JOSIP JURAJ STROSSMAYER UNIVERSITY OF OSIJEK
Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

Project task in the subject
DATA VISUALIZATION

Forest fires in France

Student: Soulaïman Marsou

Mentor: Josip Job

In Osijek, June 2023

CONTENT

1. KV1 - Defining a Project Task

1.1. Project Task

The objective of the project is to analyze the evolution of the occurrence of forest fires in French territories over the past 50 years.

Task name: Forest fire in France

Problem description: Analyze the history of forest fires in France over the past 50 years to understand their occurrence and patterns.

Task description: Create an interactive data visualization dashboard to explore and analyze forest fire data in France.

Objective of the project: Develop insights into the frequency, severity, and causes of forest fires in France using data visualization.

Link to git repository of the project: https://github.com/Zintoulou/data-visualization-forest-fires

1.2. Data

Z-1.2.1. *The dataset for the forest fires will be a csv file provided by a french public organization: https://bdiff.agriculture.gouv.fr/incendies. We will also use one svg maps of the french territory, in which each area is drown as a polygon or path: https://commons.wikimedia.org/wiki/File:Blank_map_of_France,_with_communes_and_departments.svg.*

Z-1.2.2. In the dataset, we can find the 37 columns. We can find for example: the date, the name of the nearest city, the surface of forest burned, and so on. However, not all of them have enough data. The data provided is from 1973 to 2023. In the svg map, we can identify each area by the name of the commune or the department.

1.3. Data processing

Z-1.3.1. *We began by removing all the columns from the forest fire dataset. We finally keep only 11 columns: id, year, department, code of department, commune, date of first alert, surface burned, cause, deaths, damaged building, destroyed building.*

Z-1.3.2. *To ensure precise representation of geographic locations, we meticulously compared the names of communes and departments in the dataset with those in the SVG maps. By cross-checking the names between the dataset and the maps, we created a dependable connection, enabling us to visualize and analyze the forest fire data in relation to specific communes and departments.*

1.4. Relevant display types for the data used

Z-1.4.1. *Several display types would be appropriate for this project to effectively visualize and analyze the data. For example :*

- *We can try to show the temporal evolution of forest fires over the past 50 years to provide a clear representation of how the occurrence of forest fires has changed over time.*
- *We can visualize the distribution or the spatial density of forest fires across different departments or communes in France.*
- *We can display key metrics such as the total number of fires or the surface area affected by fires. It would provide a visual comparison of fire incidents across different categories, such as departments or years.*
- *We can visualize the proportion of forest fires attributed to different causes, such as human activity, lightning, or unknown factors. They help in understanding the relative contributions of various factors to fire occurrences.*

2. KV2 - Data visualization design.

2.1. Questions that visualization answers

Z-2.1.1. The visualization of forest fire data aims to provide answers to several important questions:

- Has the number of forest fires in France increased over the past 50 years?
- Which areas in France have experienced the highest number of forest fires?
- Are there any noticeable patterns in the occurrence of forest fires?
- What are the main causes of forest fires?
- Do forest fires directly harm infrastructure and the population?
- Can we identify areas at higher risk of frequent forest fires?

2.2. Data visualization draft

Z-2.2.1. *We will try to use this kind of chart for our visualization. We can represent different useful information with the same chart.*

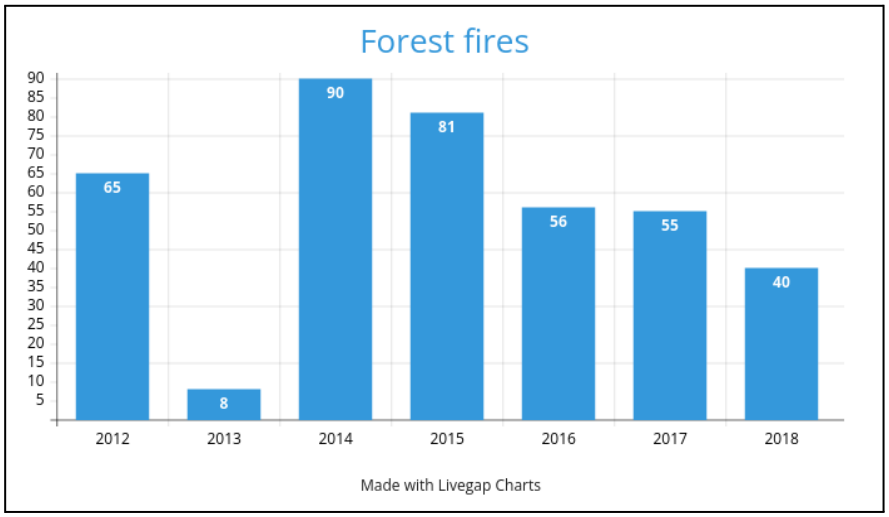


Fig: Bar chart to represent evolution of forest fires over the time.

