

# REDESIGN PROJECT

## STAT 515: Midterm Project

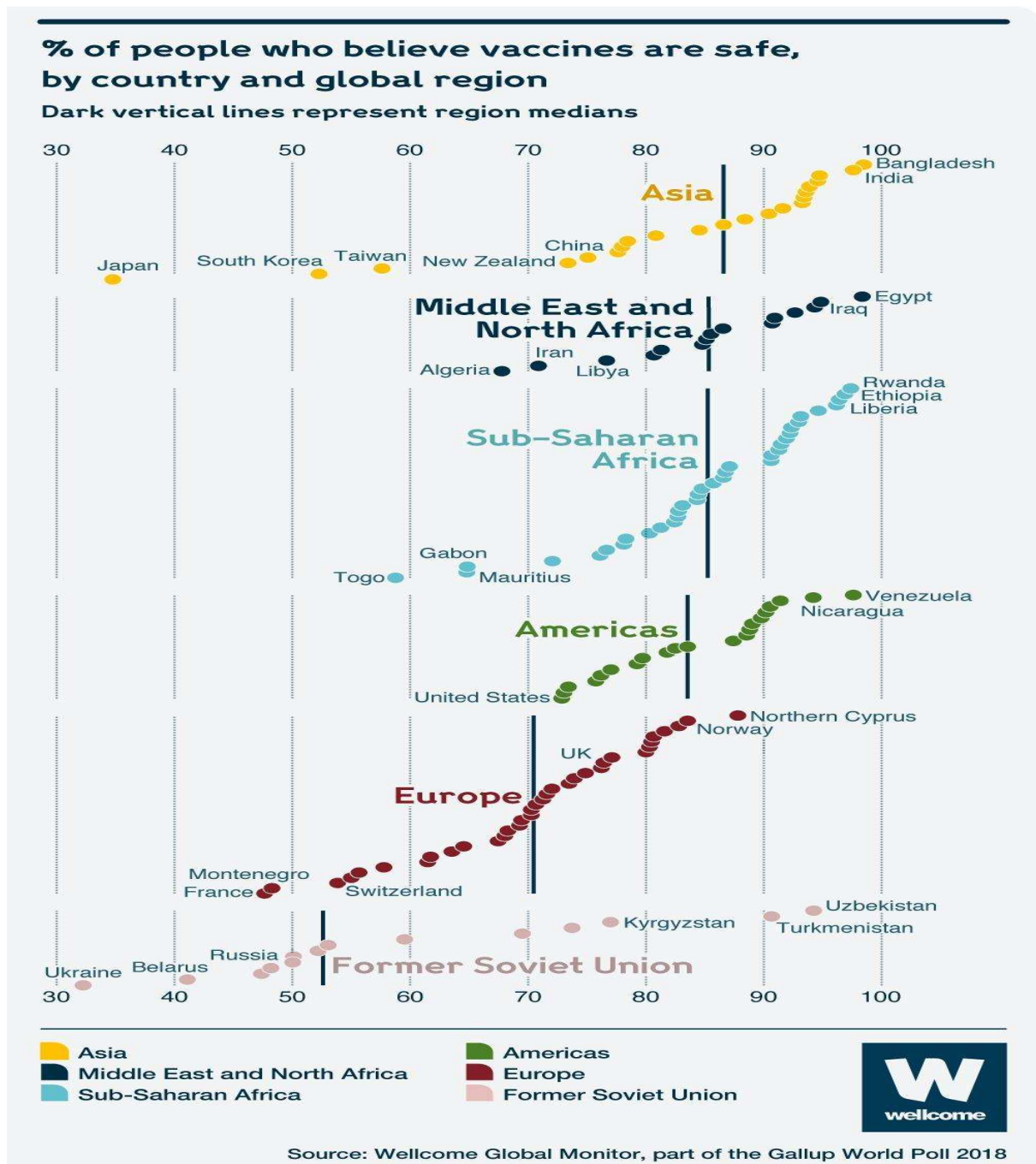
**NAME:** Sushanth Reddy Rachala

**G.NO:** G01475271

**Group No.:** 24

**INSTRUCTOR:** Dr. Isuru Dassanayake

## ORIGINAL VISUALIZATION



Link to Original Visualization:

<https://wellcome.org/sites/default/files/wellcome-global-monitor-2018.pdf>

The plot above is about the people who believe the vaccines are safe by country and global regions. It shows how many people are ready to believe vaccines are safe. This plot is taken from the Tumblr website [1] .

