



OLYMPICS STATS

PROCESS BOOK

CS 6630 - DATA VISUALIZATION

CONTENTS

- Basic Information
- Overview and Motivation
- Related work
- Questions
- Data
- Exploratory Data Analysis
- Design Evolution
- Implementation
- Evaluation

Basic Information

PROJECT TITLE: Olympics Stats

TEAM MEMBERS:

- Krunal Jain (u1140629) – krunal.bitsy@gmail.com
- Abhishek Lakshminarayana (u1145084) – abhishekl9426@gmail.com
- Vinod Reddy Gooty (u1142935) – gootyvinod12@gmail.com

REPOSITORY: https://github.com/krunaljain/dataviscourse-olympics_stats

Overview and Motivation

On April 6, 1896, the first modern Olympic Games was held in Athens, Greece, with athletes from 14 countries participating. From then the places where Olympics takes place started to change. Later, every year the number of countries participating increased. From year 1900 women started to participate in the Olympics.

The Olympics are leading international sporting events featuring summer and winter sports competitions in which thousands of athletes from around the world participate in a variety of competitions. The Olympic Games are considered the world's foremost sports competition with more than 200 nations participating. The Olympic Games are held every four years, with the Summer and Winter Games alternating by occurring every four years but two years apart.

All countries vouch for success at Olympic games, which are held once every 4 years. The performance of various countries over the years has shown some interesting trends. The aim of our project is to capture and visualize this trend and provide an insight into how different regions across the world fared in these games.

The motivation for us to consider visualizing Olympics stats is that how modern Olympics has changed over years along with several factors like men/women ration, world wars, GDP of participating countries. And coming from India, we always wonder why India doesn't perform well in Olympics. While we play with the data, we shall try to analyze why some countries have never made an impact in Olympics.

In this project we are going to make visualizations by which we will be able to see a trend in Olympics over the years. This will include medals won by countries, country ranking, men/women ratio, host nation performance. Overall, we can infer from the designs what the trend is and will be worth reckoning to know in detail. Our goal is to make visualizations synchronized with each other so that it's easy to deduce the learning from it.

Related Work

Inspiration for this project came from personal interest in Olympics sports. Throughout the course, we studied different visualization techniques and their applications in specific use cases. Through this project, we wish to incorporate these visualizations to highlight the Olympic performances of different countries over the years.

Questions

The visualization project tries to analyze certain statistics of Olympics and will help us giving details to the people who are very keen to know what's the trend. The main questions we want to answer with this project include the following:

- How European countries dominated at the beginning?
- How host nations perform better compared to previous years?
- Did world wars affect Olympics? How?
- Does GDP of country have any effect on Olympics medal?
- Men/women medal count for country over years?

Data

We are using one dataset. Here is the link to the dataset <https://www.kaggle.com/the-guardian/olympic-games> . This dataset has 3 CSV files, dictionary.csv, summer.csv, winter.csv. Since we are concentrating only on summer Olympics in this Project, we skip winter.csv file.

Dictionary file has list of country name and respective country code, population and GDP per Capita. Summer file has year column in ascending order of Olympics and respectively where it took place, different sport and its classification and 1st/2nd/3rd winner details in each classification.

We have two kind of views in our project:

- Basic Views: For these views, we can directly use the data available in the CSV file and do projections to filter out columns depending on our requirement.
- Aggregated views: For these types of views, the data is aggregated based on some parameter. So, we need to do data processing.