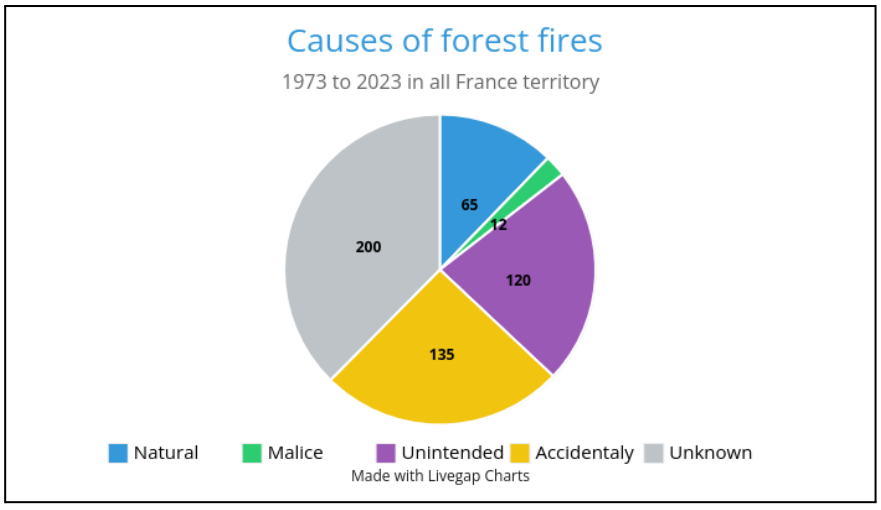


Fig: Pie chart which shows the proportion of fires attributed to different causes.

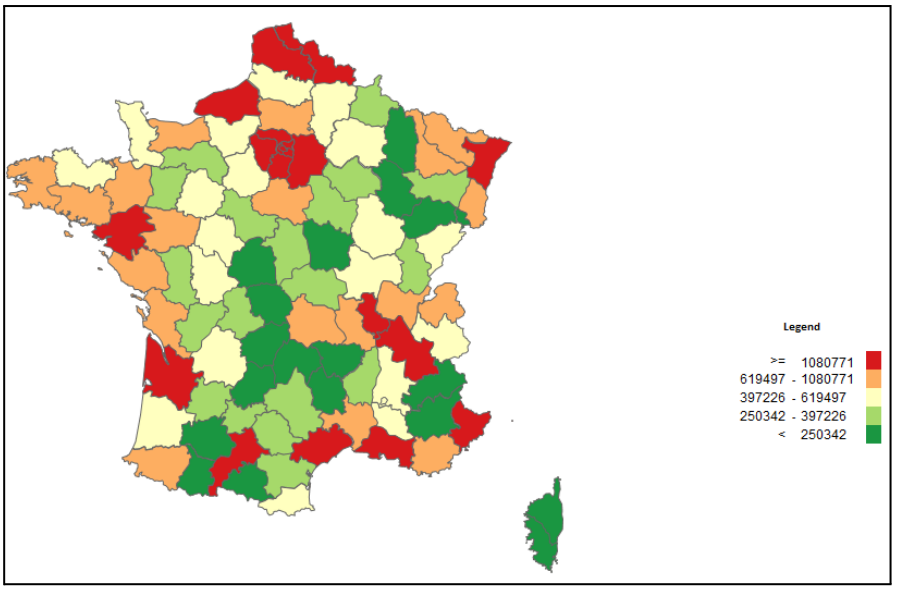


Fig: Map which shows a distribution related to forest fires in France.