The original visualization [2] is a dot plot representing the percentage of people who believe vaccines are safe, categorized by country and global region. While the intent is clear, there are several areas where the visualization could be improved for better clarity and information conveyance:

**Overplotting:** When the data points overlap this make challenging to determine specific values or to discern across different countries, particularly for nations in the same region there is notable degree of overplotting.

**Color Selections:** Not all users, particularly those with color vision impairments, can clearly distinguish between the color's used to denote distinct regions.

**Median Lines:** There might be confusion for a portion or the gridlines or axis not every observer will understand the dark vertical lines that indicate the region medians.

**Country Labels:** Because the country labels are all arranged at the same horizontal level, they may overlap and become difficult to see, particularly in cases where a large number of the nations have comparable percentages.

**Clarity of Message:** The main point or conclusions are not readily apparent from the visualization. It presents information without pointing out key points or drawing inferences for the audience.

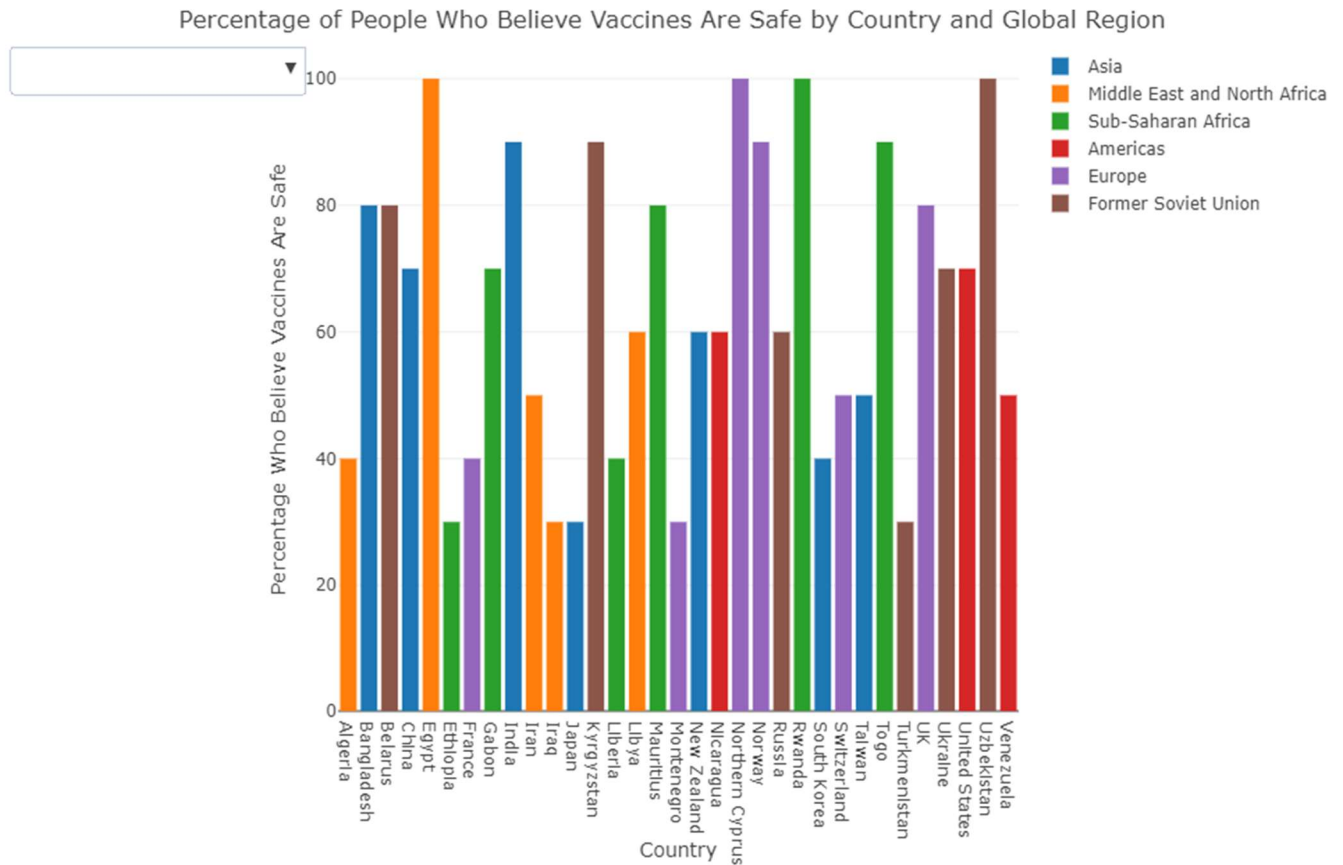
**Axis Labels and Scale:** The labels (percentages) on the x-axis are a little unclear, and the y-axis is not labelled. Furthermore, it's not immediately obvious why the x-axis begins at 30% rather than 0%, which could skew the viewer's understanding of proportion.

