Declaration for both Assignment 2(a) & 2(b)

I have read and I understand the plagiarism provisions in the General Regulations of the University Calendar for the current year, found at `http://www.tcd.ie/calendar`. I have also completed the Online Tutorial on avoiding plagiarism 'Ready Steady Write', located at `http://tcd-ie.libguides.com/plagiarism/ready-steady-write`.

Tools used for all visualisations

To create all visualisations, I used Python 3.9 along with the `Vega-Altair` library from PyPI. To process the data given to us, I used the `Pandas` and `NumPy` libraries from PyPI.

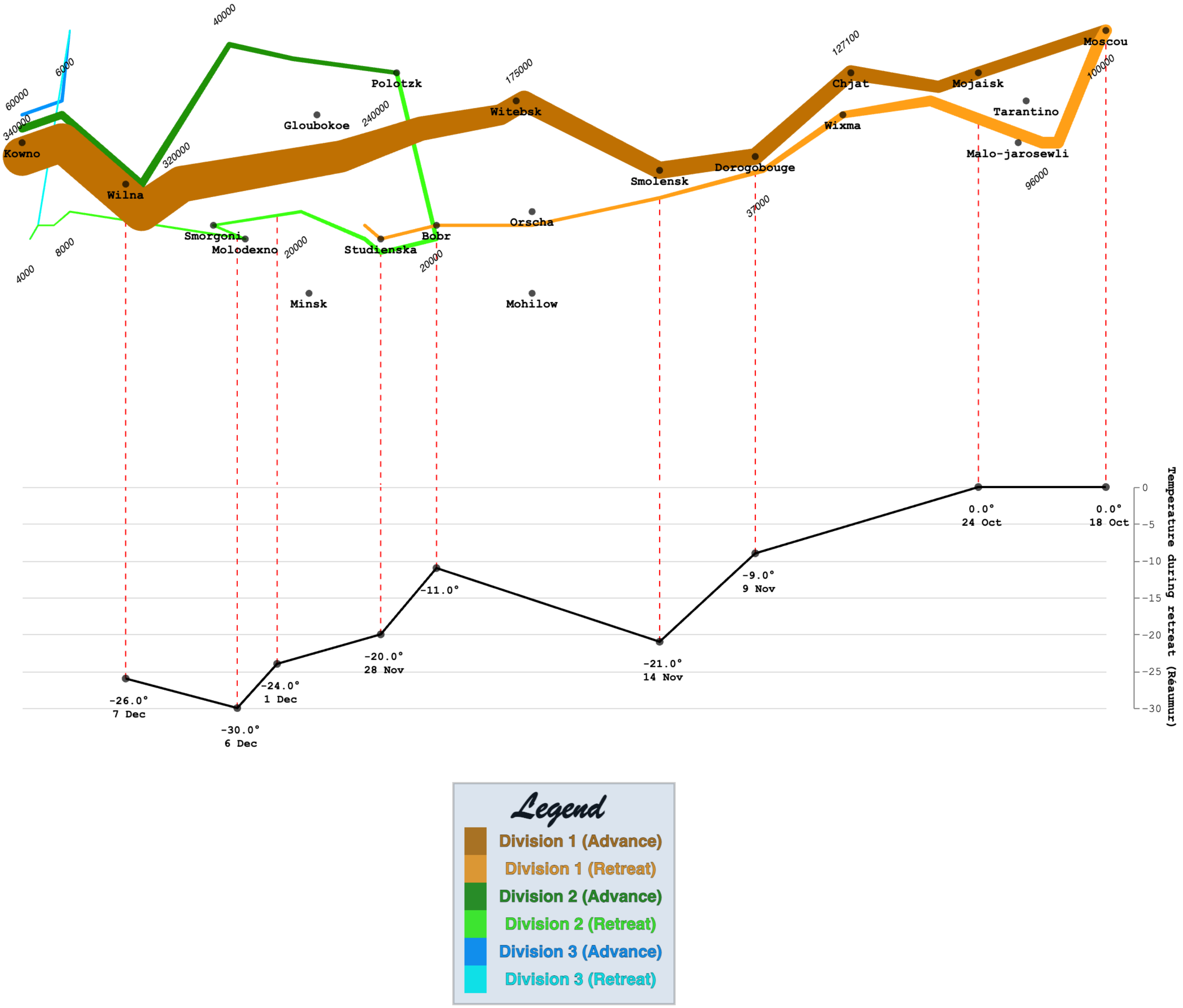
How I implemented the visualisation

Firstly, I parsed the data from my Napoleon-Russian-Campaign.xlsx file containing data given to us by Professor John Dingliana. I used this data to create a chart showing the relative size of the army along their trail with the `mark_trail()` function in Vega-Altair with the size parameter as the ”SURV” column in the dataset and ”LONP” and ”LATP” columns for the x and y axis. I set color conditions for each trail; division 1 was brown for the approach and orange for the retreat, division 2 was dark green for the approach and light green for the retreat, and division 3 was blue for the approach and light blue for the retreat. After experimenting with various colors and ways of depicting the path of the army, I chose this because it was easy to understand for the reader and showed the distinct turning points for each army division. Furthermore, I made a chart showing the position of cities with text beside them using the `mark_circle()` and `mark_text()` functions. I also used the `mark_text()` function to add survivor counts using data from every third row along the trail so that the reader can see how the army decreased in size over time. I combined these individual charts using the ”+” operator, and then created a temperature plot to show the temperature during retreat using `mark_circle()` to create the points, `mark_line()` to connect the points, and `mark_text()` to show the temperature/date when each datapoint was recorded; I used the ”LONT” and ”TEMP” columns for the x and y axis. I added red dotted lines from each temperature recording to the latitude and longitude in the retreat above it with the `mark_rule()` function. Once the survivor chart and the temperature chart were made, I created a legend using the `mark_rect()` function and combined these three charts using Vega-Altair's `vconcat()` function which vertically concatenates the charts one below another.

Specific features of relevance

Although I wanted to use Minard's general style for my visualisation, I also wanted to improve upon his work. I felt that the lack of color in his visualisation made it dull and the black lines connecting the temperature points to points on the survivor chart made it unappealing. Therefore, I made sure to add different colors for each division and direction of the army along with red dotted lines to show the connection between the temperatures and the points where they were recorded in the retreat. I ensured this didn't confuse the reader by adding a legend to show what each color meant. Furthermore, I made sure to choose darker colors for the approach and lighter colors for the retreat to mimic the tension and status of the army along their trail.

