Data Visualization - Process Book

Julien Biefer, Alexandra Korukova

June 2021

1 Process

At the beginning of the semester we were working on a completely different dataset related to the osu! online game. Because the owner of the data dropped the course, we decided to start another project. We were very motivated to make a project related to the EPFL course evaluation to motivate the students to participate. However, this idea turned out to be impossible: the required data is considered sensible and we did not succeed to retrieve it on time. Finally, we have restarted all over. This time, because of the lack of time, we have chosen well structured and simple datasets. Thus, our biggest challenge was organisational. We had to change the subject of the project two times and did not have much time left to implement the core part of the project.

1.1 Data

All of the datasets used for the project are CSV files.

The main three files used for the visualisations contain country by country (lines) and year by year (columns) outbound value, expenditures value and population. First of all, the datasets were cleaned to filter out the years for which data was unavailable and remove the information not related to our project, such as the lines containing values corresponding to the regions. Then, the tables were merged and processed to calculate the departures per capita. This was performed in the *Data Analysis* jupyter notebook (link). The generated tables were saved in the CSV format.

When we started to implement the visualisations, we realised that we needed two auxiliary datasets: one containing the map between alpha-2 and alpha-3 ISO country codes and one containing the continent for every country. The former dataset is necessary because one library we have used for the visualisations (amCharts) uses alpha-2 code to identify countries, while only alpha-3 codes were specified in the initial datasets. The second dataset is used to group countries in the bubble chart and the line plot.

Later on, datasets were reorganized several times to better suite the visualisation needs. We had to select one of the options (time-memory trade-off): either split the data into multiple files (one file per year, one file per country for every value), or put all the data together into a single large CSV. The first option would lead to a faster data retrieval for a given visualisation, but would cause many replications. With the second approach it would be slower to retrieve the data, since every time a particular value is needed, it has to be found in a large data file. On the other hand, less memory is used, because there is no data replication. After analysing these two approaches, we have decided to choose the second option, since it is simpler to implement and to maintain.

1.2 Code Challenges

When working on the first topic we had several pages and some code section that would be repeated at different places on our website. So we were looking for a javascript framework that would suit our needs. In the JS jungle, we have found Preact (`https://preactjs.com/`), a reduced and lighter version of React.js. It had enough features for our intended use. After we have changed of the topic of our project, we kept the same website and ended up with a nearly single page website that would not require a complete framework to operate. Two HTML pages would have been enough and less complicated. However, we did not change the framework to get some experience.

The code is organized as follows: the *Home* and *About* pages reside in the *./src/routes/* module. Every visualisation is implemented in it's own component in *./src/components/* module. The data handling is done by the *utils.js* from *./src/data/* module.

The challenge with Preact is to properly handle the state and props passed between each component (which are the *home* and *about* pages and the each visualization). These mechanisms enable the interactions between the various plots. For example, when the user selects the year, the map and the bubble plot is updated to display the information related to the newly selected year.

The code was hosted on Github using a different branch for each new visualization. When a visualization was ready, the corresponding branch was merged into the main branch and a new build of the website was performed. We build using Github Actions[1] and the output of the build was automatically pushed to another branch, to be served with Github Pages.

1.3 Visualisation Challenges

The Map and line chart are implemented with the amCharts library while the bubble plot was realized with D3.
An issue we had was to reach the desired result using the libraries API that are sometimes not detailed enough for specific issues. This leads to spending a lot of time in frustration to get something working (e.g. the axis label in the bubble chart, section 2.3).

Several more specific issues we have encountered are the following:

- When the heat map was first displayed, all the countries were appearing pale for an unknown reason and it was difficult to distinguish between the countries. The heat rule used to pick the color corresponding departures per capita ratio is supposed to interpolate the color between the minimum and maximum values of the ratio. All the country ratios seemed to be between 0 and 2. Later, the answer to this behaviour was found: Hong Kong with it's departure per capita ratio greater than 10. This region was too small so it was not visible on the map without zooming. We have still decided to interpolate the color gradient between 0 and 2, despite the outlier.
- Line chart: the chart becomes too heavy and slow if all the countries are displayed as lines (see Figure 1 below). Tradeoff: only display the countries from a continent the selected country belongs to the same continent.

