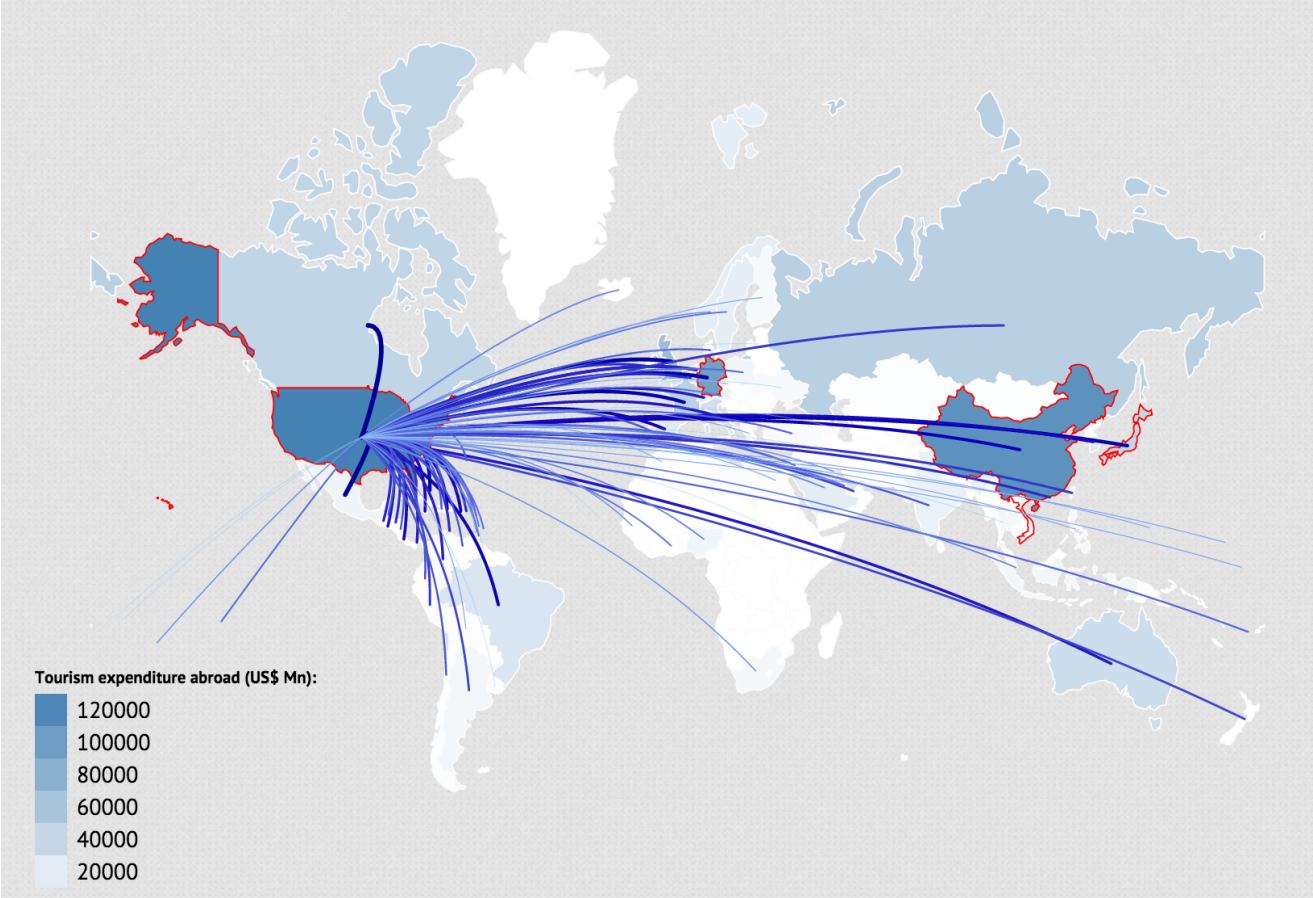


Process Book

Prepared for: CS171 Final Project

Prepared by: Masahiro Kusunoki, Nhu Nguyen, Zeno Ziemke

GitHub Page: https://maza5296.github.io/cs171-project/

Demo Video: https://www.youtube.com/watch?v=ajYWUTMBjZY&feature=youtu.be

May 5, 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 and every single 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 financial crisis or the Olympic Games) that would bring new trends across particular a time span. We were 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, our greatest motivation to visualize this topic was perhaps the fact that all three of us on the team are extremely interested in traveling!

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. We will also add peripheral visualizations that respond to selections on the map to display the data from different perspectives.

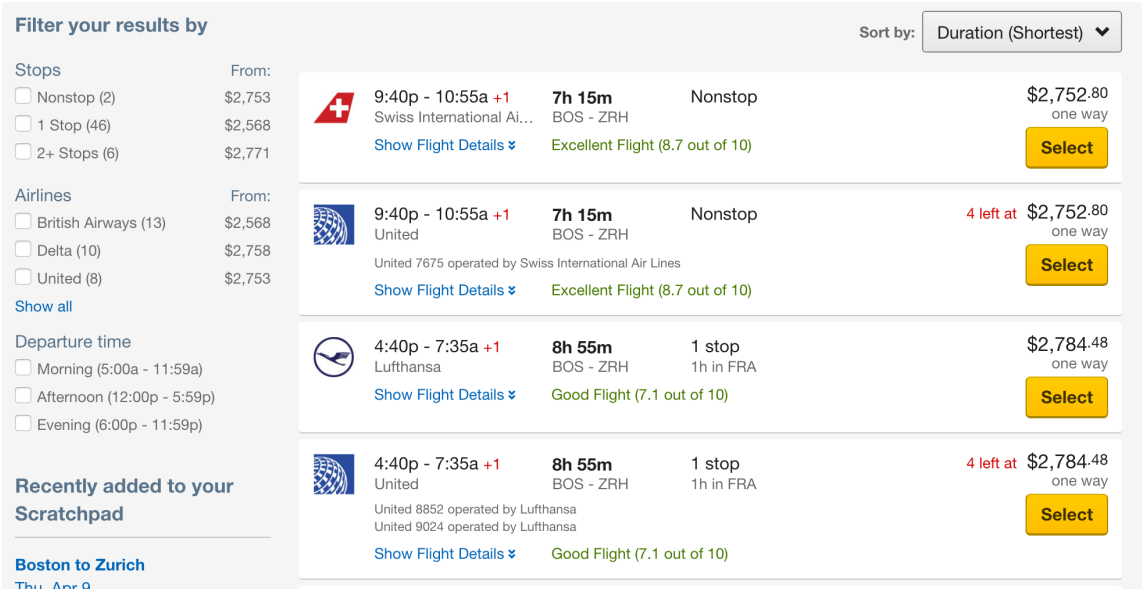
Other questions we consider are:

- Are there any correlations with other variables (e.g. tourism expenditure)?
- Which are the best connected hub airports/airlines in a given continent/country?
- What is the prices of a specific route? How can we visualize the effectively?

The first question is beneficial because it will allow us to understand the phenomenon of global transportation in more depth. We can imagine with relative ease that the countries' who's residents spend large amounts of money abroad will be more likely to have better connected airports. However, will there be exceptions to this rule? If so, can we explain this with a different mechanism? This will give us

The second question allows us to deliberate before we make certain choices of investment. For customers who are considering registering to a certain airline membership for mileage, they may consider the coverage of the airline before they actually make their decision. For example, United and Continental's merger in 2012 means that this company now have hubs in Japan and can offer many non-stop flights from Japan to airports in the US. Other information this gives is is about how well particular cities are connected, an insight we can infer from the best connected airports visualization. This may give tourist agencies information to consider when starting up a regional office for example.

