Data Visualization - Project Report

Jeevan Ravi Kumar

Pradyoth Singhenahalli Prabhu

Thanay Geeshpathy Narayanamurthy

World Energy Consumption Visualized

Overview

- Relationship between a nation's energy consumption and its economic prosperity.
- Informs observations about the economic health of nations worldwide.
- Energy consumption and economic growth are intertwined factors - Business and Policies.

Objectives

- How the world's energy consumption patterns influence the GDP of different countries.
- Analyzing the energies that drive economies and impact the daily lives of people around the globe.

About Dataset

Our complete Energy dataset is a collection of key metrics maintained by *Our World in Data*. It is updated regularly and includes data on energy consumption (primary energy, per capita, and growth rates), energy mix, electricity mix and other relevant metrics.

ref: https://www.kaggle.com/datasets/pralabhpoudel/world-energy-consumption?resource=download

Data Pre-processing

Dropped Columns

- Everything but the present top 50 largest GDP nations.
- All data before 1990

Missing Data

- Mean Imputation
- Forward Fill GDP

Outliers

- Checking IQR
- Imputed extreme values with mode.

Proposed Deliverables

World Choropleth Map

Story - Conveys the global distribution of GDP by color-coding countries on a choropleth map.

Marks - Area geospatial

Channels - Color hue represents GDP values.

Detail on demand - Tooltip on country hover displays population, GDP, and percentages of different energy sources.

Insights

- Immediate visualization of countries' economic strengths and disparities.

- Comparative analysis of GDP distribution globally.

Code Explanation

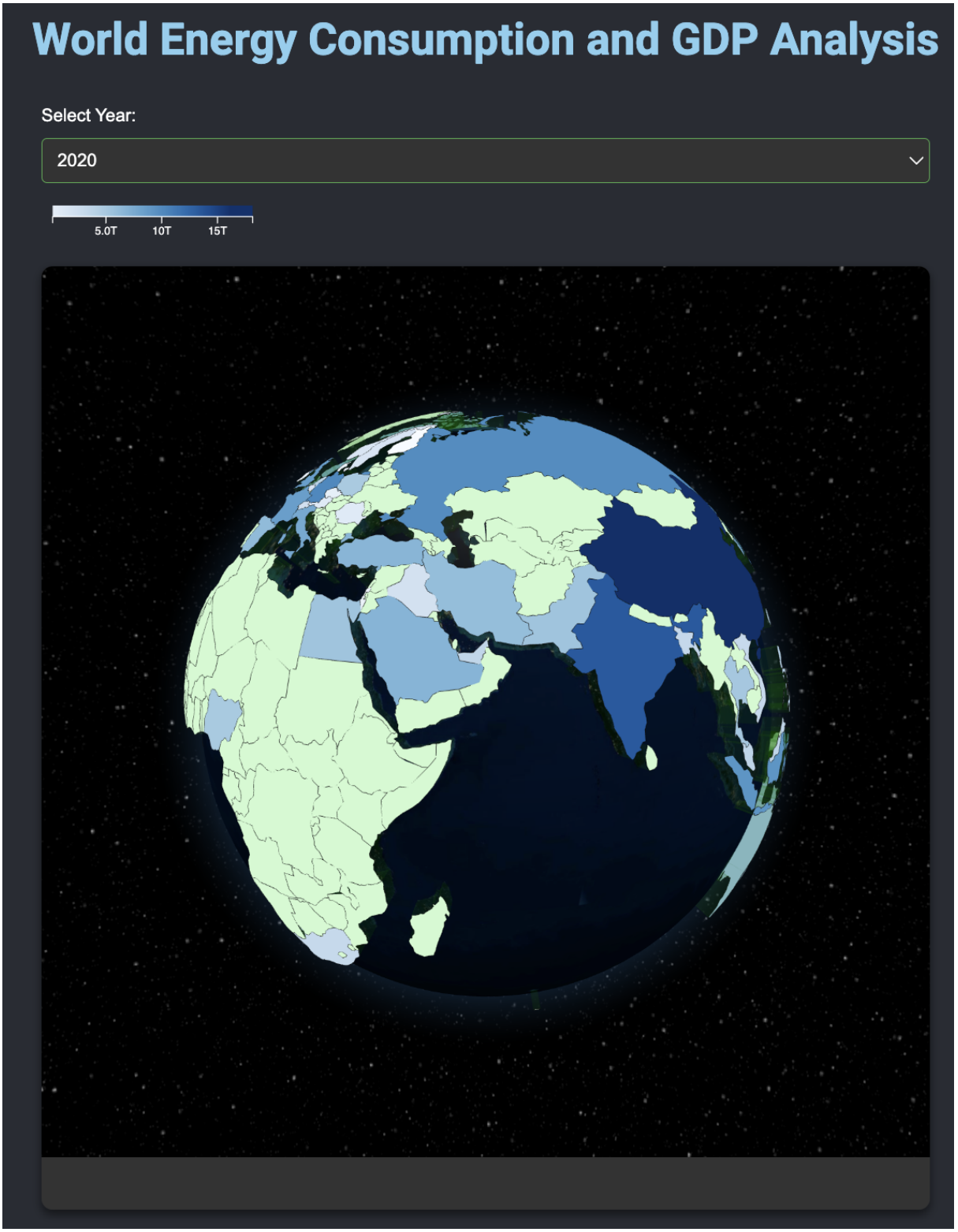
This JavaScript code fetches and processes geographical and energy data to create an interactive globe visualization using the D3.js library. Here's a brief explanation of the code:

