IT-452 Data Information and Visualization

Data Insights Dashboard: Visualizing COVID-19 Cases in Africa, GitHub Repo Structure, Fund Performance, Browser Market Share, and Regression Line Visualizer

-SAI KIRAN NANDIPATI
-ULID: SNANDIP

1. Introduction:

The project focuses on visualizing diverse datasets through five distinct types of visualizations. Firstly, the **Regression Line Visualizer** allows users to interactively plot points on a grid. Using linear regression computation, a line is drawn to minimize the distance from these points, similar to linear regression analysis. Secondly, the **Performance Comparison between Two Funds** visualization aids users in comparing trends between two funds over time. Thirdly, the **Collapsible Tree Displaying File Structure of GitHub Repositories** offers an intuitive representation of the tree structure within a GitHub repository, facilitating efficient navigation of project files and folders. Additionally, the **100% Stacked Bar Chart for Browser Market Share** enables users to comprehend browser market share trends from 2009 to 2024. Lastly, the density based **African Country-wise Covid-19 Cases per Million People** visualization provides a density-based plot to identify which African countries were most affected by Covid-19 from 2020 to 2022. These visualizations offer users varied perspectives on the datasets, empowering them to extract insights and make informed decisions based on the presented data.

2. Background and Literature Review:

The datasets utilized in this project were obtained from open sources, with hyperlinks provided to grant users access to the data sources used for visualization. For some data sources I preprocessed them using python script and cleaned them for better visualizations. The visualization techniques employed in this project draw from the knowledge imparted during the weekly lectures conducted throughout the duration of the course. Leveraging tools such as d3, JavaScript, HTML, SVG, CSS, Node.js, among

others, which were extensively covered in the lectures, these visualizations were meticulously crafted. By adhering to the methodologies and tools discussed in the course lectures, the project ensures consistency and alignment with the principles taught, thereby fostering a deeper understanding of data information and visualization concepts among users.

3. Visualization Design:

3.1 Plot 1: Regression Line Visualizer

The Regression Line Visualizer is an interactive tool designed for exploring two-dimensional regression analysis. Upon accessing the plot, users are presented with a blank plane (**as shown in fig1.a**) where they can interactively add and remove points. To add a point, users simply need to click on an empty area of the plane. As points are added, a regression line is dynamically recalculated and displayed (shown in **Fig 1.b**), reflecting the linear relationship between the plotted points. This regression line is computed using the least squares method to minimize the distance between the line and the points. Additionally, users have the option to remove points by clicking on existing ones, prompting an automatic update of the regression line to reflect the modified dataset. For users seeking a fresh start, a "Reset" button is provided, allowing them to clear all points from the plane and begin anew. This visualization serves as a valuable tool for understanding linear relationships and exploring the principles of regression analysis in a hands-on manner.

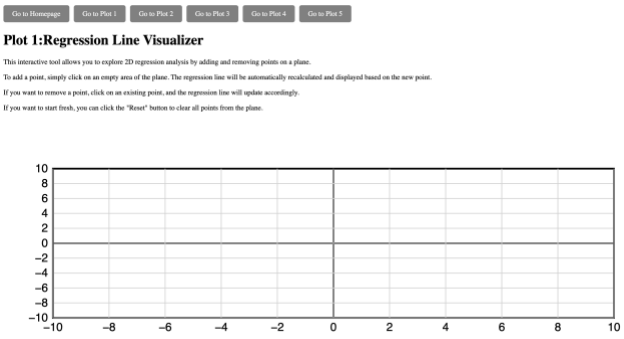

Fig 1.a Blank pane for users to plot points

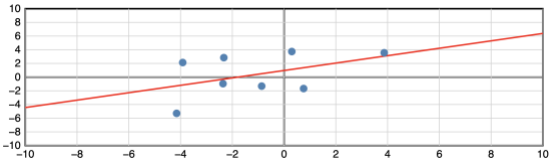

Fig 1.b Dynamic regression Line based on plotted points

3.2 Performance Comparision between 2 Funds

This plot offers a visual comparison of the performance between two different funds over a selected time period. Upon accessing the plot, users are presented with options to choose two funds from a dropdown menu, with popular options like S&P 500, Nasdaq 100 ETF, and others available for selection(as shown in **Fig 2.a**). Additionally, users can specify the time period for the comparison, choosing between 1 month, 1 year, or 2 years.

