Data Analysis and Visualization Project's Report

Project:

Interactive Data Visualization Dashboard for TB Using D3

Group members:

I211376 Irtaza Malik
I211761 Nabeeha Fazail
I211721 Hurraida noor

Introduction

This report covers the implementation of interactive data visualizations. The applications were built using **D3.js**, with additional libraries to enhance functionality. The aim was to provide intuitive and visually engaging interfaces for exploring complex datasets. Below are detailed design choices, preprocessing steps, and feature highlights.

Force-Directed Graph

- **Physics Simulation:** Utilizes D3.js's force simulation for a natural layout. Nodes repel each other, while links act as springs, creating a balanced distribution of nodes.
- **Responsive Design:** Adjusts dynamically to different screen sizes.

2. Interactivity

- **Tooltip on Hover:** Displays detailed information about a node, including its relationships.
- **Node and Link Highlighting:** Clicking on a node highlights it and its directly connected links.
- **Zoom and Pan:** Supports zooming (1x to 8x) and panning for detailed exploration.
- **Draggable Nodes:** Allows users to reposition nodes manually.

3. Filters

- **By Relationship Type:** Filters links based on their relationship type (e.g., "Collaboration," "Conflict").

- **By Node Type:** Filters nodes based on their type (e.g., "Type 1," "Type 2").
- Filters work independently or together for advanced queries.

4. Legend and Breadcrumb

- **Legend:** Displays all node types with corresponding colors. Clicking on a type highlights related nodes.
- **Breadcrumb:** Updates dynamically to track the user's current exploration path.

5. Image Export

- **Download Button:** Captures the graph as an image (PNG) using the `html2canvas` library.

6. Dynamic Updates

- Handles real-time updates when the window resizes, maintaining layout integrity.

Design Choices

1. Visualization Framework

- D3.js was selected for its robust visualization capabilities and flexible force-directed graph layout.

2. Color Scheme

- The `d3.schemeCategory10` color palette ensures nodes are distinct and visually appealing.

3. Data Interaction

- Clickable nodes and tooltips enhance user engagement.
- Filters provide tailored insights, catering to varied user requirements.

4. Performance Optimization

- The simulation uses `d3.forceManyBody()` with negative strength to minimize node overlap while ensuring quick convergence.

- Bounding constraints prevent nodes and links from moving outside the visible area.

Data Preprocessing

The data used in this graph comprises nodes and links formatted as a JSON file (`updated_graph_data.json`). Preprocessing steps include:

1. **Data Cleaning:**
    - Ensured all nodes have unique IDs.
    - Verified all links reference valid source and target nodes.
2. **Attribute Annotation:**
    - Nodes are annotated with `id` and `type`.
    - Links are annotated with `source`, `target`, and `value`.
3. **Graph Properties:**
    - Defined link distance as 50 units to maintain a readable layout.

Interactive Features Details

Tooltip

- **Purpose:** Provides contextual information.
- **Implementation:** Tooltip follows the mouse cursor and displays:
    - Node `id`.
    - Connected relationships and their types.

Node Highlighting

- **Purpose:** Emphasizes selected nodes and links.
- **Implementation:**
    - Highlighted nodes turn orange.
    - Connected links change stroke color to red.

Filters

- **Purpose:** Focuses on specific data subsets.
- **Implementation:**

    - `select` elements trigger style updates.
    - Hidden nodes/links remain part of the simulation but are visually excluded.

Zoom and Pan

- **Purpose:** Allows exploration of large graphs.
- **Implementation:** Utilizes `d3.zoom()` to scale and translate the entire graph.

Image Export

- **Purpose:** Saves the current graph view for offline analysis.
- **Implementation:** Captures the SVG using `html2canvas` and downloads it as a PNG.

Future Enhancements

1. **Dynamic Data Loading:**
    - Add options to load new datasets without refreshing the page.
2. **Advanced Filtering:**
    - Include multi-criteria filtering (e.g., filter nodes based on multiple attributes).
3. **Custom Styling:**
    - Allow users to customize node colors, sizes, and link widths dynamically.
4. **Search Functionality:**
    - Integrate a search bar to quickly locate specific nodes.