The third question is beneficial from a purely 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/

EXPLORATORY DATA ANALYSIS

Related Words & Insights Gained

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:

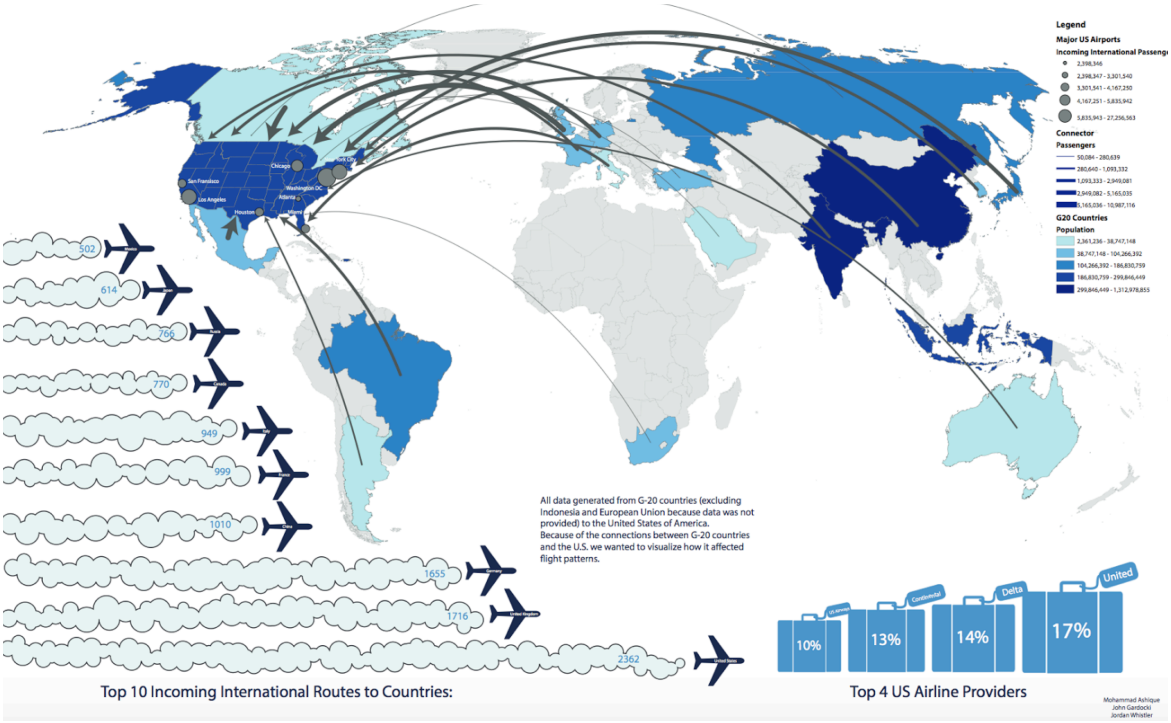
The first is a beautiful rendering of the different routes provided by major airline companies in the world. The connections are of different color and the more condensed they are, the more flights routes that are being offered. However, due to its aesthetic nature, the features for further exploration it seems have been sacrificed. This visualization does not enable us to look deeper into each specific route or choose your own route to examine. To overcome this, we created an airline route ranking table, that can be filtered, sorted and displayed. By making this feature interact with the map, we have given the user more control on exploring the visualizations further.

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

FLIGHT VISUALIZATION

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. We were particularly fond of the heat map feature in which we could visualize different explanatory variable in one map. In the example below, the different colors of the countries represent the different populations. The darker the country, the higher the population. This visualization enables us to understand why certain countries have more flights and how the two variables are correlated.

Our goal was to create a similar informative visualization, but allow the user to select which variables to map and explore for him or herself much further. For example, this only shows the flights to US, but the user may be interested in other countries too. Also, the only heat map that this can create is for population. However, theoretically, using the same visualization but just by changing the variables you pass, you can give the user great control over what they see and thus understand.



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

UN World Tourism Data

Link: http://data.un.org/DocumentData.aspx?q=arrivals+of+non+resident&id=371

This database is titled "Arrivals of non resident tourists/visitors, departures and tourism expenditure in the country and in other countries" and is in an excel format. The two most important categories for our purposes are number of passengers arriving and tourism expenditure for all the countries from the year 2005 to 2014.

> Inbound tourism | Outbound tourism | Tourism expenditure in the country | Tourism expenditure in other countries

CountryList.net

Link: http://countrylist.net/en/

This database includes all of the country names, their country codes and the continents that they belong to. This database was necessary since we needed to be able to effectively connect data points from different databases together to create an effective interactive visualization.

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.

Problems with OpenFlights Database

Whilst forging ahead with the visualizations, it became apparent that the city to city connections of the OpenFlights databases contained only the connections that either originated of or arrived in a city in the US. This description was not included in the description of the database and so we did not realize this issue until we attempted a visualization for Top Airports in the bar chart.

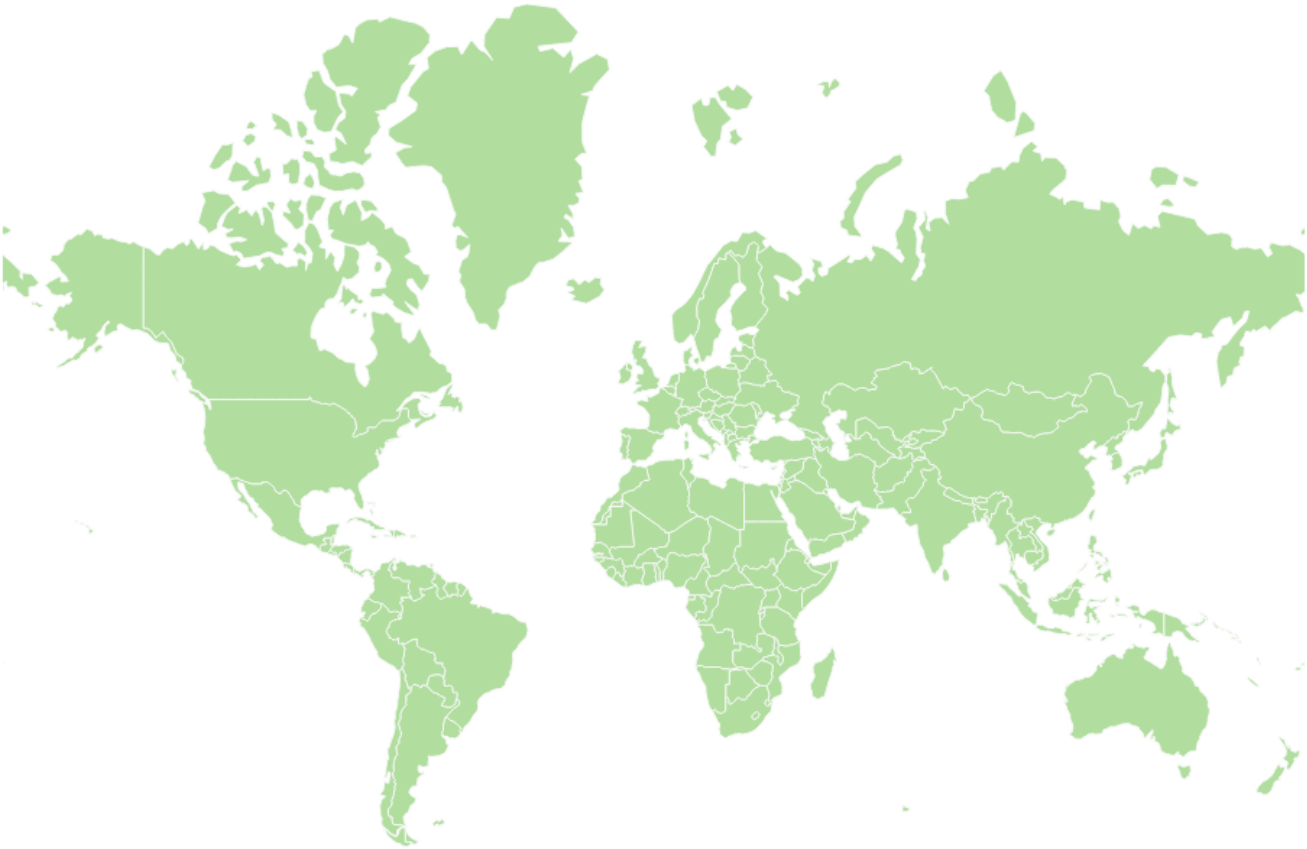
COUNTRY		Series	1995	1996	1997	1998
AFGHANISTAN						
Inbound tourism						
Arrivals - Thousands			..	..	..	..
Tourism expenditure in the country - US$ Mn		IMF	..	..	..	..
Travel - US$ Mn		IMF	..	..	..	..
Passenger transport - US$ Mn		IMF	..	..	..	..
Outbound tourism						
Departures - Thousands			..	..	..	..
Tourism expenditure in other countries - US$ Mn		IMF	..	..	..	..
Travel - US$ Mn		IMF	..	..	..	..
Passenger transport - US$ Mn		IMF	..	..	..	..
ALBANIA						
Inbound tourism						
Arrivals - Thousands	*1/2/*	VF	304	287	119	184
Arrivals - Thousands	*1/*	TF	..	..	..	..
Arrivals - Thousands	*3/*	THS	41	64	23	22
Tourism expenditure in the country - US$ Mn		IMF	**70**	**94**	**34**	**60**
Travel - US$ Mn		IMF	65	77	27	54
Passenger transport - US$ Mn		IMF	5	17	7	6
Outbound tourism						
Departures - Thousands			12	16	..	..
Tourism expenditure in other countries - US$ Mn		IMF	**19**	**25**	**13**	**22**
Travel - US$ Mn		IMF	7	12	5	5
Passenger transport - US$ Mn		IMF	12	13	8	17

