VIS-Assignment 5

Maitham Al-rubaye
a12142043

1. Motivation

Service providers that help compensate for gas emissions should offer reliable options to customers and other clients. They need to identify where emissions are highest, which forests have large amounts of carbon, and where climate disasters occur most often. The key is to show this information so users can compare multi climate indicators in a short time, explore the changes across a selected time window, and see how the indicators relate across countries.

User Tasks (from assignment A4)

I designed the dashboard to support User Group A (Service Providers):

- **A1:** Identify reliable compensation options by using emissions (dataset #03), forests (dataset #13), and disasters (dataset #14).
- **A2:** Identify promising countries for forest projects by finding large forest carbon stocks (dataset #13) with high emissions (dataset #03).
- **A3:** Identify countries for disaster-related projects by finding frequent climate disasters (dataset #14) with strong warming (dataset #23).

Selected design

I selected Dashboard A (Map-Driven Offset Explorer) from assignment A4. My design uses a 2×2 grid with four linked views which are a world map (choropleth), a time series (emissions and temperature), a scatterplot (emissions vs forest carbon), and a bar chart (disaster ranking). I chose Dashboard A because it is simpler for non experts and supports fast comparison with a clear map entry point and linked views for the time window and cross country relationships.

2. Prototyping / Design Process

Initial prototyping and data challenges

I implemented the four views as Vue.js components using D3.js. The main challenge was data quality: Dataset #03 has many missing values in the early years (1970s–1980s) and target years (2024–2029). The 2024–2029 gaps are expected because these are future target years where not all countries have submitted their expectations. There, I decided to filter for reliable years (e.g., "up to 2023; default window 2013–2023"). I implemented the countries with missing data as gray on the map with "No data" labels, and tooltips show "n/a" instead of zero.

Performance optimization (AI: VS code IDE copilot)

Aggregating values over a selected time window for almost 190 countries was initially slow (200–300 ms). I optimized this by using prefix sums for each country's time series, which enables O(1) range aggregation instead of looping over years (O(n)). This reduced aggregation time to under 10 ms.

What Worked on Real Data

- **Linked interaction across views:** Selecting countries in any view (map, scatterplot, or bar chart) highlights the same countries in the other views. This makes it easy to start from whichever view is most informative and then confirm patterns elsewhere.
- **Map (A1):** In choropleth the high emission countries have dark colors. I used border thickness for disasters to avoid clutter and to highlight countries with more disasters.
- **Time series (A2):** The chart with emissions (solid) and temperature (dashed) shows the patterns. Time brushing updates all views. The default window (last 10 years) as a starting point can be adjusted.
- **Scatterplot (A3):** Emissions vs forest carbon for comparing countries across both indicators. Point size encodes disasters. You can select a single country by single click and to select a group I implemented a rectangular brushing.
- **Bar chart (A4):** The Top-10 disaster ranking countries are sorted and selectable.

What didn't work (and changes made)

- **Forest data gaps:** Dataset #13 has more missing values than expected. I treated Missing values as N/A (not zero) in tooltips and the scatterplot.
- **Default time window:** Starting with the full range made early years hard to read due to missing values in Dataset #03. I switched the default to the last 10 years and I let users brush across ranges.

- **Multi-selection feedback:** I added “Shift+Click to multi-select” hints and selected country tags with remove buttons, making multi-selection discoverable.

The final dashboard answers the user tasks:

- **A1:** Users can quickly see where emissions are high and then check forest carbon and disaster frequency using linked selection across the other views.
- **A2:** Users can find countries with high emissions and large forest carbon by selecting in the scatterplot (or map) and verifying trends in the time series.
- **A3:** Users can identify high disaster countries using the bar chart and then confirm their warming and emissions in the other linked views.

