Report Programmeer project

For the endproject, I have made a visualization of obesity related data using a world map, a barchart and a scatterplot. The data is visible for several years ranging from 1999 to 2014 for the data map and barchart. The scatterplot is a fixed for the year 2010. The datamap maps the mean BMI per country, the bargraph shows the ratio of obese per country and scatterplot shows optional categories, containing food supply per person per day in kcal, percentage of the population that performs insufficient amounts of physical activity, the ratio of the population that is obese and the ratio of the population that is overweight against the mean BMI.

The code is separated in several files. The front end file is the html file in which all the elements come together. The html file loads the datamap, the d3 library and the JavaScript files. Next to separate sections (<div>) that are made in which texts and visualizations are placed, also interactive elements are added like a slider and dropdown menus to let the user pick between several datasets.

Three visualization are created and each has its own JavaScript file. This maintains the overview on how the visualizations are created and makes it easier to edit them. The files are named after their content.

All the visualizations are based on data from the same data file. The json file “obesity\_data .json” has been created in which several datasets have been processed. The Json file is structured in the following order:

Year --> sex --> list of countries containing country information.

The barchart and scatterplot files (obesity\_barchart.js and obesity\_scatterplot respectively) are quite similar to eachother. Both of them start off with setting the borders for the plot. After this the data is loaded, the categories are retrieved from the interactive elements, such as sex and year. The interactive elements contain an EventListener, which are able to update the plots through an update function.

These update functions start with clearing the plot, then they determine what the input values are (again; year, sex and categories) and start drawing the plots, starting off with their axis, followed by the titles and finishing with creating the bars and dots including their interactivity. Almost the whole file is included into a “draw” function, which divides the two script strictly apart. Only the interactivity is described in functions at the end of the files outside of the “draw” function, which makes them reachable for other files to use.

The data map file (obesity\_map.js) start of with a dictionary containing BMI categories with their linked colours. Next, the data is being loaded and the the selected categories from the interactive elements are retrieved, after which the title of the map is created. The interactive elements, on change, call a function that updates the map. Before the datamap is created, the data is taken from the json file and based on the selected categories, a new dataset is created. The datamap is created, containing its characteristics and a “done” option in which all the interactivity is being processed. The legenda is created and the update map function is written here.

After the update function, functions are written outside of the data loading and are callable by other files as well. Highlighting a country is included to these function, allowing other elements than the datamap ones to highlight countries in the map.

DESCRIBE CHALLENGES IN THE PROCESS

Towards the technical side of making the website, the thing that was very time consuming was getting the data in the right json format. Althought I already had some experience with creating a json file, it was a first timer to retrieve the data from multiple files. The most relieving solution was to find out that a dictionary can be directly converted to a json file and this resulted in the best json file I have worked with yet.

The design process went quite fluently since I didn’t set the bar too high due to time pressure.

The initial plan was to make a world make, a stacking barchart and a scatterplot. The world map would indicate the mean BMI’s per country and the stacking barchart would emphasize where the mean of these values laid and how this mean shifted upwards over the years in time. The scatterplot would give additional information on categories that are related to BMI; the amount of the population that had an insufficient physical activity and the the food supply per person per day in kcal for every country. These categories would be plotted against the mean BMI per country, to show their relationship on the national scale.

At first I wanted the countries in the worldmap to change colour on highlight. For the worldmap interactivity this didn’t cause any trouble, but in a linked interactivity, the highlight induced from another chart would overwrite the original colour of the country and did not change back. After some trial and error, I decided to no longer highlight a country by changing its colour, but by decreasing the opacity value of all other countries but the selected country and increase the border thickness of the selected country. In this way the country is also strongly highlighted, overwriting bugs did no longer occur and after the first group evaluation on Friday, it became clear that this might be the better way to highlight since the colour is preserved during highlighting.

Another strong change in design was the eventual choice of a normal bargraph instead of the stacking bargraph. This choice was made on the lack of good examples of stacking bargraphs and time pressure. To maintain consistency, the bargraph is coloured with the same colours as the worldmap, slightly reflecting the effect of the stacking barchart.

To the scatterplot I added 2 more categories to play around with. At first I used the other categories just to check the scatterplot since the data about the physical activity wasn’t in the right format yet, but since the added categories were visually interesting and not a total nonsense visualization, I left them in the scatterplot.

DESCRIBE WHAT I WOULD DO BETTER!