Prathamesh's recreation of Minard's visualisation of Napolean's Russian Campaign



Part 1: Variant 1

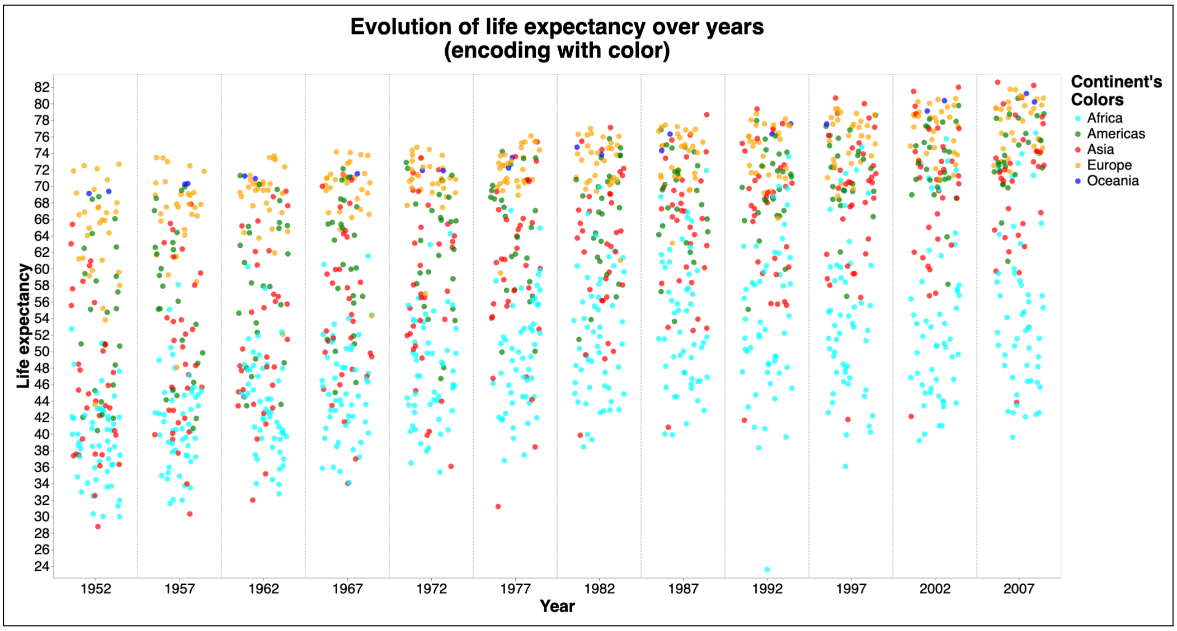


Figure 1: Encoding continent with color as the encoding channel.

I encoded `continent` using color as an encoding channel with the `mark_circle()` function and a custom color scheme for each continent. I chose color as an encoding channel as it is powerful at differentiating between categories in categorical data. Since `year` is in intervals of 5, my initial plot had straight vertical lines at intervals of 5 on the x-axis which made it difficult to comprehend any trends over time. Therefore, I leveraged a visualization technique known as ”jittering” where you add noise to the data to better see the ”cloud” of data points at each interval rather than a straight line. In terms of efficacy, I believe this was my most efficient variant for part one as it is simple yet distinct in the message behind it; it’s easily visible to the reader that life expectancy has been going up for all continents but Europe started the highest.

Part 1: Variant 2

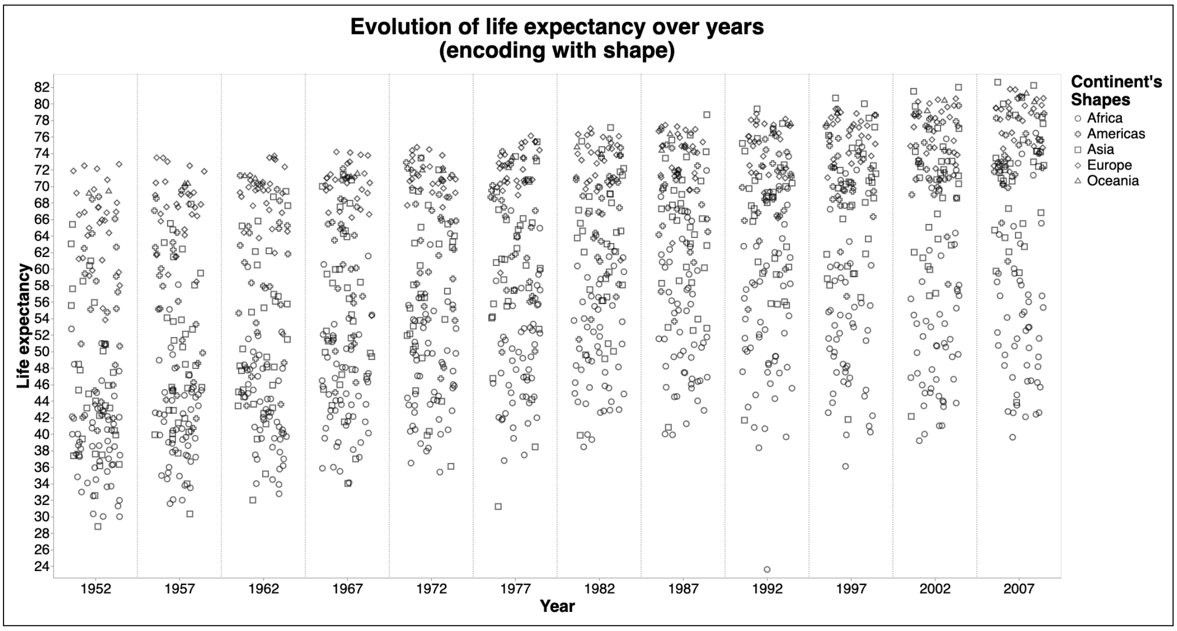


Figure 2: Encoding continent with shape as the encoding channel.

