

**First and last name:** Salma Chana

## 1. Introduction:

### 1.1 Dataset Description :

The dataset used in this project presents all Nobel Prize winners from the creation of the prize in 1901 to 2016. It can be found in the following source:

<https://data.world/sya/nobel-prize-winners>

It is composed of 18 columns and 970 rows of data, but in this project, I only used ten properties:

- **Year:**
- **Category:** the field of the prize
- **Laureate Type:** whether it is individual or shared by an organization.
- **Full name:**
- **Birth Country:**
- **Sex:**
- **Organization name:** (the university) of the winner
- **Organization Country:**
- **Age:** the age at which the winner received the award, I added this property by using the **Birth Date** and the **Year** of the award.

In order to present geographical properties, I used the following dataset:

**ne\_50m\_admin\_0\_countries.geojson**, that can be found here:

<https://github.com/nvkelso/natural-earth-vector/tree/master/geojson>

### 1.2 Purpose of visualizations:

The objective of the visualizations carried out is:

1. Discover the most awarded countries.
2. Compare the number of prizes received between countries by year, category, gender, and type (individual or organizational).
3. Show immigrant winners (born in a country other than the one in which they get the prize).
4. View the evolution of the age of the winners over the years
5. Show the top awarded universities in each country.

## 2. Technical description:

The visualizations presented in this project were realized using **D3.js** and **Vegalite**. The code in **script.js** is organized as follows:

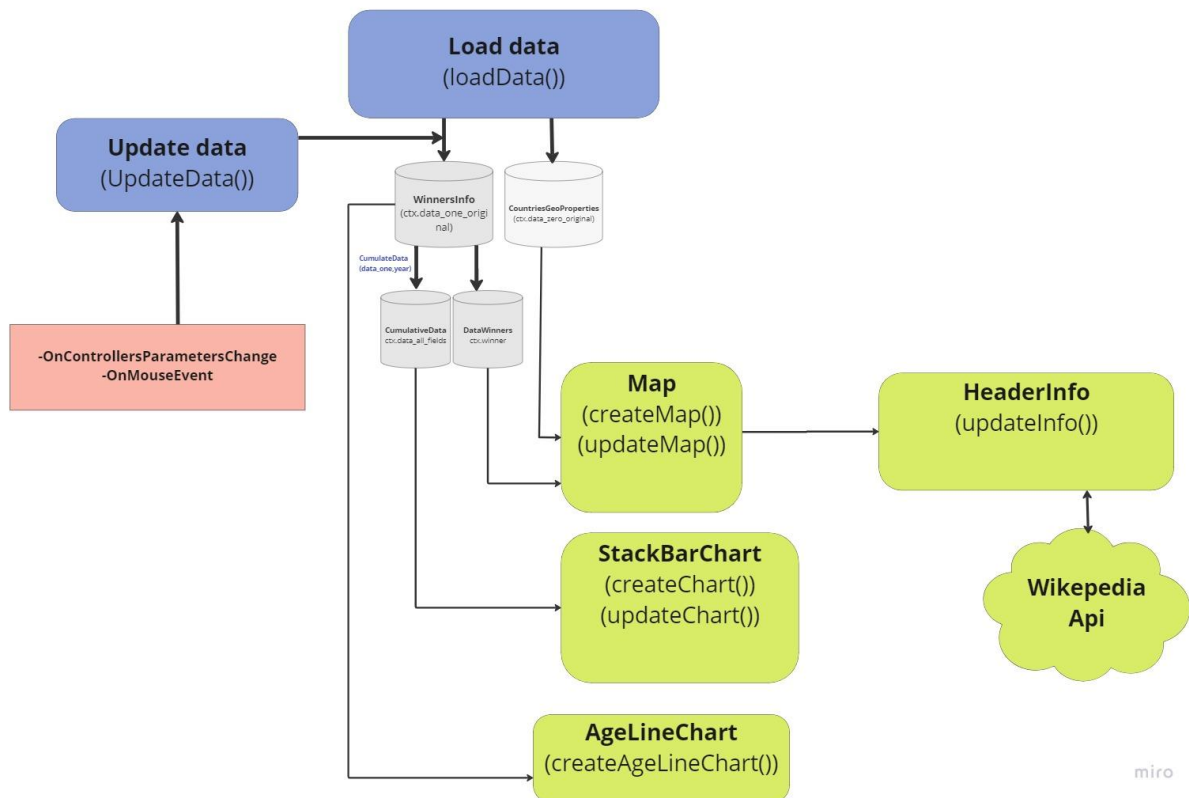


Figure 1: Code organization diagram

- One of the objectives of this project is to compare the number of awards received by a country with other countries. This information can be presented in two ways:

- Either, the number of awards received in **one year** by a country (this information will be displayed on the map).
- Or, the cumulative number of prizes won by a country **from 1901 to the selected year** (this information will be displayed on the StackBar)

- With that said, after loading the data, I extract two specific data structures (one cumulative and one non-cumulative) that will serve as inputs for the charts and the map.

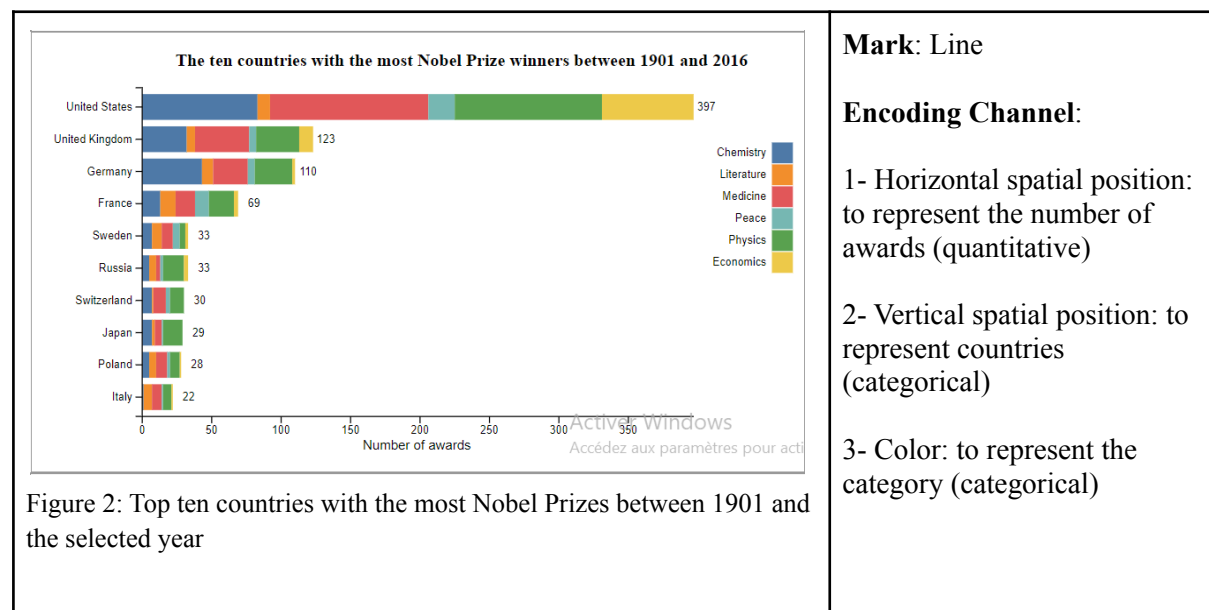
- When an event listener is registered (change in controller settings or mouse event), we call `updateData()` and start over what we did before.

- I also used `WikipediaApi` to get the **URL** of a winner's image and display it.

