Health Data Visualization - Documentation

Motivation

The motivation behind this application is to provide users with an interactive platform to explore and understand health-related data. By visualizing various factors such as economic status, environmental conditions, personal habits, demographics, health care system, and personal health, users can gain insights into the correlations and trends within the data. The application aims to empower users to make informed decisions and take actions to improve health outcomes.

Data

The data used in the application comes from this website. It includes a wide range of health-related attributes such as poverty percentage, median household income, air quality, smoking rates, elderly population, and more.

Visualization Components

The application consists of multiple visualization components, including:

- **Bar Chart**
- **Scatter Plot**
- **Choropleth Map**

Users can interact with the application by selecting different attributes and filters using the graphical user interface (GUI). The views update dynamically in response to user interactions, allowing for real-time exploration and analysis of the data.

Color Scheme

Each category in the visualization is represented by a distinct color scheme to aid in easy identification and interpretation:

- **Economical (Orange)**: Poverty percentage, median household income, and percentage of the population that completed high school. Color: `#FFA500`

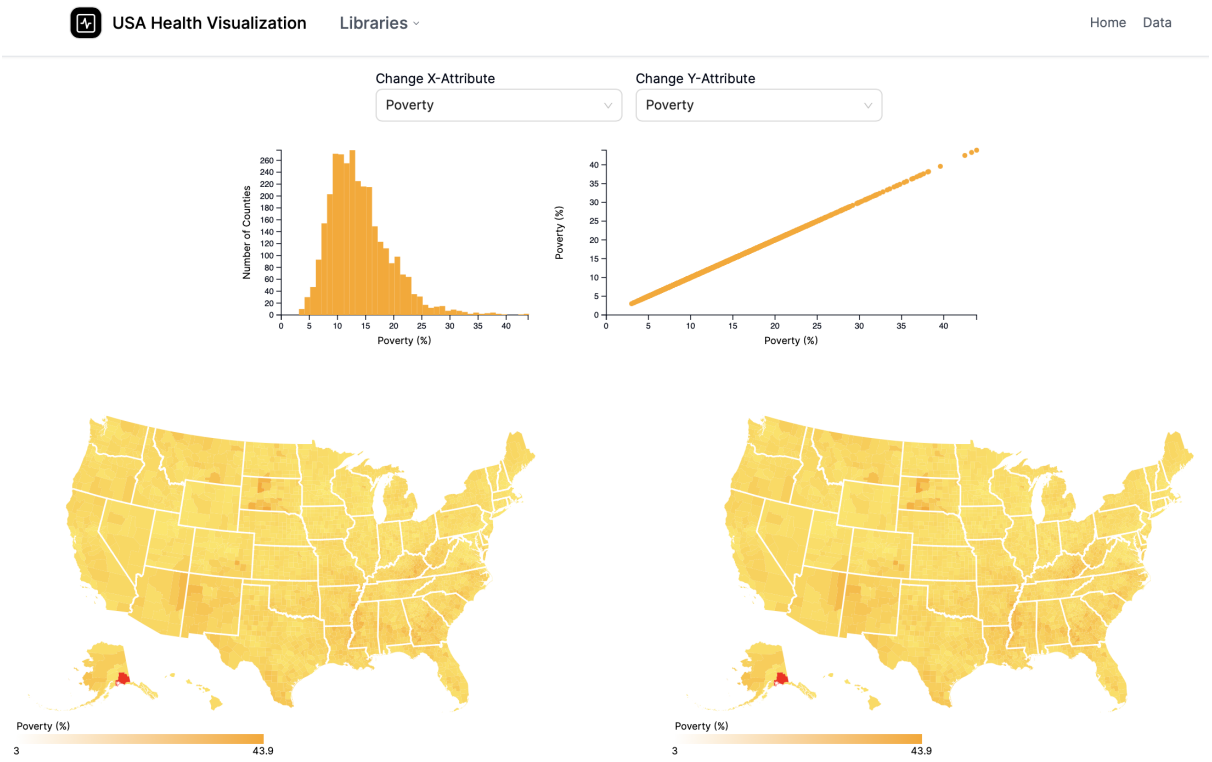
- **Environmental (Green)**: Air quality and park access. Color: `#008000`
- **Personal Habits (Purple)**: Smoking and inactivity. Color: `#800080`
- **Demographical (Pink)**: Elderly population and county type (rural/urban/suburban/small city). Color: `#FF69B4`
- **Health Care System (Blue)**: Number of hospitals, primary care physicians, and percentage of the population without health insurance. Color: `#0000FF`
- **Personal Health (Red)**: High blood pressure, coronary heart disease, stroke, and high cholesterol. Color: `#FF0000`