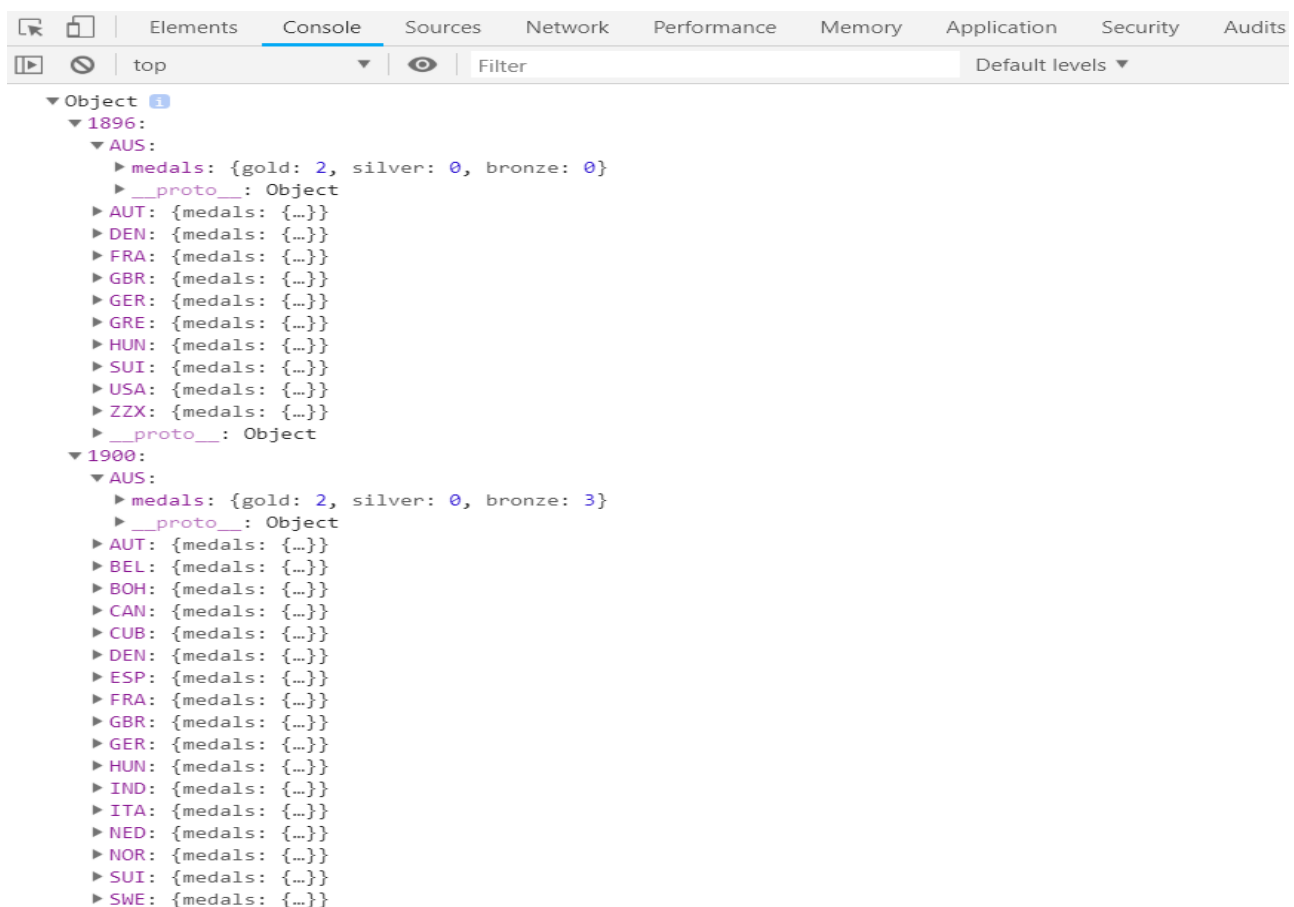
Exploratory Data Analysis

The schema of the dataset is as explained above. For each view, we had to aggregate and group data according to the desired parameters. The following are the different transformations we have done on the data depending on the requirements of the view. In

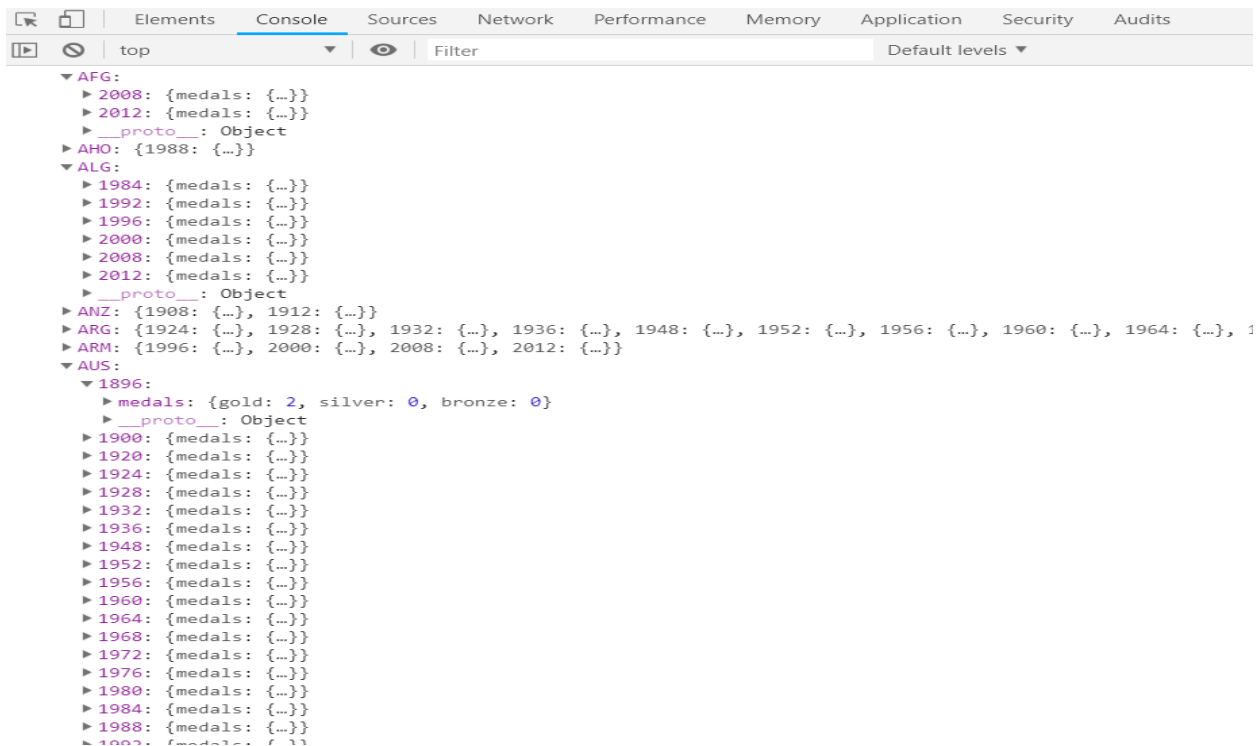
the code, we implemented a function named `aggregate`, which takes in two parameters, the `aggregate` parameter and the `grouping` parameter. Based on the values provided to this function, we can get the data required for all the views in our project.

