data visualization education. Final examinations should also be designed to assess students' comprehension of the content and coding skills for accessible data visualization on some real-world datasets.

6. References

1. Inclusive Data Visualization
2. Quarto
3. https://medium.com/the-readability-group/a-guide-to-understanding-what-makes-a-typeface-accessible-and-how-to-make-informed-decisions-9e5c0b9040a0
4. https://fossheim.io/writing/posts/apple-dataviz-a11y-tutorial/
5. https://www.dataquest.io/blog/what-to-consider-when-choosing-colors-for-data-visualization/
6. https://uxdesign.cc/making-data-visualization-accessible-a-case-study-e5fb41ac62ad
7. https://silviacanelon.com/blog/2021-09-23-data-viz-a11y/
8. https://uxdesign.cc/designing-charts-principles-every-designer-should-know-5bd3969a0150
9. https://uxdesign.cc/designing-charts-principles-every-designer-should-know-part-2-ce1e06af56fc
10. https://www.mokkup.ai/blogs/exploring-inclusive-principles-in-data-visualizations/
11. https://dl.acm.org/doi/pdf/10.1145/3457875