PROCESS BOOK
Global Migration Flow
Bao Chau Pham, Ricardo Gonzalez, Lucas Idstrom

Motivation

Migration, both immigration and emigration, are becoming more and more significant topics especially in the past few years where political figures have made it a top priority. However, we have always found it hard to see exactly how much people are moving around in the world. How many are leaving countries, how many are coming in, and most importantly how has that changed over time? Through some research, we have found data that we can use to visualize those questions and hopefully be able to satisfy our curiosity as well as others.

Objective

The main objective of this project is to easily visualize the immigration patterns of countries around the world. We are curious to see what countries people tend to immigrate to as the years go by. One of the most interesting aspects of this type of visualization would be relating it to potential factors that may have sped up or slowed down the usual immigration flow of several countries. Therefore, an important part of this project is to learn about real world events that affect the immigration patterns of people, this could be in the form of war, natural disasters, or even government policies. We would like to create a visualization that is intuitive to the user but that also presents all the necessary immigration data to paint a picture of what a country might be like. Users would be able to learn about fairly recent immigration data on several countries worldwide while maybe also learning a little bit about their recent history.

Data Processing

The data, while somewhat organized already, will need to be cleaned further for our usage. Currently the data is housed in multiple Excel sheets, so it will need to be stripped of non-essential fluff and then converted to csv files. Because we have 45 separate files, it is right on the edge of being manually done and finding solutions to

speed up the process. We will experiment with how much each file takes manually and assess if other processes are needed. Within the data, there is also some cleanup that needs to be done. Some countries do not have data for each year, and so those will need to be shown as zeros. Each country also has different criteria for how they count their immigrants. Some are based on birth country, some on current residence, and others on citizenship. While we will use the data the same from each country, we will need to show how each country counts somewhere in the visualization. Finally, we also need to group the data by continents so that we can compute the total numbers for each continent.