Consider the example of the primary view of the project, i.e the country wise distribution of medals for each Olympic. This view would require the aggregation of our primary dataset based on the year parameter and then grouping by countries to get medal count for each country for a specific year. Assume the dataset csv to be loaded in **Olympics data** object. We can get the data in the desired form by simply calling the `aggregate` function with the `aggregate` parameter as `year` and grouping parameter as `country`. We came up with a robust design and implementation of this function to get the data for all the views.

Example 1: Aggregated by years and grouped by countries



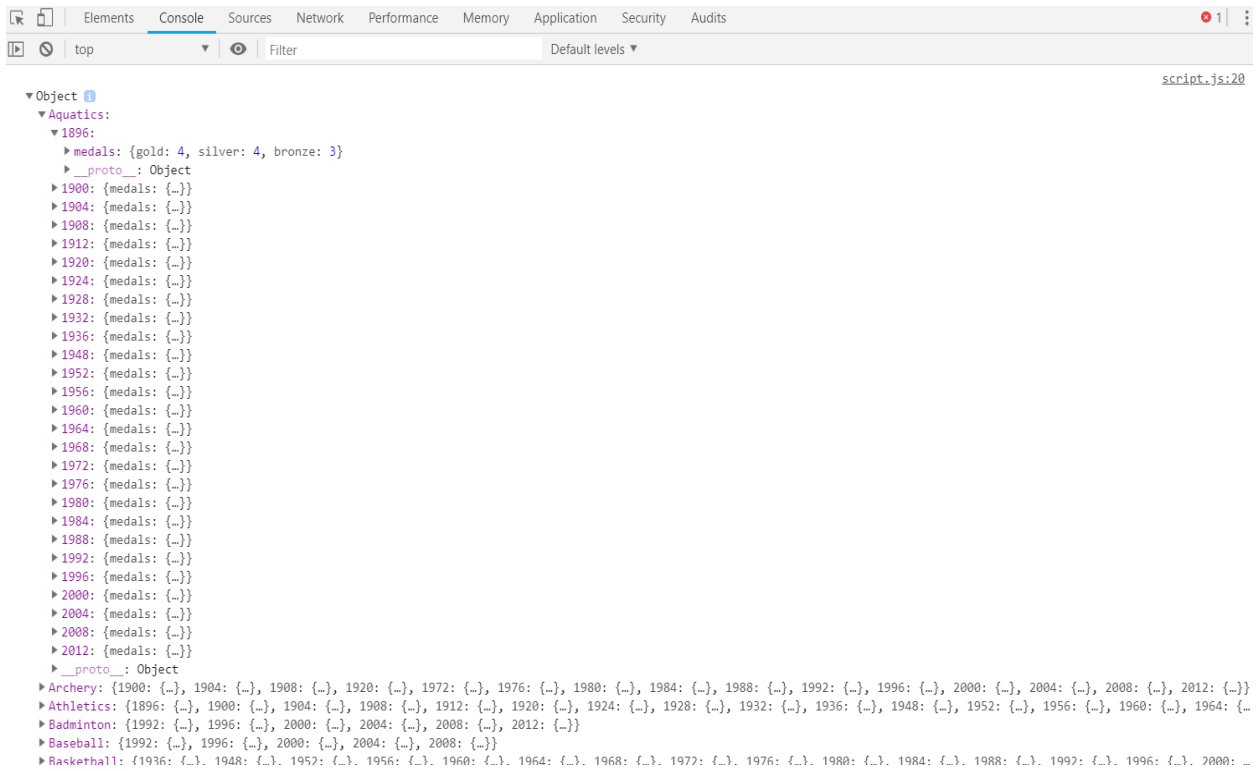
Example 2: Aggregated by countries and grouped by years



The screenshot shows the Chrome DevTools Console with the 'Console' tab selected. The top toolbar shows 'top' as the selected scope and 'Filter' as the search filter. The 'Default levels' dropdown is also visible. The console displays a large, nested object structure representing aggregated data by country and year. The structure is as follows:

- AFG:**
 - 2008: {medals: {...}}
 - 2012: {medals: {...}}
 - __proto__: Object
- AHO:** {1988: {...}}
- ALG:**
 - 1984: {medals: {...}}
 - 1992: {medals: {...}}
 - 1996: {medals: {...}}
 - 2000: {medals: {...}}
 - 2008: {medals: {...}}
 - 2012: {medals: {...}}
 - __proto__: Object
- ANZ:** {1908: {...}, 1912: {...}}
- ARG:** {1924: {...}, 1928: {...}, 1932: {...}, 1936: {...}, 1948: {...}, 1952: {...}, 1956: {...}, 1960: {...}, 1964: {...}, 1968: {...}, 1972: {...}, 1976: {...}, 1980: {...}, 1984: {...}, 1988: {...}, 1992: {...}, 1996: {...}, 2000: {...}, 2004: {...}, 2008: {...}, 2012: {...}}
- ARM:** {1996: {...}, 2000: {...}, 2008: {...}, 2012: {...}}
- AUS:**
 - 1896: {medals: {gold: 2, silver: 0, bronze: 0}, __proto__: Object}
 - 1900: {medals: {...}}
 - 1920: {medals: {...}}
 - 1924: {medals: {...}}
 - 1928: {medals: {...}}
 - 1932: {medals: {...}}
 - 1936: {medals: {...}}
 - 1948: {medals: {...}}
 - 1952: {medals: {...}}
 - 1956: {medals: {...}}
 - 1960: {medals: {...}}
 - 1964: {medals: {...}}
 - 1968: {medals: {...}}
 - 1972: {medals: {...}}
 - 1976: {medals: {...}}
 - 1980: {medals: {...}}
 - 1984: {medals: {...}}
 - 1988: {medals: {...}}
 - 1992: {medals: {...}}
 - 1996: {medals: {...}}
 - 2000: {medals: {...}}
 - 2004: {medals: {...}}
 - 2008: {medals: {...}}
 - 2012: {medals: {...}}
 - __proto__: Object