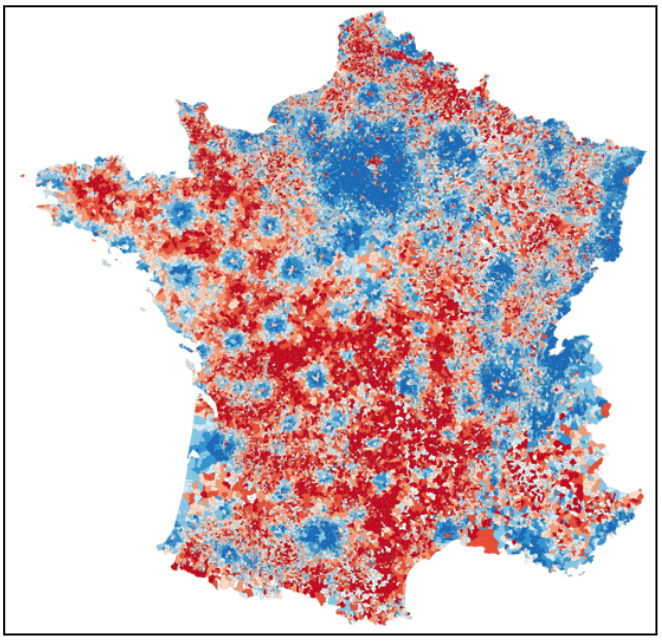


Fig: Map which shows a certain distribution in France with more details.

2.3. Existing solutions and examples

Z-2.3.1. *For the choropleth map, we found the Data Viz Project's collection of choropleth maps at [https://datavizproject.com/data-type/choropleth-map-2/]. This collection provides various examples and design ideas for creating choropleth maps. We will study these examples to understand different ways of representing data in specific regions using colors.*
We have also found a library of 100 chart examples at [https://100.datavizproject.com/]. This resource offers a wide range of chart types, such as pie charts, line charts, and bar charts. By referring to these examples, we can learn about different chart designs and choose the most suitable one for displaying forest fire data effectively.
We can also use all of the examples on: https://d3js.org/

Z-2.3.2. *On the following website, we can find an example of a map of France in which we can zoom to see more details for:https://simplemaps.com/resources/svg-fr*

We will also use all the labs we studied in the course of Data Visualization as examples, in which we can find code examples of Pie chart, Linear Chart, and bar chart.

Z-2.3.3. *All of these examples don't use the D3 library. As examples, we will mainly use all of the 4 labs we did.*

2.4. Customizing data

Z-2.4.1. I couldn't use Data Wrangler to process the data because it didn't read the file correctly.

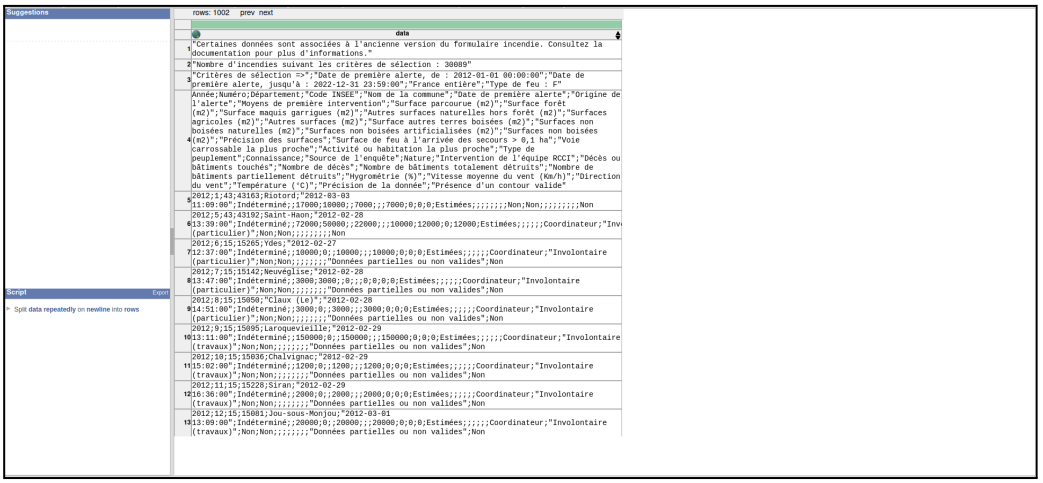


Fig: Screenshot of the result of importing the csv file on the data wrangler app.

Instead, I converted the file into json on an online web app.

Z-2.4.2. First thing to do was to remove all the extra columns we didn't need. I used this website https://jsoneditoronline.org/#right=local.faveci and run a specific filter to obtain the final json file with the 11 columns that I will use for my visualization.