At this point, we switched database to the UN World Tourism Data. However, this database was difficult to handle since there were many inconsistencies such as the number of columns per country, the units of the values etc). This required a lot of cleaning up, parsing the data using a python code, which are awesome TF helped us to with ;)

DESIGN EVOLUTION

Flight Map

We used a 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.



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

FLIGHT VISUALIZATION

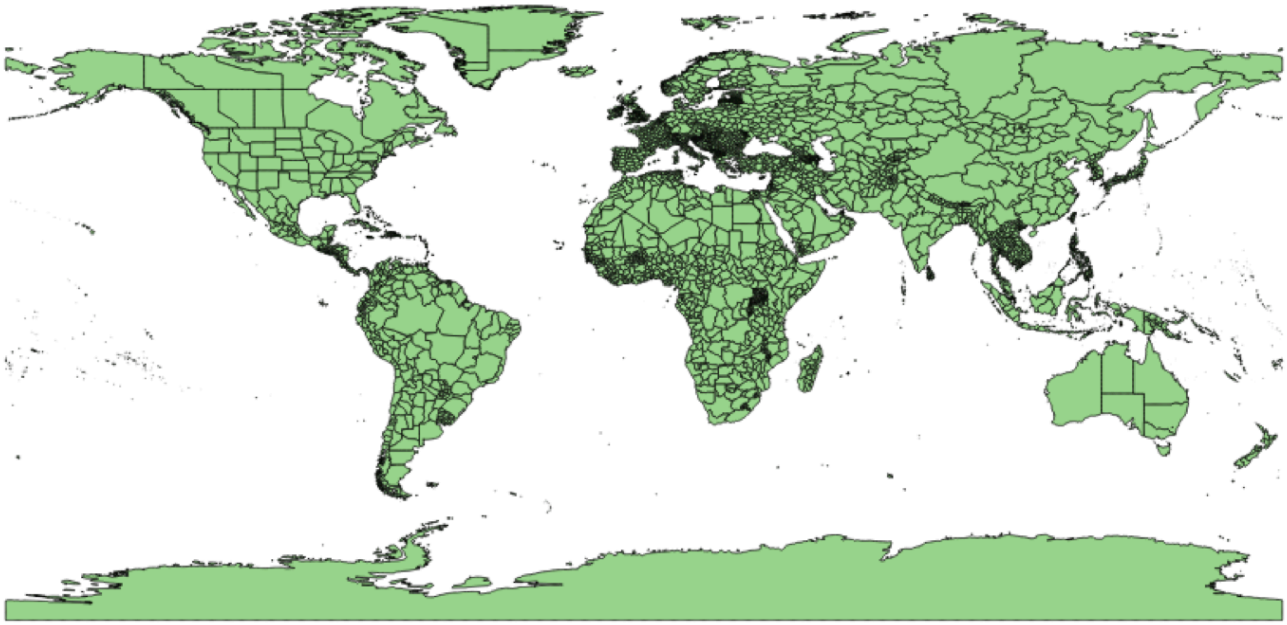
We then went on to map the city to city connections. Below is a screenshot of one of the earlier attempts.

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.

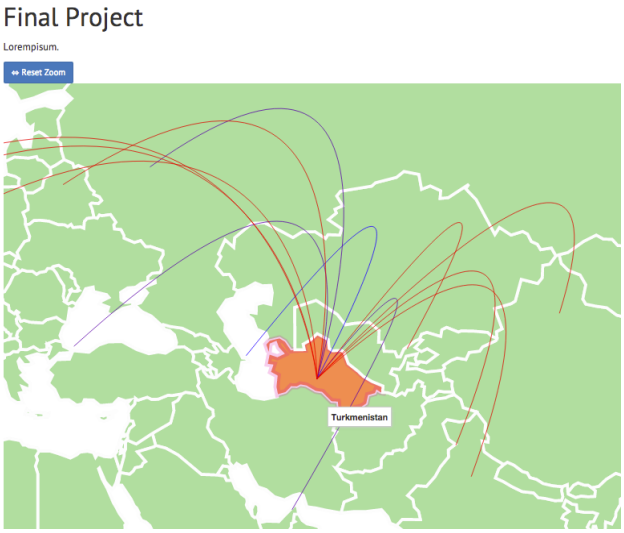
Next was the issue of efficient filtering. Particularly because the database consisted of duplicate city to city connections. For example, the flight from JFK to MUC would also be included in the database as MUC -> JFK and while in principle they may be different routes, we do not need to redundantly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