3. Implementation Details

Views and Functionality

Figure 1 (page 6) shows the dashboard layout:

- **Header:** Title, filter buttons (All/Africa/Americas/Asia/Europe/Oceania), time window display, selected country tags and a button to clear selections.
- **View A1 (top-left):** World map with emissions as fill color (Inferno scale), disasters as outline thickness. Click to select; Shift+Click for multi-select. Hover shows tooltip. Legend explains encodings.
- **View A2 (top-right):** Dual-axis line chart with emissions (solid, left axis) and temperature (dashed, right axis). Drag to brush time window. Brushed region highlighted in blue. Legend shows selected countries.
- **View A3 (bottom-left):** Scatterplot with emissions (x) and forest carbon (y). Size = disasters. Color = Inferno scale (same as map). Drag rectangle to brush or draw new one. Selected countries have a blue outline. Legend explains size and color.
- **View A4 (bottom-right):** Top 10 bar chart by disasters. Bars colored by emissions (Inferno same as others). Click to select. Country names on left; counts labeled on bars.

How Implementation Addresses Tasks

- **A1:** The map (A1) provides a global overview of emissions with disaster context, and linked selection lets users highlight the same countries across views. The scatterplot (A3) relates emissions to forest carbon, and the time series (A2) shows how these patterns change within the selected time window.
- **A2:** Users brush the scatterplot (A3) to target countries with high emissions and high forest carbon; the map (A1) confirms where these countries are located, and the time series (A2) shows their emissions trends in the selected window.

- **A3:** The bar chart (A4) ranks countries by disaster frequency; selecting a country propagates to the map (A1) and time series (A2) to check emissions and warming patterns for the same selection.

Technical Decisions

Framework: Vue.js 3 for reactive state; D3.js v7 for all visualizations; TopoJSON for map geometry.

Data processing:

- Loaded four CSVs in parallel; standardized column names (ISO3, Country, Region).
- Melted wide-format (years as columns) to long format for easier aggregation.
- Pre-computed prefix sums for O(1) time-window queries.
- Forest "latest value" logic handles sparse data.

Color consistency: Same Inferno sequential scale for emissions across A1 (map fill), A3 (scatter fill), A4 (bar fill). Selected countries get blue.

Missing values: Never shown as zero. Map uses gray fill with "No data" legend; scatterplot shows gray dots with dashed outline; bar chart shows "n/a"; tooltips display "data missing" or "n/a".

Interactions:

- Select country/countries click (toggle), Shift+Click (add). Tags in header with "×" remove buttons.
- Time brushing by dragging on A2 sets a window for all views, shown as blue overlay.
- Spatial brushing by dragging a rectangle on A3 selects countries in region or single selection by clicking on a single one.
- Region filtering buttons in header to filter all views to region.
- Hover tooltips as a dark background, positioned near the cursor.

Responsive: All views use ResizeObserver; CSS Grid for 2×2 layout fits to window size.

4. Discussion

What I Learned

- sums can improve performance too much (200ms to <10ms for aggregation).
- Color linking across views created strong visual connections, as users immediately understood "black = high emissions" everywhere.
- Missing data handling by never showing as zero, and by using neutral colors and "n/a" labels.
- Interaction feedback (tags, tooltips) is important for usability.

What I Would Do Differently

- Reduce the risk of confusion from the dual-axis time series by adding clearer axis labeling and a short note explaining that the two lines are different units/scales.
- Add save button feature storing state in URL query string for sharing/bookmarking.
- Test with real users from service providers to understand the usability issues.

Can Users Solve the Tasks?

Yes, users can solve tasks A1–A3 because the dashboard supports:

(1) fast cross country comparison (map + ranking),

(2) time window exploration (brushing in A2),

(3) linked selection across views to connect emissions (#03), forest carbon (#13), disasters (#14), and warming (#23) for the same countries.

