Assignment 2 – Graphics Fundamentals & Visual Encoding Channels

Module: CS7DS4/CSU44056 Data Visualization

Student Name : Indrajeet Kumar
Student Number : 23345983

Part A : Graphics Fundamentals - Minard's Map Of Napoleon's Russia Campaign

The Minard dataset was imported from a CSV file using the Python pandas library, and it was stored in a DataFrame called 'data' with the following code:
data = pd.read_csv('minard.csv')

After importing the dataset, we then prepared the data by dividing into segments based on by dividing into segments based on the direction (advance or retreat) and divisions of the French army (1,2,3). These segments are used to create different elements of the Minard's map visualization such as city markers, campaign events and temperature graph.

Once the data is prepared , we created a scatter plot using plotly library which represents the cities that the French army passed through during Napoleon's Russia campaign. In this scatter plot, each city represented by a black dot , which allowing them to be clearly distinguished from other map elements.

After that , we initially divided the datasets into different subsets based on the direction of movement : "Advance" and "Retreat". We created custom function 'create_trace()' which is used to create traces for campaign events. For every segment, such as "RETREAT Division 2" or "ADVANCE Division 1," the **create_trace()** method is invoked. These calls produce unique traces that record the particular events in each section. The **fig.append_trace()** function is used to add the traces—which stand in for the campaign events—to the map. This method creates a visual representation of the campaign's development by superimposing the lines on top of the city scatter plot. Distinct colors and lines width are used to differentiate between "Advance" and "Retreat" events. Advance events are represented with orange lines with an line width of 15 and on the other hand Retreat events are represented with blue lines and a line width of 12.

We created a separate plot to visualize the temperature data during the campaign. This graph illustrates the fluctuating temperatures experienced by the French army as they advanced and retreated. Temperature data is represented using square markers, and each marker displays the temperature value and date.

The temperature graph and the main map are arranged in two rows of the code, which makes use of Plotly to create subplots. Plot and paper background colours are chosen to produce a striking contrast. The resulting plot provides an interactive visualization of Minard's map which is displayed in Figure 1.

Figure 1 : Visualization of Napoleon's Russia Campaign using Minard's Map

Part B : Visual Encoding Channels

1. Visualize the evolution of life expectancy over the years (1957 – 2007)

In the 1st part, We created three different scatterplot visualizations to explore changes in the life expectancy across continents from 1957 to 2007.

In the first scatterplot, we use color encoding to represent continents. We added a small amount of random noise to the **'year'** values to prevent overlapping data points along the x-axis. Each continent is identified by a distinct color, making it easy to distinguish their life expectancy trends. Figure 2 illustrates the life expectancy trends spanning the years from 1957 to 2007, with color encoding used to convey the information.

Figure 2: Life Expectancy over the years (1957-2007) – Color Encoding

In the second scatterplot, we use various shapes (such as circles, squares, diamonds, triangles, pentagons, and crosses) to differentiate continents. And random noise to the 'year' values to prevent overlap which help us compare the life expectancy trends by shape. Figure 3 illustrates the life expectancy trends spanning the years from 1957 to 2007, with shape encoding used to convey the information.

The third scatterplot employs size encoding to represent the continents. Each data point is sized according to its corresponding continent, and the 'sizes' parameter is used to control the range of sizes. Random noise is added to the 'year' values. Figure 4 illustrates the life expectancy trends spanning the years from 1957 to 2007, with size encoding used to convey the information.