FLIGHT VISUALIZATION

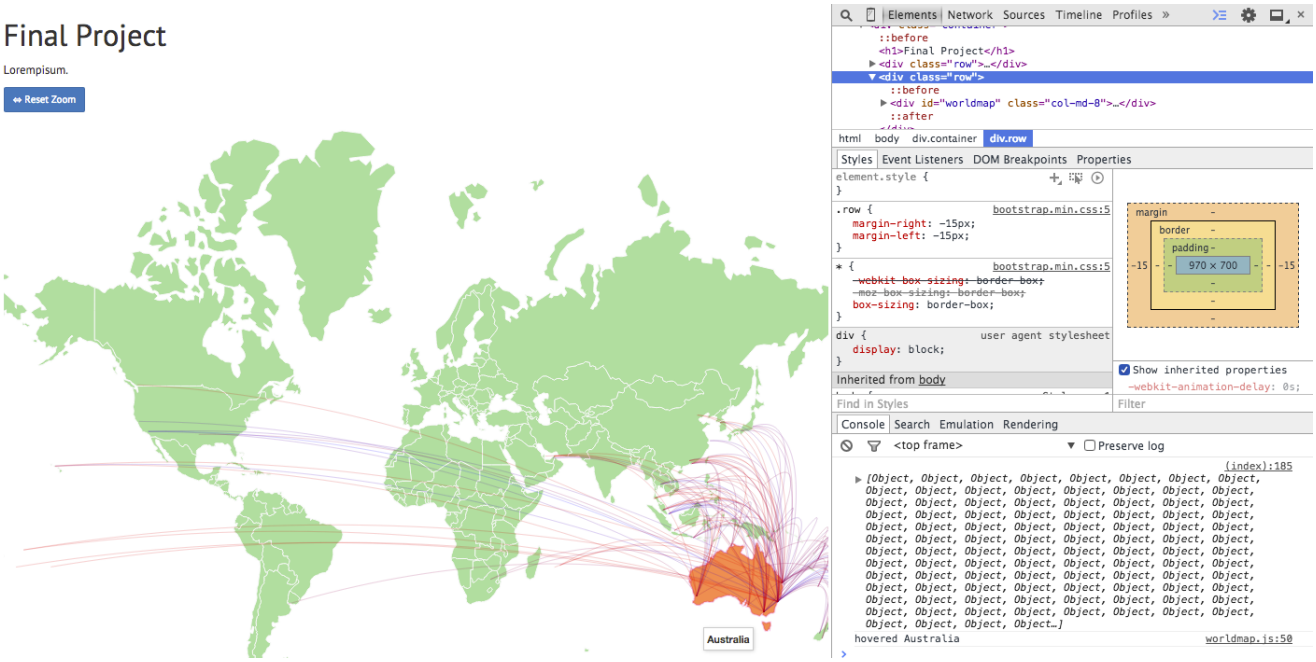


Here are examples of the zoom functions.



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.

We then generated custom maps from this datasource:

http://www.naturalearthdata.com/downloads/10m-cultural-vectors/10m-admin-1-states-provinces/

This contains information for the individual state names and boundaries within a specific country. To apply these to the visualization, we extracted the individuals countries with command line tools and then converted to .shp format so that TopoJSON could interpret it. Here are the steps that we used:

CUstom Maps:
http://www.naturalearthdata.com/downloads/10m-cultural-vectors/10m-admin-1-states-provinces/

We start with a shp file that contains a map segmeneted by provinces/states from :
http://www.naturalearthdata.com/downloads/10m-cultural-vectors/10m-admin-1-states-provinces/

The following commands extract a specific country from the dataset and converts it into json format:cd data

```
custom_maps/ne_10m_admin_1_states_provincesogr2ogr -f
GeoJSON -where "ADM0_A3 IN ('VNM')" VNM.json
ne_10m_admin_1_states_provinces.shp
```

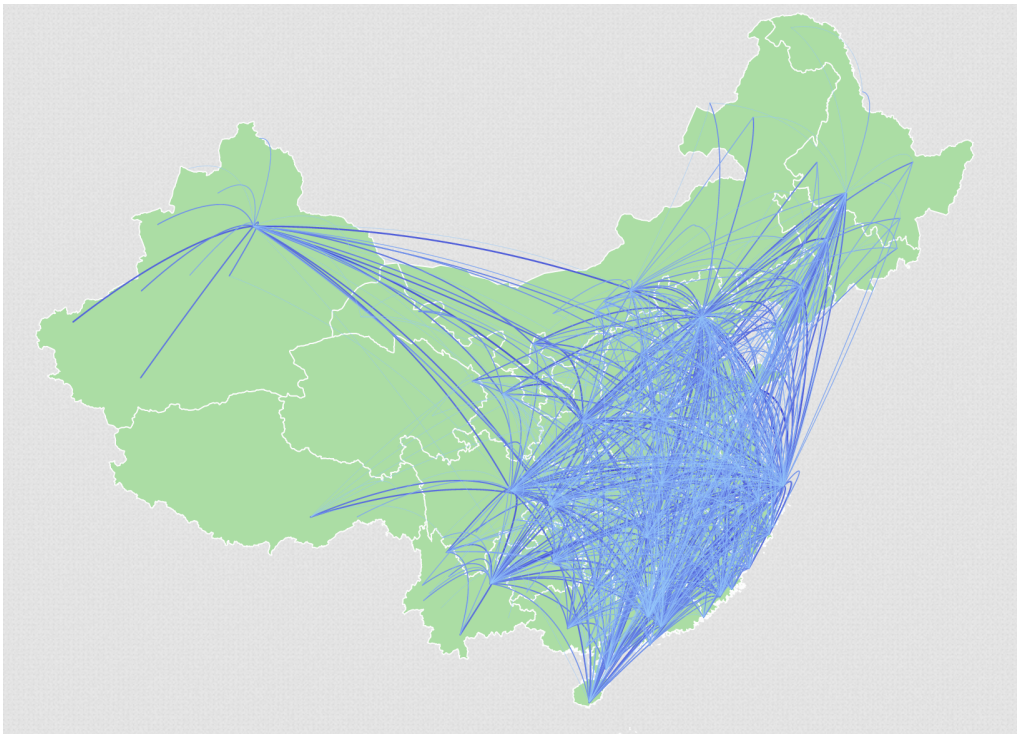
The next command converts the file into topo.json a compressed format:

```
topojson --id-property adm1_code -p name=NAME -p name -o CHN.topo.json CHN.json
```

We then insert this into datamaps.all.js, make country code lower case
add country code to selectableCountries[] in worldmap.js
add boiler palte code for projection in datamaps
pan and zoom to set country (zoom out first)
Click Route Visualiation title to get projection values and add to projection part
Repeat last step if necessary

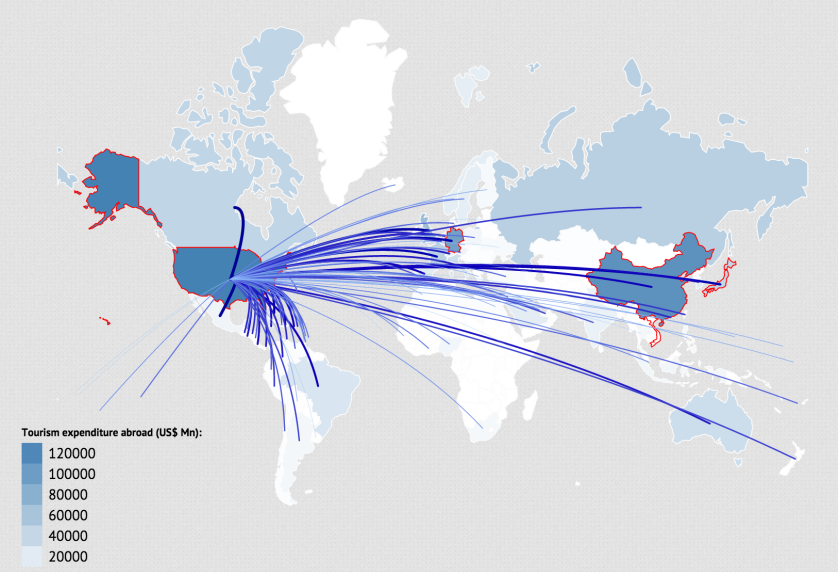
```
} else if ( options.scope === 'chn' ) {
    projection = d3.geo[options.projection]()
    .center([66.166667, 34.4444])
    .scale(2430*0.2895732015486609)
    .translate([width / 2-164-313,height / 2+280-235]);
```

Using this process, we are able to create visualizations for flights within each country and also map on the boundaries and the names of the states/provinces/prefectures. This process cannot be automated; thus given more time, we would have created more visualizations for all of the countries. At this point, we have implemented 5.