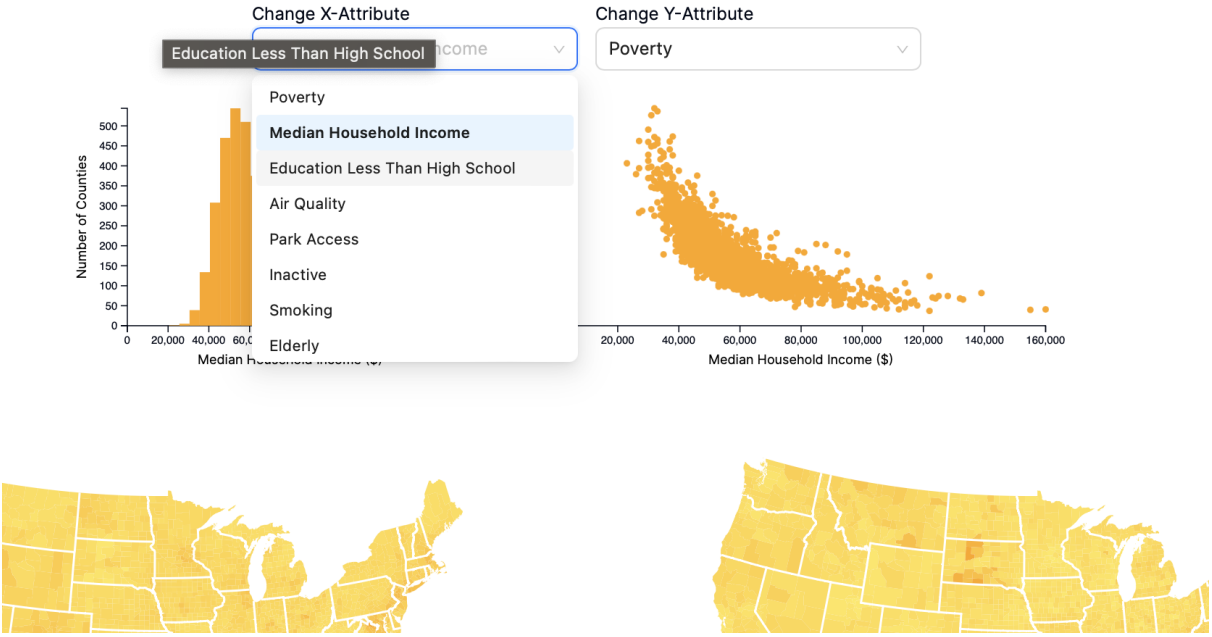
Dropdown Selection

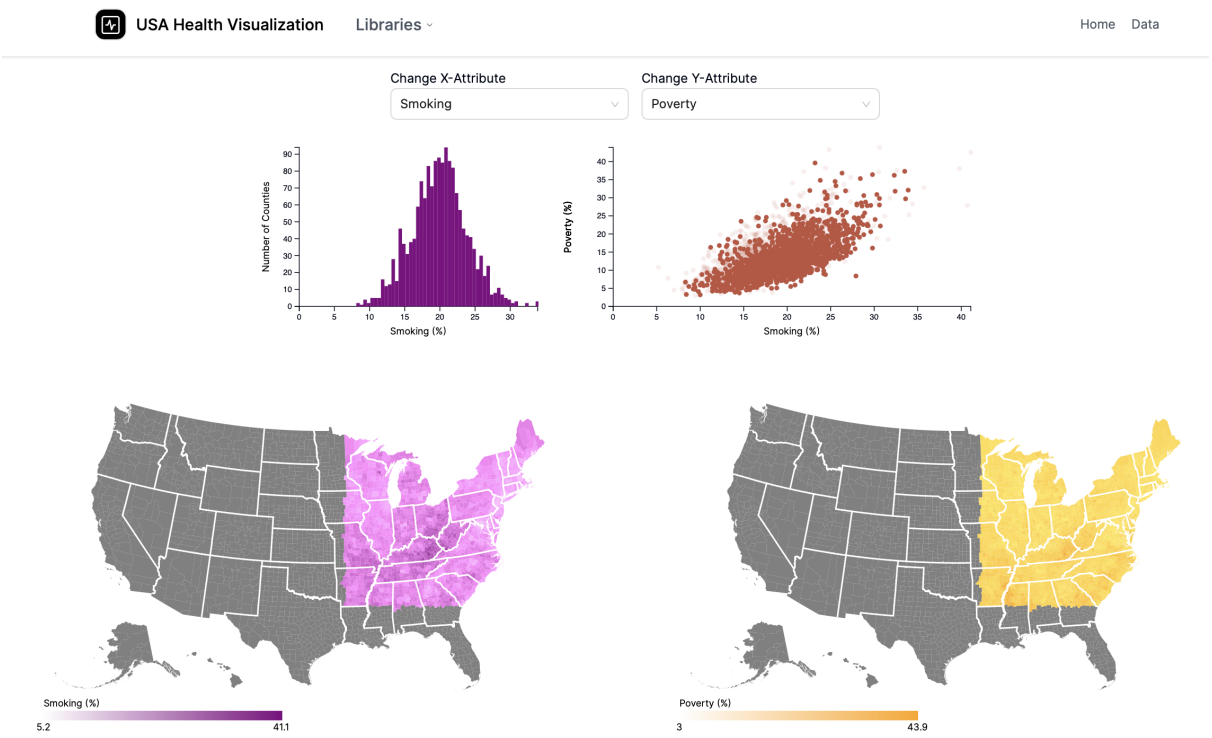
The dropdown selection of categories in the health data visualization application provides users with a straightforward way to choose which aspect of health-related data they want to explore. By selecting a category from the dropdown menu, users can quickly switch between different views of the data, focusing on areas such as economics, environment, personal habits, demographics, health care system, and personal health. This feature simplifies the user experience and enables efficient navigation through the dataset.