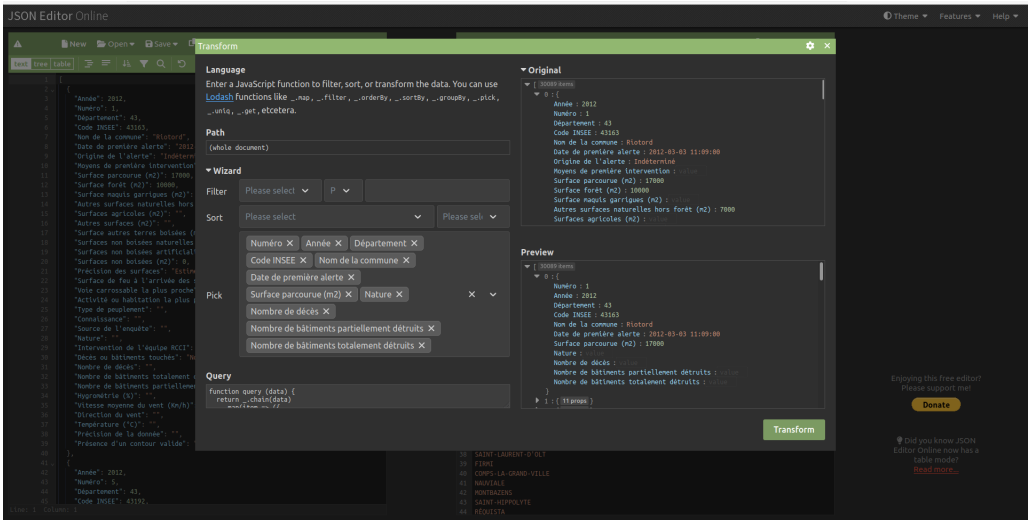


Fig : Json with 37 columns to json with 11 columns.

Z-2.4.3. Second thing to do was to make a connection from the names given in the dataset and the names given in the svg map. Indeed, with "Name" as the name of the commune and "Code" as the code INSEE of the commune, we have two different ids : "Name" and "Code NAME". To resolve this issue, I chose to create two json files to make a connection between the two different patterns. Each file

has all the names used in the dataset. The first one has all the names given in the dataset, and the second one has all of these names converted for the svg map. Practically, we used the same website and launch this function for the conversion :

```
function query (data) {
  let chars_get = _.chain(data)
    .map(item => [item?.["Code INSEE"],item?.["Nom de la commune"]])
    .value();
  let chars = [...new Set(chars_get)];
  for (let i = 0; i < chars.length; i++) {
     chars[i][1]=chars[i][1].toUpperCase().split(/[()]+/)
     if (chars[i][1].length == 1){
      chars[i][1] = chars[i][1][0].normalize('NFD').replace(/[\u0300-\u036f]/g, '').replace(/\s/g, '')
     }else{
      chars[i][1] = chars[i][1][1].normalize('NFD').replace(/[\u0300-\u036f]/g, '')
                    +" " +
                    chars[i][1][0].normalize('NFD').replace(/[\u0300-\u036f]/g, '').replace(/\s/g, '')
     }
     chars[i] = chars[i][0].toString() + " " + chars[i][1]
  }
  return chars
}
```

Fig : Filter function to retrieve all the names of the communes and converted them to match with the svg. Pattern “<name>” to “<code> <nameUppercase>”.

Z-2.4.4. As a result, we obtain our dataset completed, and our connection between the names in the dataset and in the svg.