The plot dynamically updates to display the performance data of the selected funds over the specified time period. The visualization is presented as a line chart, where each line represents the performance trend of one of the selected funds over time. (as shown in **Fig 2.b**)

Interactive elements allow users to change the selected funds and time period, triggering automatic updates to the displayed data. Furthermore, users can hover over points on the chart to view specific details about each data point, such as the date, fund name, and corresponding performance value.

This visualization serves as a valuable tool for investors and financial analysts, enabling them to easily compare the historical performance of different funds and make informed investment decisions based on the displayed data. Additionally, the inclusion of interactive features enhances user engagement and facilitates a deeper understanding of the performance trends observed in the plot.

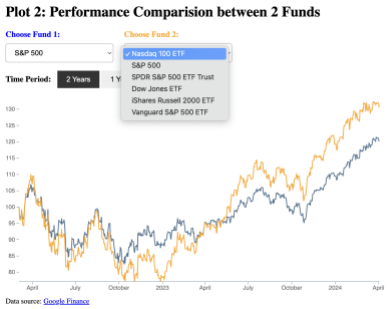

Fig 2.a: Stocks trend comparison between two selected funds over year

Fig 2.b: Stocks trend comparison between two S&P 500 , Nasdaq

3.3 Collapsible tree visualization

This visualization features a collapsible tree visualization that illustrates the folder structure of any GitHub repository. Here's an overview of the key components and functionalities:

- **Description:**

The visualization presents a hierarchical representation of folders and files within a GitHub repository(Fig 3.a). Blue circles depict directories, which users can expand or collapse to reveal or hide their contents. Similarly, files are represented by circles.

- **Functionality:**

- Users can input the owner and name of the GitHub repository they wish to visualize into the respective fields.

- Upon clicking the "Fetch Data" button, the visualization fetches the repository data from GitHub using the GitHub Contents API. This process may take some time due to the multiple requests involved.

- Once the data is fetched, the collapsible tree structure is dynamically generated and displayed in the SVG area.

- Users have the option to click on directory nodes to expand or collapse them, allowing for exploration of the repository's folder hierarchy.

- Additionally, users can click the "Save Data" button to replace the default structure being displayed with the newly fetched repository structure.

- **Implementation:**

- The collapsible tree visualization is implemented using D3.js, a powerful JavaScript library for data visualization.

- The GitHub repository data is fetched asynchronously using the `fetch` API, and the visualization is drawn using D3's tree layout.

- The `drawTree` function dynamically generates the tree structure based on the fetched data, assigning attributes to nodes and links, and handles the animation and transition effects during updates.

- GitHub utility functions (`makeRequest` and `saveDataToFile`) facilitate data retrieval from and storage to the GitHub repository.

- **Interactivity:**

- Users can interact with the visualization by expanding and collapsing directory nodes, enabling them to navigate through the folder structure of the GitHub repository.

- Feedback is provided to users through loading indicators, which inform them of the progress of data fetching and visualization rendering processes.

Overall, Plot 3 offers a user-friendly and interactive way to explore and visualize the folder hierarchy of GitHub repositories, aiding developers and users in understanding the structure and organization of project files.

Fig 3.a : **Hierarchical representation of folders and files within a GitHub repository D3**

3.4: Stacked Bar Chart

- **Description:**

The visualization presents a stacked bar chart (as shown in **fig 4.a**)where each bar represents a specific year, and the segments within each bar represent the proportion of market share held by different browser types. The chart is stacked vertically, with each segment representing a browser type and the total height of each bar representing 100% of the market share for that year.