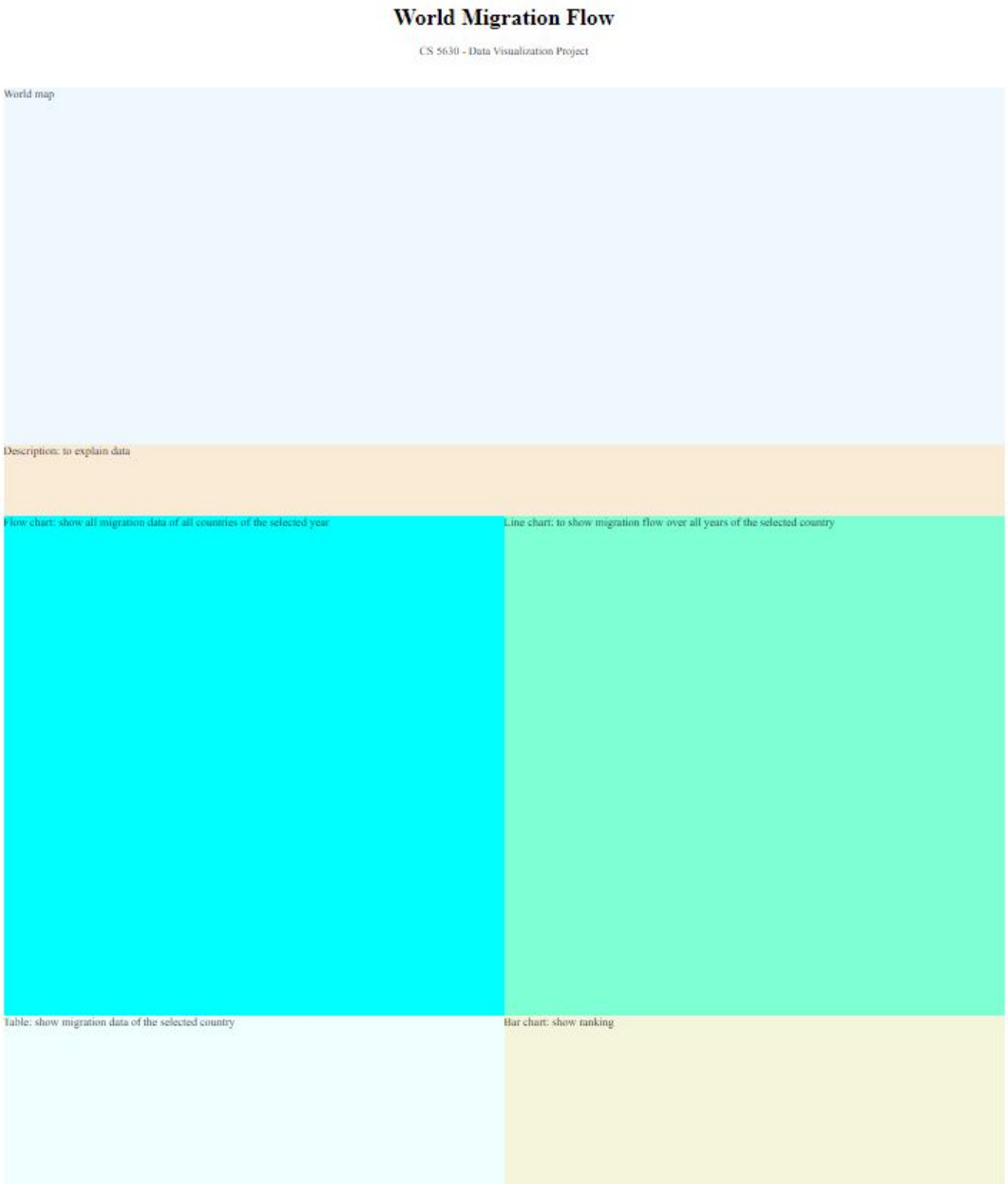
Implementation and Evaluation

Data Wrangling

The dataset we downloaded has 43 csv files. Each file is a dataset of each country in 43 countries who record migration flows. The first step for preparing the data we needed was that we created a root csv file, then copied all the data from 43 csv files to the root file. We found a problem which was that this dataset didn't have country IDs which are used as keys to map with the GeoJSON file. So, we used some functions in MS Excel to create an ID column and copy IDs from the GeoJSON file. The second problem was that the GeoJSON file didn't have the coordinates of countries, so we obtained another file that has coordinates data, then we also needed to map IDs between these files. As a result, we created a clean dataset for our project.

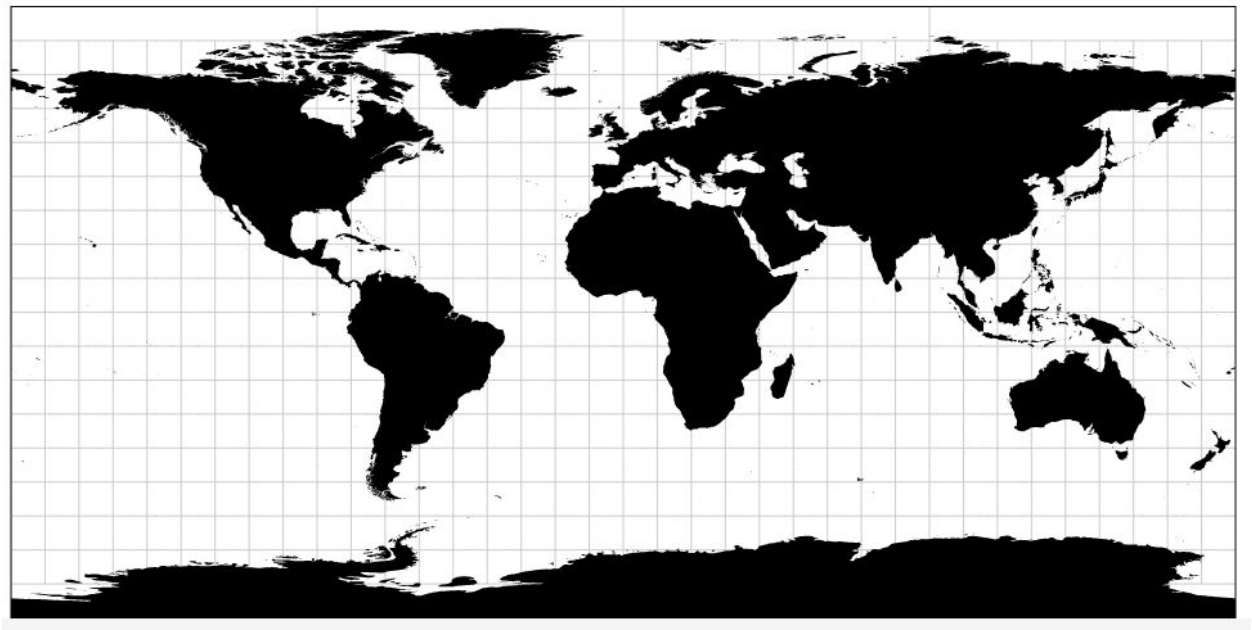
Web layout initialization

We set up the layout using CSS Grid Layout. We considered between Grid Layout and Bootstrap 4, and decided to go with Grid Layout because Bootstrap is not very flexible. The below picture is our initial layout.