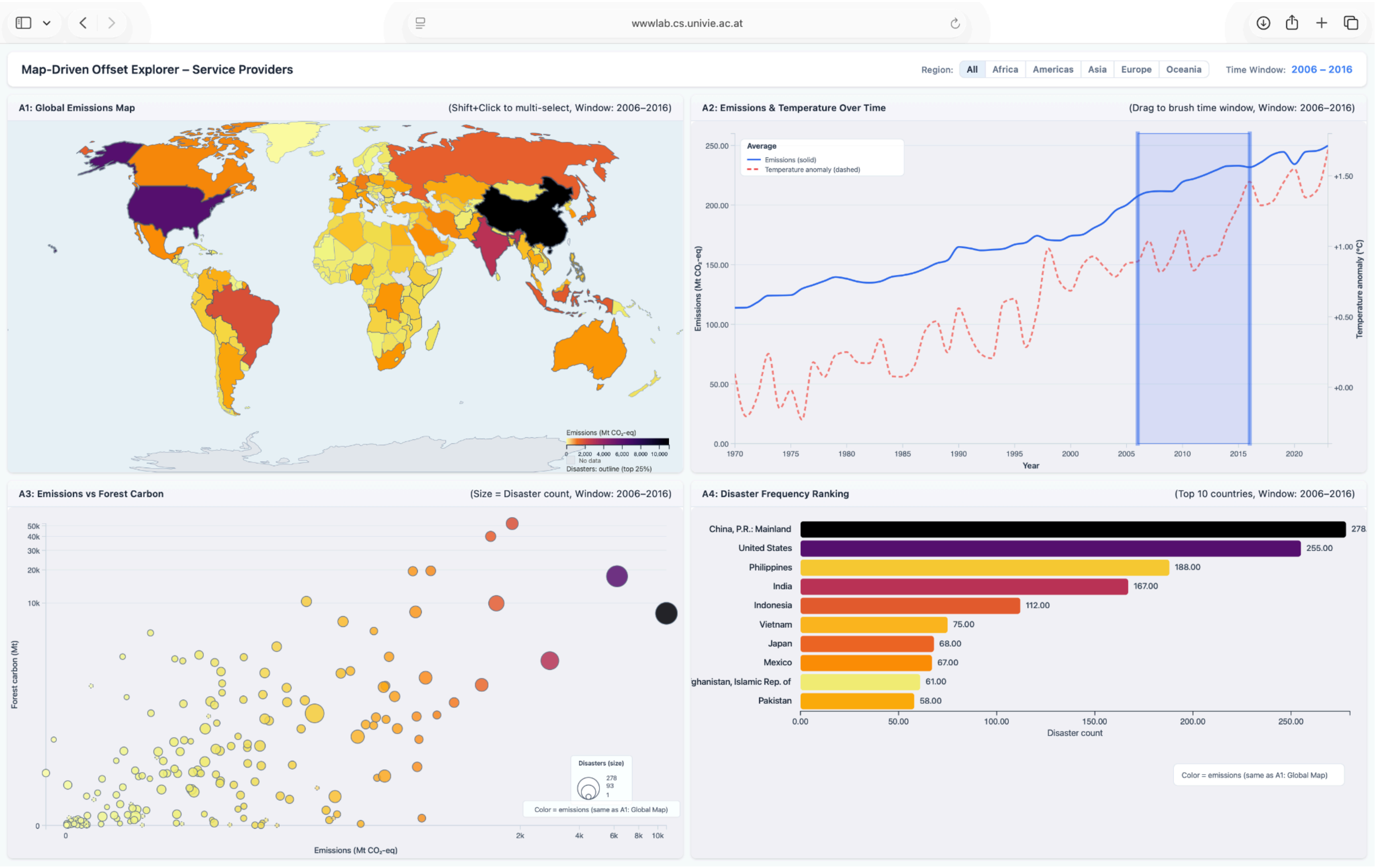
5. Conclusion

This assignment implemented Dashboard A from A4 as a fully functional Vue.js + D3.js dashboard. The implementation demonstrates effective encodings (position, color, size, outline thickness), linking (shared selection and time brushing), overview detail workflow (map/bar chart → scatter/time series), and attention to usability (no scrolling, tabs, pop-ups, clear labels and legends).

The prototyping process showed data quality issues (missing values in forest and target-year emissions) and performance bottlenecks (solved with prefix sums). The final dashboard addresses the original user tasks and provides a foundation for further refinement (user testing / accessibility checks).

AI use:

I used GitHub Copilot in VS Code for code completion and refactoring suggestions (e.g., prefix sums optimization). I verified and tested all generated code and troubleshootings.



Map-Driven Offset Explorer – Service Providers

Region: All | Africa | Americas | Asia | Europe | Oceania

Time Window: 2006 – 2016