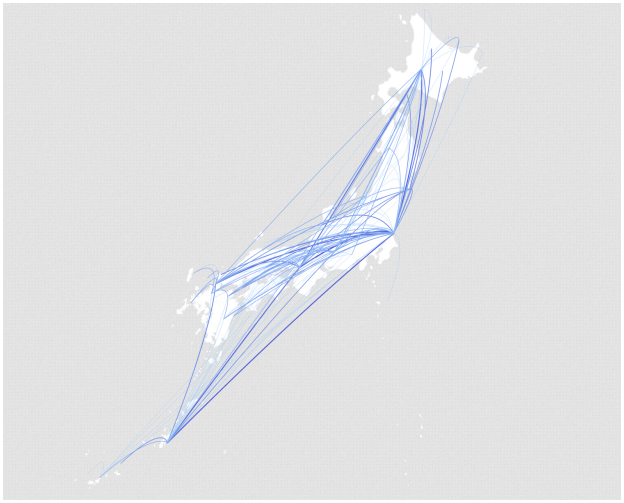


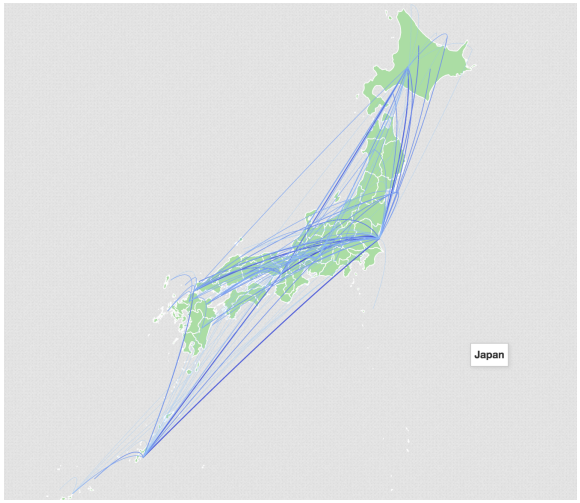
Heat-map

Creating the heat map was the final process of creating the map. All of the data had been sorted for the line chart (dataset was the UN World Tourism). Thus only implementation required was to create a color scale in D3 and then map all of the different countries onto this. The D3 legend was also taken from an online example and so the process was relatively smooth. source: http://bl.ocks.org/mbostock/3202354



The only problem we encountered was that after we implemented this, the individual country visualization became counter intuitive. This is because, a color map only makes sense in relation to the other colors of the other countries. However, once we zoom in, this relationship is no longer apparent. Furthermore, the color makes it very difficult to see the individual states/prefecture borders that we took time to implement. Hence, we reverted the color for individual country visualization back to green.





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:

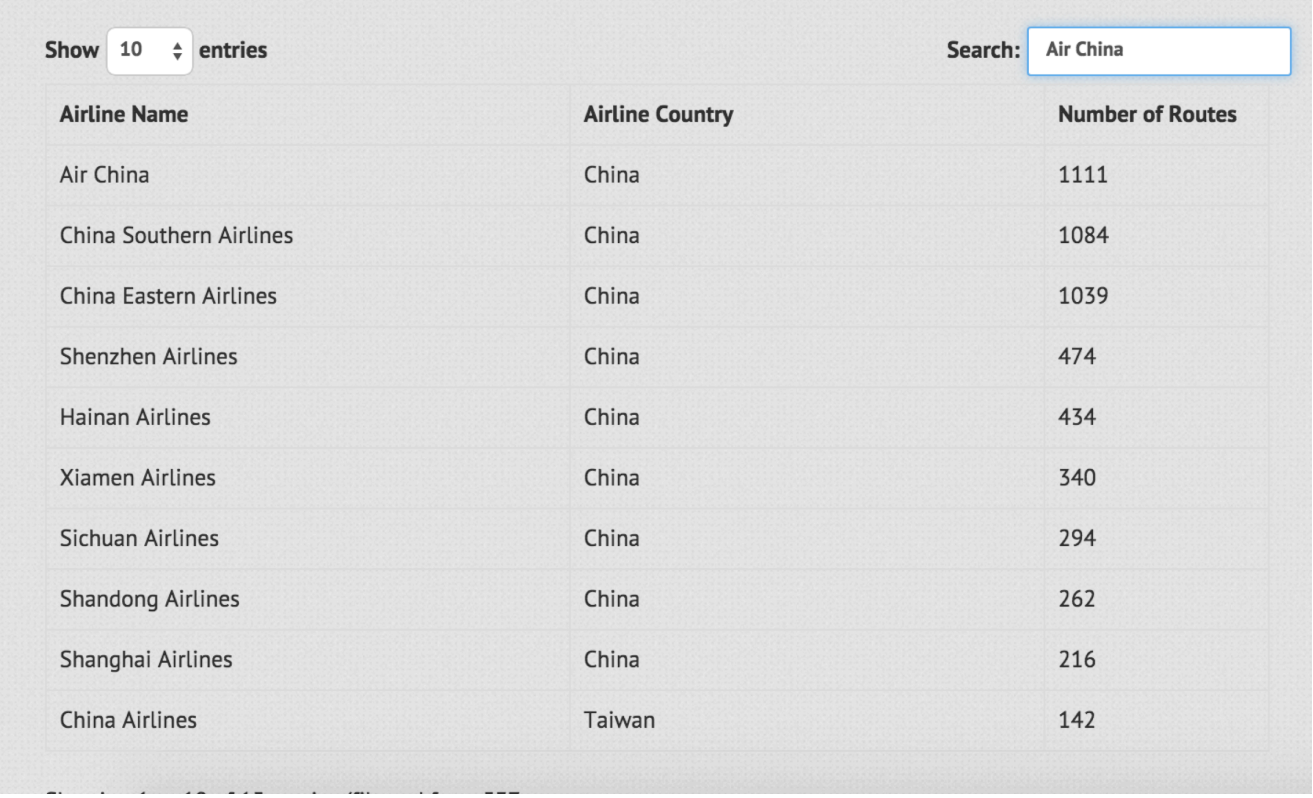
Using the completeTable.csv file, we modified the data such that we have an array of objects, and each object contains the name of the airline, number of routes it offers, country where the airline is from, and continent. The airlines are always sorted in descending order. However, we can add other sorting 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 then used the boostrap.js library to help us add features for limiting the size of the visualization. You can now control how many entries are displayed on each page and you can use the buttons at the bottom to scroll through. We were also able to search the database where users can type in the name of the country or the airline name to search for information about that country or airline. The pictures below show search by country name and search by airline name, respectively:

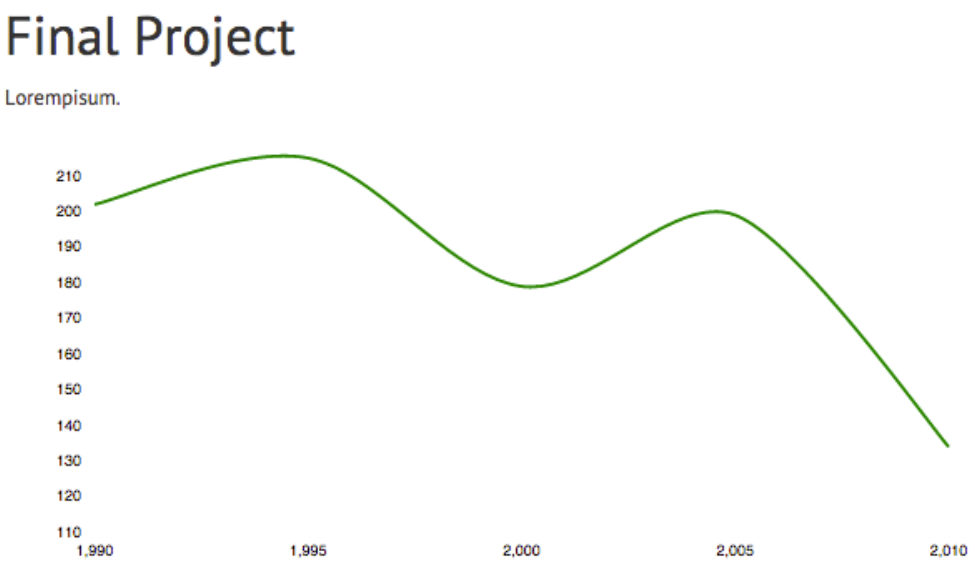


We also use event handlers to update the table when the user click on a clickable country or on a continent name. The event handle passes the updated data to the onSelectionChange function in the table.js file, which update the table with the new data. More details about how this function works is in the video.

