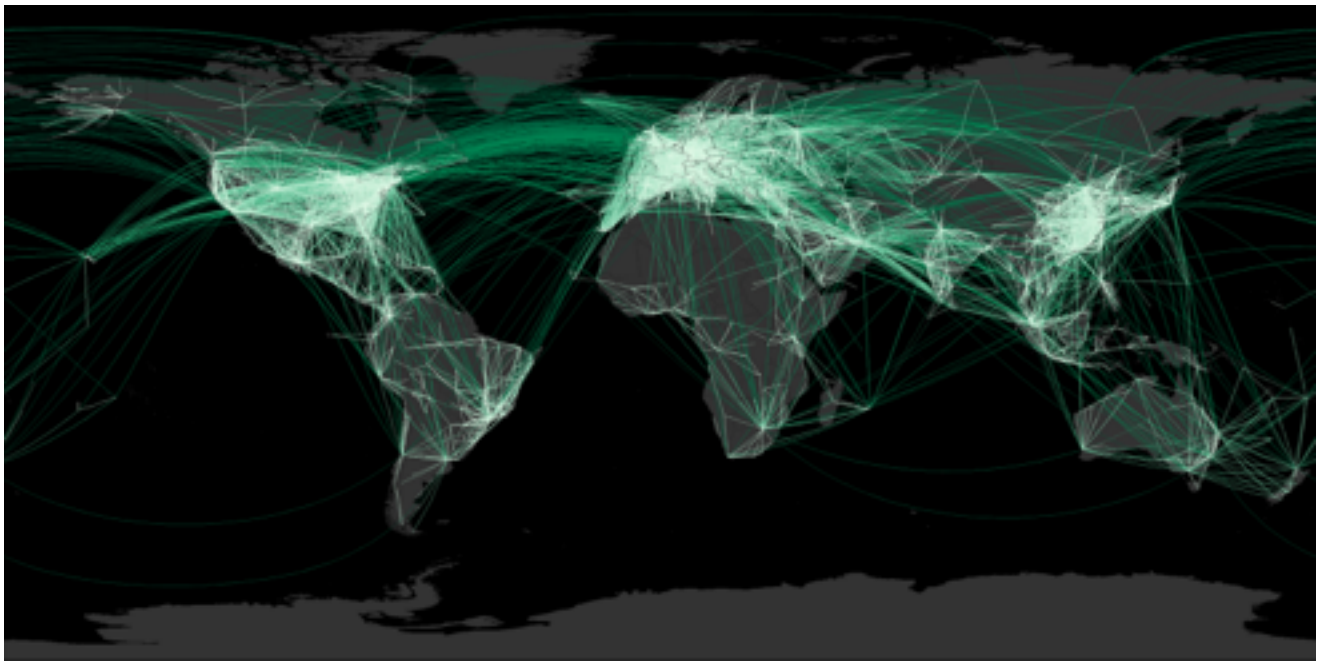


source: Visualizing geodetic information with JEQL, http://lin-ear-th-inking.blogspot.com/

# Process Book

Prepared for: CS171 Final Project

Prepared by: Masahiro Kusunoki, Nhu Nguyen, Zeno Ziemke

GitHub Repository: https://github.com/nhunhu9/cs171-pr-flightdata

April 17, 2015

# OVERVIEW

## Background and Motivation

In 2010, the number of commercial departures topped 30 million annually. This number calculates to over 80,000 departures per day, and with the number ever increasing the figure is expected to have reached double that amount by 2030[1]. This is an incredibly resourceful dataset considering that each one of these flights have been carefully recorded and documented. Hence we felt that there would be great potential in visualizing this data.

Trends in the frequencies of commercial flights are also linked to other important factors such as economic standing, cultural attraction or political affinity of two countries. They may also be linked to specific events (such as the FIFA World Cup or the Olympic Games) that would bring new trends across particular a time span. We would be interested in seeing whether we can find other factors that correlate with the phenomenon of global transportation and if so, whether we will be able to predict new trends off of different variables in the future.