World map implementation

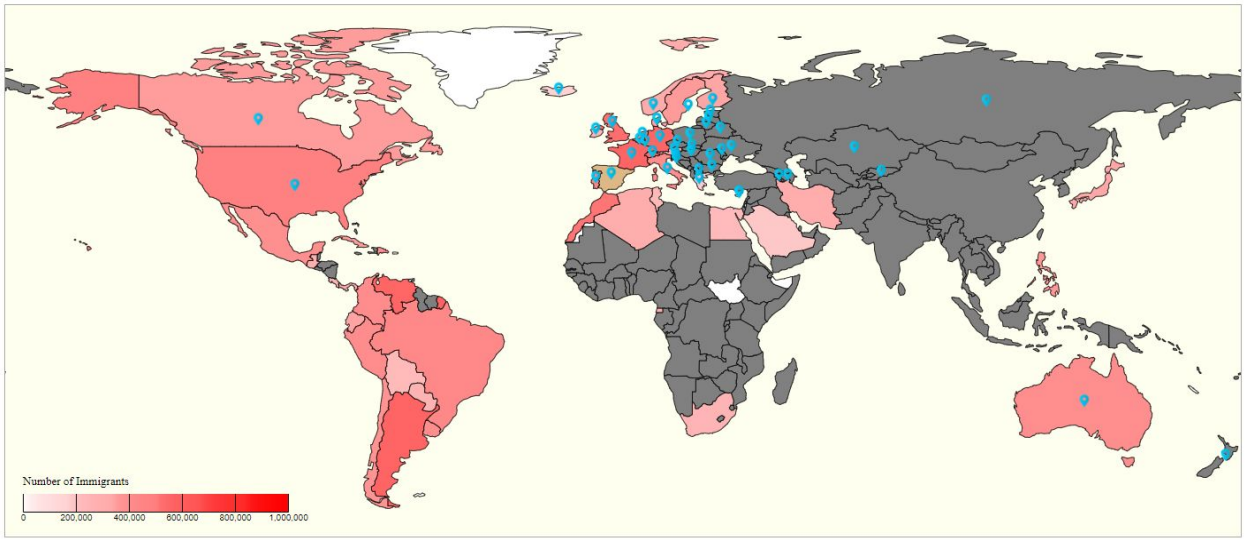
- To draw the map, we need to decide which projection we should use. We considered geoEquirectangular, geoMercator, and geoOrthographic. Then we decided to use geoEquirectangular because geoMercator is not very accurate on area scale. geoOrthographic makes the map like a globe and it is not realistic to show relationship data on a globe. Our initial map was as follows:



- For the 44 countries which have recorded migration data, we need to show users that these countries can be clicked, so we decided to put an icon marker on each of those countries. The icon we used is like the one below:

- For other countries which are included in the migration data, at first we drew a circle as a marker for each country, but they didn't look aesthetic, so we decided we didn't need to draw markers for those countries because they would be filled with colors.
- In our initial design, we wanted to draw lines to show the relationship between countries, but when we drew them on the map, they looked like a mess because there were too many lines since one country such as the U.S. had migration flows from a lot of countries. So we decided to go with using colors to show the migration data. When a user clicks a country that has a marker icon, the map will color the countries which have migration flows with the selected country, and the color saturation shows the data intensity.
- For the color scale, at first we used scaleLinear, but the colors didn't draw at all for some countries, we realized that the min and max values in data have a huge gap, so we decided to use scaleLog. In addition, some countries have no data, so we filled those

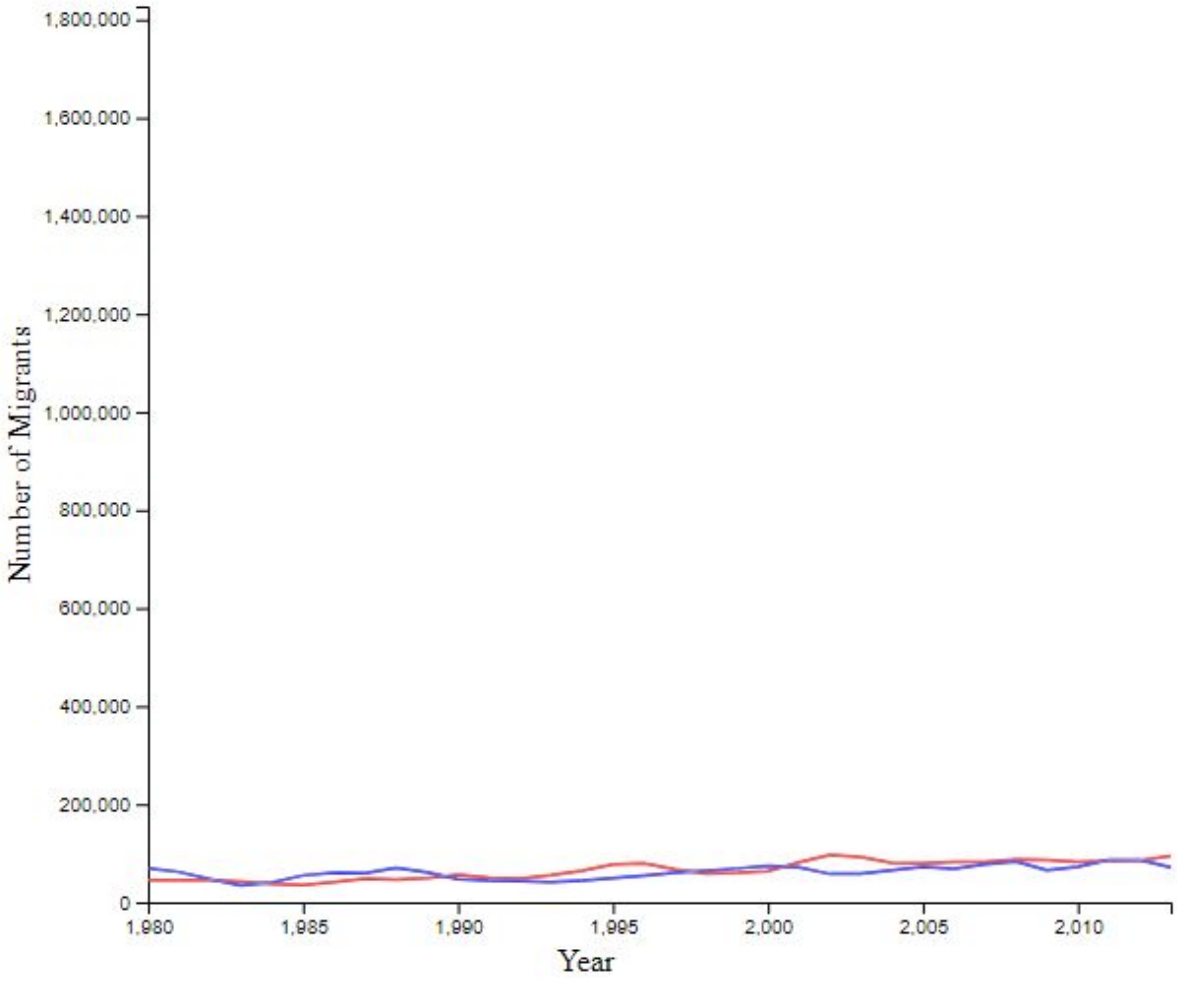
countries with gray color. Besides, we also added the legend for color scale. As a result, our map was as follows:



- For the toggle button, we just applied what we learned from homework to draw the button and make it work. For the dropdown button, we load the years from data, and put them to the dropdown box's values. When users interact with these buttons, an event will emit and all related charts will update accordingly. We also set the default values to "Immigration", and "1990", so that when the web first loads, the migration type will be "immigrants" and the year is 1990.

Line Chart Implementation

- At first, the chart used the scale with the max which was the maximum number of the overall dataset. But it didn't look very nice because of the huge gap in numbers, as a result, some countries with small numbers showed their lines close to the x-axis.



- Then we decided to use max as the maximum number of the selected country over all the years, instead of using the maximum number of all the countries. This required us to re-compute the scale each time users select a country. And the result looked better as we expected.

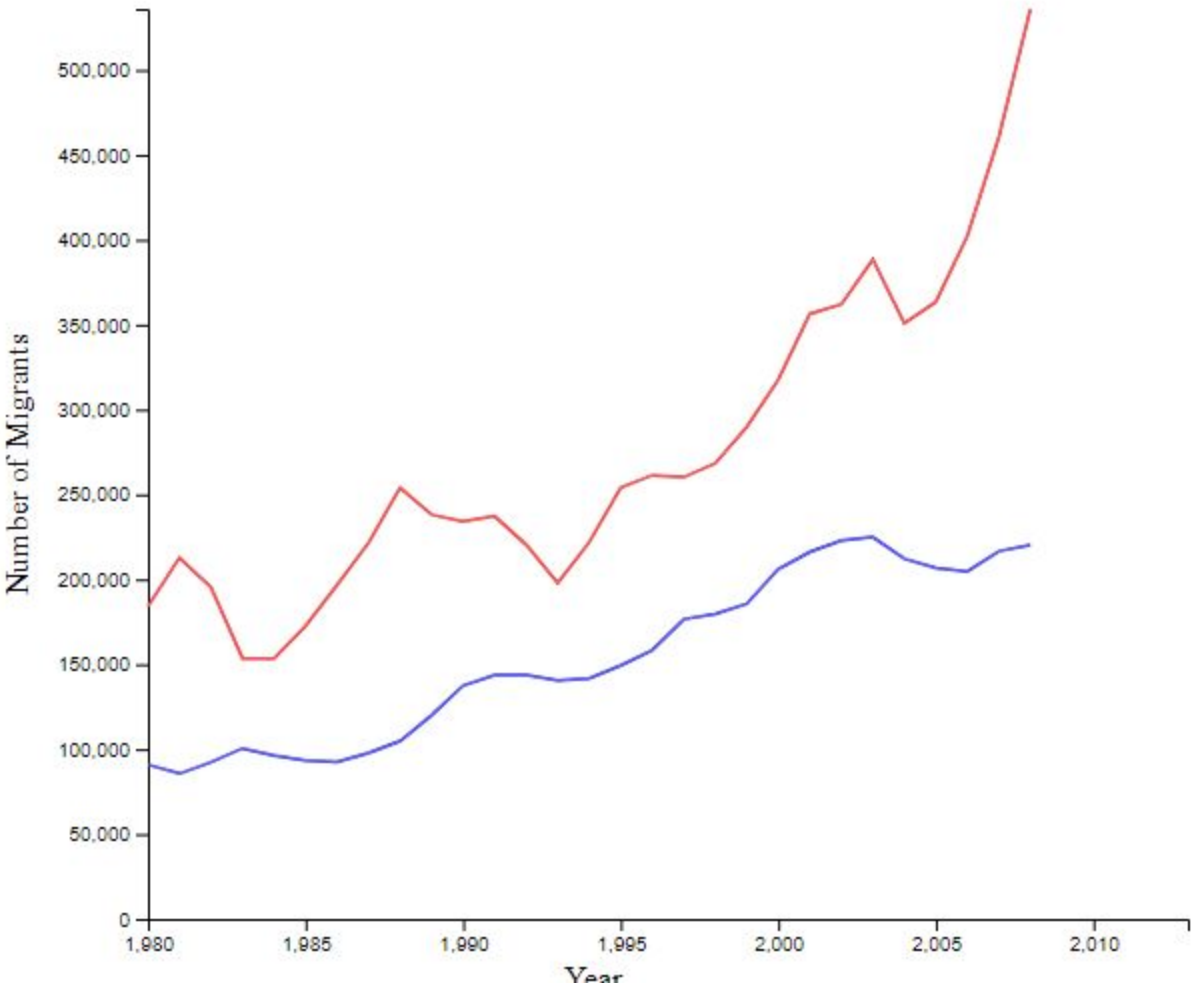
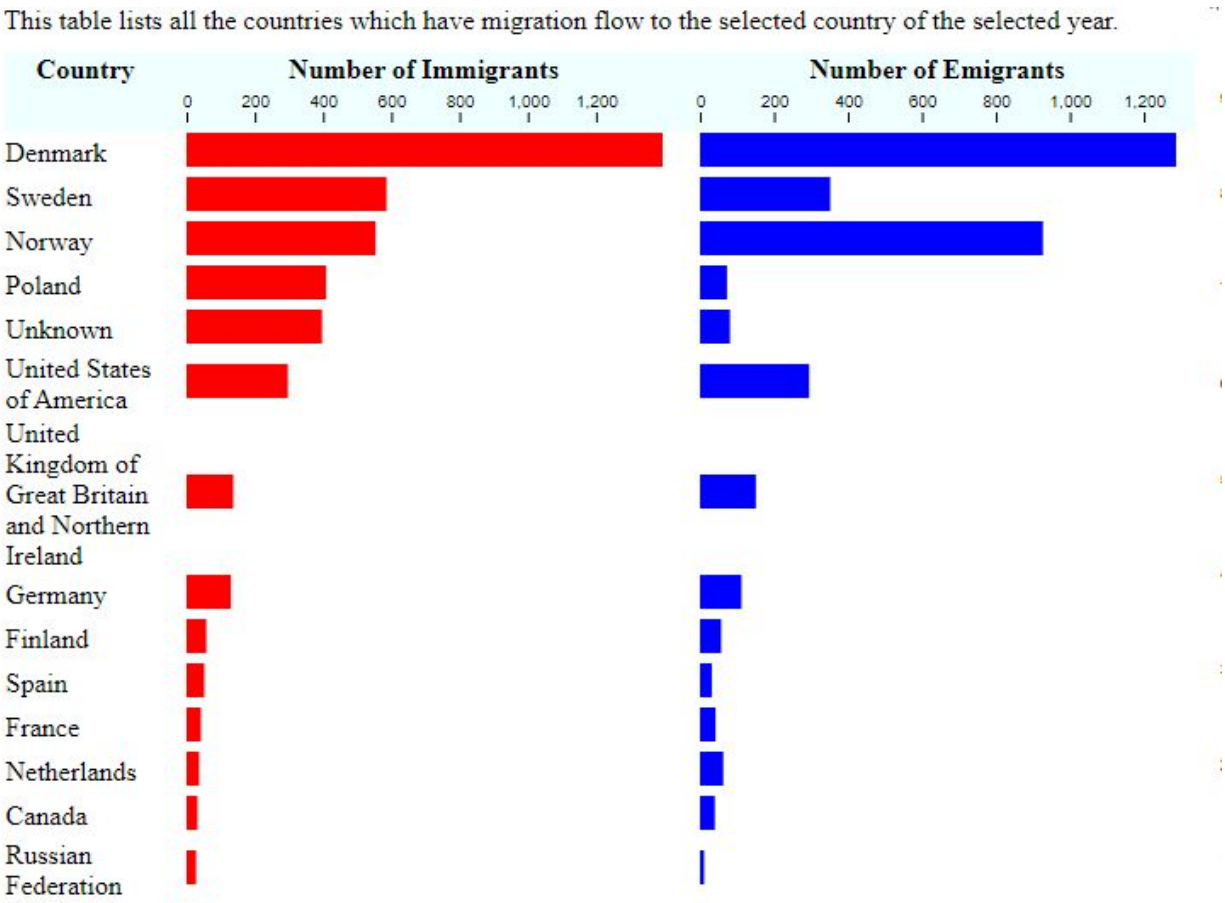