I encoded `continent` using shape as an encoding channel with the `mark_point()` function and a custom shape scheme for each continent. I used shape as it is distinct for categorical data and limitations such as a limited range and ordering do not matter for our small number of continents which makes it effective. The only downside was that densely populated areas caused shapes to overlap which may cause confusion.

Part 1: Variant 3

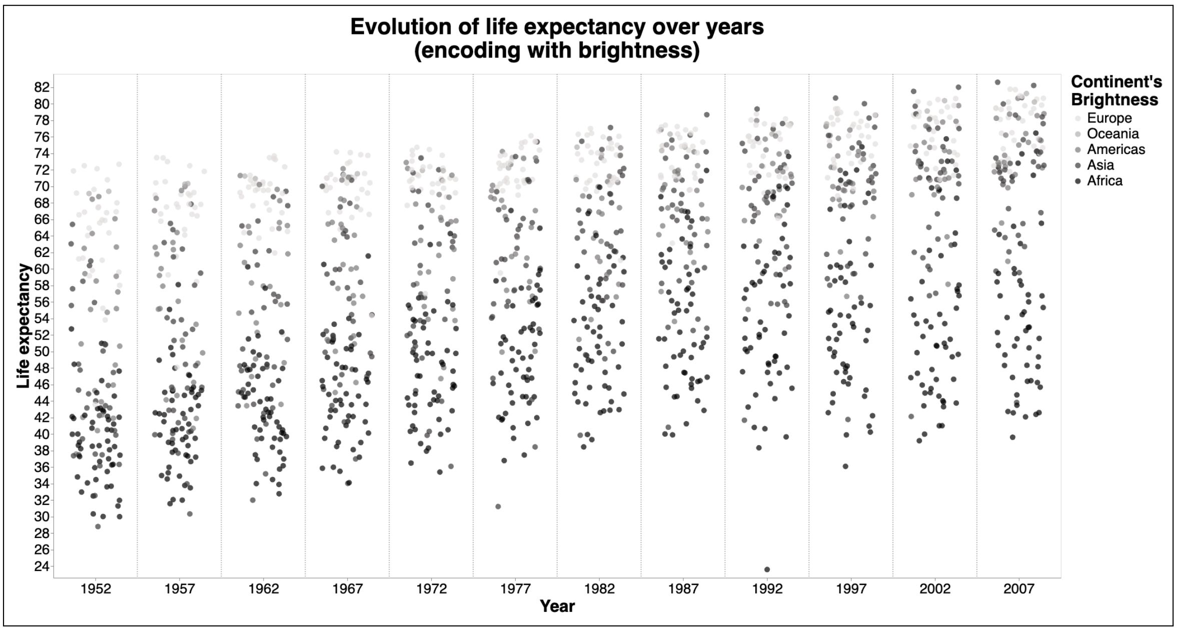


Figure 3: Encoding continent with brightness as the encoding channel.

I encoded `continent` using brightness as an encoding channel with the `mark_circle()` function and a custom brightness scheme for each continent. I used brightness as I was curious how it would cope with a small number of categories such as in `continent`. It is limited for categorical data but with only 5 shades of brightness I felt that it is still somewhat effective but is likely the least effective out of my 3 variants for part 1.