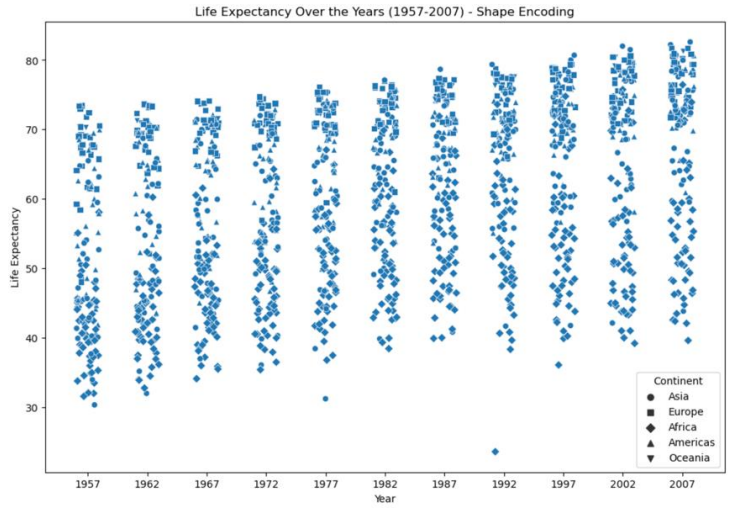

Figure 3 : Life Expectancy over the years (1957-2007) – Shape Encoding

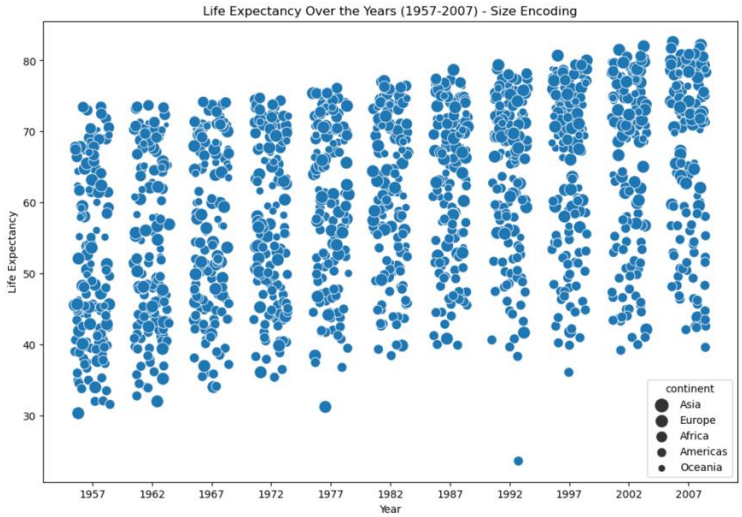

Figure 4 : Life Expectancy Over the years (1957-2007) – Size Encoding

2. Visualize the correlation between wealth and health

In the 2nd part, We visualize the correlation between wealth (GDP) and health (life life expectancy) while encoding the quantitative attribute population (pop) using three different encoding channels (Brightness , Position and Texture encoding).

In the first variant, we encoded population using brightness. The 'hue' parameter is used to represent population, with the darker color indicating higher population and lighter color indicating lower population. Figure 5 illustrates Correlation Between GDP, Life Expectancy, and Population, with brightness encoding to convey the information.

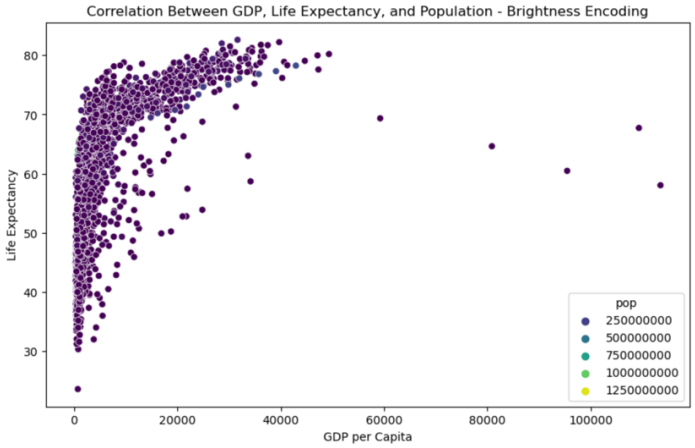

Figure 5 : Life Expectancy Vs GDP Per Capita – Brightness Encoding

In the second variant, we encoded population using position. Data points are positioned on the chart based on their population, allowing to see how GDP and life expectancy relate to the size of the population. Higher positions on the y-axis represent larger populations. Figure 5 illustrates Correlation Between GDP, Life Expectancy, and Population, with position encoding to convey the information.

Figure 6 : Life Expectancy Vs GDP Per Capita – Position Encoding

In the third variant, we use texture encoding to represent population. Different textures, such as slashes, backslashes, vertical bars, hyphens, and plus signs, are used to indicate different population levels. Each texture corresponds to a specific range of population, offering a unique way to visualize the relationship between GDP, life expectancy, and population which is displayed in figure 7.