However, perhaps most importantly, what drew our attention to this topic the most is the fact that we are all interested in traveling and flying and would like to learn more about how global transportation has evolved to look like what we currently experience today.

## Primary Objectives

The primary objective of this project is to visualize how different cities in the world have been connected over time. What have been the global trends for the establishment of flight routes and which routes have been the most (or least) popular? To answer these questions, we will plot the data of all the commercial flight routes from “OpenFlights” database, and map these routes onto a map of the globe. If we add a feature for filtering depending on the year, then this can show us over time how new flight routes have been established (or terminated) and how the world has become more connected through commercial flights over time.

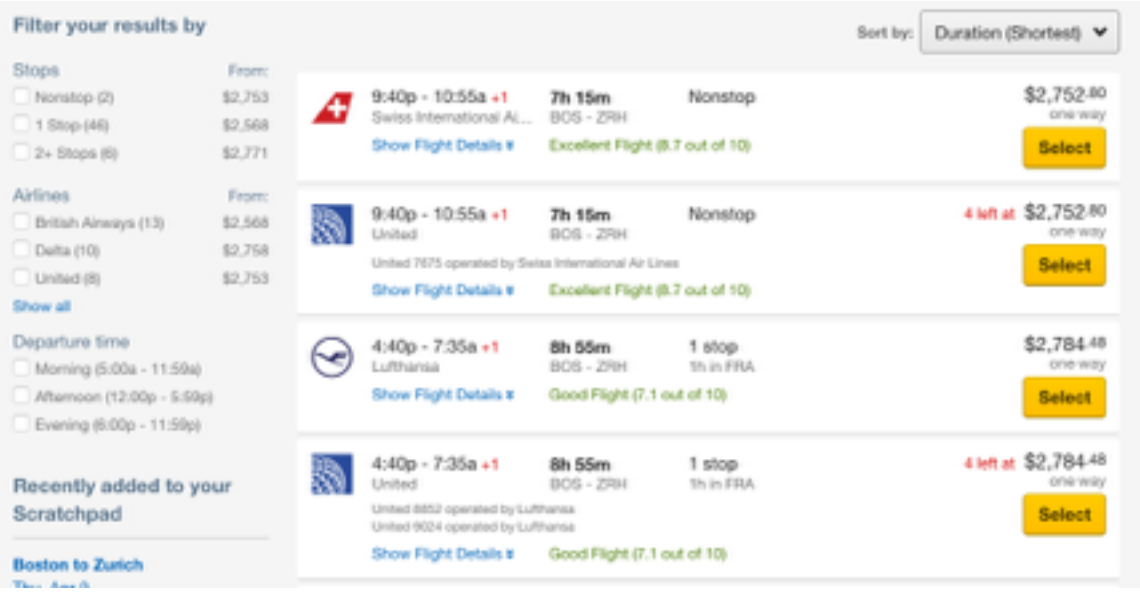
Other questions we may consider are:

- Are there any correlations with other variables (e.g. GDP, population)?
- What are the prices of the different routes at different times of the year?

The first question is beneficial because it will allow us to understand the phenomenon of global transportation fly in more depth. We can imagine with relative ease that the countries’ population would be correlated to the number of routes that exist to the country, but perhaps there may be exceptions to this rule and if so, we can explore other variables that explain this.

## FLIGHT VISUALIZATION

The second question is beneficial from a more practical standpoint. When we try to search online for our own flights to book, currently, many of the search websites provide a very static and stale looking response. These search results could be improved greatly if we did not simply represent each option independently but rather compared these different options to each other adding in visualization for factors such as geographic locations of transits, times of departure and daytime/nighttime analysis. This would be an optional feature for the project, but hopefully we can integrate Google's flight search API to return some flight price comparisons integrated into the visualization.