### 3. Design of the visualization:

#### 3.1 Stack Bar Chart displaying top ten rewarded countries:

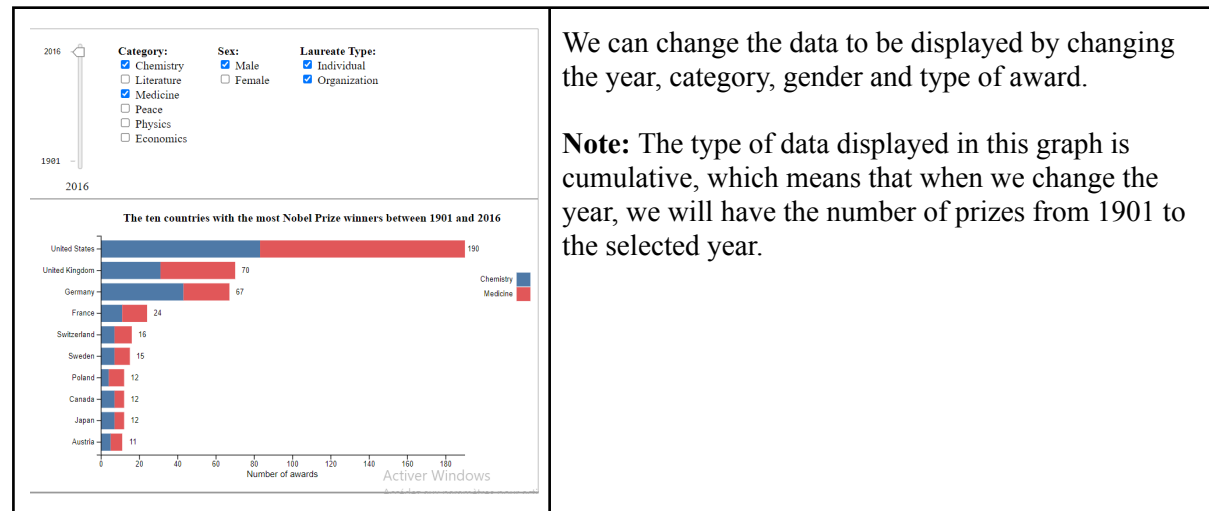
## Marks and encoding channel:



The advantage of a stacked bar chart is that we can compare multiple categorical data, and therefore answer the following questions:

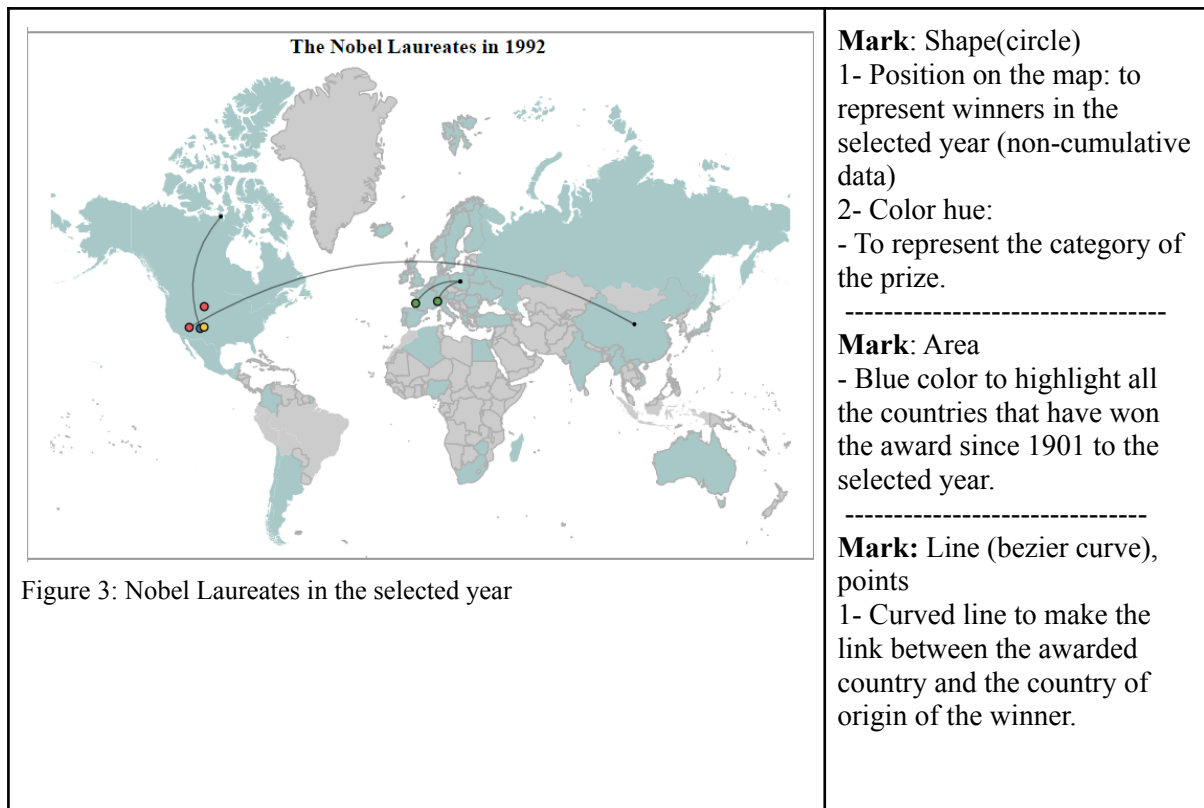
- Which countries receive the most Nobel prizes?
- Which is the most rewarded field in a country?