**Consistency in Design:** New Zealand is categorized as Asia, which may cause confusion for certain viewers as it does not correspond with their geographic classification.

**Absence of Interaction:** It is a static image and cannot be interacted with. It is not possible for viewers to filter by region or click on a country to acquire additional information.

**Absence of Context:** There is no explanation provided in the visualization for possible differences in vaccination safety beliefs. The audience might not comprehend the fundamental causes of the variations in the absence of other information or annotations.

# REDESIGN PLOT 1



**Library Calls:** The script begins by loading necessary R libraries. `plotly` [3] is used for creating interactive plots, `dplyr` [4] is a grammar of data manipulation, and `readr` [5] is for reading CSV data into R.

**Data Reading:** The `read_csv` function reads the "vaccine\_belief\_data.csv" file into an R data frame named `vaccine_data`.

**Regions Setup:** The script extracts unique regions from the `Region` column of the `vaccine_data` data frame and adds an 'All' option to allow viewing data for all regions at once.

**Dropdown Menu Creation:** It then creates a dropdown menu for the plot with buttons using `lapply` [6]. Each button corresponds to a region, plus one for 'All', which will allow the user to filter the data shown on the plot by region.

**Data Preparation for Plotting:** Since `plotly` doesn't natively support the kind of interaction required for this plot directly, the script prepares the data by creating a separate data frame for each region using `lapply` and subsetting the original data.

**Initial Plot Creation:** An empty `plotly` object `p` is initialized.

**Adding Traces:** The script loops through each region's data frame and adds a bar trace to the plot for each one. A trace in `plotly` is essentially a single plot element or series. The `visible` parameter is set to `TRUE` by default so all regions will be displayed initially.

**Dropdown Menu Addition:** The dropdown menu is then added to the layout of the plot using the `layout` function. The `updatemenus` [7] option is configured with the list of dropdown buttons created earlier.

**Final Plot Layout Configuration:** The title and axis labels are set, completing the layout configuration.

**Displaying the Plot:** Finally, the script displays the interactive plot by calling the `final_plot` object.

The resulting plot will allow users to interactively select which region's data they want to view. By default, all regions will be visible, but selecting a specific region from the dropdown will hide all other regions' data, showing only the selected region's data.

## Redesign (Interactive Bar Chart with plotly):

**Interactive Visualization:** The redesign allows users to interact with the data, such as filtering by region.

**Enhanced Data Exploration:** Users can click on elements to see more details or isolate regions for a clearer view.

**Customization:** The script can be modified to add features such as hover text, which can provide more context for each data point.

**Adaptability:** It can handle new data or changes in the dataset more dynamically, allowing for the plot to be updated without needing to redesign the graphic.