2 Before & Now

2.1 Map

The sketch of the map (see Figure 2 (a)) is very similar to the actual plot (Figure 2 (b)). The color gradient corresponds to the departures per capita ratio for all the

[1] Thanks to this guide (link).

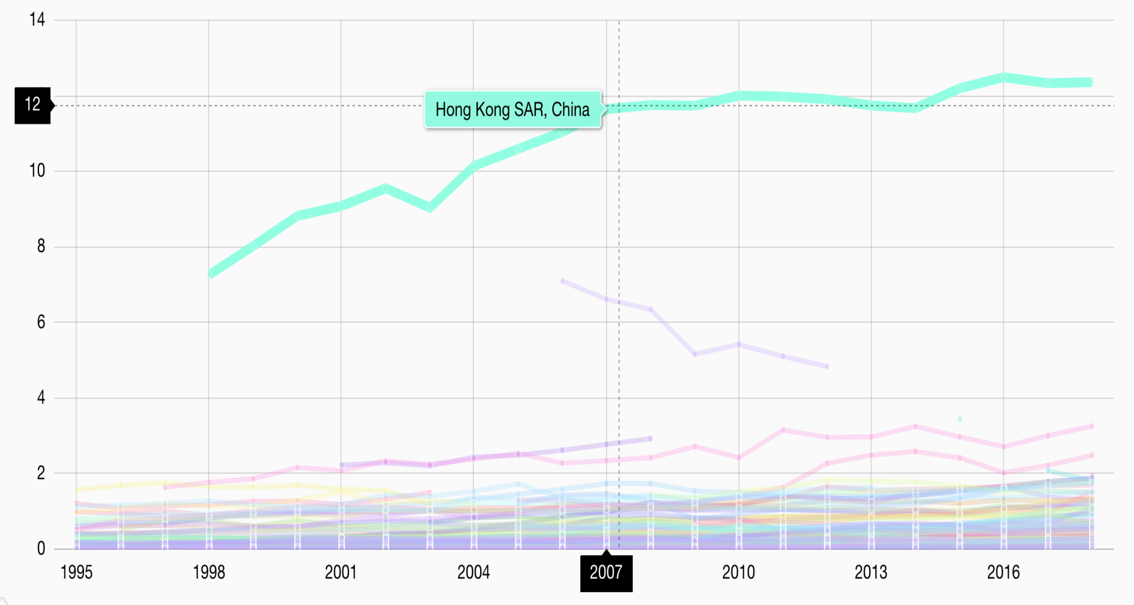


Figure 1: Linechart

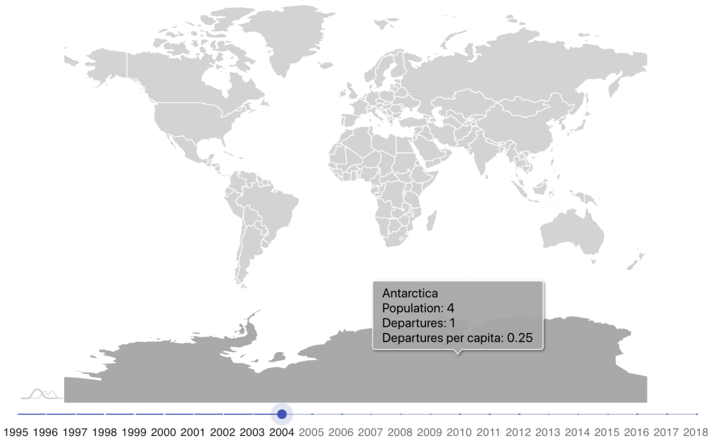
countries. If the data for a given country is missing, this country appears light gray on the map.