Interactive Map

1. Overview

This project features an **interactive world map** created using D3.js, TopoJSON, and a combination of geographic and relationship data. The map visualizes tuberculosis (TB) incidence data across the globe, with dynamic features like filtering, region-specific exploration, temporal animation, and relationship-based entity highlighting.

2. Key Functionalities

- **Geographic Visualization:** Displays countries on a world map, rendered using TopoJSON data.
- **Data Overlay:** Plots data points (representing entities) on the map with pins based on geographic coordinates.
- **Year Selection:** Filters data based on the selected year from a dropdown menu.
- **Region Filtering:** Filters the map to display entities specific to a chosen region.
- **Relationship Highlighting:** Highlights entities with specific types of relationships, such as "Close" or "Distant."
- **Temporal Animation:** Automates the transition across years, providing an animated view of data evolution.
- **Exporting:** Enables downloading the current map view as an SVG image for sharing or reporting.
- **Zoom and Pan:** Provides a smooth zoom and pan functionality to explore details at different scales.
- **Tooltip Interaction:** Displays detailed information for each entity when the user hovers over a pin.

---

3. Design Choices

- **Map Projection:** The **Mercator projection** was chosen for simplicity and wide usage in global mapping applications.
- **Color Coding:** Entities are color-coded by relationship type for easy visual differentiation (e.g., green for "Close" and red for "Distant").
- **Interactivity:** The inclusion of dropdown menus, tooltips, zoom, and filtering enables user exploration and detailed analysis.
- **Responsiveness:** The map dynamically adjusts to the browser window size, providing a consistent user experience across devices.

---

4. Data Preprocessing Steps

The data preprocessing involved:

1. **Geospatial Data (TopoJSON):**
   - Countries' boundaries were extracted from the **world-110m.json** file and converted to a usable format with D3.
2. **TB Data (CSV):**
   - **cleaned_tb_data_with_lat_lon.csv** was preprocessed to ensure geographic coordinates and TB incidence data were in the correct format.

    - Redundant rows were removed, and missing geographic details were supplemented.
3. **Relationship Data (JSON):**
    - **updated_graph_data.json** was refined to include source-target pairs and relationship types.
    - Validation ensured consistency between entity names in the relationship data and geographic data.

---

5. Interactive Features

- **Dropdown Filters:**
    - Users can filter entities by year, region, and relationship type.
- **Zoom and Pan:**
    - Built-in D3 zoom behavior allows users to navigate the map seamlessly.
- **Entity Highlighting:**
    - Clicking on an entity (pin) highlights related entities based on the selected relationship type, with enhanced visual cues (opacity and border changes).
- **Temporal Animation:**
    - The "Play" button cycles through years, showing changes in data over time, while the "Pause" functionality gives users control over the animation.
- **Tooltip Display:**
    - Hovering over an entity displays detailed metadata like TB incidence rate, region, and year.
- **Export Map as Image:**
    - The map's current view can be downloaded as an SVG image using the "Export" button.

---

6. Legend and Accessibility

- **Legend:** A visual legend helps users quickly interpret relationship types. Clicking on legend items dynamically updates the map to focus on the selected relationship type.
- **Accessibility:** The interface is designed with simple controls and descriptive labels for intuitive navigation.

---

7. Future Enhancements

- **Dynamic Data Updates:** Add real-time data integration for updated TB statistics.
- **Enhanced Filtering:** Introduce multi-select filters for granular analysis.
- **Mobile Optimization:** Improve usability on smaller screens by introducing collapsible menus and touch-friendly interactions.

Timeline Visualization

Overview

The **Enhanced Timeline Visualization** is an interactive web application built using D3.js to display tuberculosis (TB) data trends across different age groups, years, and countries. It allows users to explore relationships between TB incidence and deaths through dynamic, intuitive visualization.

Functionalities

1. Interactive Visualization

- **Scatter Plot**: Displays TB incidence (X-axis, logarithmic scale) vs. TB deaths (Y-axis, linear scale).
- **Dynamic Radius**: Circle size represents the magnitude of deaths for a selected age group or cumulative deaths for all age groups.
- **Color Coding**: Circles are color-coded to distinguish between different entities (countries or regions).