1. **Initialization:**
   - `mainYear` is set to 2020 initially.
   - The code uses `Promise.all` to fetch two sets of data asynchronously:
     - The first set is a GeoJSON file representing country boundaries.
     - The second set is a CSV file containing energy-related data.
2. **Data Processing:**
   - The code filters the energy data to obtain information for the `mainYear` and extracts unique years for creating a dropdown menu.
   - The dropdown menu is populated with the unique years, and an event listener is added to update the visualization when a different year is selected.
3. **Globe Visualization Update:**
   - The `updateGlobeVisualization` function is defined to update the globe visualization based on the selected year.
   - The function filters energy data for the selected year, maps data by country, and prepares for GDP color scaling.
   - A legend is created with a color gradient representing GDP values.
   - The legend is updated dynamically.
4. **Globe Configuration:**
   - The `Globe` function is called to create and configure a globe visualization.
   - The globe is updated based on the selected year, and country polygons are highlighted based on GDP values.
   - A click event is added to the country polygons to redirect to a pareto.html page with specific query parameters.
5. **Default Scroll Position:**

- The default x-axis scroll position is set to the center of the visualization.

6. **Rendering:**
   - The globe visualization is rendered on the HTML element with the ID "globeViz."
7. **Overall Execution:**
   - The entire process is triggered when the page is loaded, and the initial globe visualization for the `mainYear` (2020) is displayed.

Note: The code includes some HTML template strings for displaying information about hydroelectricity, low carbon, oil, and renewables for a clicked country. The displayed information is based on energy data for the selected country and year.



Pareto Chart

Story - Illustrates the relationship between a country's GDP and energy consumption over time.

Marks - Bar chart for GDP. Line chart for energy consumption per capita.

Channels - Bar length represents GDP. Line position represents energy consumption per capita.

Detail on demand - Tooltip provides energy consumption per capita for a specific year.

Insights

- Correlation between GDP and energy consumption trends.
- Comparative analysis of energy consumption patterns for selected countries.

