Visualizing Airline Flight Delay and Cancellation Data

Lebedev Georgiy Konstantinovich, Jiang Yi, Tsakalidou Ioanna
Data Visualization, Milestone 2

1. Sketches

U.S. Airport Origin Stats

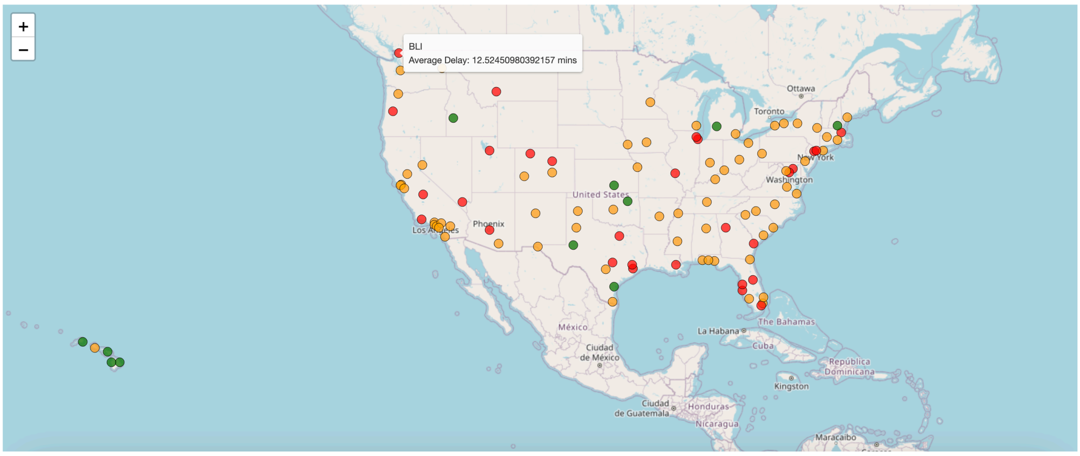

Figure 1: U.S. Airport Origin Stats Web Visualization

The main page shows a map of the U.S., with airports color-coded based on average delay durations. Red indicates higher average delays, while green signifies better arrival rates. This visualization gives an immediate overview of airport-level performance across the country.

2. Tools and Lectures

Tools: For building the interactive map we were using Leaflet.js. For the supplementary charts (delay cause, time-of-day effects, seasonality), we were using: D3.js. For preprocessing and aggregating the raw flight data, we used Pandas, NumPy, geopandas, and scikitlearn. For airport geolocation mapping, we used GeoJSON.

Relevant Lectures: Maps and Graphs were key in choosing effective projections and color schemes for delay visualization. Data and Interactions were used for planning filtering interactivity. Uncertainty and Multivariate Data influenced our approach to visualizing delay distributions and multifactor causes.

3. Work Breakdown

Features Implemented So Far:

- **Interactive U.S. map**: Showing delay severity at each origin airport via colored dots.
- **Responsive interface**: Users can zoom, pan, and interact with airports.
- **Data backend**: Preprocessed CSVs used for quick loading and lightweight interaction.
- **Data abstraction**: Aggregated delays and cancellation causes grouped by airport.

Next Steps:

- Implement temporal filters (by month, weekday, time of day).
- Visualize route-level delays using arcs or lines.
- Integrate plots for delay causes and cancellation trends.
- Refine UI styling and performance optimizations.

We will breakdown the work as follows:

- Complete the map (Currently only starting stations are shown, we would like to show ending stations)
- Implement time/date selection of the data
- flights/hour graph
- Flights demographic graph
- Non-visualization styling (plain HTML/CSS) of the site
- stretch goal, allow selections over the graphs to alter the data presented in the other graphs (Currently, only selections on the map update the data for all visualizations)
- stretch goal, allow multiple selections of the data to be visualized in different colour palettes so that users can compare e.g. different time ranges.

4. Prototype

We have implemented a working prototype hosted at: https://com-480-data-visualization.github.io/com-480-team-kraska/. We use a geojson file with all the necessary data which we read and build the map points from in JS.

To debug the website locally run npm start. To push the website update run npm deploy. To change the .geojson file based on which the map is built check out the tail of processing/flight_data_exploration.ipynb. To change the map generation code check out website/src/MapView.js.