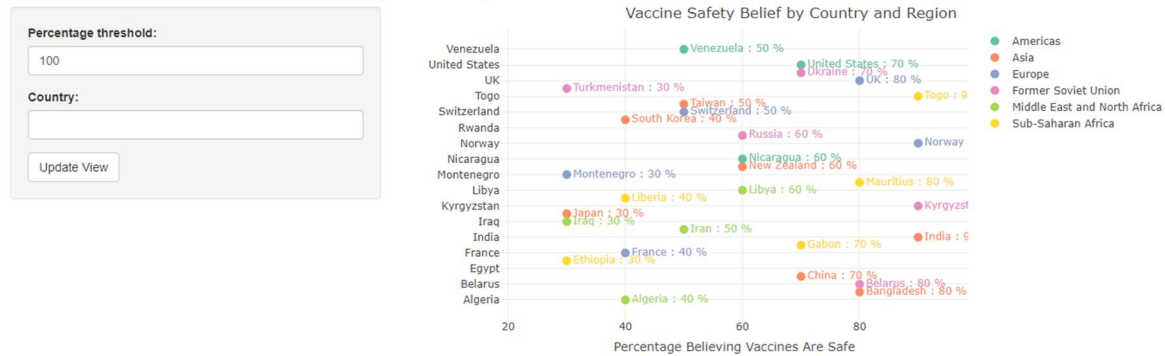
**User Engagement:** An interactive plot can lead to higher engagement from the audience, as they can manipulate the data presentation.

**Clarity and Focus:** Users can focus on one region at a time, making it easier to analyze and interpret the data in segments.

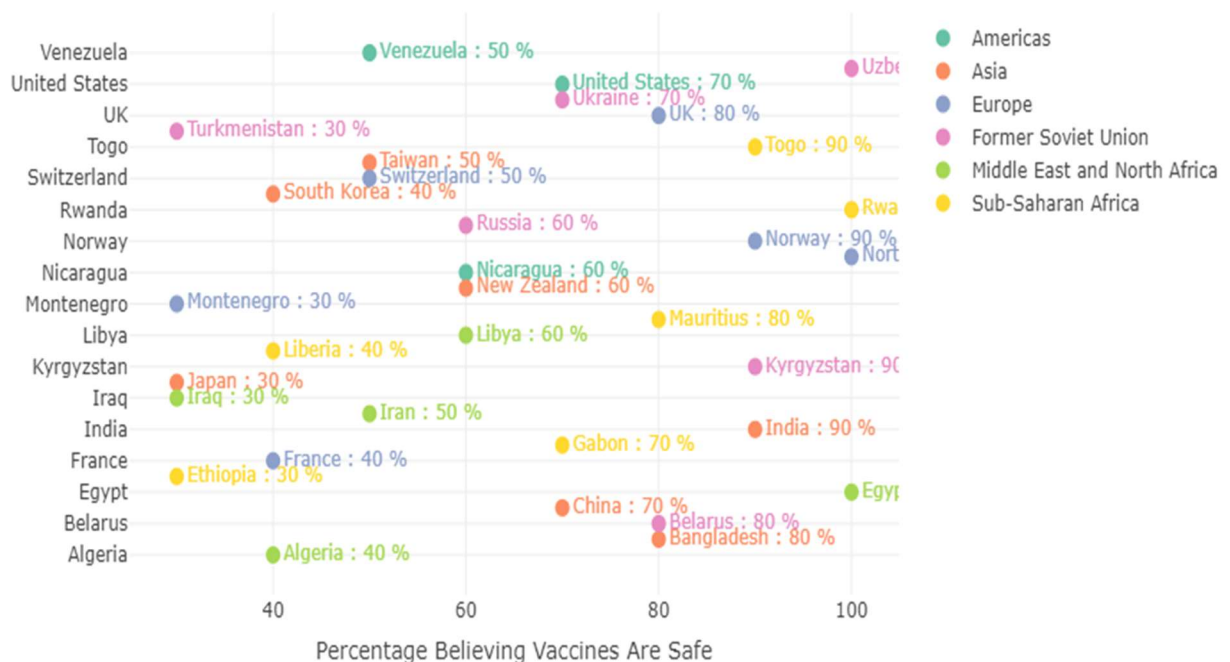
# REDESIGN PLOT 2

http://127.0.0.1:3771 | Open in Browser | Publish

Vaccine Safety Belief by Country and Region



Vaccine Safety Belief by Country and Region



## Libraries Used:

**Shiny** : This is the core library for building interactive web applications with R. It is used to define both the UI and server components and manage the reactivity and session state of the application [8].