Below are the interactions possible with the world map:

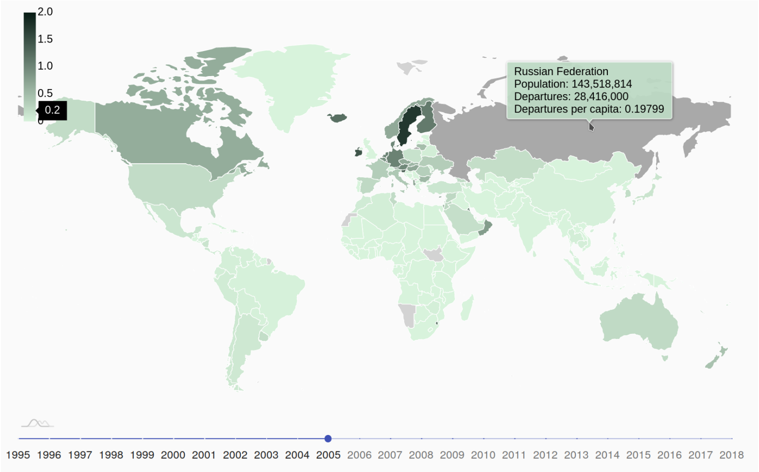
- The year can be selected using the slider on the bottom of the map
- When the mouse passes by the country, it appears in dark grey and the tooltip displayng the population, the number of departures and the departures per capita pops up
- The user can zoom the map to see information about small countries or islands

2.2 Line chart

This plot displays how the departures per capita ratio evaluates year by year. We initially planned to display a single line chart per country (see Figure 1 (a)), but then we realised that it would be interesting to compare how the ratio changes between countries. Since the line chart becomes too complex and slow when all countries are plotted, we have decided to plot the subset of countries. Therefore, once a country is selected by the user on a map plot, all the countries from the same continent are displayed on the line chart. This results in four different plots: one for Africa (Figure 1 (b))one for Asia (Figure 1 (c)), one for Europe (Figure 1 (d)) and one for Oceania (Figure 1 (c)).