source: Search results from http://www.expedia.com/


source: Search results from https://www.hipmunk.com/

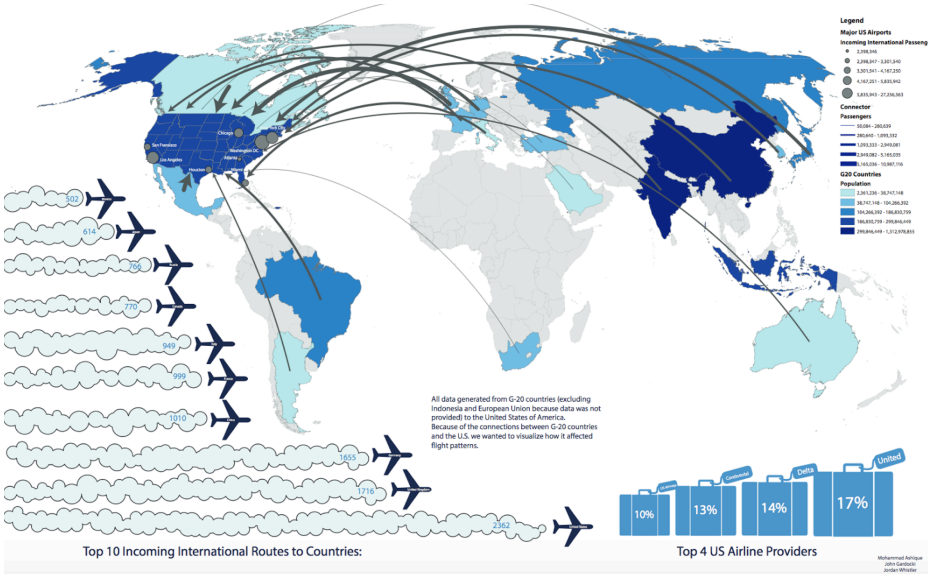
## Related Words

Our inspiration has come from various sources. Many people have in the past created visualizations that map the routes for flight data. However, most of these are static visualizations that do not allow for the exploration by the user. The following are two examples:

The first is a beautiful rendering of the different routes provided by major airline companies in the world. However, due to its aesthetic nature, the features for further exploration it seems have been sacrificed.


source: Visualization Analysis and Resource,
http://spatialanalysis.co.uk/2012/06/mapping-worlds-biggest-airlines/

The second is a colorful and well designed implementation that maps flights used by passengers traveling to the united states. A color code has been used to tell a story about where people come from and what the trends are. Here, the ideas are very good and the result is clear and well designed. However, this can only tell a very specific story and does not give the user any tools to explore the data further.




source: International Passengers Traveling to the United States of America,
http://www.visualizing.org/visualizations/international-passengers-traveling-united-states-america

# DATA SOURCES

## OpenFlights Database

Link: http://openflights.org/data.html

This database is in csv format and contains hand collected information about flight routes, airports and airlines. It contains 58000 flight routes with the following fields:

> airport departure | city departure | country departure | long. departure (decimal) | lat. departure (decimal) | airport arrival | city arrival | long. arrival (decimal) | lat. arrival (decimal) | airline name | airline country based | route ID | number of stops | distances (km.) | domestic

It also contains information about the passenger volumes between cities (3000 routes) with the following fields:

> departure city name | arriving city name | number passengers 1990 | number passengers 1995 | number passengers 2000 | number passengers 2005 | number passengers 2010

## General Economic Data

Link: http://data.worldbank.org/

This dataset contains detailed economic data like GDP, Import/Export, Currency Rates, etc. for all major countries of the world. The data is available on yearly and quarterly basis.