Brushing

Brushing, on the other hand, facilitates data exploration by allowing users to interactively select and filter specific data points or regions of interest within individual charts. As users brush over a chart, the application dynamically updates all other visualization components to reflect the filtered data, ensuring that users can analyze correlations and patterns across multiple aspects of the dataset simultaneously. This synchronized approach to updating charts enhances the coherence and usability of the visualization.


USA Health Visualization
Libraries
Home
Data
Change X-Attribute
Poverty
Change Y-Attribute
Poverty
Number of Counties
Poverty (%)
Poverty (%)
Poverty (%)
3
43.9
Poverty (%)
3
43.9


Change X-Attribute
Education Less Than High School
Change Y-Attribute
Poverty
Poverty
Median Household Income
Education Less Than High School
Air Quality
Park Access
Inactive
Smoking
Elderly
Number of Counties
Median Household Income ($)



Discoveries

Through the application, I was able to discover various insights, such as:

- Correlations between economic status and health outcomes.
- Impact of environmental factors on personal habits.
- Disparities in health care access across different demographics.
- Trends in personal health conditions based on geographical location.

Process

For this project, we used the following libraries and technologies:

- **D3.js**: For data visualization and manipulation.
- **React.js**: For building the user interface and managing component state.
- **Next.js**: For server-side rendering and routing.
- **TypeScript/JavaScript**: Language the code is written in
- **Shad/cn UI: Customizable UI components**

My codebase is structured using modular components, following best practices for React development. Users can access and run the application by cloning

the repository from GitHub and following the instructions in the README file.

Future Works

In future iterations of the project, the aim is to:

- Enhance the user interface and add more interactive features.
- Incorporate additional data sources to provide a comprehensive view of health-related factors and maybe have an upload option for file for the visualization to be more dynamic
- Implement machine learning algorithms for predictive analytics and personalized recommendations.
- Optimize performance and scalability for handling large datasets.
- Addressing browser compatibility issues with D3.js and React.js integration and optimizing rendering performance for complex visualizations.

Challenges

During the development process, we encountered several technical difficulties, including:

- Managing state and data synchronization between different visualization components (explained below)

Synchronized Updates Across Charts

One of the key features of the visualization application is the synchronized updates across all charts. Whenever a user interacts with a visualization component, such as brushing over a chart, the changes are propagated to all other charts in real-time, ensuring consistency and coherence in the visualization.

Mechanism:

- **Central Data Store:** We utilize a central data store to maintain the current state of the visualization data.
- **Subscriber Pattern:** Each visualization component subscribes to the central data store, listening for updates.
- **Event Handling:** When a user interacts with a visualization component, the corresponding event triggers an update in the central data store.

- **Broadcasting Changes:** The central data store broadcasts the updated data to all subscribed visualization components, prompting them to refresh and display the latest data.

In summary, the application offers a user-friendly interface for exploring health data. With features like category selection and brushing, users can efficiently analyze data across different dimensions, facilitating informed insights and understanding. Despite the challenges, I learned valuable lessons and gained insights into data visualization techniques and web development best practices.