VINUNIVERSITY

Data Visualization - COMP4010

Project 2: Final Report

Interactive Data Exploration and Visualization Using a Shiny R Application

Nguyen Duc Loi | Nguyen Thai Uyen | Nguyen Duy Minh

I. Introduction.

In the era of data-driven decision-making, data visualization is indispensable in nearly every field. Beyond data wrangling and selecting the best type of visualization, selecting the right aesthetics is a crucial part of data visualization. However, this process may involve a lengthy trial-and-error approach and a great deal of tinkering with visualization code, which can prove daunting to those new to ggplot, R or data visualzation in general. Our Shiny R application aims to simplify this by providing an intuitive interface that allows users to browse and upload CSV files from their local computer, select variables of interest, and visualize differences across datasets. Additionally, the app offers customizable themes specifically designed for colorblind users, enhancing accessibility and user experience. Moreover, the application includes a real-time data display page, providing dynamic insights into live data streams.

This Shiny R app bridges the gap between raw data and actionable insights, empowering users to make informed decisions based on comprehensive and easily interpretable visualizations. By integrating user-friendly interfaces with robust analytical capabilities, the app caters to both novice users and experienced data analysts, making data exploration more intuitive and efficient.

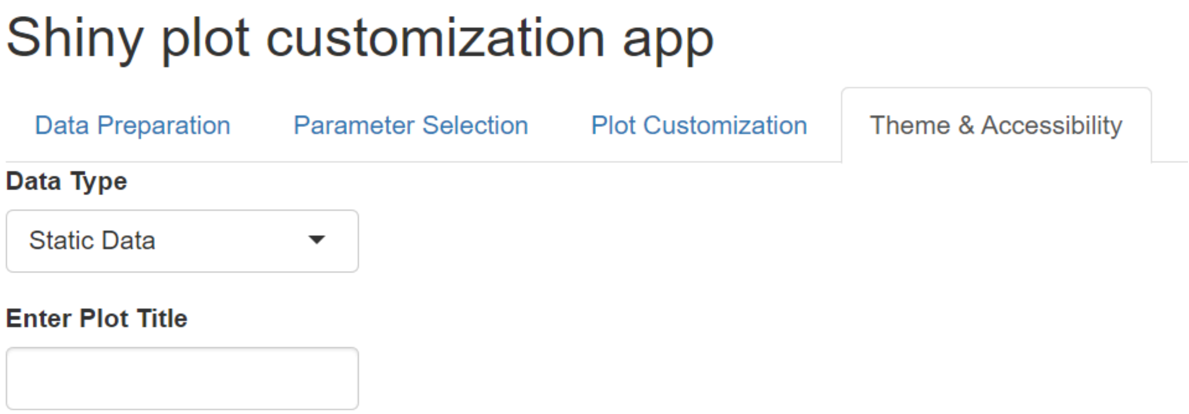
II. Objectives and features.

Our Shiny R application is designed to leverage the power and flexibility of R, enhanced with HTML beautifiers to create an intuitive and visually appealing user interface. The primary objectives of our app are:

- **Facilitate Data Upload and Exploration**: Allow users to work on their dataset.
- **Ease of use:** Simplify the design process for visualizations by providing themes and adjustable features.
- **Enhance Accessibility:** Provide customizable themes specifically designed for colorblind users.

To accomplish this, our app features the following:

- **Data Upload:** Enable users to browse and upload CSV files from their local computer effortlessly.
- **User-Friendly Interface**: Ensure the application is easy to use for both novice and experienced data analysts. In addition our have been divided into multiple tabs and each tab is for a specific stage in data visualization. This makes sure the visualization pipeline follows a logical procedure such as Data loading > Parameter Selection > Plot Customization > Theme & Accessibility