- **Functionality:**

The data for the visualization is sourced from a CSV file named `new_bar.csv`. The chart is dynamically generated using D3.js, a JavaScript library for data visualization. Users can interact with the chart by hovering over each segment to view detailed information about the browser type and its corresponding market share percentage for the selected year. A legend is provided to help users identify the different browser types represented by each color segment in the chart. The tooltip provides additional context and information when users hover over individual segments of the chart.

- **Implementation:**

The visualization begins by setting up the SVG canvas and defining the dimensions and margins for the chart. The data is parsed from the CSV file and processed to convert it into a format suitable for plotting. Scales are defined for the X and Y axes to map the data values to the visual representation. Bars are created using the stacked layout, with each segment representing a different browser type. Event listeners are added to enable interactivity, such as displaying tooltips on hover and highlighting segments. A legend is added to help users interpret the colors used in the chart.

- **Interactivity:**

Hovering over each segment of the chart triggers a tooltip that displays detailed information about the browser type and its market share percentage. The tooltip's position dynamically adjusts to follow the cursor as users move over the chart. Segments are highlighted on hover to provide visual feedback and improve readability.

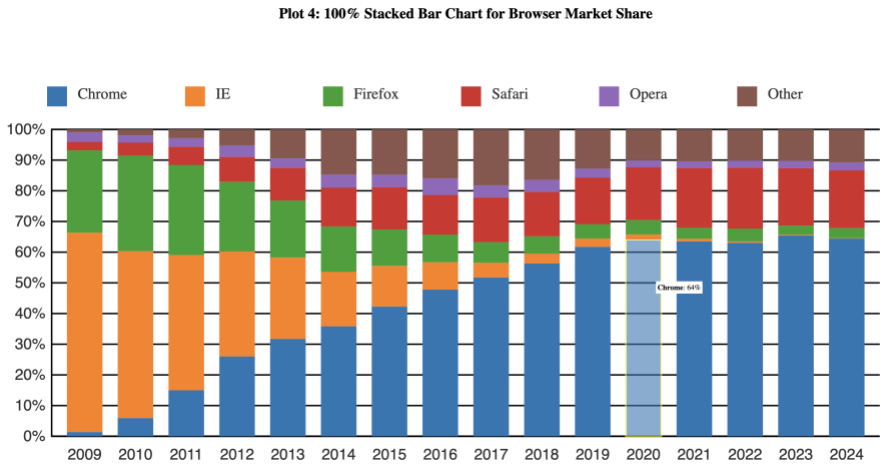

Fig 4.a: Stacked Bar Chart for Browser Market Share

3.5: Density Plot

- **Description:**

Plot 5 presents an interactive choropleth map/color density based visualization visualizing the spread of COVID-19 cases per million people across African countries(as shown **in Fig 5.a**). The map dynamically updates based on the selected date using a slider, allowing users to explore the progression of COVID-19 cases over time.

- **Functionality:**

The visualization leverages D3.js, a powerful JavaScript library for creating interactive data visualizations. It imports GeoJSON data representing African countries and COVID-19 case data from a CSV file. The choropleth map color-codes each country based on the number of COVID-19 cases per million people, providing a visual representation of the severity of the pandemic across different regions.

- **Implementation:**

The code initializes the choropleth map, sets up the SVG canvas, and defines scales for mapping data values to visual attributes such as color. It dynamically updates the map based on the selected date using a slider input, fetching the corresponding COVID-19 case data for that date and updating the map accordingly.

- **Interactivity:**

Users can interact with the visualization by adjusting the date slider to explore COVID-19 case trends over time. Hovering over each country on the map displays a tooltip with detailed information about the country's name and the number of COVID-19 cases per million people. This interactivity enhances user engagement and facilitates data exploration.

- **Data Source:**

The COVID-19 case data is sourced from a reliable dataset, ensuring the accuracy and credibility of the information presented in the visualization. A link to the data source is provided, allowing users to access additional information and verify the data's authenticity. Overall, Plot 5 provides a visually compelling and informative representation of COVID-19 spread across African countries, empowering users to gain insights into the pandemic's impact and track its progression over time.