Example 3: Aggregated by sport and grouped by years



The screenshot shows the Chrome DevTools Console with the 'Console' tab selected. The top toolbar shows 'top' as the selected scope and 'Filter' as the search filter. The 'Default levels' dropdown is also visible. The console displays a large, nested object structure representing aggregated data by sport and year. The structure is as follows:

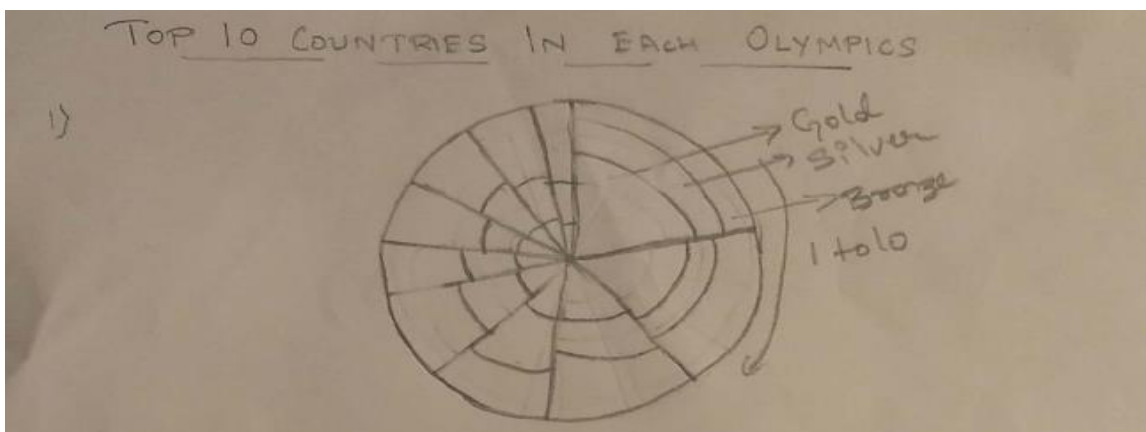
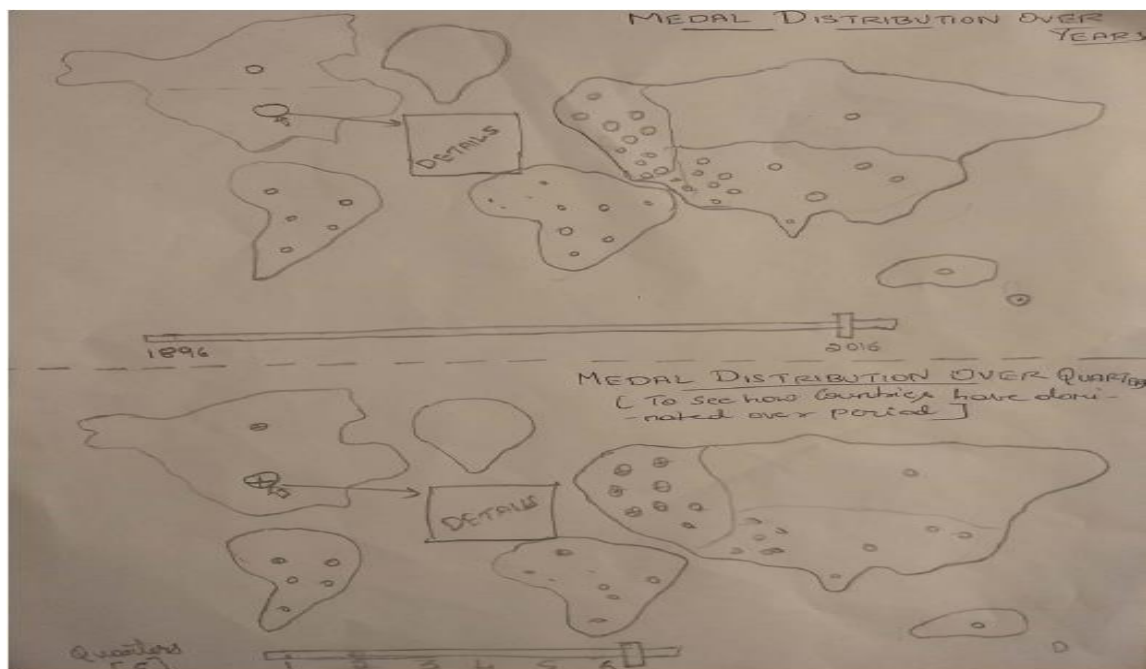
- Object**
 - Aquatics:**
 - 1896: {medals: {gold: 4, silver: 4, bronze: 3}, __proto__: Object}
 - 1900: {medals: {...}}
 - 1904: {medals: {...}}
 - 1908: {medals: {...}}
 - 1912: {medals: {...}}
 - 1920: {medals: {...}}
 - 1924: {medals: {...}}
 - 1928: {medals: {...}}
 - 1932: {medals: {...}}
 - 1936: {medals: {...}}
 - 1948: {medals: {...}}
 - 1952: {medals: {...}}
 - 1956: {medals: {...}}
 - 1960: {medals: {...}}
 - 1964: {medals: {...}}
 - 1968: {medals: {...}}
 - 1972: {medals: {...}}
 - 1976: {medals: {...}}
 - 1980: {medals: {...}}
 - 1984: {medals: {...}}
 - 1988: {medals: {...}}
 - 1992: {medals: {...}}
 - 1996: {medals: {...}}
 - 2000: {medals: {...}}
 - 2004: {medals: {...}}
 - 2008: {medals: {...}}
 - 2012: {medals: {...}}
 - __proto__: Object
 - Archery:** {1900: {...}, 1904: {...}, 1908: {...}, 1920: {...}, 1972: {...}, 1976: {...}, 1980: {...}, 1984: {...}, 1988: {...}, 1992: {...}, 1996: {...}, 2000: {...}, 2004: {...}, 2008: {...}, 2012: {...}}
 - Athletics:** {1896: {...}, 1900: {...}, 1904: {...}, 1908: {...}, 1912: {...}, 1920: {...}, 1924: {...}, 1928: {...}, 1932: {...}, 1936: {...}, 1948: {...}, 1952: {...}, 1956: {...}, 1960: {...}, 1964: {...}, 1968: {...}, 1972: {...}, 1976: {...}, 1980: {...}, 1984: {...}, 1988: {...}, 1992: {...}, 1996: {...}, 2000: {...}, 2004: {...}, 2008: {...}, 2012: {...}}
 - Badminton:** {1992: {...}, 1996: {...}, 2000: {...}, 2004: {...}, 2008: {...}, 2012: {...}}
 - Baseball:** {1992: {...}, 1996: {...}, 2000: {...}, 2004: {...}, 2008: {...}}
 - Basketball:** {1936: {...}, 1948: {...}, 1952: {...}, 1956: {...}, 1960: {...}, 1964: {...}, 1968: {...}, 1972: {...}, 1976: {...}, 1980: {...}, 1984: {...}, 1988: {...}, 1992: {...}, 1996: {...}, 2000: ...}