Shiny plot customization app

Data Type: Static Data

Enter Plot Title

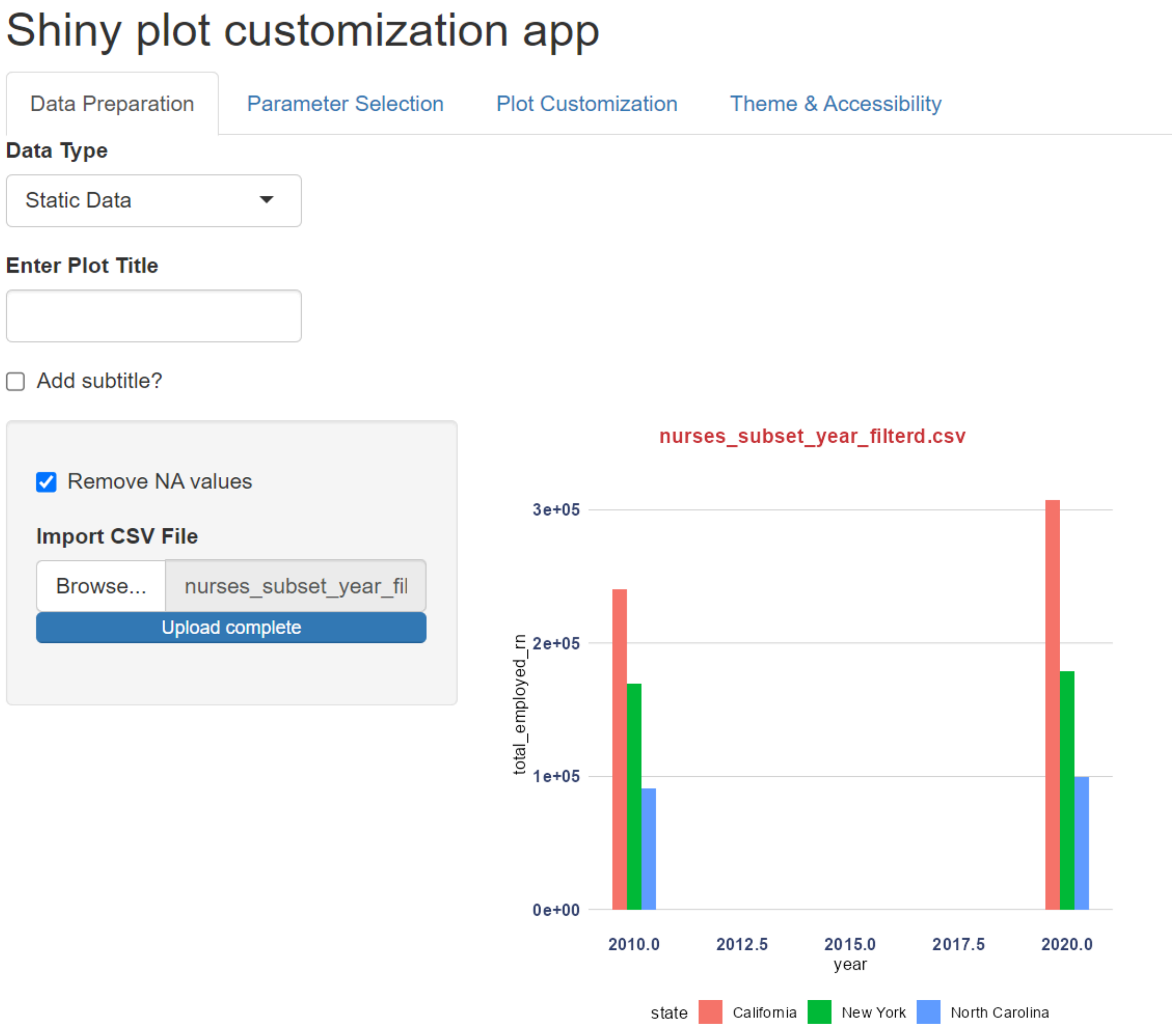
- **Variable Selection and Analysis**: Allow users to select variables of interest and visualize differences across datasets.
- **Comprehensive Data Visualization**: Offer a variety of visualization options to support detailed data analysis.
- **Real-Time Data Interaction**: Incorporate reactive functions to display and analyze real-time data.

By combining the foundational knowledge from our course with innovative solutions we have developed ourselves, our app provides a robust platform for data-driven decision-making. The integration of reactive functions, in particular, allows users to interact with and gain insights from real-time data, enhancing the overall utility of the application.

III. Core Functionalities - Implementation Details.

1. CSV File Browsing and Variable selection.

Users can browse and upload CSV files from their local computer, enabling them to quickly access and analyze their data. Once users successfully upload a file, they can choose which type of variable to plot out, and there is also an option for them to plot a second chart if needed.