Table Implementation

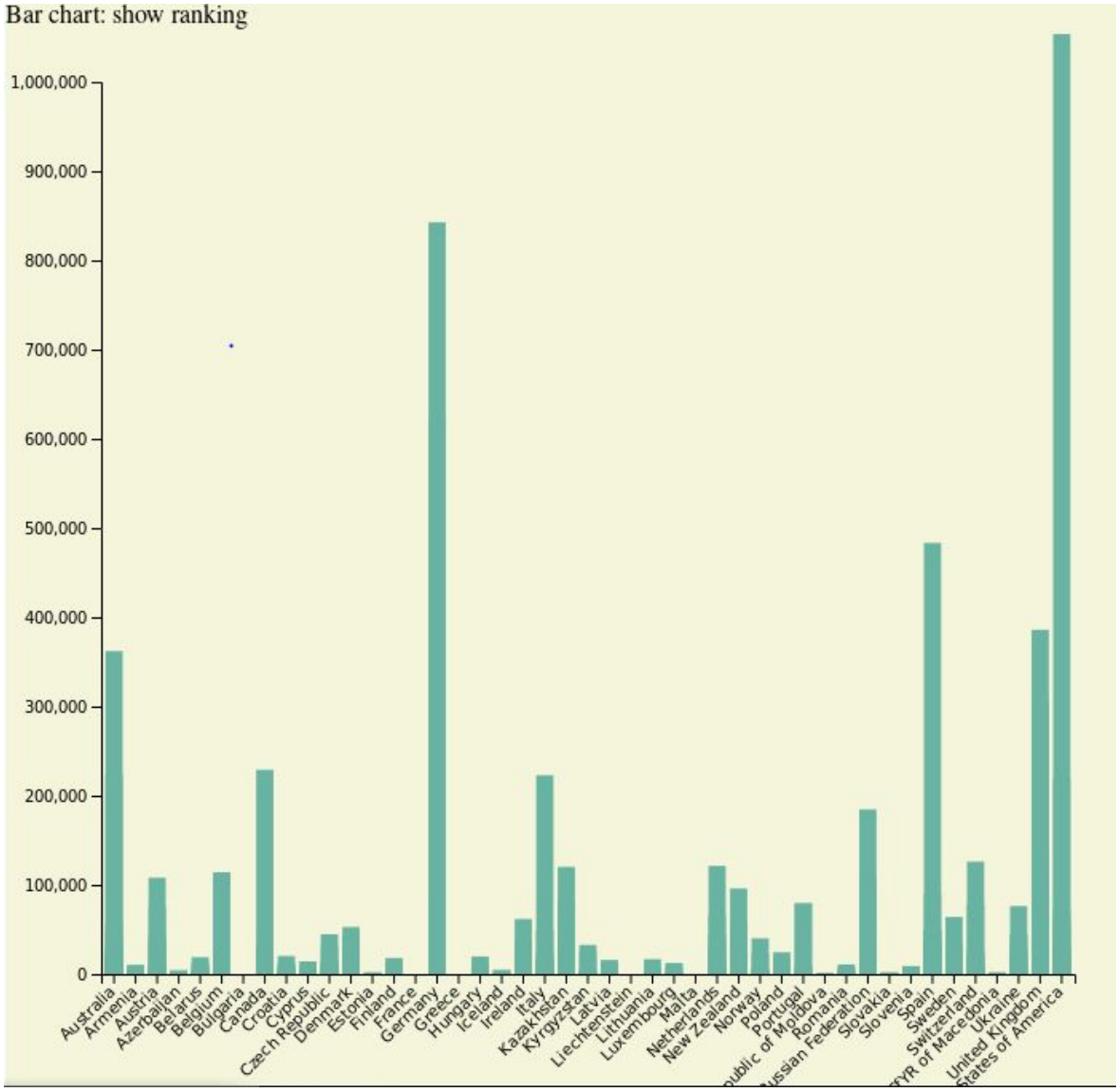
- This table chart showed migration data of all the countries which have migration flows to the selected country. We also implemented the sorting feature which was that when users click one of the headers, the table sorts the rows respected to the header's value.

This table lists all the countries which have migration flow to the selected country of the selected year.



Ranking Implementation

- The current design of the immagration/emigration ranking plot is a bar chart. This might still change as there might be other designs that look better or better represent the data, such as a lollipop chart or maybe even a circular barchart. The current state of the ranking bar plot does not have an updating feature for the data so that when a different country is selected, migration type is toggled, or year is changed the data changes accordingly. This will be added next. It is also missing highlighting the bar that corresponds to the selected country. An idea that is also in the works is adding an animation to the plot so that there is a nice transition whenever a different country is selected in the world map.

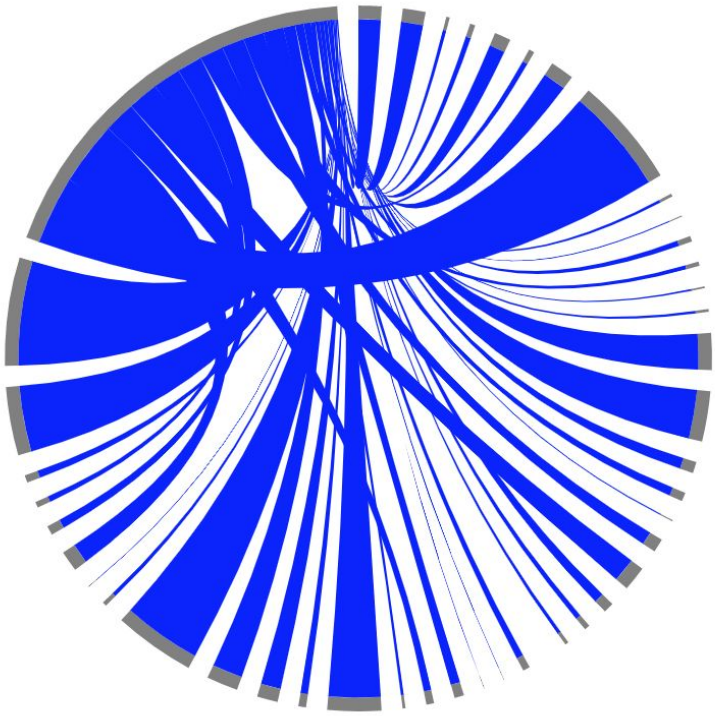


- An issue came up that one of our team members could not be reached out to, and we didn't have enough time to complete the ranking chart. So, we decided to drop this chart out of the project.