2. Filters

- **Year Selector**:
    - A slider and previous/next buttons allow users to select a specific year to view data.
    - "Play" functionality animates year progression, updating the visualization automatically.
- **Age Group Selector**:
    - Dropdown menu allows users to focus on specific age groups or view cumulative data.

    - Options include: All age groups, 15–49 years, 50–69 years, 70+ years.

3. Legend

- **Interactive Legend**:
    - Displays unique entity codes with corresponding colors.
    - Clicking on a legend item toggles the visibility of associated data points.

4. Tooltip

- **On Hover**: Displays detailed information about the selected data point, including:
    - Country/Entity
    - Year
    - TB Incidence (per 100,000 population)
    - TB Deaths

5. Export

- **Export to SVG**:
    - Allows users to download the current visualization as an SVG file.

Design Choices

1. Visualization Style

- **Scatter Plot**:
    - Chosen for its ability to represent two continuous variables (incidence and deaths) simultaneously.
- **Logarithmic X-axis**:
    - Handles the wide range of TB incidence values effectively.
- **Dynamic Radius**:
    - Visual emphasis on the magnitude of deaths within or across age groups.

2. User Experience

- **Interactivity**:
    - Tooltips and legend interactivity enhance user engagement and ease of data interpretation.
- **Responsive Design**:
    - Scales dynamically to fit various screen sizes.

3. Scalability

- **Data Binding**:
    - Utilized D3’s data-binding capabilities to dynamically update visual elements based on selected filters.

Data Preprocessing

1. Data Source

The data is sourced from `cleaned_tb_data_with_lat_lon.csv`, which contains the following key fields:

- **Year**: Data is available from 2000 to 2019.
- **Entity**: Country or region.
- **TB Incidence**: Estimated incidence of TB per 100,000 population.
- **TB Deaths**: Disaggregated by age groups: 15–49, 50–69, 70+, under 5, and 5–14 years.

2. Preprocessing Steps

1. **Data Parsing**:
    - Converted numeric fields (e.g., `Incidence`, `Deaths`) to numbers.
2. **Age Group Aggregation**:
    - Created a cumulative deaths column for "All Age Groups."
3. **Validation**:
    - Ensured no missing or corrupted values for essential fields.

Interactive Features

1. Play/Pause Animation

- Animates through the dataset year by year.
- Users can pause at any point or skip to a specific year.

2. Tooltip

- Highlights specific details on hover.
- Positioned dynamically to avoid overlapping with the visualization.

3. Legend Interactivity

- Allows users to focus on specific entities by toggling their visibility.

4. Filtered Updates

- Dynamically filters and updates data based on user-selected age groups or years.
- Smooth transition animations for seamless updates.

Challenges and Solutions

1. Logarithmic Scale on X-axis

- **Challenge**: Handling zero values in logarithmic transformation.
- **Solution**: Excluded or adjusted data points with zero TB incidence.

2. Data Overlap

- **Challenge**: Overlapping circles obscured smaller data points.
- **Solution**: Used dynamic radius scaling and tooltip interactivity.

3. Export Functionality

- **Challenge**: Maintaining SVG quality in exports.
- **Solution**: Leveraged XMLSerializer to produce scalable, high-quality SVG output.

Future Enhancements

1. **Data Integration**:
    - Include additional health metrics (e.g., comorbidities, vaccination rates) for richer analysis.
2. **Advanced Interactivity**:

    - Implement brushing and linking to allow users to zoom into specific data ranges.
3. **Dynamic Legends**:
    - Add filters to dynamically generate legends for subsets of data.

Hierarchical Tree Map with Country and Year Selection

Overview

This application visualizes tuberculosis (TB) data in a **hierarchical tree map** format, allowing users to interactively explore TB statistics by **country**, **year**, and **age group breakdown**. It provides insights into TB incidences and deaths across different demographics, offering a clear representation of hierarchical relationships within the dataset.