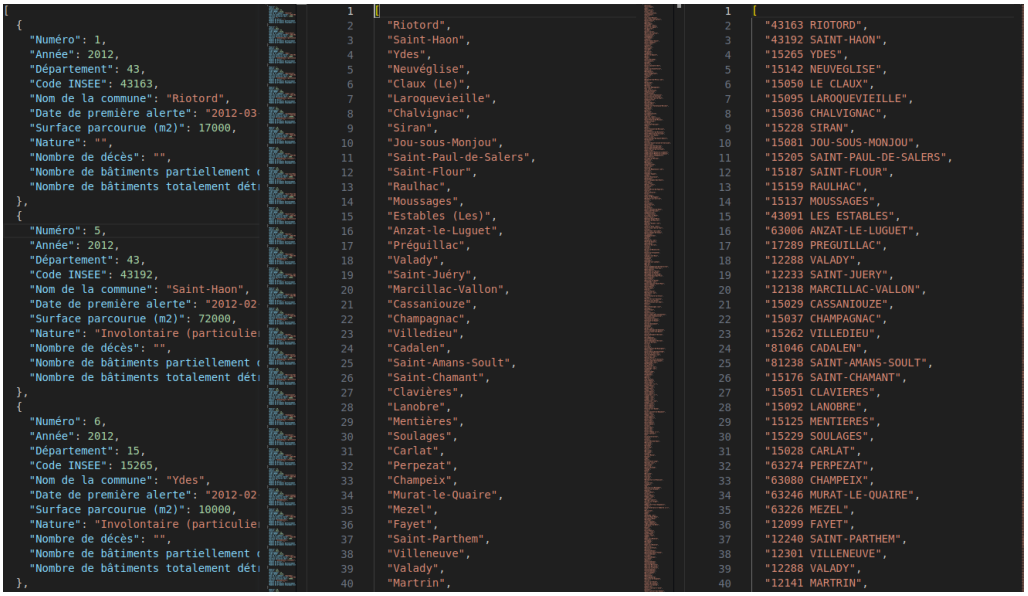


Fig : All the final files: the dataset in json, and the connection between the names in the dataset in the svg map.

2.5. Colors and data

Z-2.5.1. We use blue, red and orange for the barcharts, but it doesn’t add information to the visualization.

Z-2.5.2. We also used d3.schemeCategory10 to visualize the piechart. It doesn’t add information to the visualization.

Z-2.5.3. For the spatial distribution shown through the maps, we use a linear gradient color from yellow to red. The yellow area are the ones with less forest fires reported, and the red areas are the ones with the most forest fires reported.

3. KV3 - Creating prototype data visualization

3.1. Basic functionalities and behaviors

Z-3.1.1. A bar chart should show the distribution of the forest fires in each department throughout the years. Another bar chart will show the total surface burned each year. A last bar chart will show the total damages for each year. A pie chart should show the main causes of those fires.

Z-3.1.2. All of this will be static data, except for the pie chart. We will be able to choose either or not to show the amount of missing data about the cause of fires.

3.2. Advanced functionalities and behaviors:

Z-3.2.1. A dynamic map should give the distribution of the forest fires in each department throughout the years. A static map should give a more precise distribution for each commune.

Z-3.2.2. The dynamic map should dynamically change the data to correspond with a certain year. A loop will be launched and each 1 second the data of the next year will be shown.

Z-3.2.3. We should also be able to either display cumulative data or specific data. We should be able to stop the loop and restart it.

3.3. Implementation of basic functionalities

Z-3.3.1. For the basic functionalities, we try to program the bar chart which shows the distribution of forest fires. We have to first retrieve the data. For this, I have to launch an asynchronous function and wait for it to finish the loading. Then, I obtain a javascript array in which I can retrieve the data. I loop over all fires and count for each year how many we have. Finally, to display the data we use the same code as in the labs.

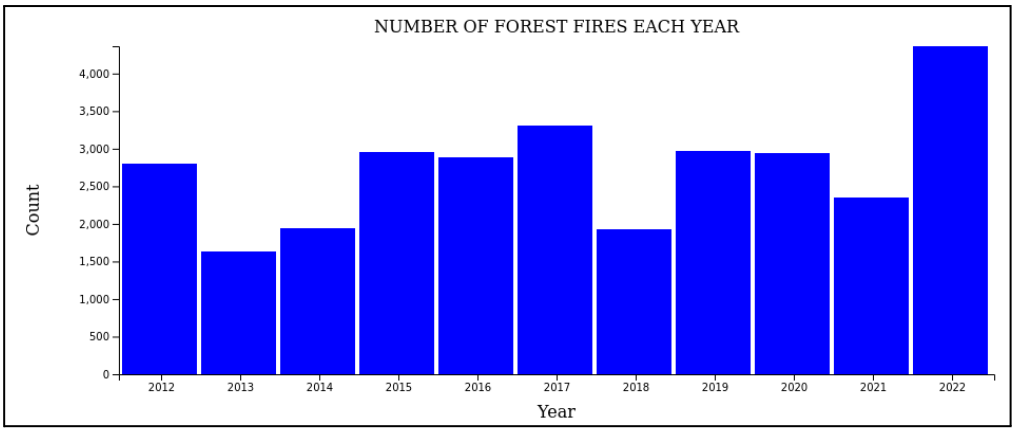


Fig: Result of the code described above.