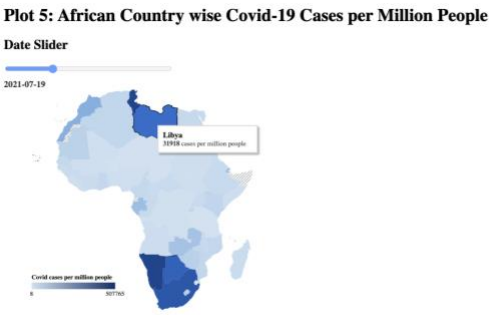

Fig 5.a: African Country wise Covid-19 Cases per Million People on 2021-07-19

4. Conclusion:

In this project, we embarked on a journey to visualize diverse datasets through five distinct types of visualizations. From exploring linear relationships to comparing fund performance and understanding COVID-19 trends, each visualization offered unique insights and empowered users to make informed decisions.

By leveraging tools such as d3, JavaScript, and HTML, we crafted interactive and intuitive visualizations that facilitated data exploration and understanding. Through transparency in data sourcing and adherence to visualization principles, we ensured credibility and consistency in our approach.

In conclusion, these visualizations serve as powerful tools for extracting insights from complex datasets, fostering a deeper understanding of data information and enabling informed decision-making. As we continue to innovate in the realm of data visualization, let us embrace the potential of these tools to drive positive change and exploration.

My Contribution:

As part of the IT-452 Data Information and Visualization project, I contributed significantly to the development of diverse visualizations aimed at providing insights into diverse datasets by improving my skillset in coding part. Here's a breakdown of my contributions to each visualization:

1. Regression Line Visualizer (Plot 1):

- Designed and implemented the interactive tool for two-dimensional regression analysis.
- Developed functionality for users to plot points on a blank plane and dynamically recalculate and display the regression line using the least squares method.
- Implemented features allowing users to add and remove points interactively, with automatic updates to the regression line.
- Created a "Reset" button for users to clear all points and start afresh, enhancing user experience and facilitating hands-on exploration of linear relationships.

2. Performance Comparison between Two Funds (Plot 3.2):

- Contributed to the visualization design for comparing trends between two funds over a selected time period.
- Implemented interactive elements such as dropdown menus for fund selection and a time period selector.

- Developed functionality for dynamically updating the plot to display performance data of selected funds over the specified time period.
- Enabled hover-over interactions to view specific details about each data point, enhancing user engagement and understanding of performance trends.

3. Collapsible Tree Visualization (Plot 3):
- Played a key role in the development of the collapsible tree visualization illustrating the folder structure of GitHub repositories.
- Contributed to the implementation of user input fields for specifying repository owner and name.
- Developed functionality for fetching repository data from GitHub using the GitHub Contents API and dynamically generating the collapsible tree structure.
- Implemented interactive features allowing users to expand and collapse directory nodes for exploration of the repository's folder hierarchy.

4. Stacked Bar Chart (Plot 3.4):
- Contributed to the description and functionality of the stacked bar chart presenting browser market share trends.
- Assisted in parsing data from the CSV file and processing it for plotting.
- Implemented event listeners for hover-over interactions to display tooltips with detailed browser market share information.
- Contributed to the enhancement of user experience through dynamic adjustments of tooltip position and segment highlighting.

5. Density Plot (Plot 3.5):
- Played a significant role in the development of the interactive choropleth map visualizing COVID-19 cases per million people across African countries.
- Contributed to the implementation of functionality for dynamically updating the map based on selected dates using a slider input.
- Developed interactive features allowing users to explore COVID-19 case trends over time and view detailed information about each country on hover.
- Ensured reliability and credibility of COVID-19 case data by sourcing from reputable datasets and providing links to data sources for verification.

Overall, my contributions to these visualizations aimed to empower users to extract insights and make informed decisions based on the presented data. Through adherence to visualization principles and utilization of tools such as d3.js and JavaScript, I facilitated a deeper understanding of data information and visualization concepts among users.