## Airplane Crashes and Fatalities Since 1908

Link: https://opendata.socrata.com/Government/Airplane-Crashes-and-Fatalities-Since-1908/q2te-8cvq

This csv database contains airplane fatalities from 1908 to 2009 with the following fields (~5000 incidents):

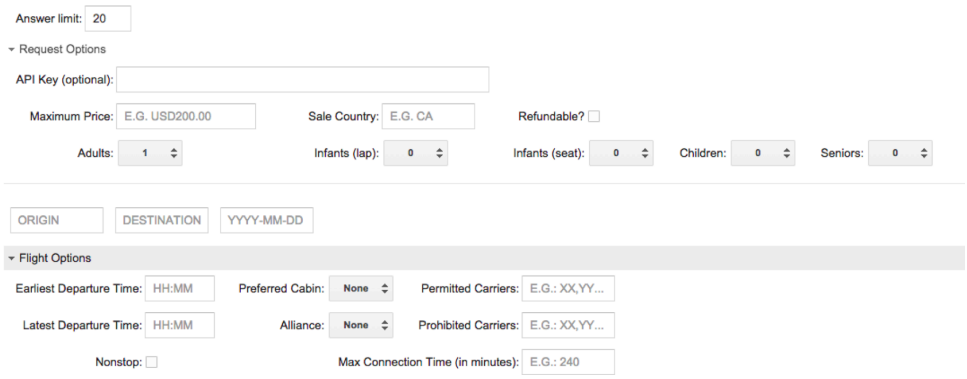
> departure airport, departure city, departure country, long. departure (decimal), lat. departure (decimal), arrival airport, arrival city, arrival country, long. departure (decimal), lat. departure (decimal), airline name, airline country, active airline, distance between airports (km),domestic flight

## FLIGHT VISUALIZATION

### Flight Fares from QPX Express API/ (50 free requests/day)

Link: https://developers.google.com/qpx-express

This Google API provides flight fares. The JSON based API offers the following options for searching flight rates:




It returns routes including up-to-date fare information in JSON format. The API is restricted to 50 free requests/day. Additional requests costs $0.035 US per request. We expect that the 50 free requests/day will be sufficient so that at least 1-2 users can try out our visualization on the demo day.

# DATA PROCESSING

### General

Most of our information is available in clean CSV formats. We have done some basic format conversions. For example converting date formats or number formats for all datasets.

| city departure | city arrival | number passengers 1990 |
|---|---|---|
| Denver,CO | London,United Kingdom | 44807 |
| Denver,CO | Manchester,United Kingdom | 1685 |
| Denver,CO | Melbourne,Australia | 178 |
| Denver,CO | Mexico City,Mexico | 10977 |
| Denver,CO | Mazatlan,Mexico | 13715 |
| Denver,CO | Tokyo,Japan | 377 |
| Denver,CO | Puerto Vallarta,Mexico | 14729 |
| Denver,CO | San Jose del Cabo,Mexico | 4990 |
| Denver,CO | Sydney,Australia | 56 |

However, in the case of the dataset that included the number of passengers across time, more effort was required. This was because, only the cities were included in the locational data and the countries were tied into the same datapoint. We needed to split up the data at the comma value, and if this was a city in the US, since the data only consisted of the state abbreviation, we appended US.

## FLIGHT VISUALIZATION

### Flight Search with Google API

To use the flight fare API we will need to integrate the API calls into our application.
Task TODO

### Airplane Crashes and Fatalities

The Airplane Crashes and Fatalities will require additional cleanup:
One issue was that this dataset does not provide geolocation data for the crash locations. If we have time we will try to automatically get the longitude/latitude for the locations of the crashes (they are saved in text format in the data set, for example: “Fort Myer, Virginia”) using the Google Maps API. This would allow us to plot the crashes on a world map.
Furthermore this dataset contains incidents for all kind of incidents including military plane incidents. Thus we will needed to filter out the commercial airline incidents.
Task TODO

# DESIGN EVOLUTION

## Flight Map

D3 library to create map. This will be the center of the visualization. The library used was Mike Rostock's topojson library. Thiis an extension of GeoJSON that encodes topology. Rather than representing geometries discretely, geometries in TopoJSON files are stitched together from shared line segments called arcs. We chose this because this has implemented many of the features for mapping arcs onto the globe, that we wished to utilize.

Currently, this is a simple representation of the map, with no extra features such as mouse hover identification or zoom functioning.