3.4. Implementation of basic behavior

Z-3.4.1. For the basic behavior, we can describe how we code the pie chart. Indeed, we wanted the user to be able to change the pie chart to display it with or without the count of fires with unknown causes. For that, we just add a button with an onclick event function that will remove all the elements of the pie chart with *d3Element.exit().removes(), then we create again the pie chart with or without the unknown fires depending on the previous visualization.*

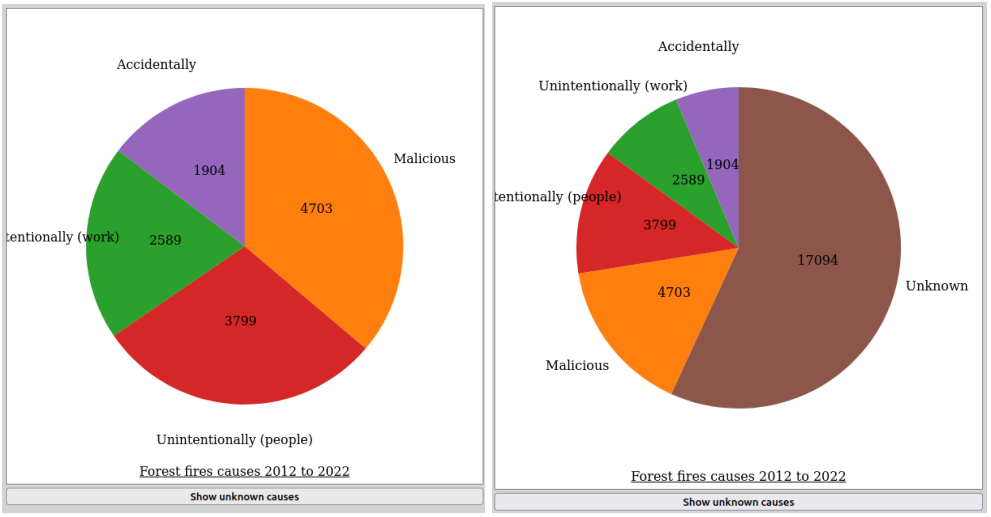


Fig:Pie chart without and with the unknown values.

4. KV4 - Creating the final data visualization

4.1. Implementation of basic functionalities

Z-4.1.1. For the final visualization, we completed all the basic functionality in which we display interesting simple data.

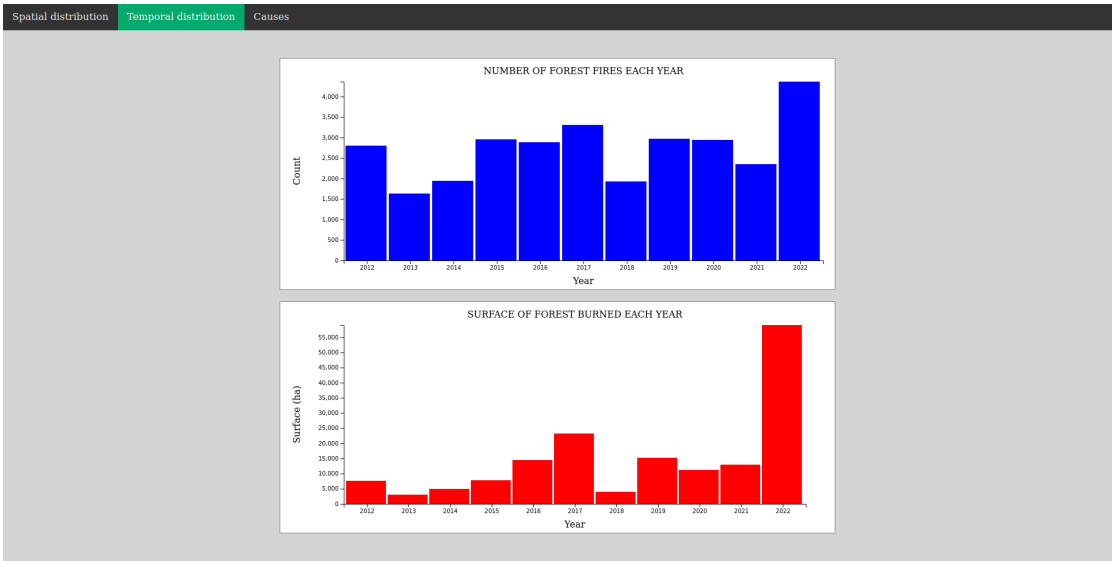


Fig: Static bar charts show some distributions throughout the years.