Part 2: Variant 1

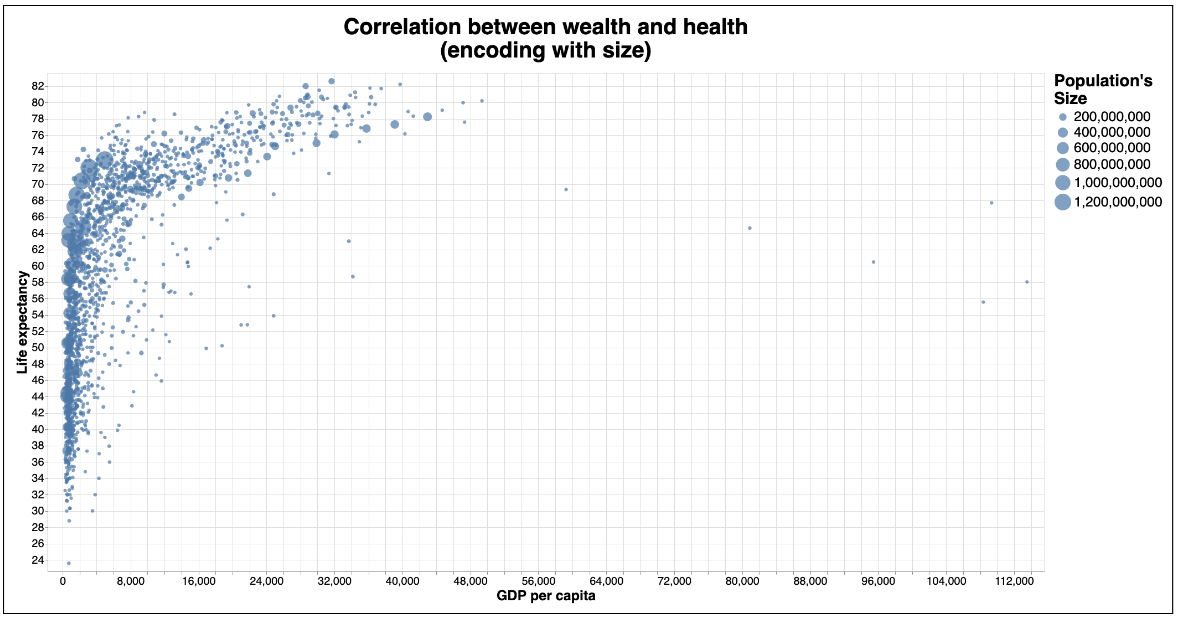


Figure 4: Encoding population with size as the encoding channel.

I encoded `pop` using size as an encoding channel with the `mark_circle()` function and setting the size encoding as the population count. I used size as it vividly shows differences in a quantity (i.e. `pop`) among data points which makes it effective. I believe this was my most efficient visualization for part 2 because of its simplicity; it's clear to see that as GDP per capita increases, life expectancy goes up but population goes down.