**Plotly:** This library is used for creating interactive plots. It integrates well with Shiny, allowing the plots to be reactive to user input [3].

**Readr:** Part of the tidyverse, it's a fast and friendly way to read rectangular data (like csv, tsv, and fwf). It is used here to read the CSV file containing the data [5].

**Dplyr:** Also, part of the tidyverse, it's used for data manipulation. It is used to filter the data according to the user's inputs in the Shiny app [4].

### **Redesign with Shiny and Plotly:**

**User Interface (UI):** The fluidPage function from Shiny creates a web page that automatically adjusts to the size of the user's browser, making it responsive and mobile-friendly. It includes a title panel, a sidebar for inputs, and a main panel for displaying the plot.

**Interactivity:** The sidebar includes interactive elements: a numeric input for setting a threshold on the percentage of people who believe vaccines are safe, a text input for filtering by country name, and an action button to trigger the update of the view.

**Server Logic:** The server part of the Shiny app contains the instructions to read the data, filter it according to user inputs, and render a Plotly plot that updates with the filtered data.

**Reactivity:** The app uses reactive programming principles to update the output whenever the user changes the input parameters and clicks the update button.

## Differences from the Original and Previous Redesign:

**Input Controls:** This Shiny app allows users to set a threshold for the percentage of people who believe vaccines are safe and to filter by country name, which provides a more refined control over the data displayed than the previous dropdown menu from the Plotly-only redesign.

**Reactive Updates:** The data visualization updates only after the user clicks the "Update View" button, which can prevent unnecessary calculations and make the app more efficient.

**Server-Side Processing:** The data filtering and plotting happen on the server side in real-time, which can handle larger datasets and more complex manipulations than client-side JavaScript in a standard Plotly plot [9].

## CONCLUSION

The project overview and subsequent redesign proposals for visualizing beliefs about vaccine safety across different countries and regions demonstrate a thoughtful approach to enhancing data visualization for clarity, accessibility, and interactivity.

The first redesign, leveraging Plotly, introduces interactivity through a dynamic dropdown menu, allowing users to filter data by region. This approach significantly improves the user's ability to engage with and explore the dataset, facilitating a more personalized exploration of vaccine beliefs. By transforming a static visualization into an interactive bar chart, the redesign addresses issues of clarity and accessibility, making the data more digestible and the insights more apparent.

The second redesign, incorporating both Shiny and Plotly, advances the project by adding server-side data processing and more sophisticated user input controls, such as numeric and text filters. This not only enhances the level of interactivity but also allows for a more nuanced and tailored data exploration experience. The use of Shiny for building an interactive web application introduces a platform for complex, real-time data manipulation and visualization, showcasing the power of integrating multiple tools to achieve a highly responsive and user-friendly interface.

## REFERENCES

- [1] "Tumblr," [Online].
- [2] "wellcome," [Online]. Available: <https://wellcome.org/sites/default/files/wellcome-global-monitor-2018.pdf>.
- [3] "Plotly," [Online]. Available: <https://plotly.com/r/bar-charts/>.
- [4] "Dplyr," [Online]. Available: <https://dplyr.tidyverse.org/>.
- [5] [Online]. Available: <https://readr.tidyverse.org/>.
- [6] "lapply," [Online]. Available: <https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/lapply>.
- [7] [Online]. Available: <https://plotly.com/r/reference/layout/updatemenus/#layout-updatemenus>.
- [8] "shiny," [Online]. Available: <https://shiny.posit.co/r/getstarted/shiny-basics/lesson1/index.html>.
- [9] [Online]. Available: <https://rstudio.github.io/DT/server.html>.