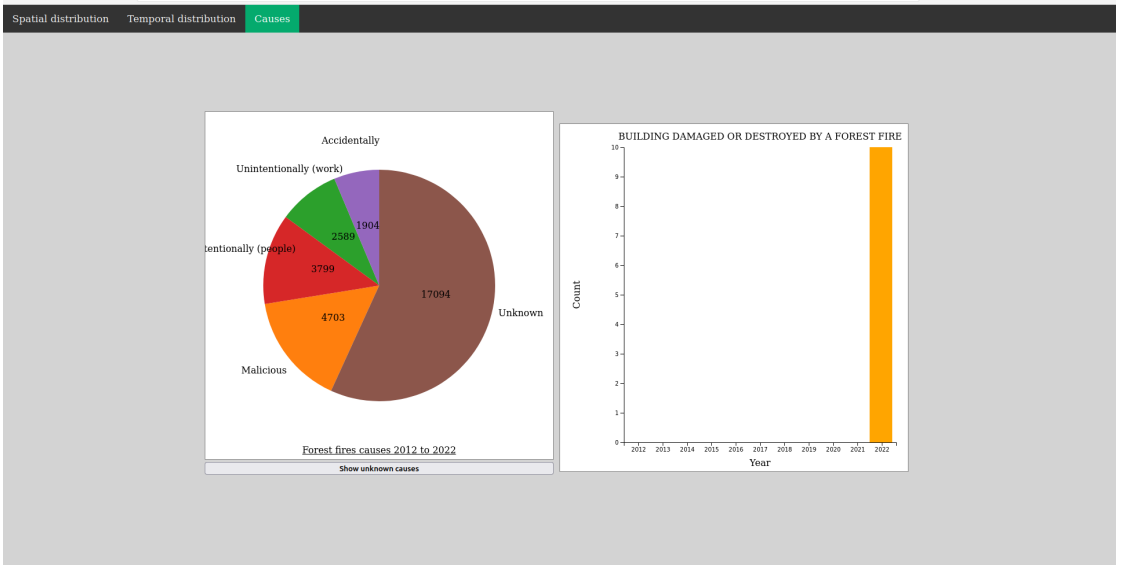


Fig: The famous pie chart which shows the causes and a bar chart which shows the damages.

4.2. Implementation of advanced functionality

Z-4.2.1. We implemented two visualizations of maps as we described in the previous part. One with a dynamic map, and one a static map.

Z-4.2.2. The static map was particularly complicated to implement. Indeed, there are approximately 34 000 communes in France. Therefore, we have possibly 34 000 different polygons to fill with a different color. Implement advanced functionalities. Prove by describing an example code. But filling thousands of svg paths takes a lot of resources and it didn't work when I wanted to fill all the data on time. To overcome this, we launch asynchronous functions that will fill only a fixed number

of polygon and wait for it to finish. Then, the asynchronous function continue to fill all polygons until it finishes everything. Finally, to make it beautiful during the loading time, we display a loading css animation on the screen.