Code Explanation

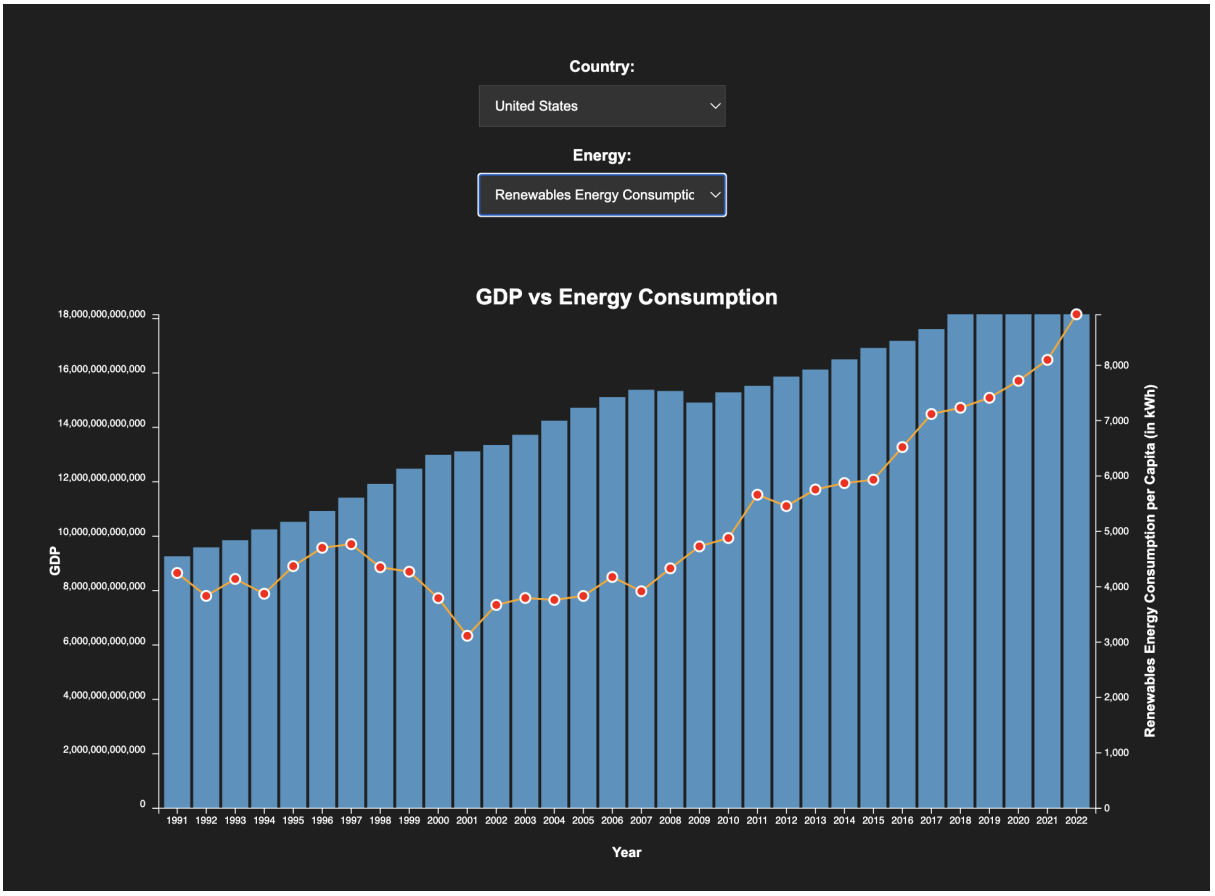
This JavaScript code uses the D3.js library to create an interactive chart that compares GDP and energy consumption for different countries over time. Here's a breakdown of the code:

1. **Data Loading and Preprocessing:**
   - The code loads data from a JSON file ("data/world_clean_dataset.json") using `d3.json`.
   - The "year" and "gdp" fields in the data are converted from strings to numbers using `forEach` and the unary `+` operator.
2. **Dropdown Options:**
   - Unique country names are extracted from the data and logged to the console.
   - A mapping of consumption types is defined, and a reverse mapping is created for line chart processing.
   - Dropdown options for country and consumption type are populated and logged to the console.
3. **Chart Configuration:**
   - Chart dimensions, color scales, and SVG elements are set up.
   - The color scale for bars and lines is initially based on the primary energy consumption.
4. **Chart Update Function (`updateChart`):**