Users can also click on the name of any airline to see the airline's network on the map. This feature, in addition to the data shown in the table, will help the users understand better the magnitude and geographical focus of each airline.

Passengers over time

We implemented a line chart that displays the transition of the number of passengers over time.

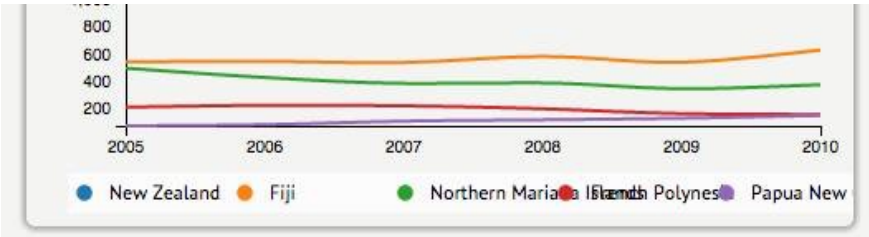


Here is our initial implementation. However, upon trying to add the datapoints, it became clear (as pointed out earlier in the data processing section) that the locational data (the origin and the destination) only showed the cities. This would mean that to even aggregate or filter for countries, we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

We thus created this simple function to map the continents from the country list database. We had to be exceedingly caution that all of the cases, commas and spaces in the country names matched perfectly and so we created many helper functions (in stringHelpers.js) to make this process easier.

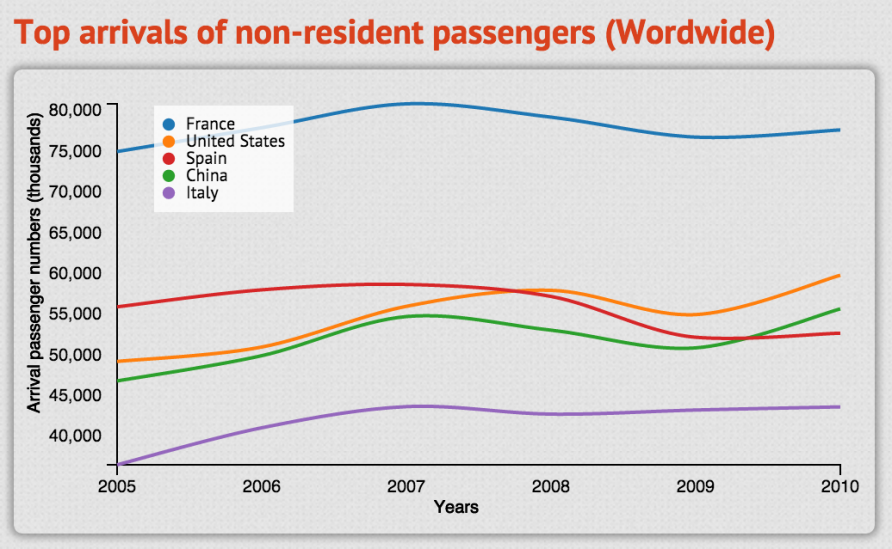
```
UNData = Object.keys(_UNData).map(function(d){
    return {
        name: d.toTitleCase(),
        continent: (countryList[d.toTitleCase()]) ? countryList[d.toTitleCase()] : "Unknown",
        data: _UNData[d]
    }
});
```

```
String.prototype.capitalizeFirstLetter = function() {
    return this.charAt(0).toUpperCase() + this.slice(1);
}

String.prototype.toTitleCase = function() {
    return this.replace(/\w\S*/g, function(txt){return txt.charAt(0).toUpperCase() + txt.substr(1).toLowerCase();});
}

String.prototype.removeComma = function() {
    return(this.replace(/,/g,''));
}
```

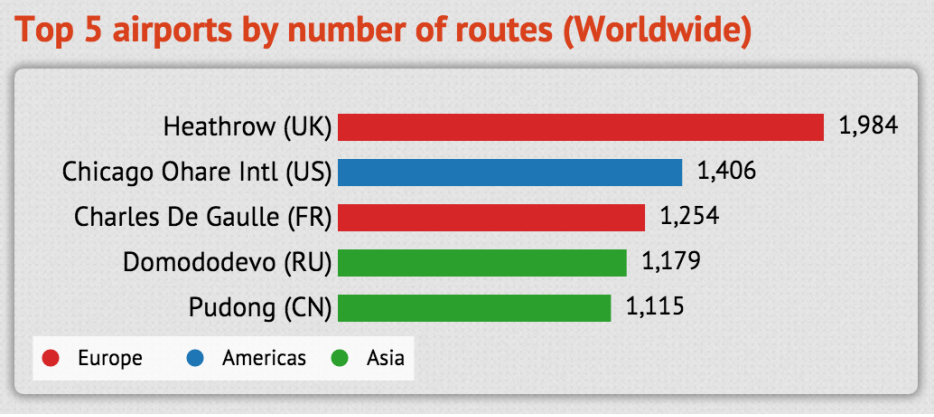


Another problem we encountered was when we were creating the legends. Legends were already required for the bar chart but the format for this one was more difficult because some countries had long names and thus created overlap.

We ended up creating two d3.legend libraries, one for vertical implementation and one for horizontal, with the feature to pass arguments to create specific spacing added to the library. The resulting implementation looks like this.



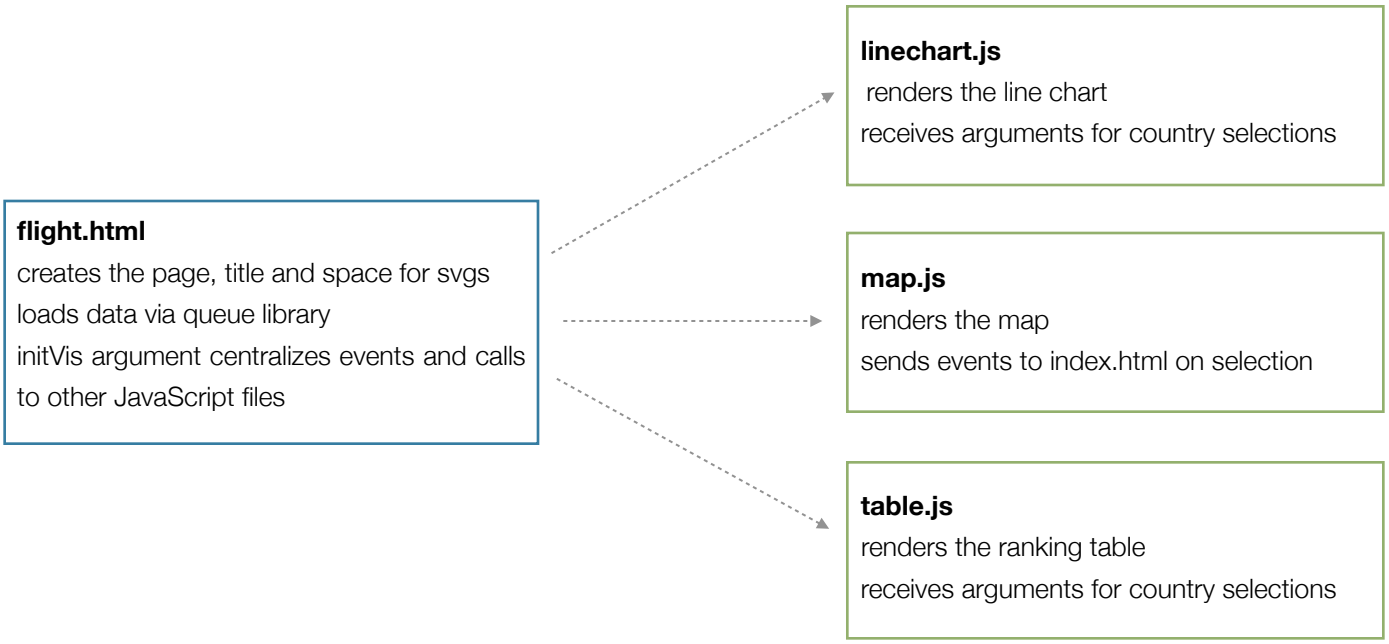
This contrasts with the legend implementation for the barchart, which is horizontal.