## Interaction:



## 3.2 Map displaying winners in a single year:

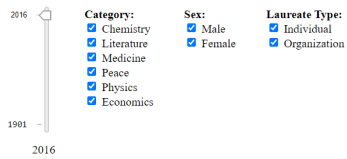
## Marks and encoding channel:



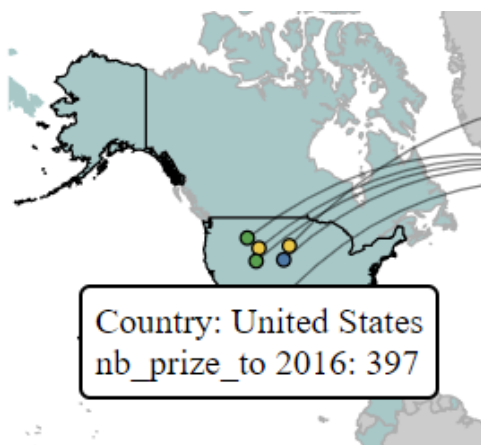
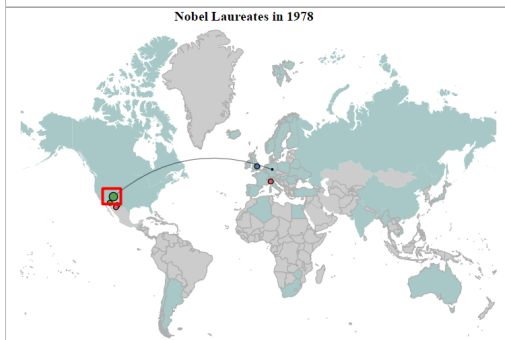
I choose the map because it helps answer the following questions:

- Who are the winners in a given year?
- Which countries are awarded in a given year?
- Which countries have been awarded since 1901 in a given year?
- What is the country of origin of a winner?