Design Evolution

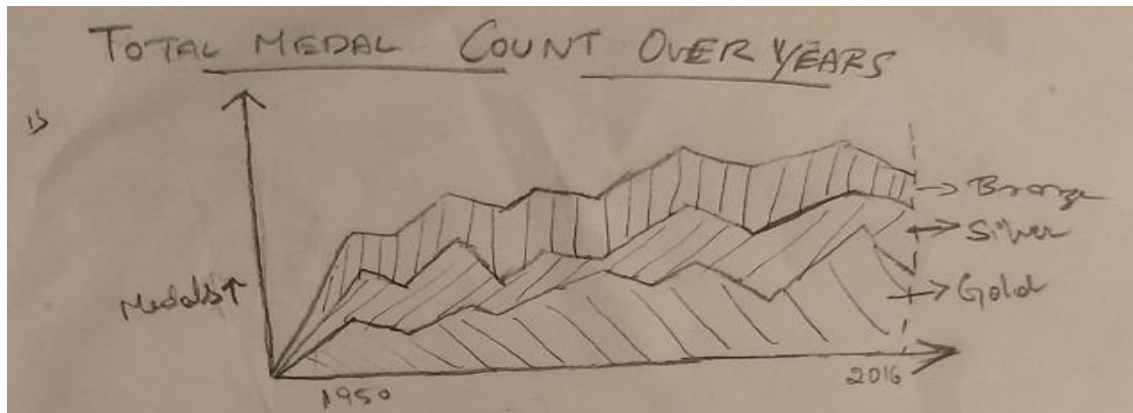
Proposal Design

In our proposal we had 5 views.

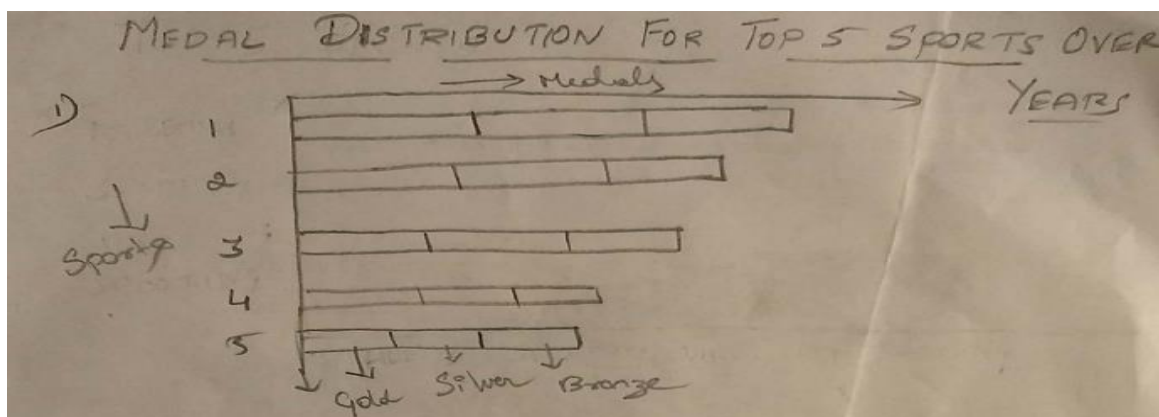
- ❖ **Map:** Once we launch the website our first visualization is a map which shows country with medal count. We use a circle to denote the countries that participated in that particular year and circle size represents the medals won.
- ❖ **Pie Chart:** We use this chart to view top 10 countries in selected year. Here each pie represents a country and each pie is divided into 3 parts representing different medal types.