## FLIGHT VISUALIZATION

We then went on to map the city to city connections. Below is a screenshot of one of the earlier attempts.

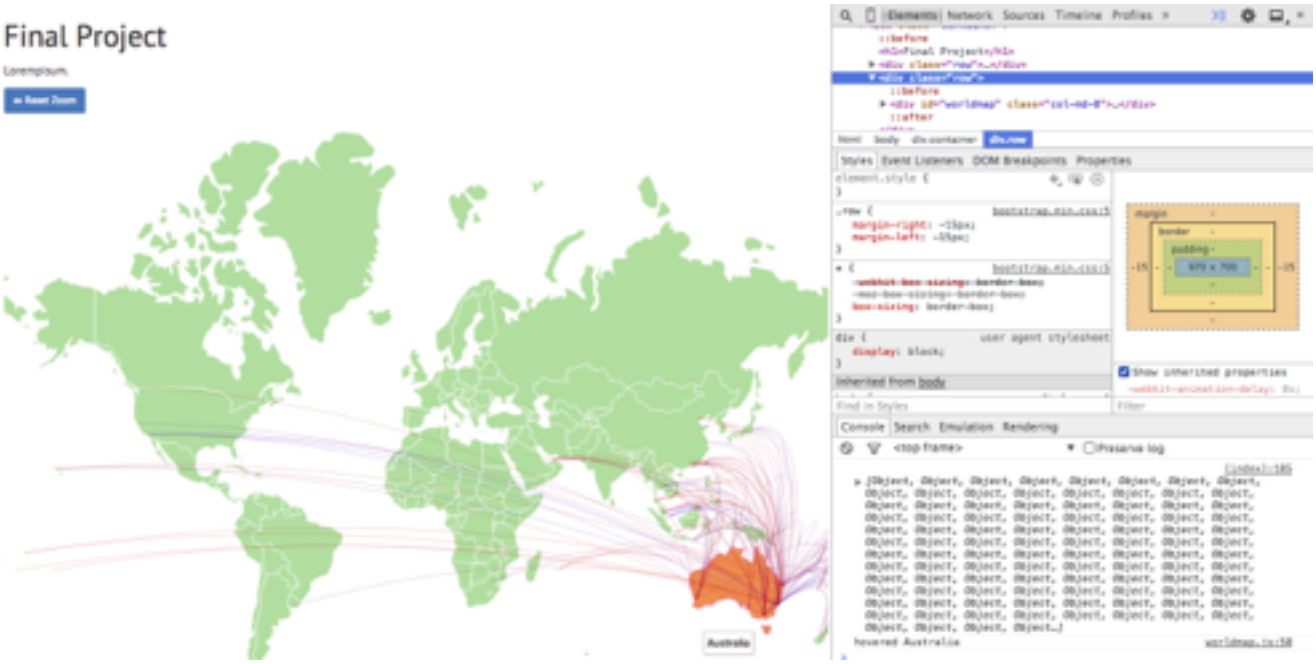
There were some issues. At first, there were far too many lines to visualize at once. The default landing visualization used to show every single flight route in the world all on the same map. However, the page took very long to load and even made the browser crash sometimes. That is why we no longer have a function for visualizing all the flights in the world and require by default a filter argument.

Next was the issue of efficient filtering. Particularly because the database consisted of duplicate city to city connections. For example, the flight from JFK to MUC would also be included in the database as MUC -> JFK and while in principle they may be different routes, we do not need to redundantly map both of these on the visualization.

Finally, we was the issue of congestion, particularly in the small countries. Particularly in Europe were there were many small but well connected countries, the visualization became very congested when we did not allow the user to zoom in. Since there were not preset features for Pan & Zoom in the library, we implemented this ourselves so that the user may now use click, scroll or drag on their mouse to move around the map and see what they are interested in  in more detail. (We used functions from https://github.com/markmarkoh/datamaps/pull/122/files)

Here are examples of the zoom functions.


