

Visualising World Population Against World Happiness

Data Visualisation And analytics

Michael Soutar | F20DV | 13/04/2023

# **INTRODUCTION**

In this report you will find the justification and reasoning behind why I choose to select world population and compare the contrasts between the world happiness index as an attempt though it unfortunately didn’t work as planned. Originally, I wanted to select world happiness and compare it to the weather but the data just wasn’t available unlike that of the world population. Searching around I found dataset from Kaggle but choose to use the dataset for the world population from “Our Would in Data” as it match what sort of information I wanted to showcase for the world population.

Now the data presented had to be complex in natural; which I had originally overlooked. Hence why there is 3 separate javascript files in the code as: LineChart, PieChart and BoxChart as I had started on those 3 before realizing I had to do something a bit more complicated. So I elected to keep the line-chart and the world map but unfortunely the world-map didn’t work as intended and I feel if I gave myself more time I would have figured it out eventually.

## **design choice reasoning**

So as mentioned before I elected to do a line chart and the world map but as mentioned the world map didn’t work as intended so all I was left was the line chart. The line chart is broken into two sections. One as a basic line chart that show the increase in population over a number of years and the other as a other line chart that when selected, the user can highlight on certain years and the first chart will update accordingly. This means if a user wants to select the years of 1950-1970 the first graph will update showing more concise data. The world map that I had intended to implement was to have it that when the user clicks on the country on the map the line chart would update to whatever country that the user had selected. As mentioned this didn’t work so the only way a to change which country to view is by going onto the code and altering the code from say “Asia (UN)” as it is currently to say “France” and the graphs will update that way. The graphs can also be reset via the “reset” button that is on the bottom of the page which means that if you’ve gone to far in the years and wish to focus on the years 1990-2000 then the button can be pressed and the user can easily move the slider back along.

## **cODE**

As mentioned, I attempted to create a number of different charts to highlight the data, this includes all the filles associated with the project. Bar-Chart.js, Line-Chart#3.js, Pie-Chart.js, World-Map.js and World-Map#2. Even though Line-Chart.js is the only working file I felt it was necessary to include the other files to show that I have worked on them in terms of their implementation but unfortunately just couldn’t get them working. The Line-Chart that I did managed to get working, I chose to reuse a lot of the same code that I had used for the first coursework as I knew it was working and just needed to change a number of the variables to match up with the new dataset that I had. So the only pieces of code that actually have of any relevance is the Line-Chart.js, index.html and style.css. Before I go further with the design choice the style.css it links to the overall design choice of the dashboard. The wrapper that holds the information for the line chart is there, there is a secondary wrapper there for the world map but as it doesn’t work, it was a bit redundant to keep it there, but I feel that showcasing all work done even if it doesn’t work to it’s intended purpose is better than nothing. Then everything else in the style sheet is self-explanatory as it just links to what has been created in the Line-Chart.js file.

### **LINE-CHART.JS**

The chart displays population data for a selected country over time ranging from the years that are presented in the csv file. The constants at the top of the code define the size and margins of the chart. Then, the code loads a CSV file containing the population data for multiple countries. Next, the code creates two SVG elements: one to hold the main line chart, and one to hold a smaller chart used for selecting a specific range of years. The scales for the x and y axes of both charts are defined. The function MakeLineChart is defined to create the line chart for the selected country. It first filters the CSV data to only include the selected country, then sorts the data by year in descending order. The x and y domains are set based on the data, and the line chart is created using the D3 line generator. Finally, the x and y axes are added to the chart.

## **FUTURE THOUGHTS AND CONCLUSION**

Previously mentioned a number of my code aspects didn’t work as intended which is a shame because if I had given myself more time to implement them all then I may have been able to get it working.

## **APPENDIX**

Here I will store all the images associated with the codeA screenshot of a computer

Description automatically generated with medium confidence

Text

Description automatically generated

## **rEFERENCES**

1. <https://ourworldindata.org/world-population-growth>
2. <https://d3-graph-gallery.com/graph/backgroundmap_basic.html>
3. https://d3-graph-gallery.com/line.html