- This function is responsible for updating the chart based on the selected country and consumption type.
- Existing chart elements (bars, line, axes) are removed.
- Data is filtered for the selected country, and x and y scales are created.
- Bars for GDP and a line for energy consumption are drawn with transitions.
- Tooltip functionality is implemented for mouseover events on bars and dots.

5. **Initial Chart Update:**
   - The chart is initially updated with the first country and consumption type in the lists.
6. **Event Listeners:**
   - Event listeners are added to update the chart when the user selects a different country or consumption type.
   - A tooltip div is created and positioned dynamically based on mouse events.
7. **Labels and Titles:**
   - Labels and titles are added to the chart, including a title, x-axis label, and y-axis labels for GDP and energy consumption.
8. **Y-Axis Label Update Function (`updateYAxisLabel`):**
   - A function is defined to update the y-axis label based on the selected consumption type.
9. **Initial Y-Axis Label Update:**
   - The y-axis label is initially updated with the first consumption type in the list.
10. **Y-Axis Label Update Event Listener:**
    - An event listener is added to update the y-axis label when the user selects a different consumption type.

The code creates a dynamic and interactive visualization that allows users to explore the relationship between GDP and various energy consumption types for different countries over time.



Bubble Chart

Story - Compares fossil fuel consumption and GDP for the top 5 most populous countries.

Marks - Dots/Bubbles

Channels - Color and size of bubbles

Tooltip - Tooltip displays specific values for fossil fuel consumption, renewable energy consumption, and GDP.

Insights

- Comparative analysis of fossil fuel and renewable energy consumption.
- Relationship between energy consumption, GDP, and environmental sustainability.