Shiny plot customization app

Data Type: Static Data

Enter Plot Title:

□ Add subtitle?

☑ Remove NA values

Import CSV File: nurses_subset_year_fil

Upload complete

Implementation details:

- **File Input Widget:** The app uses Shiny's fileInput widget to allow users to browse and select CSV files from their local computer.
- **Data Handling:** Once uploaded, the CSV file is read into R using *read.csv()*, ensuring efficient data loading and processing.
- **Data Validation**: Basic validation checks are performed to ensure the uploaded file is in the correct format and contains valid data.
- **Dynamic UI Elements:** Shiny's *selectInput widget* is used to dynamically generate drop-down menus based on the columns of the uploaded CSV file.

2. Data Visualization.

An indispensable part of a R Shiny app is to provide various visualization options, including bar charts, scatter plots, and line graphs, to help users interpret their data effectively.

Implementation details:

- **Visualization Libraries:** The app leverages powerful R visualization libraries such as ggplot2 and plotly to create various types of plots (bar charts, scatter plots, line graphs).
- **Customization Options:** Users can customize the appearance of the plots, such as changing colors, labels, and themes, using additional UI controls (sliderInput, colorInput).

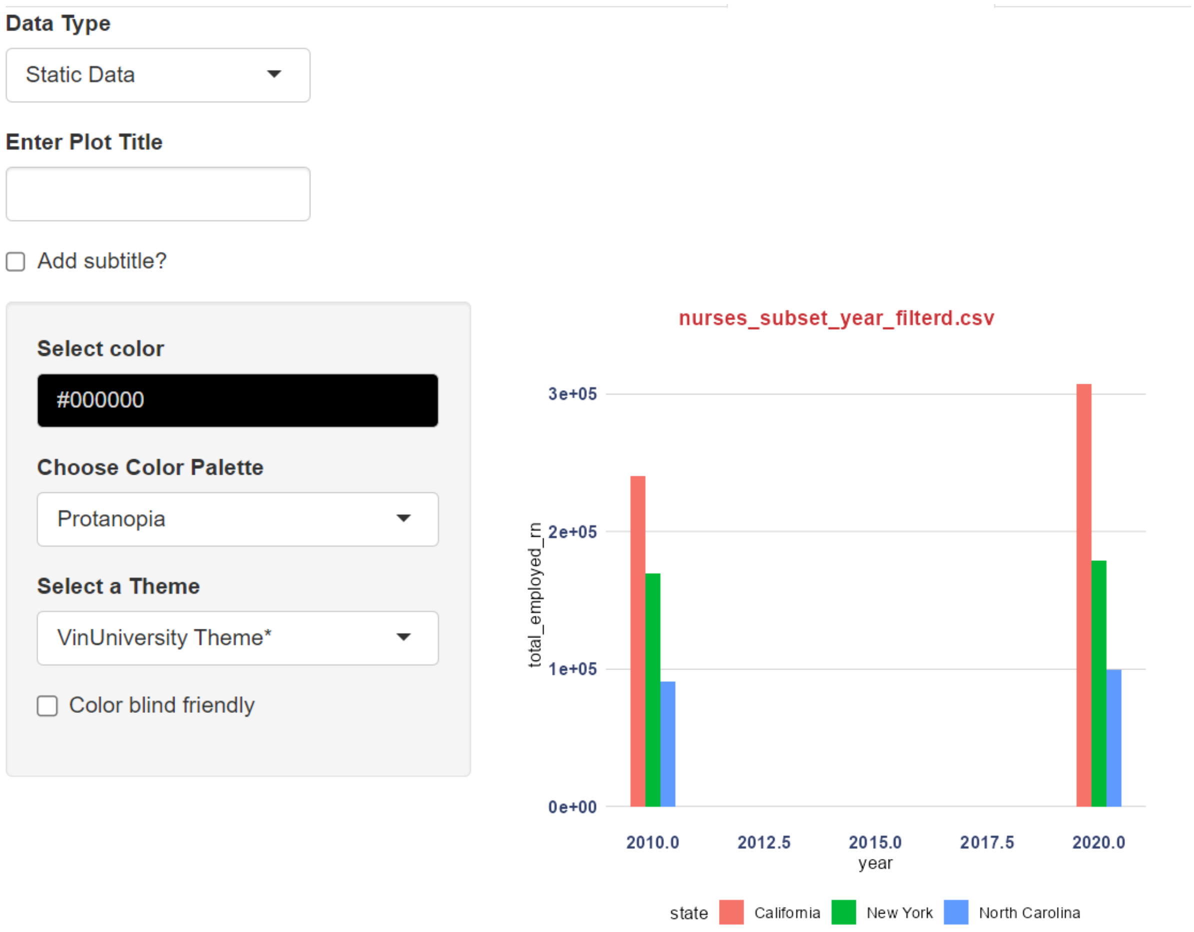
By incorporating these detailed implementation strategies, our Shiny R application provides a comprehensive and user-friendly platform for data exploration and visualization. Each functionality is designed to be robust, efficient, and accessible, ensuring that users can derive meaningful insights from their data with ease.