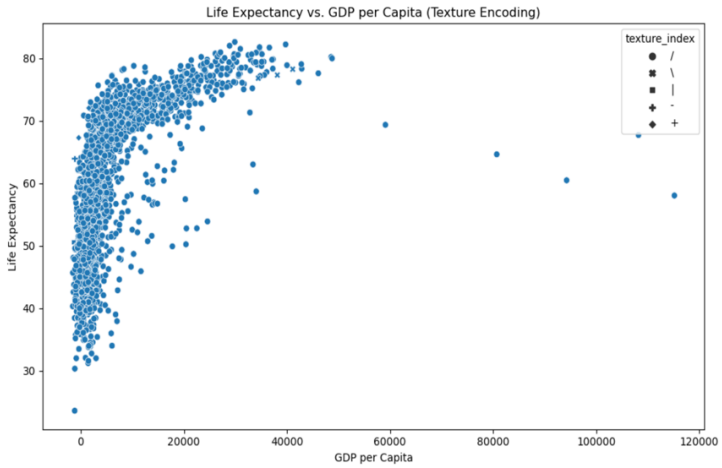

Figure 7 : Life Expectancy vs GDP Per Capita – Texture Encoding

3. Freeform exploratory visualization:

In the 3rd part, we created a created a scatterplot using the given dataset with multiple attributes Life Expectancy, Year, Continent, Population, and GDP per Capita. It utilizes the 'matplotlib' library to create the plot.

We defined two dictionaries 'colors' and 'marker_shapes' to assign colors and marker shapes to different categories. For example it assigns red to Africa, blue to Asia, and uses shapes like circles, squares, and triangles to represent different GDP categories The code also scales the 'pop' (population) attribute to adjust the size of markers using the minimum and maximum values in the dataset. This ensures that the marker size represents population appropriately. We divided the GDP per Capita into 'Low,' 'Medium,' and 'High' categories using the pd.cut function from the pandas library. After that we iterated through the dataset using a for loop, applying a small random offset to the 'year' values (jittering) for better visualization. It then plots a scatter point for each data point. The color, marker shape, and marker size are determined by the attributes 'continent' and 'gdp_category,' creating a visual encoding of the data. And also Custom legend elements are also added to indicate the colors and shapes for continents and GDP categories. Finally, the plt.show() function displays the scatterplot with all the specified elements and annotations. Figure 8 illustrates the visual analysis visual analysis of multiple attributes, including Life Expectancy, Year, Continent, Population, and GDP per Capita.

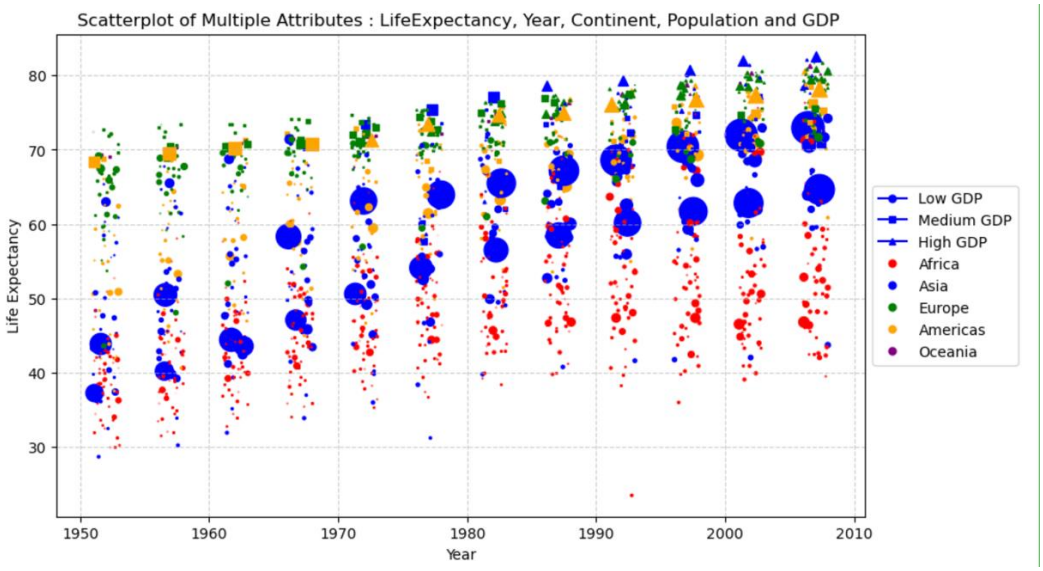

Figure 8 : Visual Analysis of multiple attributes including Life Expectancy, Year, Continent, Population, and GDP per Capita.