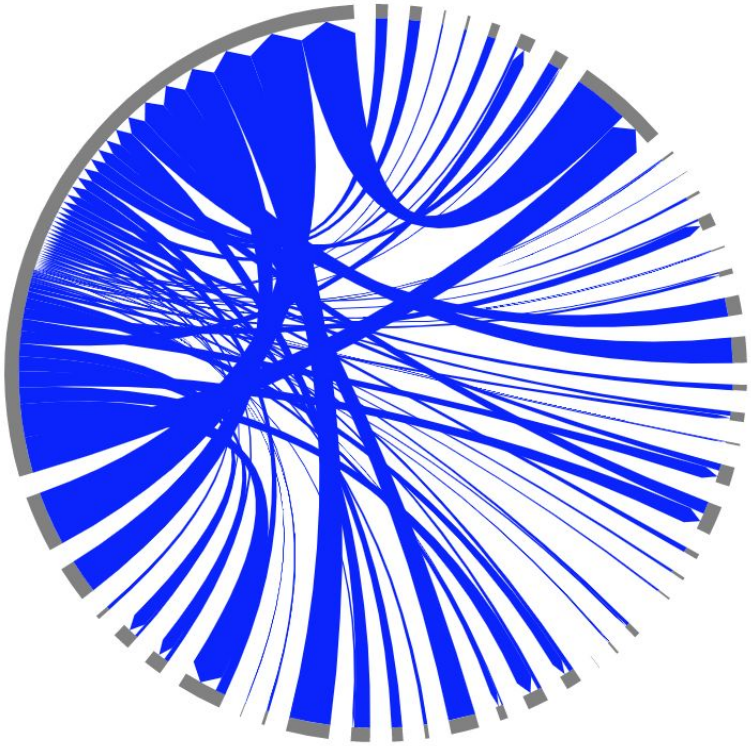
Flow Chart Implementation

- To start with the flow chart, I first had to decide the best way to show all of the data that we have. My first thought would be something like a chord diagram, but I also wanted to explore other options. So I looked around at things such as sankey diagrams, but ultimately went with the chord diagram. The first part was getting it drawing:

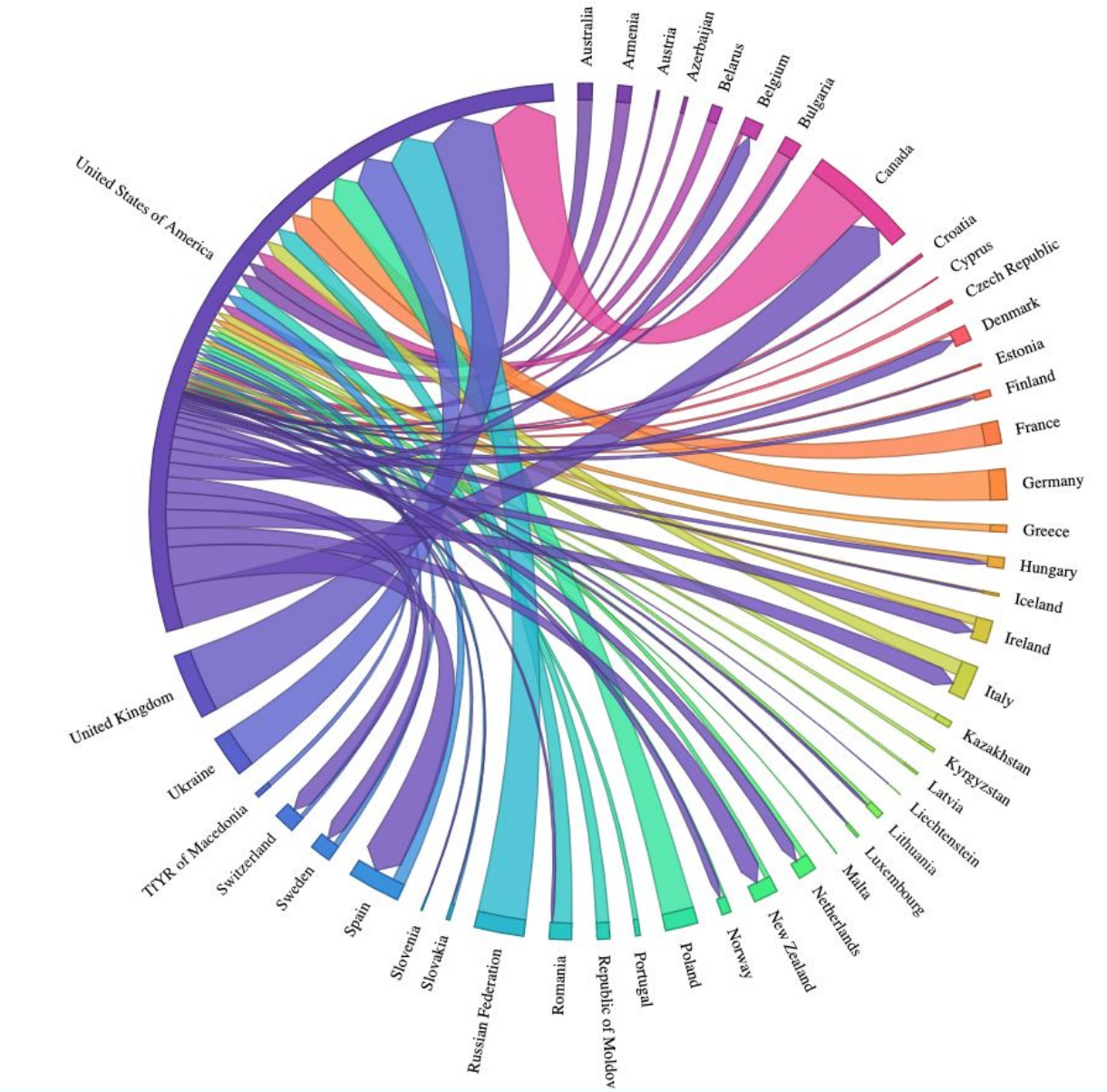
- After getting it drawn, I had to create a matrix from our data that would be turned into the chords. This was not easy as we have so much data. I decided here to just do the 45 countries that we have the main data for.