3. Theme Customization for Colorblind Users.

Other than applying what we explore ourselves, we also make use of what we learned in the lab, and our app includes customizable themes that are designed to be colorblind-friendly.

Implementation details:

- **Predefined Themes:** The app includes several predefined colorblind-friendly themes that can be applied to all visualizations. We use *selectInput()* for user to select their preferred themes. User can choose from various common theme such as **light**, **dark**, **void**, **minimal**, **classic** and **grey**. Moreover, we also include a special theme called **VinUniversity** (*theme_vinuni()*)Theme with the color palette that support branding of our university (see more detail about this theme in the server.R)



Data Type

Static Data

Enter Plot Title

□ Add subtitle?

Select color

#000000

Choose Color Palette

Protanopia

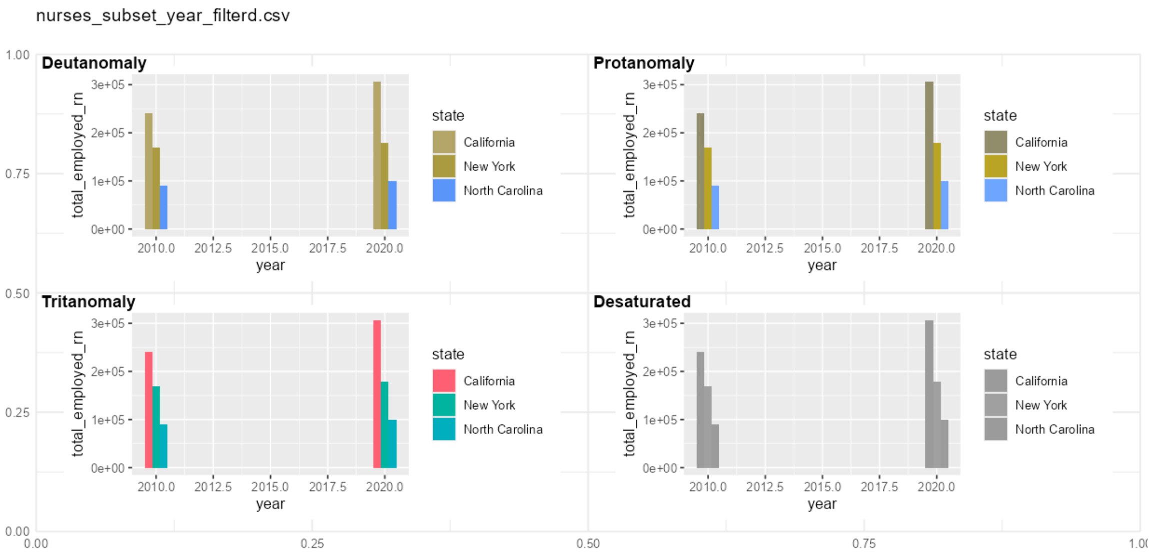
Select a Theme

VinUniversity Theme*

□ Color blind friendly

nurses_subset_year_filterd.csv
3e+05
2e+05
1e+05
0e+00
total_employed_rn
2010.0
2012.5
2015.0
2017.5
2020.0
year
state 🟥 California 🟩 New York 🟦 North Carolina

- **Dynamic Theme Switching:** Users can switch between themes using a selectInput widget, with immediate visual feedback as the theme is applied to the current plots.
- **Accessibility Enhancements:** The themes are designed based on research and best practices to ensure maximum readability and contrast for colorblind users. We apply the best practice to support colorblind by using colorblindr library, which support different strategies for colorblind such as **Deutanomaly**, **Protanomaly**, **Tritanomaly**, and **Desaturated** that make the applicant more user-friendly for the colorblind.