Code Explanation

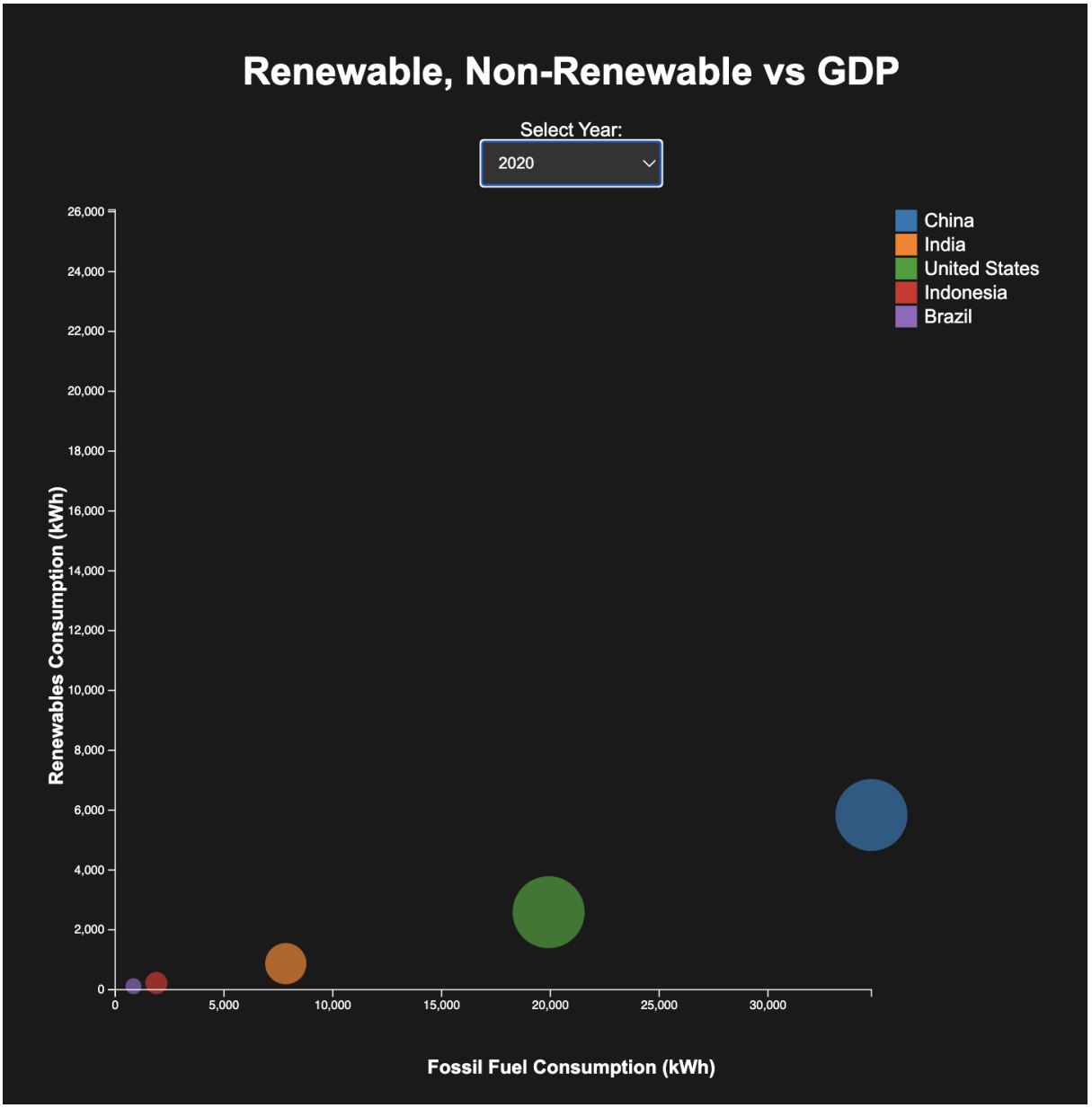
This JavaScript code uses the D3.js library to create a bubble chart that visualizes the relationship between fossil fuel consumption, renewables consumption, and GDP for the top 5 most populated countries in a given year. Here's a breakdown of the code:

1. **Set up SVG Canvas:**
   - Define canvas dimensions (`width`, `height`) and margins.
   - Append an SVG element to the body with specified dimensions.
2. **Load Data:**
   - Load data from a CSV file ("data/world_clean_dataset.csv") using `d3.csv`.
3. **Extract Unique Years:**
   - Extract unique years from the dataset and set a default selected year (`selectedYearDefault`).
4. **Create Year Dropdown:**
   - Populate a dropdown menu (`#yearFilter`) with unique years using D3's `selectAll`, `data`, `enter`, and `append` methods.
5. **Update Visualization Function (`updateVisualization`):**
   - This function is responsible for updating the bubble chart based on the selected year.
   - Filter data for the selected year and select the top 5 most populated countries.
   - Define scales for the x-axis (fossil fuel consumption), y-axis (renewables consumption), and bubble size (GDP).
   - Update the x and y axes with transitions.
   - Update existing or append new circles (bubbles) representing countries with transitions.
   - Add tooltips for mouseover events on bubbles.
   - Add legends for each country.
6. **Initial Visualization:**
   - Call `updateVisualization` with the default selected year.
7. **Add Event Listener:**

- Add an event listener to the year dropdown to update the visualization when the user selects a different year.

8. **Define Tooltip:**
    - Define a tooltip to display additional information on mouseover.
9. **Add Axes:**
    - Add x-axis and y-axis to the chart.
10. **Add Axis Labels:**
    - Add labels for the x-axis and y-axis.
11. **Styling:**
    - Set styles for text and legends, such as font weight and text color.

The code creates an interactive bubble chart that allows users to explore the relationship between fossil fuel consumption, renewables consumption, and GDP for the top 5 most populated countries in a selected year. The chart is updated dynamically when the user selects a different year from the dropdown menu.