**Interaction:**



**Full name:** Arno Allan Penzias  
**Category:** Physics  
**Age:** 45  
**University:** Bell Laboratories  
**Country:** United States  
**Birth Country:** Germany  
**Year:** 1978

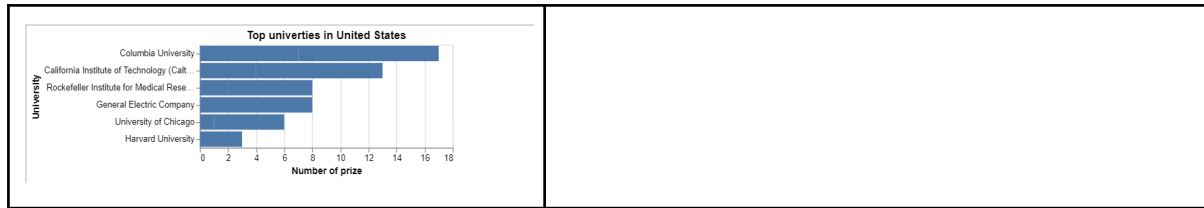


Like the first chart, we can filter the data through the controllers.

By hovering the mouse over a country, you can see the number of awards obtained by this country until the selected year.

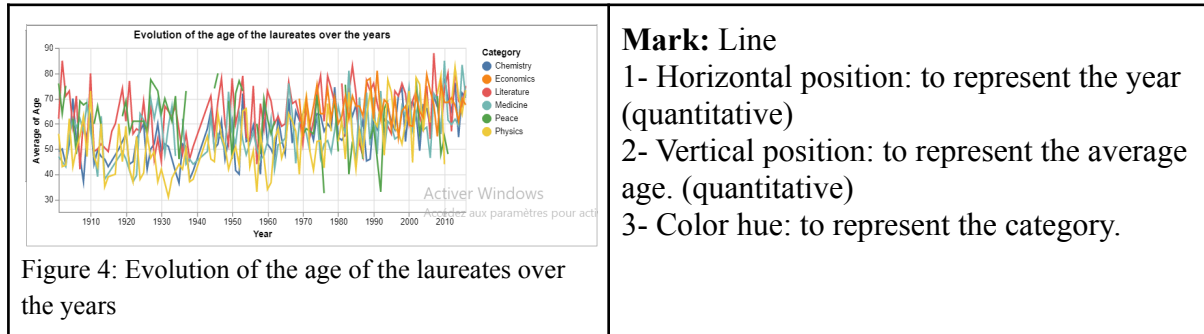
By hovering over a winner (the circle showing the winner), the information header above will update to show the winner's image, name, category, age, university, and country of birth.

By clicking on a country, a bar graph representing the most awarded universities in that country will be displayed



### 3.3 Multiple Line Chart to show the evolution of age:

#### Mark and encoding channel:



I chose a multi-line graph because it is suitable for representing the evolution over the years, so we can answer the following question;

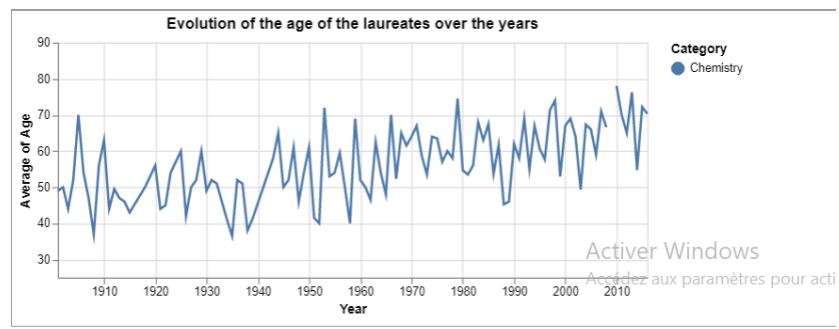
- What is the category with the highest average age? (Literature)
- What is the category with the lowest average age? (Peace)

#### Interaction:

We can choose the category to display from the controllers :

#### Category:

- ☒ Chemistry
- ☐ Literature
- ☐ Medicine
- ☐ Peace
- ☐ Physics
- ☐ Economics



#### Conclusion:

During this project, I learned how to design a visualization that answers certain questions, how to use different encoding channels to make the data easier to read.

The main challenge I faced was to display cumulative and non-cumulative data in an understandable way.

I would also like to mention that the PC tps helped me a lot to progress in my project, especially PC7 and PC9.