Functionalities and Interactivity

Key Features:

1. **Country Selection**:
    - A dropdown menu allows users to select a specific country.
    - On selection, the visualization updates to display TB data for the chosen country.
2. **Year Selection**:
    - Clicking on a year within the tree map drills down into that year's data, showing a breakdown of TB incidences and deaths by age group.
3. **Age Group Breakdown**:
    - When viewing a specific year, the data is divided into:
        - **Incidence**: Total TB cases.
        - **Deaths**: Categorized by age groups (e.g., Under 5, 15-49 years, 70+ years).
4. **Breadcrumb Navigation**:
    - A breadcrumb trail at the top of the visualization allows users to navigate between levels (country → year → age group).
    - Clicking on any breadcrumb node zooms back to that level.

5. **Interactive Zooming**:
    - Clicking on a node (e.g., a year or age group) zooms into its sub-hierarchy for detailed analysis.
6. **Dynamic Textual Representation**:
    - Nodes display both absolute values and percentages of their parent node for context.
    - This includes total deaths as a percentage of the total TB incidence.

User Interaction Highlights:

- **Tooltip-Free Design**: All information is displayed directly within the nodes, reducing the need for external tooltips.
- **Drill-Down and Zoom**: Users can click on nodes to explore hierarchical relationships or return to the previous view using breadcrumbs.
- **Dynamic Visualization**: Changes in country or zoom level dynamically update the tree map without refreshing the page.

Design Choices

1. Hierarchical Tree Map:

- **Why?** Tree maps provide a compact and hierarchical representation, making it easy to compare values and percentages across categories and subcategories.
- **Structure**:
    - Root Node: The country.
    - Level 1: Years.
    - Level 2: Breakdown into incidence and deaths by age group.

2. Color Scheme:

- D3's `schemeCategory10` is used to differentiate nodes based on their depth in the hierarchy.
- Colors provide visual cues for node categories and improve user experience.

3. Breadcrumb Navigation:

- Breadcrumbs ensure smooth navigation between hierarchical levels, enhancing usability.

4. Dynamic Layout:

- The visualization resizes and adjusts dynamically to fit the screen dimensions.

Data Preprocessing Steps

1. **Dataset Overview**:
   - The hierarchical JSON dataset includes:
     - Country-level data.
     - Yearly statistics for each country, including:
       - Incidence (total TB cases).
       - Deaths, categorized by age groups.
2. **Transformations**:
   - Data is structured hierarchically:
     - `Root` → `Years` → `Incidence` and `Deaths`.
   - Percentages are calculated dynamically during rendering:
     - Percentage of total deaths relative to incidences.
     - Percentage of age-group-specific deaths relative to total deaths.
3. **Missing or Incomplete Data**:
   - Years or age groups with missing values are excluded from the visualization.
4. **JSON Format**:
   - Input data is in a hierarchical JSON format for easy integration with D3.js.

Interactive Features

1. Zooming and Drilling:

- Clicking on nodes zooms into the next level of data.
- This feature simplifies hierarchical data exploration.

2. Dynamic Data Transformation:

- On selecting a country, the dataset is filtered, and the visualization updates to display relevant data.
- When a year is selected, the tree map dynamically focuses on that year’s breakdown.

3. Breadcrumb Navigation:

- Provides a clear path for returning to parent nodes without losing context.

Technical Implementation

1. D3.js Library:

- D3 is used for:
    - Constructing the tree map layout.
    - Applying transitions for smooth zooming and updates.
    - Handling events for interactivity.

2. HTML and CSS:

- **HTML**: Provides the structural layout, including the country dropdown and breadcrumb navigation.
- **CSS**: Ensures responsive design and clear visual hierarchies.

3. JavaScript Logic:

- Manages state (selected country, selected year).
- Handles transformations and dynamic rendering.

Sunburst Chart for TB Data

Overview

The **Enhanced Sunburst Chart** is a data visualization tool designed to display hierarchical tuberculosis (TB) data. This chart allows users to explore nested data interactively and understand relationships within the dataset, such as the distribution of TB cases or deaths by various demographic levels.