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.



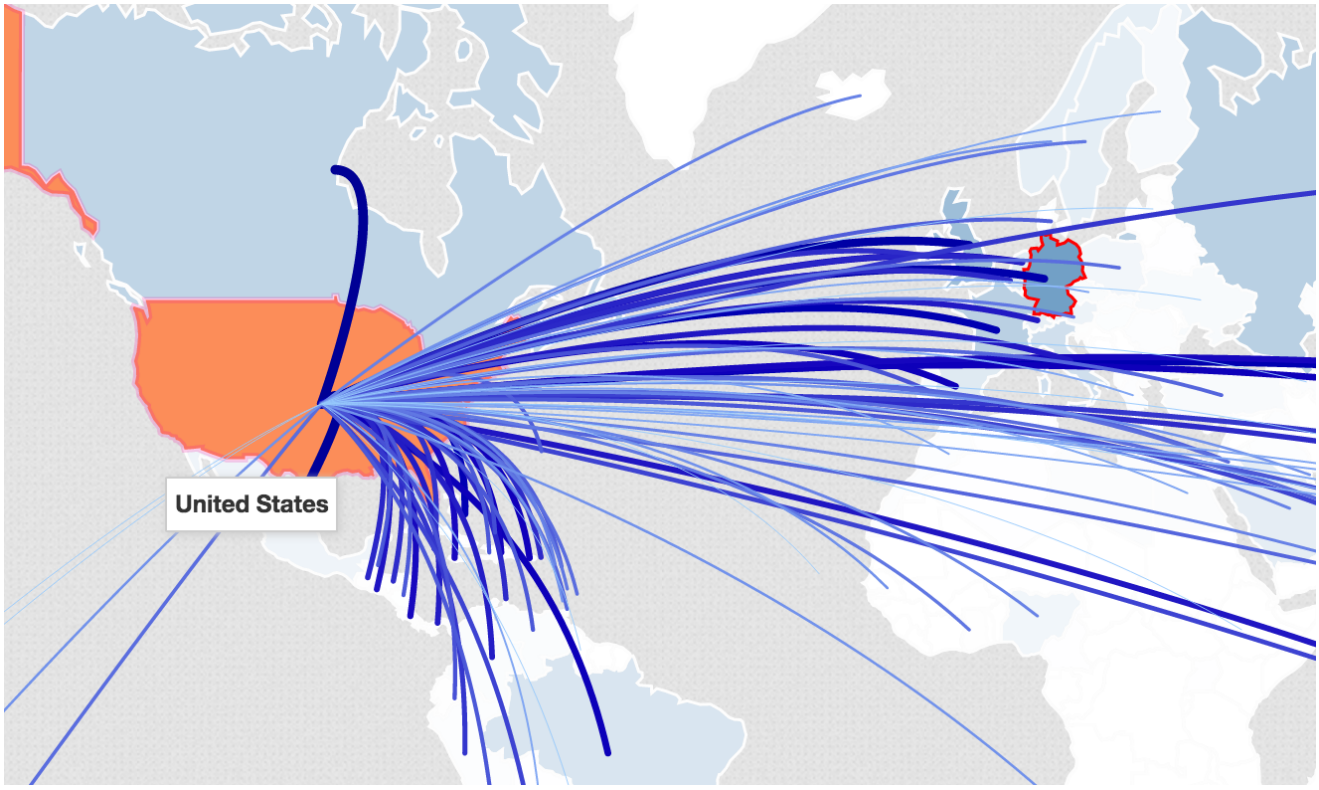
Other files are located in the format specified in the readme.

INTERACTIONS

Map Mouseover

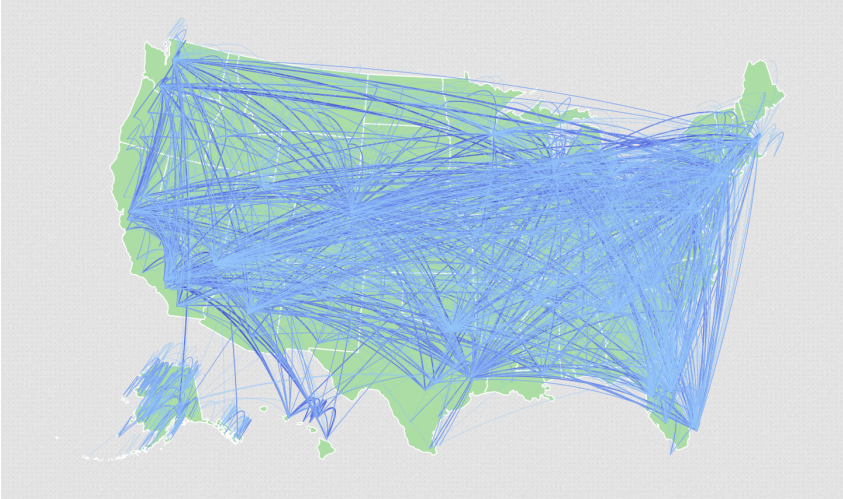
This first and most important interaction is the mouseover feature in the map. This enables the user to view all of the flights routes that are offered to the country. We did not map every single route as a separate line, but rather aggregated them by route. Thus, the lines also communicate the number of flight routes offered. In order to do this, we use two different elements of design: color and weight.

To begin with, we had planned to use two colors, red and blue, to color code the lines. However, since this is not a dataset that has a clear break even point (such as negative loss or positive profit), this did not make sense. Thus, we employed a color scale, in which the lines with more routes will be expressed in darker colors and heavier lines.