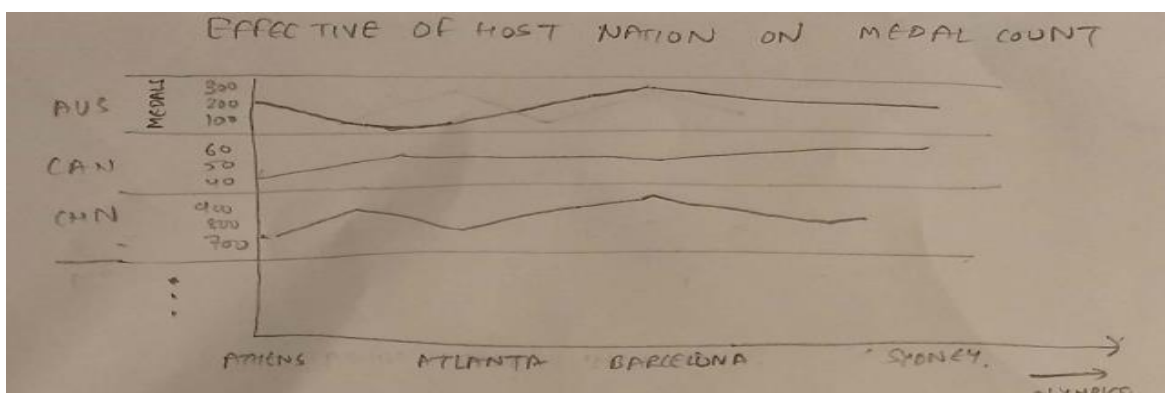
- ❖ **Area Chart:** This is used to visualize overall medal count in Olympics over the years. In this, x-axis represents year of Olympics and y-axis represents medal count. We have 3 areas in this chart with each representing different medal types.



- ❖ **Stacked Bar Chart:** Used to visualize top 5 sports (highest medal distribution) in each Olympics. Where x-axis represents Medal count and y-axis represents 5 sports. We have 3 stacks in this bar with each representing different medal types.



- ❖ **Line Chart 1:** Host nation performance can be visualized using this chart. Multiple line chart is stacked with each other to compare with other country performance.

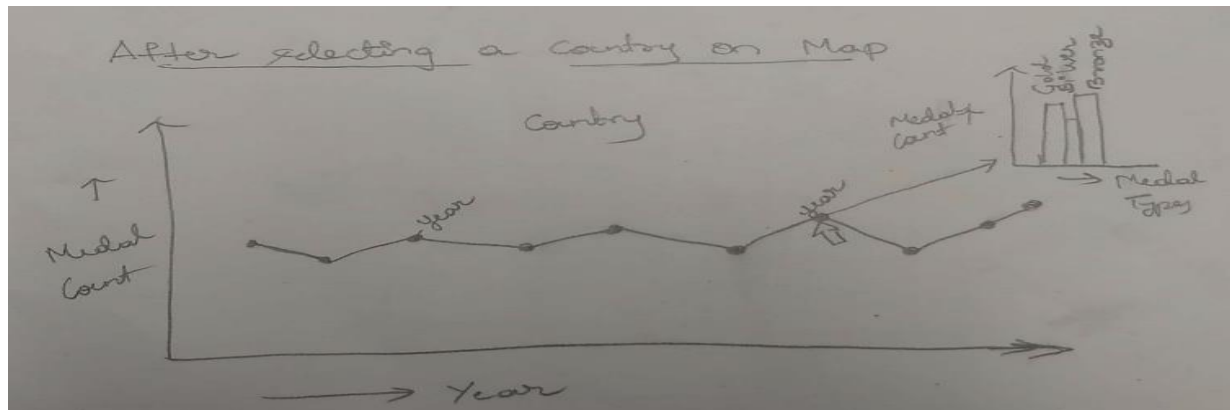


Design Changes

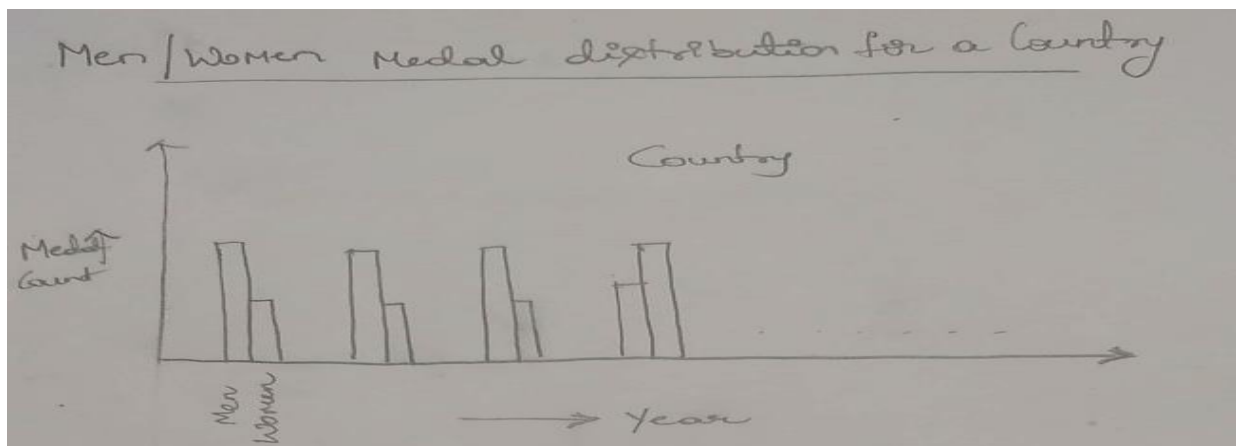
After getting feedback from team during feedback hours in class and discussing with our mentor **Mengjiao Han**, we decided to make some changes to the design and the interaction.