Functionalities and Interactivity

Key Features

1. **Interactive Sunburst Visualization**:
    - **Hierarchical Structure**: The chart uses nested arcs to represent the hierarchical nature of the dataset.
    - **Zooming and Drilling**:
        - Clicking on a segment zooms into that part of the chart for a closer view.
        - This enables detailed exploration of subcategories (e.g., drilling from country-level data to specific demographics).
2. **Tooltip Support**:
    - A tooltip appears on hovering over a segment, showing:
        - The name of the segment.
        - The value associated with it.
3. **Reset Zoom**:
    - A button allows users to reset the zoom and return to the root view of the dataset.
4. **Export as Image**:
    - Users can export the current view of the chart as an **SVG file**, enabling sharing or further usage.
5. **Legend with Interactivity**:
    - A legend at the bottom provides a color-coded reference for the top-level categories.
    - Clicking a legend item toggles the visibility (opacity) of related segments, allowing users to highlight or suppress specific categories.

Design Choices

1. Sunburst Chart:

- **Why Sunburst?**
    - Sunburst charts excel at representing hierarchical data in a compact, radial format.
    - The circular layout helps maximize screen space, especially for multi-level datasets.
- **Arc Segments**:
    - Each segment represents a hierarchical node.

    - Size of segments correlates with their value, providing intuitive visual scaling.

2. Color Scheme:

- D3's `schemeCategory10` is used for vibrant and distinguishable colors.
- Colors are assigned based on the depth of the hierarchy, ensuring clarity across levels.

3. Responsive Design:

- The chart is centered and adjusts its dimensions based on screen size for optimal viewing.

4. Tooltip Design:

- Tooltips provide instant contextual information without overwhelming the user interface.
- They appear near the cursor, offering precise details about the hovered segment.

Data Preprocessing Steps

1. Dataset Overview:

- The dataset is hierarchical, following this structure:
    - Root node (e.g., global data).
    - Level 1: Categories (e.g., regions or countries).
    - Level 2: Subcategories (e.g., age groups, TB incidences, or deaths).

2. Transformations:

- Each node in the hierarchy includes:
    - A name (e.g., "Incidence", "Under 5 Years").
    - A value (e.g., number of cases).
- Missing values or zero values are filtered to prevent rendering empty segments.

3. Hierarchical Aggregation:

- The root node aggregates the total value of all descendants.

- Subcategories are recursively summed to ensure accurate representation.

4. JSON Format:

- The data is stored in a hierarchical JSON format compatible with D3’s partition layout.

Interactive Features

1. Zooming and Drilling:

- **How It Works**:
    - Clicking on a segment adjusts the chart’s scales to focus on that segment and its children.
    - The transition is animated for smooth user experience.
- **Benefits**:
    - Enables detailed exploration of data subsets.
    - Reduces visual clutter by isolating relevant segments.

2. Legend Interaction:

- **How It Works**:
    - Clicking a legend item toggles its related segments' visibility.
    - Active items are highlighted, while inactive items appear faded.
- **Benefits**:
    - Users can focus on specific categories or hide irrelevant data.

3. Tooltip Display:

- **How It Works**:
    - The tooltip follows the cursor and displays relevant segment data.
- **Benefits**:
    - Provides clear and immediate information without disrupting the chart view.

4. Reset Zoom:

- **How It Works**:
    - A button restores the initial view, displaying all segments.

- **Benefits**:
  - Simplifies navigation, especially after multiple zoom interactions.

5. Export Chart as SVG:

- **How It Works**:
  - Serializes the current SVG into a downloadable file.
- **Benefits**:
  - Users can save and share the visualization for reports or presentations.

Technical Implementation

1. D3.js Library:

- D3 is used for:
  - Partitioning the dataset into hierarchical segments.
  - Rendering the sunburst chart with scalable arcs.
  - Managing transitions and user interactions.

2. SVG and Arc Generators:

- SVG elements form the basis of the visualization.
- D3's `arc` function is configured to calculate segment dimensions.

3. Scales and Transitions:

- Linear scales map data values to angular and radial dimensions.
- Transitions are applied to zoom interactions for smooth animations.

4. CSS Styling:

- Tooltips, legend items, and the chart itself are styled for clarity and responsiveness.
- Hover effects and active states enhance visual feedback.