Part 2: Variant 2

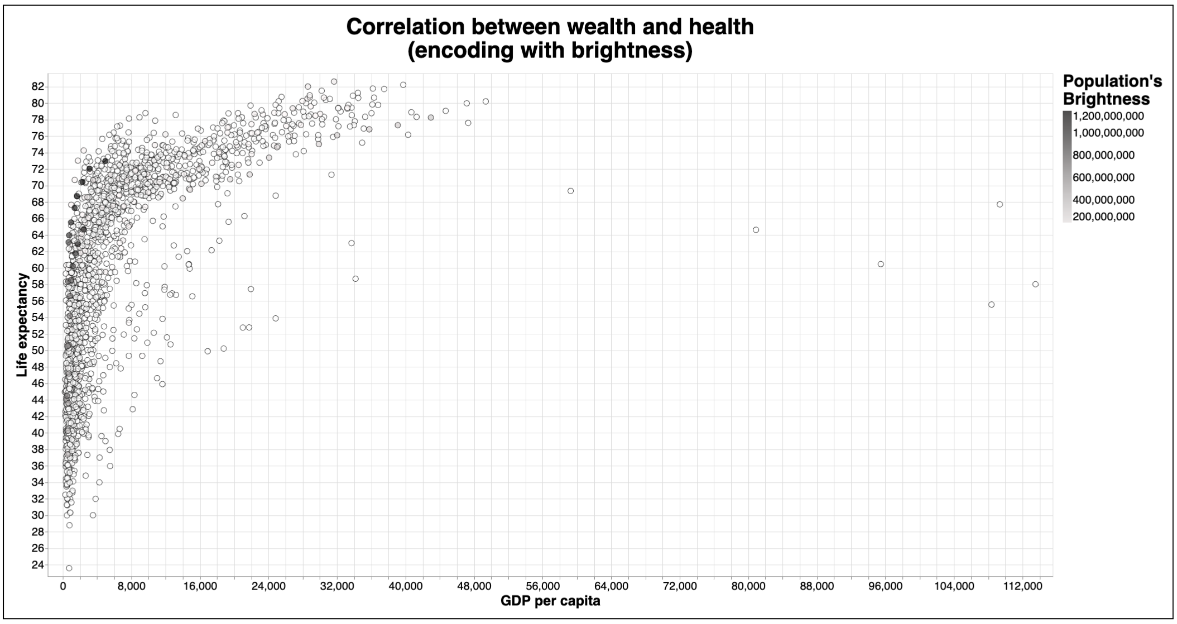


Figure 5: Encoding population with brightness as the encoding channel.

I encoded `pop` using brightness as an encoding channel with the `mark_circle()` function and a custom brightness scheme for different ranges of population. I used brightness since the main objective of encoding population is to show trends and not exact values. This visualization is effective since it's easy to decipher that a higher life expectancy tends to a higher population (darker points until mid 70's) and a higher GDP per capita.

Part 2: Variant 3

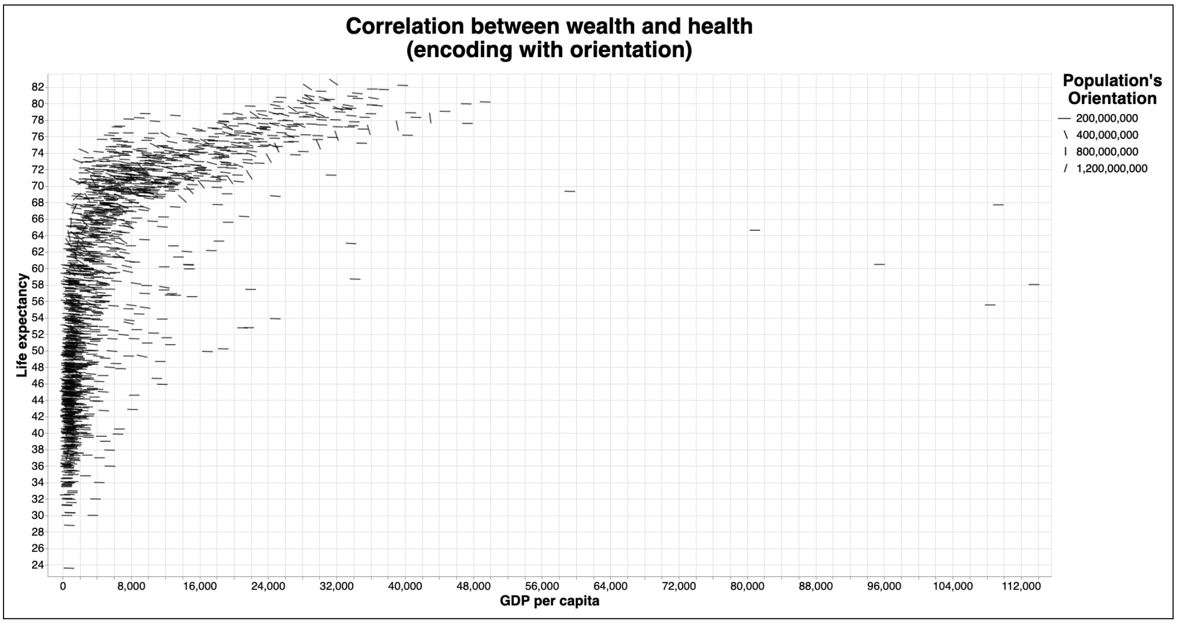


Figure 6: Encoding population with orientation as the encoding channel.