Additions

- **Line chart 2:** Used to visualize medal count over years for that country.

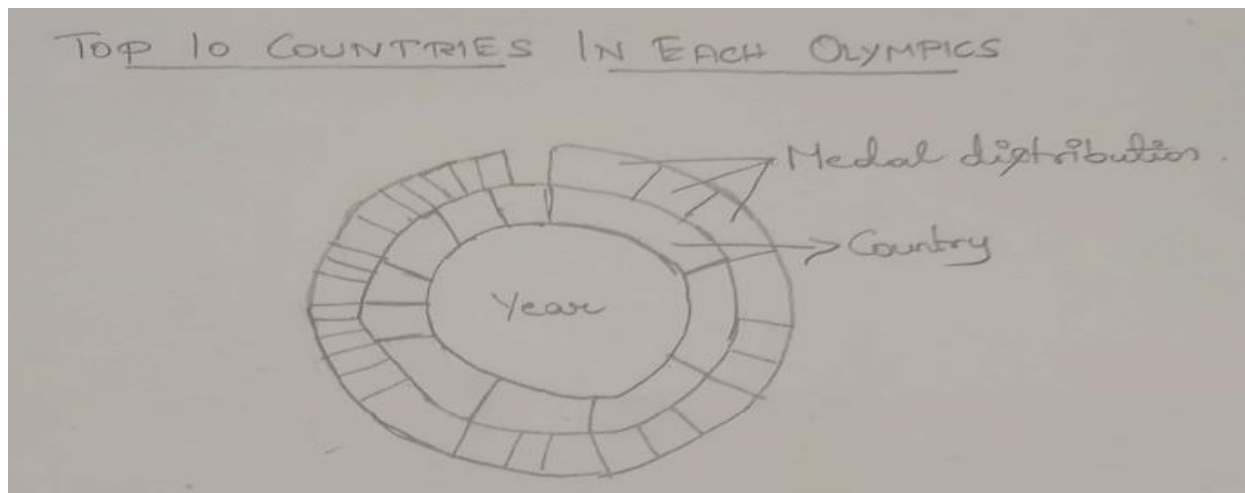


- **Bar chart:** Used to visualize men/women medal count for a country over years. X-axis represents years and Y-axis represents medal count.

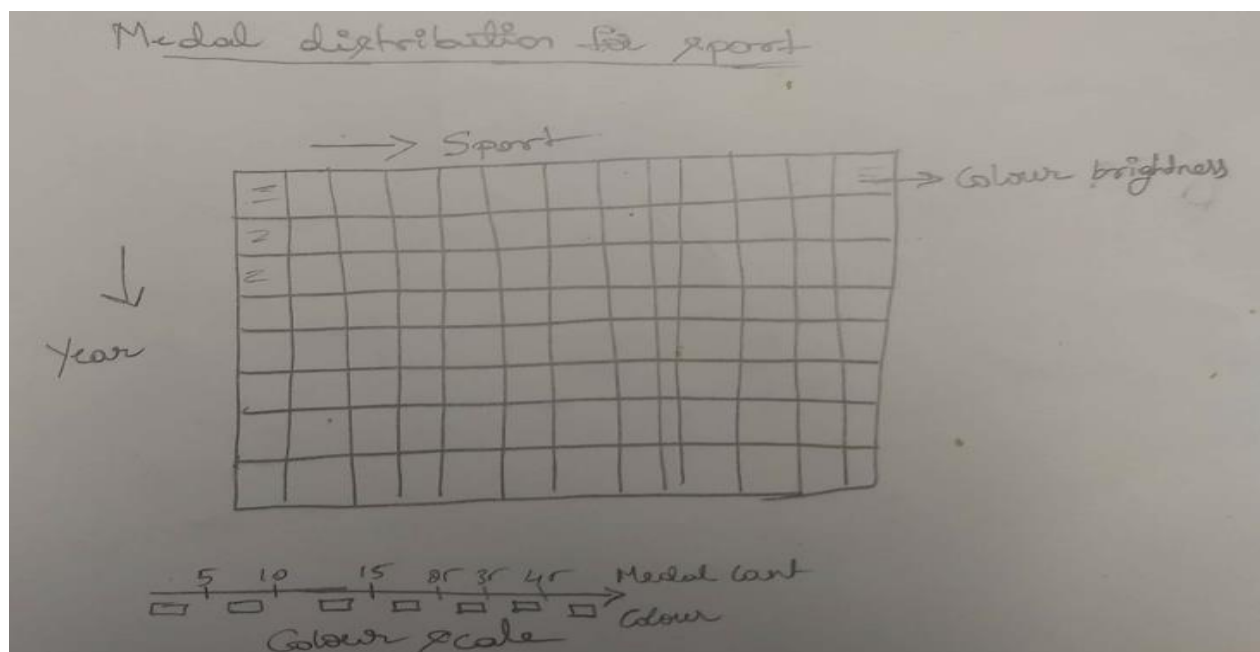


Changes

- **Pie Chart to Sunburst:** After getting suggestion from mentor we planned on visualizing top 10 countries in each Olympics using sunburst instead of pie chart.