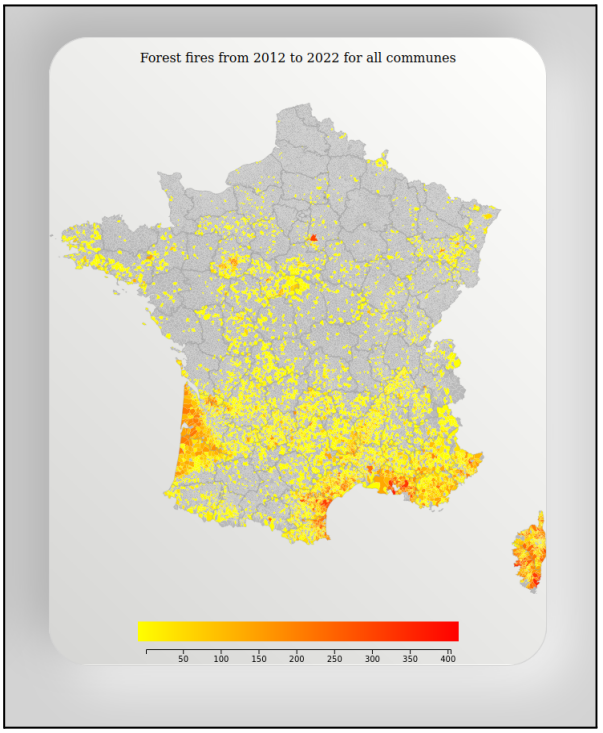


Fig: Fill thousands of polygon without making the web browser crash.

4.3. Implementation of advanced behavior

Z-4.3.1. The dynamic map can have its behavior changed by the user. The user can choose to pause the loop or he can choose to display cumulative data instead of specific one and vise-versa.

Z-4.3.2. To implement that, it's just a button event where we change a global boolean value. Then each time the map will load, the boolean will tell the program which range of data to count.

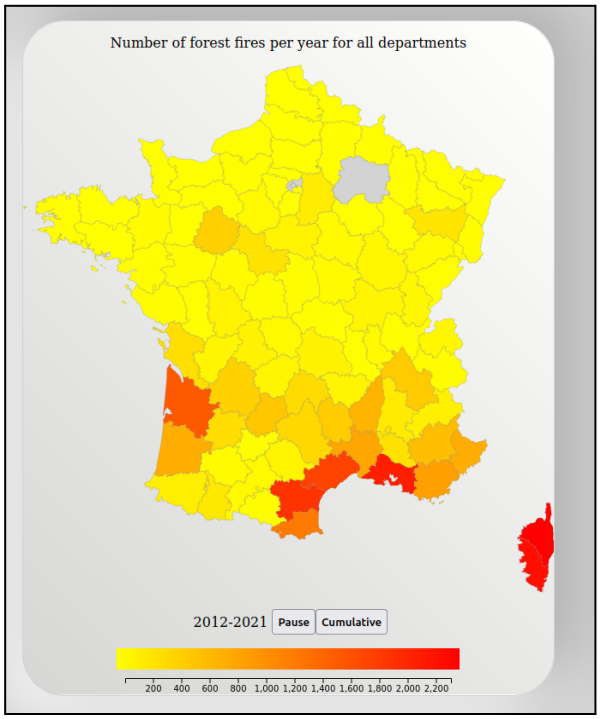


Fig: Dynamic map which changes during the loop.

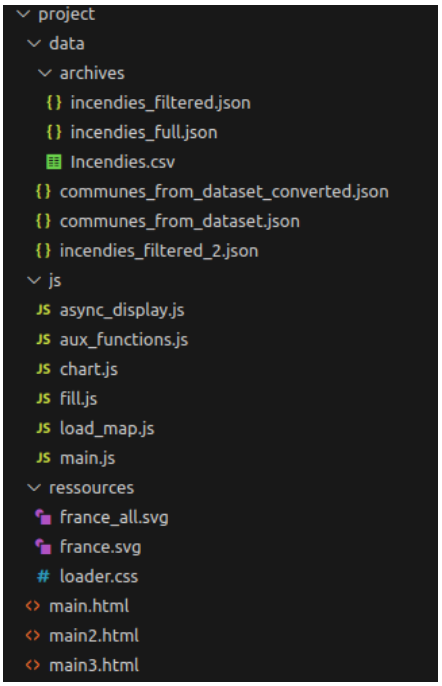
5. KV5 - Completing the project task and writing documentation

5.1. Possible modifications and refinements of the solution - in agreement with the teacher

Nothing to report.

5.2. Preparation of documents - project documentation

Z-5.2.1. Project hierarchy:



Z-5.2.2. List of technologies used: HTML, CSS, JS, D3 library.

Z-5.2.3. Setup instructions: Clone the repository from git. I suggest using the live server of VS code because it's easy to use and it's enough for this project. You just have to run this repository on a server and it should work.

Z-5.2.4. Instructions for use: Nothing special to do. You just have to access, through your server, one of the three files "main.html", "main2.html", "main3.html".

Literature

None

Annex I

Link to git repository of the project: https://github.com/Zintoulou/data-visualization-forest-fires