Our first filter function was implemented using a mouseover argument; when the mouse hovers over a country, we represent all of the flights from that country. Initially we thought of using the mouse click argument. However, we wanted to keep this for different functions (such as searching for flights between two countries or zooming into a certain country to show the domestic flights there). In the console, you can also see how we have logged the hovered country.



## Airline Companies Ranking Table

We decided to have a table where we rank the airlines based on the number of routes they offer. In addition to the map, which is visually engaging but can be confusing, the table is there to help the user see more clearly which airlines are most active in the air traffic. We also want the user to be able to click on the map and see the rankings change in the table (haven't implemented yet). We will also have a search box that can be used to search for airline by names (also haven't implemented).

Implementation:

Using the completeTable.csv file, we modified the data such that we have an array of objects, and each object contains the name of the airline, number of routes it offers, country where the airline is from, and continent. The airlines are always sorted in descending order. However, we can add other sorting methods if they can improve user experience. We created the table shown below:


Filter by: Americas Africa Asia Europe Oceania

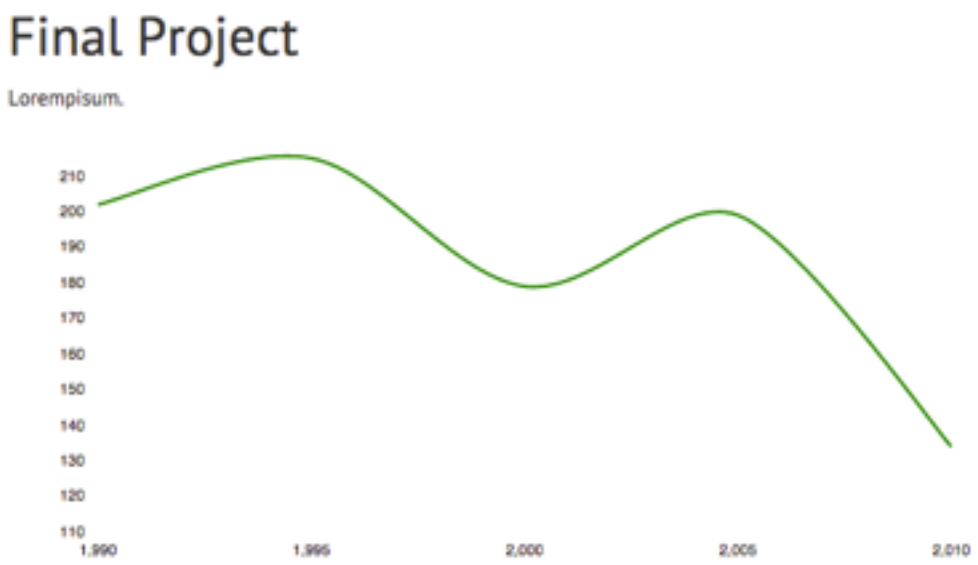
Airline Ranking

| Airline Name | Airline Country | Number of Routes |
|---|---|---|
| United Airlines | United States | 2521 |
| Delta Air Lines | United States | 1975 |
| Continental Airlines | United States | 1946 |
| Ryanair | Ireland | 1479 |
| US Airways | United States | 1463 |
| American Airlines | United States | 1326 |
| Air China | China | 1111 |
| China Southern Airlines | China | 1084 |
| China Eastern Airlines | China | 1039 |
| Lufthansa | Germany | 1009 |
| Southwest Airlines | United States | 936 |
| Alitalia | Italy | 793 |
| Air France | France | 793 |
| Air Canada | Canada | 766 |
| easyJet | United Kingdom | 733 |
| Air Berlin | Germany | 657 |
| KLM Royal Dutch Airlines | Netherlands | 581 |
| TAM Brazilian Airlines | Brazil | 557 |
| Iberia Airlines | Spain | 552 |
| All Nippon Airways | Japan | 533 |
| British Airways | United Kingdom | 485 |
| Shenzhen Airlines | China | 474 |
| Turkish Airlines | Turkey | 469 |
| Alaska Airlines | ALASKA | 464 |
| Hainan Airlines | China | 434 |
| Air India Limited | India | 422 |
| Scandinavian Airlines System | Sweden | 393 |
| WestJet | Canada | 390 |
| Jet Airways | India | 374 |
| Wizz Air | Hungary | 368 |
| Aeroflot Russian Airlines | Russia | 367 |
| City Connexion Airlines | Burundi | 360 |
| Qantas | Australia | 360 |
| Japan Airlines | Japan | 352 |