I encoded pop using orientation as an encoding channel with the mark_point() function and setting the angle based on the size of the population. I used orientation for the encoding channel because the population does not need to be known exactly for each data point. I divided the range of angles into 4 intervals as any more granularity would be harder to perceive. I learned that this encoding channel is challenging to use reliably for portraying a message with a lot of data, so I would consider this my least effective visualization in part 2. I learned that it is difficult to show trends with orientation as it takes more time for a reader to understand what is going on. When there are other encoding channels that display a trend with a quicker perception time such as size, it makes it difficult to justify this encoding channel for a lot of cases.

Part 3

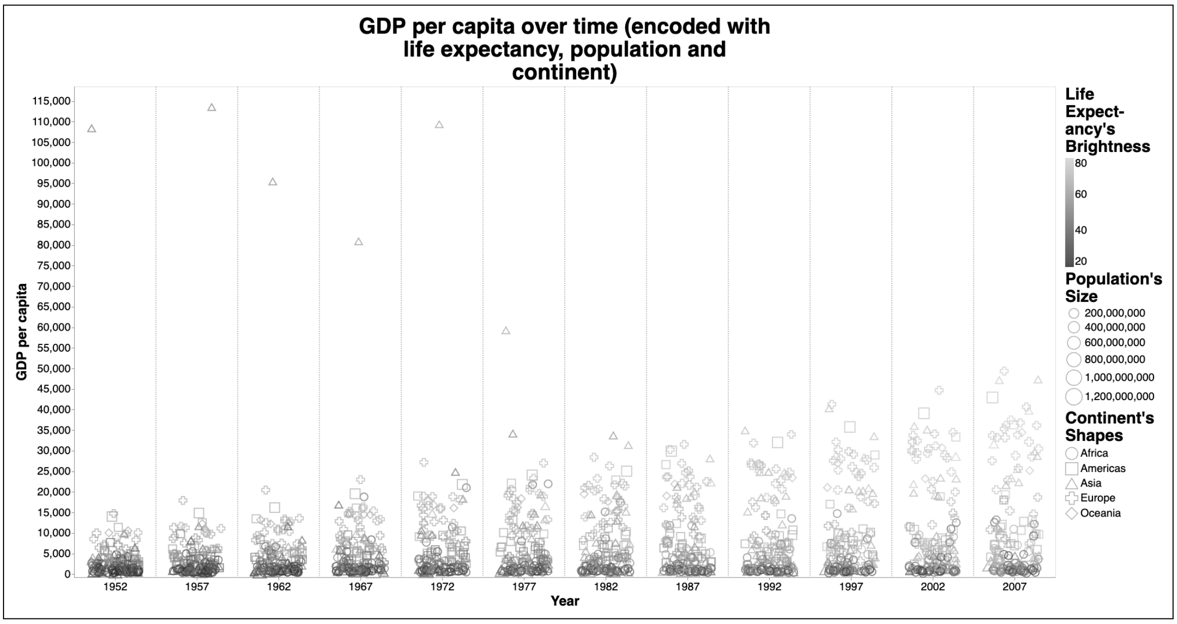


Figure 7: Encoding GDP per capita, year, life expectancy, population size and continent.

I encoded gdpPercap and year on the x-axis and y-axis with position as the encoding channel. lifeExp is encoded through brightness, population is encoded through size and continent is encoded through shape which were all done through the same methods mentioned previously. In terms of efficacy, I think this visualization is effective because it is easy to see the trend of GDP per capita going up over time for all countries (as the data points go up along the y-axis over time). The beauty about this visualization is that if a user wishes to, they will find more information the more they analyze the visualization. For example, after further investigation we can see that life expectancy is getting longer over time (more grey compared to black previously). Also we can see that countries with a bigger population (such as a square representing the ”Americas”) have consistent increases in GDP per capita over time in an almost linear fashion. This complexity hidden under a simple visualization makes it effective at portraying multiple stories with various underlying meanings.