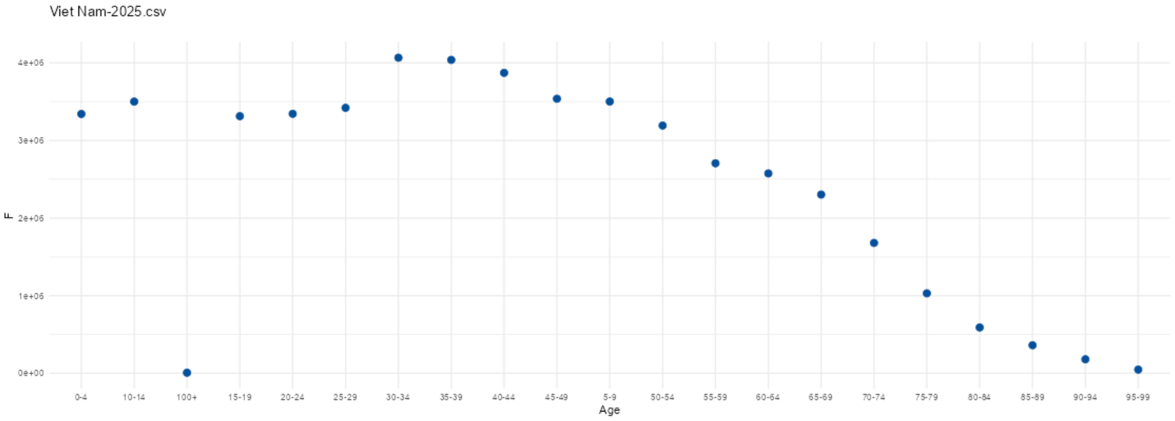
4. Real-Time Data Display.

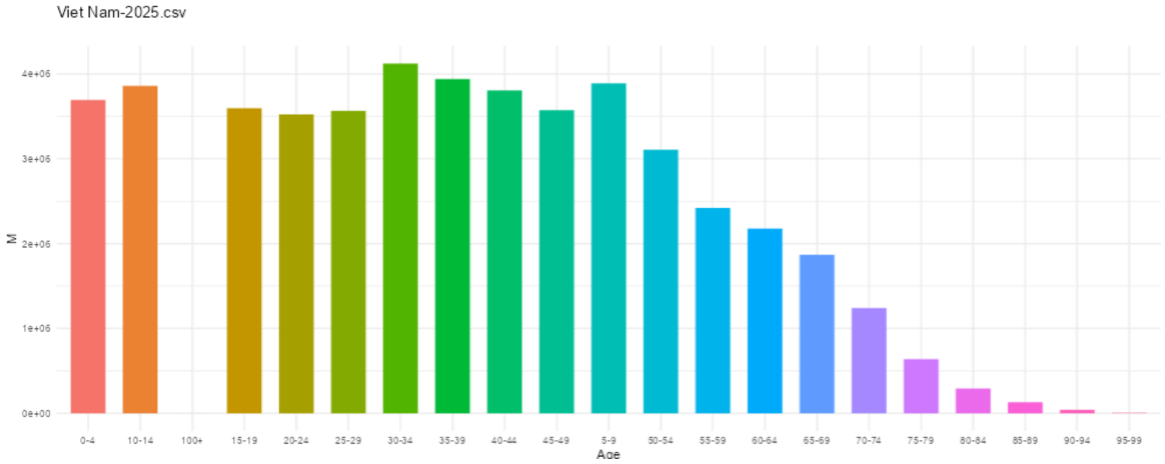
The application features a real-time data display page, leveraging reactive functions to provide up-to-date insights and analysis. Otherwise, in this page, we successfully implement a search table for data and a button for screenshot, which makes it more practical.