Stacked Area Chart

Story - Displays the composition of energy production for a selected country over time.

Marks - Areas represent different energy sources.

Channels - Color-coded areas represent energy types.

Tooltip - Tooltip reveals the percentage of the fuel compared to total, over hovered energy source.

Insights

- Trends in energy source distribution over the years.
- Comparative analysis of the contribution of each energy type.

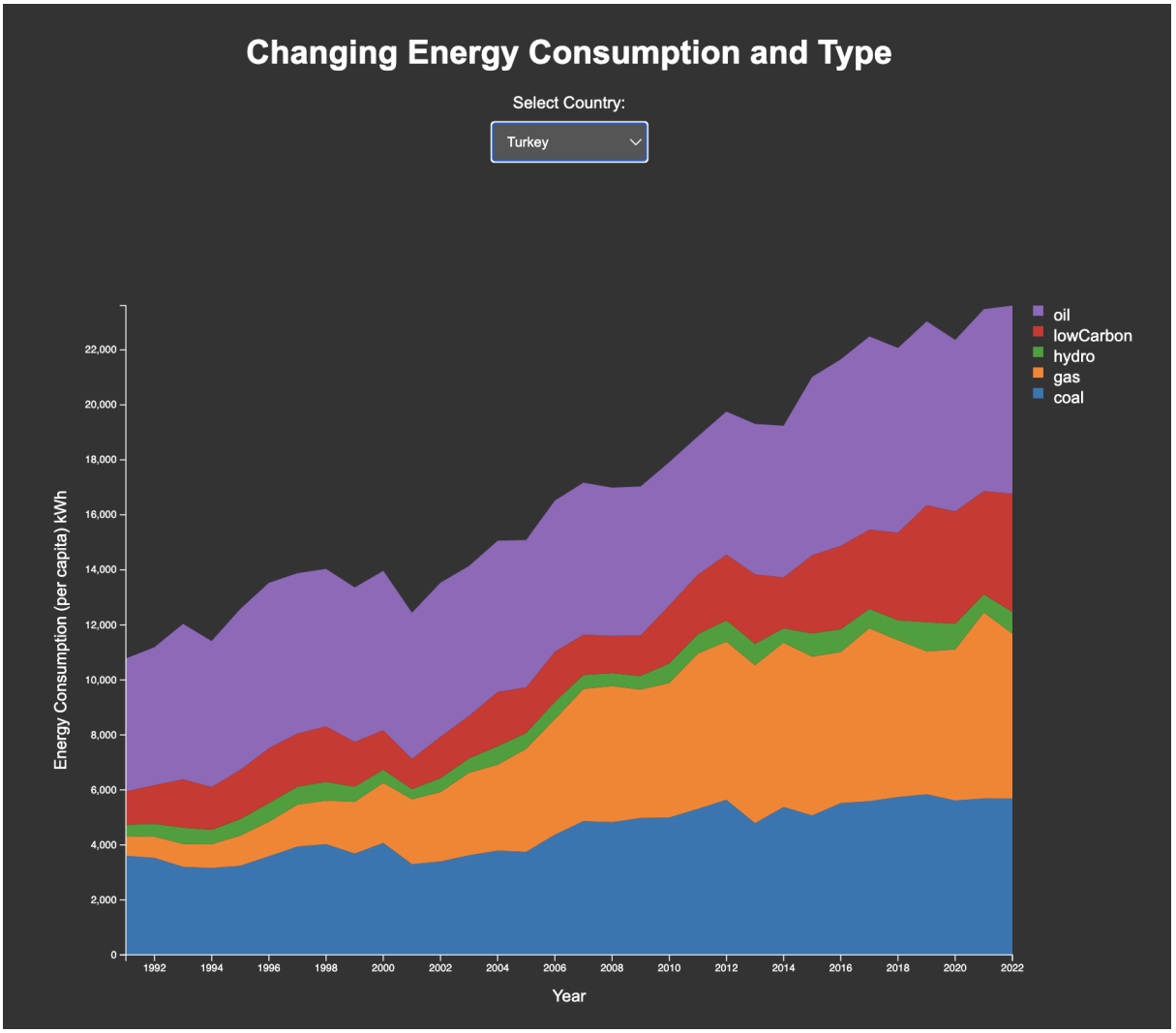
Code Explanation

This JavaScript code uses the D3.js library to create a stacked area chart visualizing the per capita energy consumption of different fuel types over time for a selected country. Let's break down the code:

1. **Set Up Chart Dimensions:**
   - Define the margin, width, and height for the chart.
2. **Append SVG Element:**
   - Append an SVG element to the body, considering the defined dimensions and margin. Also, create a `<g>` (group) element within the SVG and translate it by the margin.
3. **Parse Date/Time:**
   - Use D3's `timeParse` function to parse the date format "%Y" (year).
4. **Load Data:**
   - Load data from a CSV file ("world_clean_dataset.csv") using `d3.csv`.
5. **Extract Unique Countries:**
   - Extract unique country names from the dataset.
6. **Populate Country Dropdown:**
   - Create a dropdown menu (`#countrySelect`) and populate it with the unique country names.
7. **Initialize and Update Chart:**
   - Initialize the chart with the first country in the dropdown.
   - Add an event listener to the dropdown to update the chart when a new country is selected.
8. **Update Chart Function (`updateChart`):**
   - Filter the data based on the selected country.
   - Convert data types, parsing the year and converting energy consumption values to numbers.

9. **Stack Data:**
    - Use D3's stack function to stack the data based on keys (fuel types: coal, gas, hydro, lowCarbon, oil).
10. **Color Scale:**
    - Define a color scale for different fuel types.
11. **Set Up Scales:**
    - Set up time and linear scales for the x and y axes.
12. **Area Generator:**
    - Create an area generator function for the stacked area chart.
13. **Remove Existing Areas:**