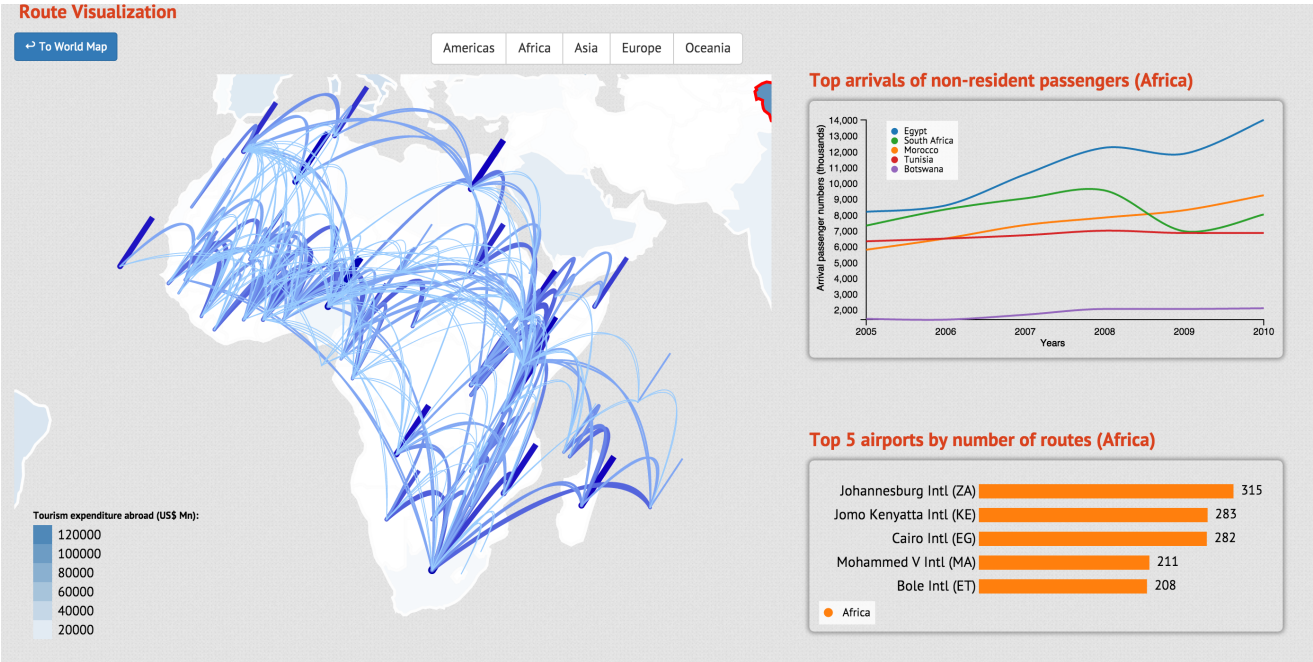
Domestic Flights

The second important interaction is when you click into a specific country. The user may not only be interested in international flights, but also in domestic flights; hence this feature. Also, as explained previously in the implementation of the heatmaps, this map will use the default green color, since the heat map will make no sense if there is no other country to compare it against.



Continent Selection

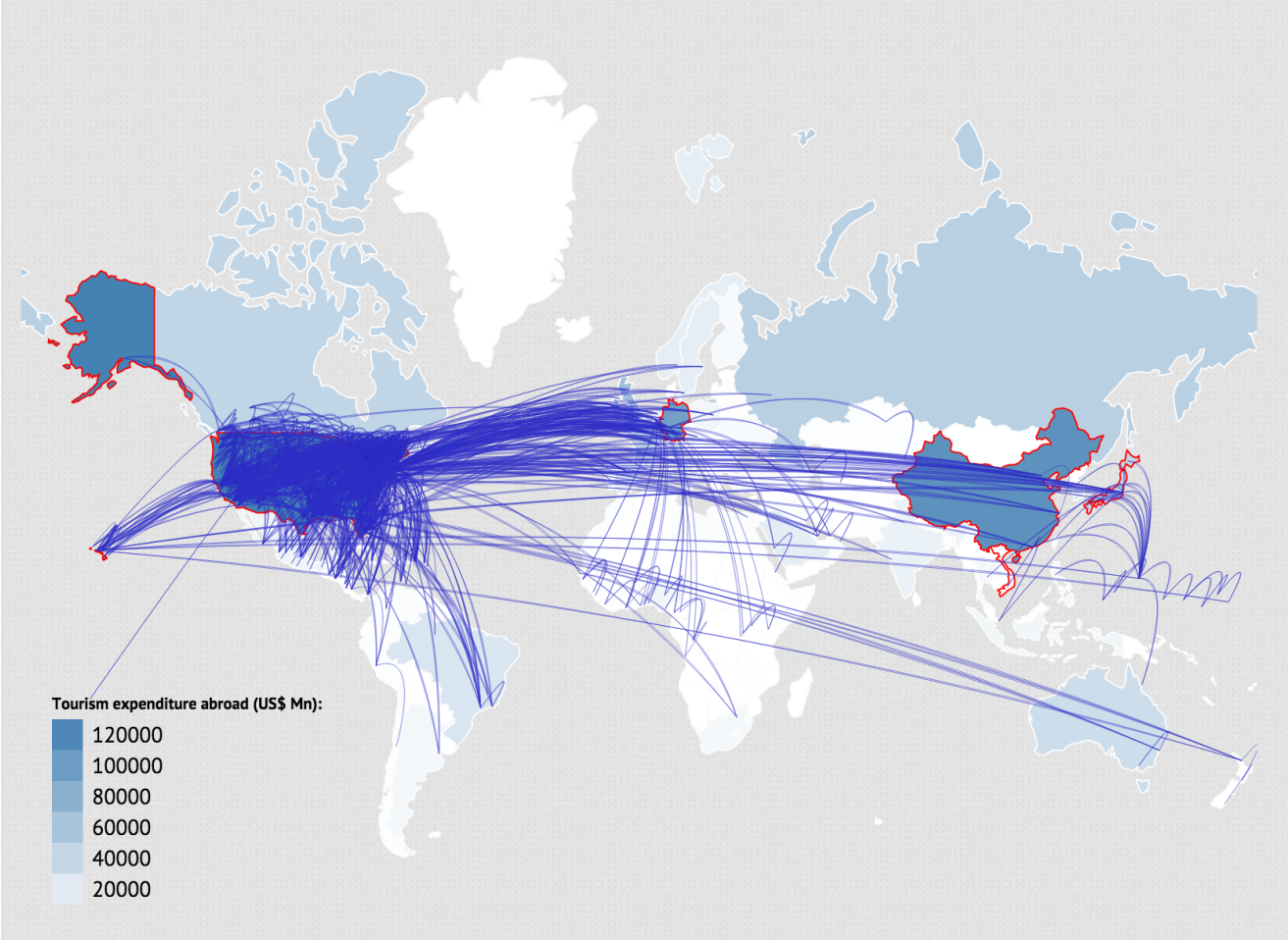
We have also enabled the user to filter all visualizations for a specific continent selection. The connection lines readjust the domains for their thickness and colors to display the relative information in that specific continent.



Specific Airlines

We have also implemented a feature whereby the user can map only routes offered by a specific airline carrier. This visualization enables the user to understand better the coverage of each carrier and where he or she could potentially travel if they are to use their routes.

Here, we are no longer interested in the number of the routes offered as opposed to the coverage of each airline. Thus, the line thicknesses and colors are calibrated to their average value and there is not differentiation for the visualization of the lines.



EVALUATIONS

What did you learn about the data by using your visualizations?

We have created a story feature to tell our most compelling findings.

One of them is to point out the inequalities of the regions within specific countries like China: ”East vs West" seems to be a paradox occurring not only on the global scale, but also even more locally within the China. It has been recently reported that the gap between China’s rich and poor is now one of the world’s highest, surpassing even that in the U.S. Our visualization enables us to also understand the scale of geographic inequalities between the cities, in which the urban east is far better connected than the rural west.

Another is that political relationships between countries are also translated into the flight routes: For example, we found that there are extremely few flights that connect Israel with nearby Arab countries. For the majority of the other countries, the flight connections are very frequent in their neighbors; clearly Israel seems to be an exception to the rule. This data can speak directly to the political relationships between the countries in the Arab world; an extremely interesting find!

To see all of the stories, please refer to the visualization webpage!

How did you answer your questions?

Our initial goal was to understand how different cities in the world have been connected over time. We were certainly able to to answer our questions, sometimes in ways that we did not expect.

One thing to point out is that this visualization does not tell one specific message; rather it is designed to empower the user to explore into the issues/regions that he or she is most concerned about. We have given them the mouseover, filter and search functions (while aggregating, color coding and updating in the background) and we believe that there are sufficient means for the user to achieve this goal.

How could you further improve it?

Given more time, we would have given the user even more tools to fine tune their exploration. For example, we could have given them the options to change the variables on the heat map (to show tourism expenditure within the specific country as opposed to outside of the country) which may tell a very different story.

We would also have created more maps for the individual countries to visualize the domestic flights to broaden the scope and possibilities for the user to explore.

However, we are confident that in the limited time that we had, we have created a visualization that is robust, informative and grants the user the most important tools to understand the data that we have visualized.

THANK YOU FOR READING

HAPPY VISUALIZING ;)

from Zeno, Masa and Nhu