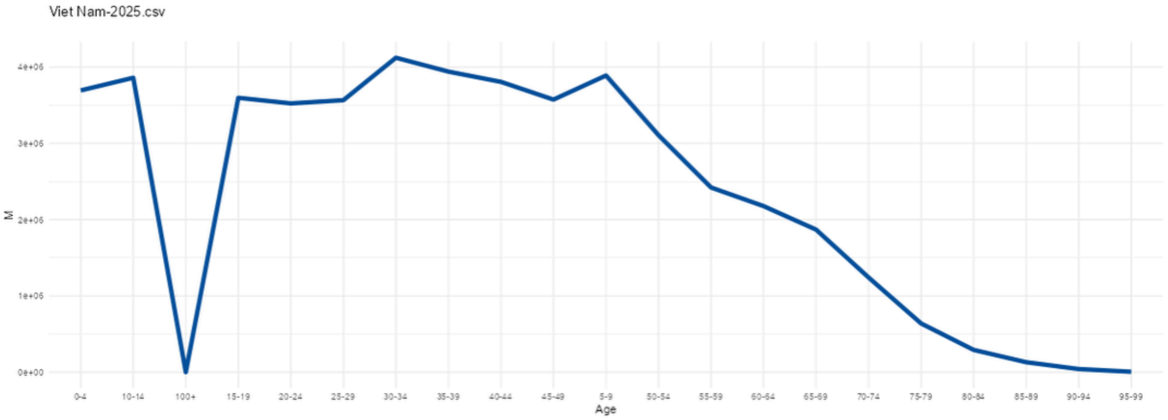
Implementation details:

- **Reactive Polling:** The app uses Shiny's reactive polling features, we use *reactiveFileReader()*, to periodically check for new data and update the display in real time.
- **Live Updates:** The visualizations and data summaries are automatically refreshed in response to new data, ensuring that users always have access to the latest information.
- **Data Summary**: We have an option for users to show a summary for the data using *summaries().*
- **Take Screenshot:** Using *shinyscreenshot::screenshot()* from webshot library allows users to take a screenshot of the whole dashboard. We also implement a simple notification in the corner to notify users about the screenshot by using *showNotification().*

  Here are some of the plot results







IV. Limitations - Potential Enhancements.

Compared to what we proposed at the beginning of the project, we have implemented several meaningful and valuable features. However, there are still goals that we have not yet achieved, and there are limitations in our current implementation that we aim to address in future updates.

1. **Limited chart options:**
    - **Current State**: The application currently offers only three types of charts for users to choose from: scatter, line, and bar charts.
    - **Impact**: While these chart types are versatile, they are insufficient for visualizing all types of data in real-world scenarios. Complex datasets might require more specialized visualization techniques such as histograms, pie charts, heat maps, or box plots.
    - **Potential Enhancements:** Future versions of the app could include histograms, pie charts, and heat maps. This would allow users to choose the most appropriate visualization for their specific data, improving the app's flexibility and usability.

2. **Imperfect Real-Time Data Handling**:
    - **Current State**: The real-time data feature is functional but not flawless. Specifically, if a user is searching for specific data in the table below a chart, any new data update causes the entire chart to reload, which resets the search filter and the user's place in the table.
    - **Impact**: This issue disrupts the user experience, making it challenging to maintain focus on specific data points when the dataset is frequently updated.
    - **Potential Enhancements**: Implement more sophisticated reactive programming techniques to prevent the search filter and table state from resetting during data updates. This could involve optimizing the way data updates are handled or using incremental data loading strategies to update only the necessary parts of the visualization without affecting the entire component.

V. Conclusion.

Our Shiny R application represents a significant advancement in user-friendly data exploration and visualization. It allows users to upload CSV files, select variables, and generate insightful visualizations with ease, all while offering customizable themes for colorblind users and real-time data displays.

While we have achieved many of our initial goals and introduced advanced features, there are still areas for improvement. Expanding chart options and refining real-time data handling are top priorities. Future updates will also focus on enhancing data processing capabilities, improving the user interface, and enabling integration with external data sources.

Overall, our application demonstrates the powerful synergy of R and Shiny in creating an accessible and robust tool for data-driven decision-making. We remain committed to addressing current limitations and implementing future enhancements to further empower users in their data analysis endeavors.