(a) world map sketch



(b) world map - actual plot

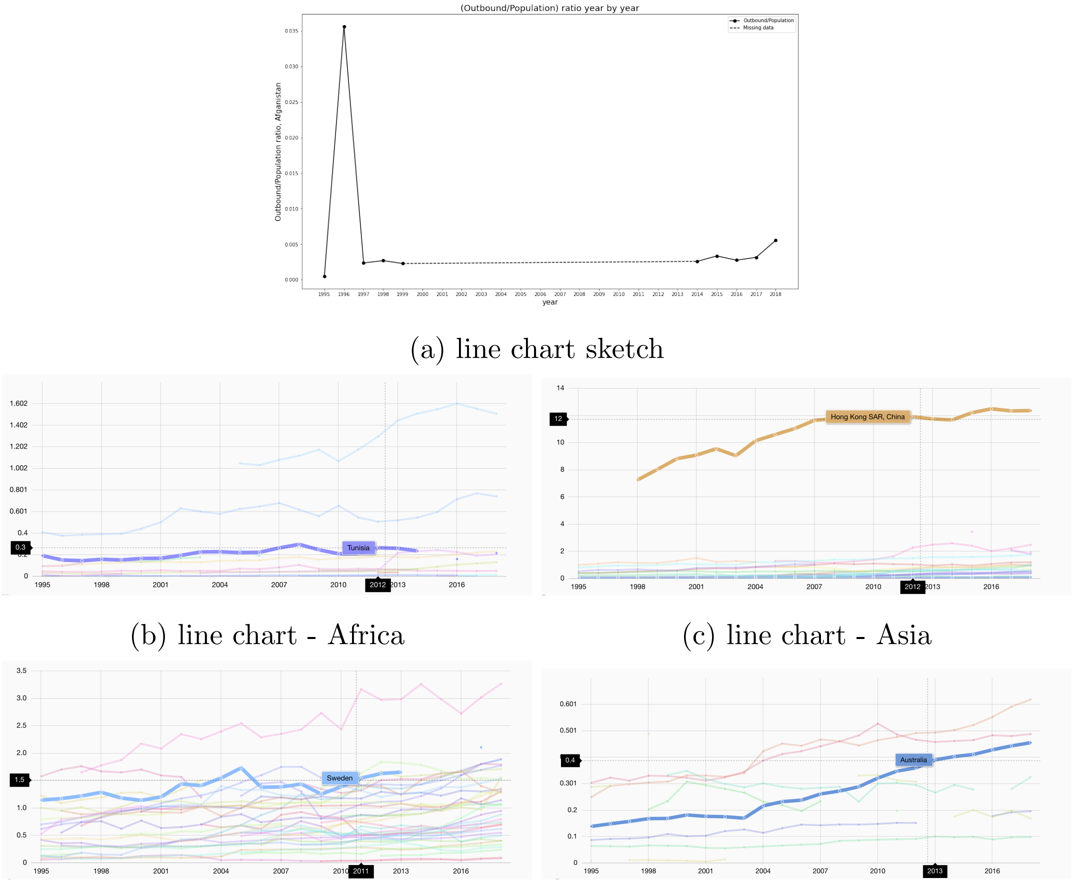
Figure 2: Maps: sketch and actual plot

Notes:

- These four regions are not officially called continents, but se call them so for the sake of simplicity
- Antarctica was excluded from the plot and all datasets

To make the plot more comprehensive, some interactions were implemented:

- The line corresponding to the selected country appears thin and is brought to the front
- The tooltip displaying the country name and the values appears when the mouse passes by the given line



(a) line chart sketch

(b) line chart - Africa

(c) line chart - Asia

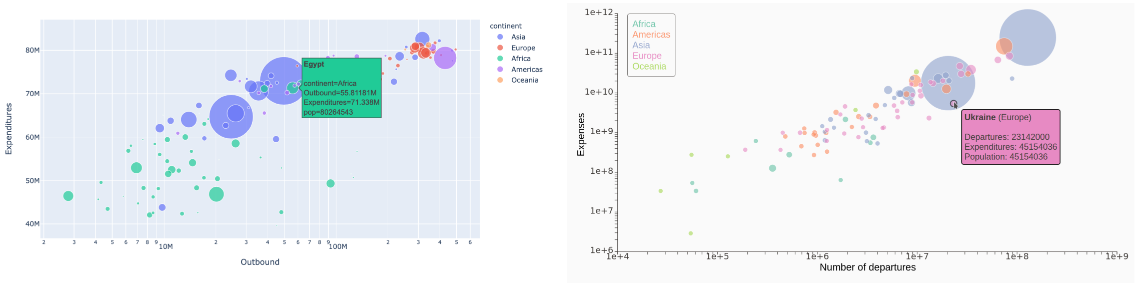
(d) line chart - Europe

(e) line chart - Oceania

Figure 3: Line chart: sketch and actual plot

2.3 Bubble chart

The bubble chart was created using the D3.js library. It displays various countries in a plane according the number of departures and expenses for a considered year, selected with the slider of the Map described in section 2.1. The size of the bubble is proportional to the population for the given year. The color allows to group countries from the same continent altogether. The circle are represented on a logarithmic scale due to the large values of departures and expenses as well as the large differences between the countries. The bubbles were sorted from the largest population to the smallest one before displaying to that the countries with large populations, corresponding to large bubble do not hide less populated countries or prevent the user to access the related tooltip. On the interaction side, a tooltip is displayed when hovering the bubbles to display the country name and continent as well as the values that characterize this bubble: expenditures, outbound and population. If another year is selected on the slider under the Map, this chart will reload and display the corresponding data.



(a) Bubble plot sketch

(b) Actual bubble plot

Figure 4: Bubble plot: sketch and actual plot

On the figure 4, we can notice that the result is quite similar to the sketch except for the axis' values that we had trouble to display in a non-scientific way. We would have liked to have it in powers of 10.

3 Peer Assignment

Here is a breakdown of the work we did for this project:

- Alexandra did
  - the sketches
  - the data pre-processing and the generation of the CSV files
  - the map and line-chart
- Julien did
  - the setup the website, the deployment with Github Actions and hosting on Github Pages
  - the bubble plot
- They both together
  - wrote this ProcessBook
  - recorded the screencast
  - found and patched errors in the website and in the visualizations