- **Stacked Bar chart to Heat Map:** Here instead of displaying top 5 sports we planned on displaying all sports over the year using heat map instead of stacked bar chart.



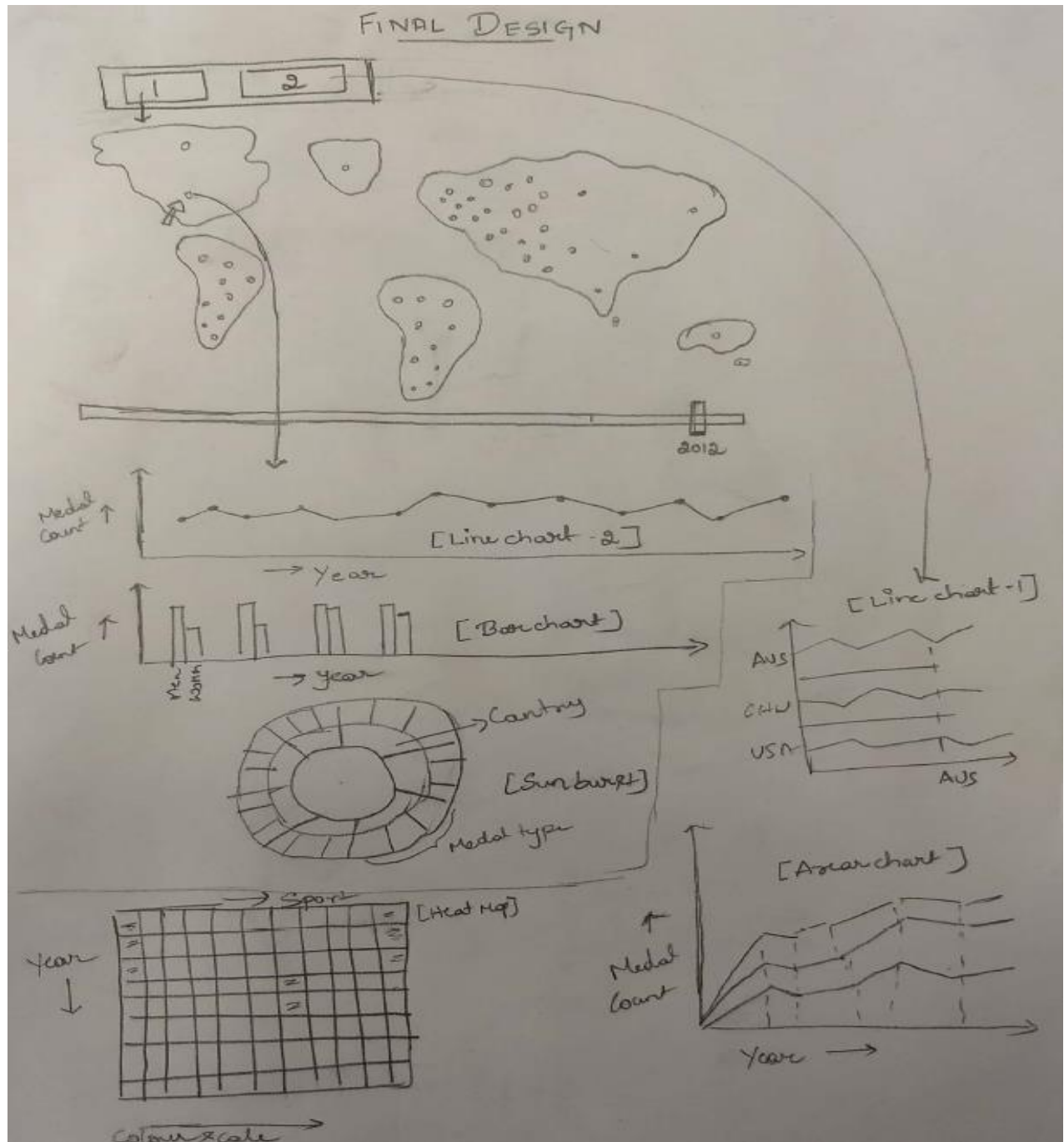
Final Design

Below is the final design of the Project. Which has two options:

- On launch, default option-1 will be selected, and a map will be displayed with country and their medal count for a default year. Once you click on a country, three visualization will be displayed.
 - Line Chart 2
 - Bar Chart
 - Sunburst
- On selecting option-2, three visualization will be displayed,

- Line Chart 1
- Area Chart
- Heat Map

All the visualizations will be interactive and will plan on using brush after implementing the below design.



Implementation

We implemented different data processing phases to make the data compatible to the views. We also had different implementation approach for the maps. We started with the Google Geolocation API and fetched the latitude and longitude information from the country name. We were able to get the data using the API, but it had restrictions on the number of free accesses. As a result, we had to change the approach and used a local json file to get the path coordinates similar to the HW4 approach. Also, we faced issues with making the data compatible with the map. We converted the topojson to geojson and created the path elements.