- After doing this I realized it didn't really make sense. The bars were both trying to represent immigration to and from a country, which didn't show up well because they just didn't make sense. What I needed was directional flows.

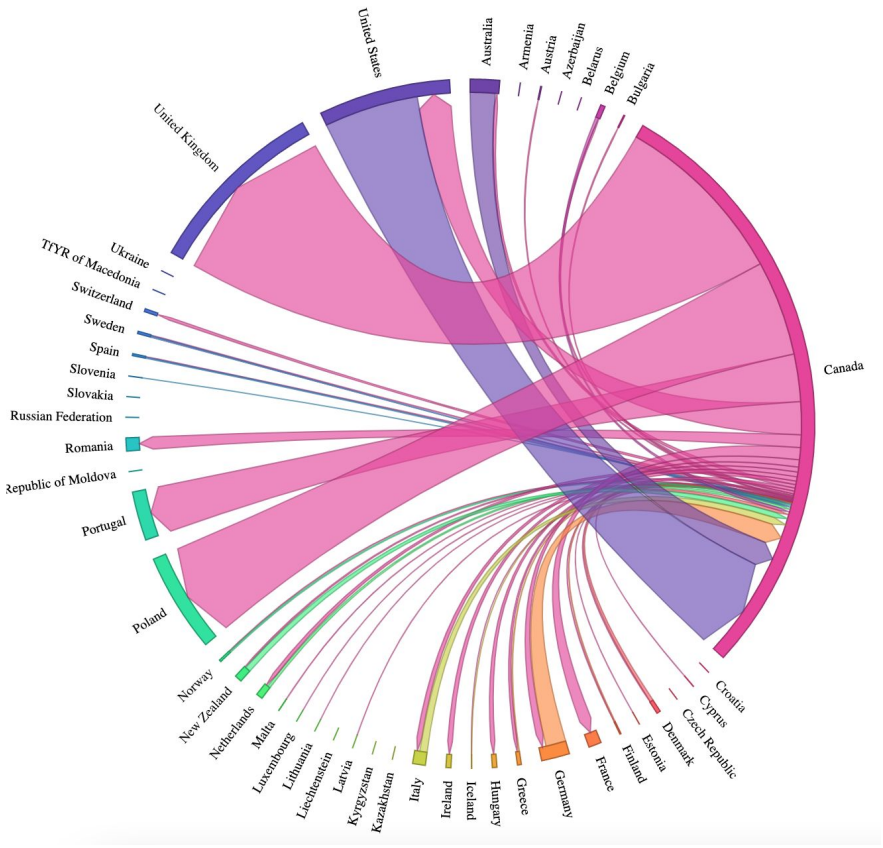


- While it doubles the amount of flows, it now makes sense as the flows are directional and you can see the amount of in and out for a country. Next I added color to the graph, as well as names and transparency to finish it off. While the current flow just shows one selected country and one year, we will be adding the ability to change the year, country, and highlights for the flow
- After getting it drawing, the next thing I had to do was get it to update depending on the



year and the country selected. I thought this would be somewhat straightforward but the

chord diagram deals with data differently than any other diagram that we have worked with before. So I had to look at what few examples I could find as well as the documentation to figure out how to update the data.



- This also allowed me to do transitions. Because certain countries are missing data for certain years, it makes it difficult to transition beautifully. However, for the most part the transitions look good.
- The final step was to get the mouseover event added so that certain paths could be isolated for a country. This was also a little difficult as the paths are not grouped under the countries, so there is some indexing that must be done to find the correct paths.