    - Remove existing area paths to update with new data.
14. **Create Tooltip:**
    - Create a tooltip element for displaying information on mouseover events.
15. **Add Areas to the Chart:**
    - Append new area paths to the chart based on the stacked data.
16. **Add Mouseover and Mouseout Events:**
    - Implement mouseover and mouseout events to highlight areas and display a tooltip with information.
17. **Add Axes:**
    - Remove existing x and y axes and add new ones based on the updated scales.
18. **Add Labels:**
    - Remove existing x and y labels and add new ones.
19. **Add Legend:**
    - Remove the existing legend and add a new one based on the fuel types and color scale.

The resulting chart is a stacked area chart showing the per capita energy consumption of different fuel types over time for a selected country. Users can interact with the chart by selecting different countries from the dropdown menu.



Conclusion

In conclusion, this comprehensive report has successfully implemented a diverse set of visualizations, showcasing the versatility of data representation. The stacked area chart offers a nuanced view of per capita energy consumption across various fuel types over time, providing a detailed insight into the energy landscape of selected countries. The bubble chart, focusing on fossil fuel and renewables consumption, elegantly visualizes the top 5 most populated countries' contributions to the global energy scenario. The Pareto chart provides a clear and prioritized display of contributing factors, aiding in efficient decision-making. Lastly, the globe visualization utilizes geographic data to enhance our understanding of worldwide energy patterns. Each visualization employs D3.js, demonstrating the power and flexibility of this

library in creating engaging and informative data visualizations. Collectively, these visualizations serve as valuable tools for comprehending and communicating complex energy-related data to a diverse audience.

References

1. https://globe.gl/
2. https://zebrabi.com/guide/how-to-customize-pareto-chart-in-d3-js/
3. https://d3-graph-gallery.com/bubble
4. https://observablehq.com/@d3/stacked-area-chart/2