We also implemented the filter function, where the user can choose to see the ranking of airlines in a specific continent. We will also implement sorting and filtering more interactively to respond to a mouseover or a click event on the main map.

## Passengers over time

We also plan to implement a line chart that displays the transition of the number of passengers over time.



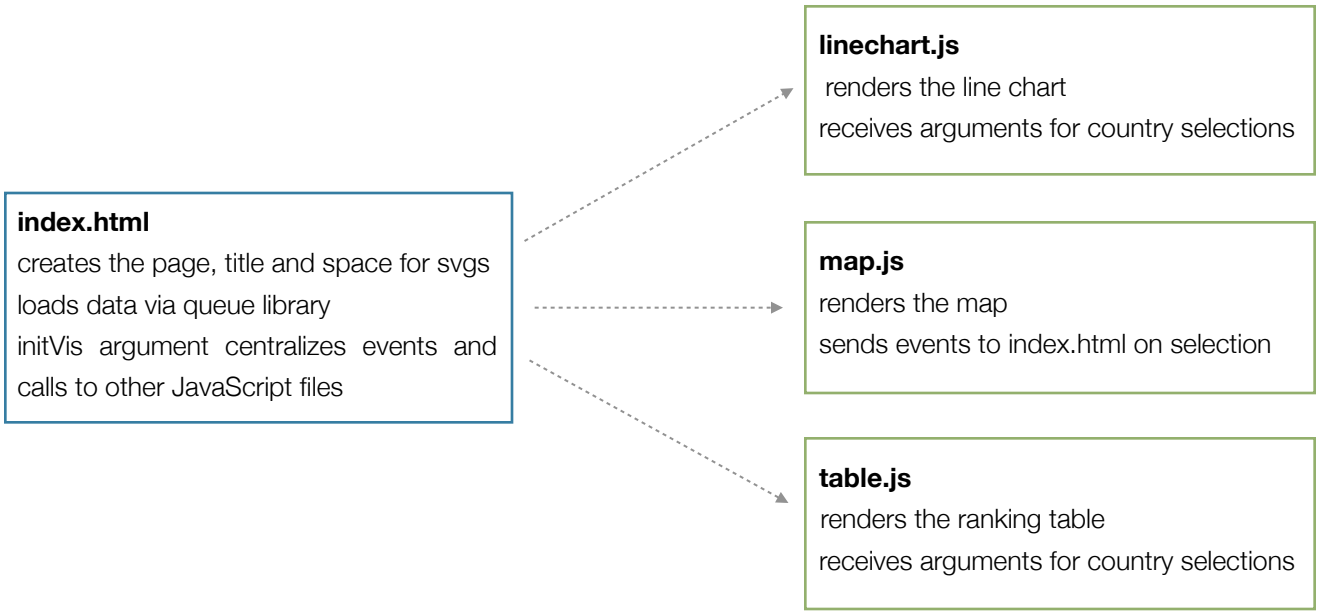

Here is our initial implementation. However, upon trying to add the datapoints, it became clear (as pointed out earlier in the data processing section) that the locational data (the origin and the destination) only showed the cities. This would mean that to even aggregate or filter for countries, we cannot do this because the data does not have these associations.

We are still working on this.

# IMPLEMENTATION

## Structure

The structure of this implementation closely follows that of homework 3 in that we have a central file called index.html which provides a structure for the other visualizations to interact amongst themselves. The page rendering (such as creating titles or spaces for svgs) is done by this file while the javascript files all feed their content into their respective places on the page.




## Interactions

Have not been implemented yet.