***Lab 1 - D3.js***

***Objectives:***

* *Understand how to load data with D3 and bind them to their graphical marks, properties, and layouts,*
* *Know how to use D3 selections, scales, and axes.*
* *Visualize geospatial data.*
* *Implement map controls, pop-ups, and data-driven styling.*

***Prerequisites:***

* *You have read chapter 3 (p. 52-62) and chapter 5 (p. 67-72, p. 79-87) in D3 -* [*interactive Data Visualisation for the Web*](https://books.google.co.ma/books?id=NGwvDwAAQBAJ&printsec=frontcover&redir_esc=y#v=onepage&q&f=false)*.*

***Resources:***

* *D3.js* [*site & documentation*](https://d3js.org/)
* [*Dashingd3js*](https://www.dashingd3js.com/)
* [*Gallery D3*](https://observablehq.com/@d3/gallery)
* [*Leaflet Documentation*](https://leafletjs.com/reference.html)*,* [*Quick start*](https://leafletjs.com/examples/quick-start/)
* *Some D3* [*examples*](https://bl.ocks.org/mbostock)
* *Meeks, E. (2015). D3. js in Action. Shelter Island, NY: Manning****.***

***Deliverables:***

* *A functional map-based visualization of earthquake data.*
* *Clean and commented code.*
* *A brief report explaining your implementation choices and any challenges faced.*

***TODO:***

***Part 1: Visualizing Earthquake Data with D3.js and Leaflet***

In this part, you will create an interactive map highlighting global earthquake occurrences. Using geospatial data, you'll visualize the epicenters, magnitudes, and depths of these events.

***Tips:***

* *Refer to the JS,  D3 and Leaflet documentation for any doubts.*
* *Collaborate with peers, but ensure your implementation is original.*

***Step 1: Setting Up the Project***

1. Project Structure:

* Create a new folder named ***Lab1*** with subfolders: ***css***, ***js***

1. Setting up HTML:
   * Create an ***index.html*** with a basic structure.
   * Link *D3.js*, *Leaflet,* and your custom ***js*** file (***lab1.js***).
   * Add a ***div*** with the id ***map*** for rendering the map.
   * Your structure should look like:

Lab1/

├── ***css/***

├── ***js/***

└── *index.html*

***Step 2: Loading and Processing Data***

1. Fetching Earthquake Data.
   1. Fetch data from USGS Earthquake [data feed](https://earthquake.usgs.gov/earthquakes/feed/v1.0/geojson.php) (GeoJSON format).
   2. Store the fetched data in a variable for further use.
2. Processing Data.
   1. Convert magnitude and depth data from strings to numbers.
   2. Filter out data without necessary attributes.

***Step 3: Visualizing Data on the Map***

1. Initialize the Map.
   1. Using Leaflet, initialize a map centered on Morocco. You can use these longitude and latitude parameters : **[31.110094, -8.37]**.
   2. Add a street view layer to the map.
2. Integrating *D3* with *Leaflet*.
   1. For each earthquake data point, create a circle marker on the map.
   2. Use *D3* scales to determine the size of the circle based on earthquake magnitude.
   3. Color-code circles based on depth using the ***getColor*** function.

Use these color codes:

* For ***depth*** exceeding 90 : ***rgb(255,13,13)***
* For ***depth*** between 70 & 90 : **rgb(255,78,17)**
* For ***depth*** between 50 & 70 : **rgb(255,142,21)**
* For ***depth*** between 30 & 50 : **rgb(250,183,51)**
* For ***depth*** between 10 & 30 : **rgb(172,179,52)**
* For ***depth*** between 0  & 10 : **rgb(105,179,76)**

1. Adding Interactivity.
   1. Add popups to each circle marker showing detailed information about the earthquake, such as magnitude, depth, location, and the time it occurred.
2. Map Controls.
   1. Implement a legend explaining the depth color codes.

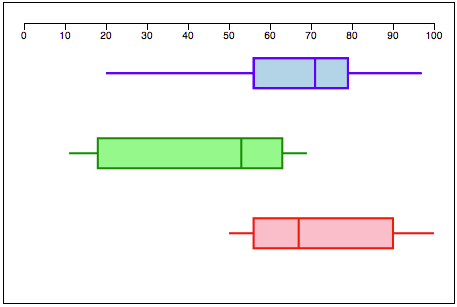
***Une image contenant texte, carte, atlas, diagramme

Description générée automatiquement***

**Advanced Challenges *(for those who finish early or want an extra challenge)*:**

* Implement a time slider to visualize earthquakes over time.
* Allow users to filter earthquakes based on magnitude or depth.

***Part 2: Box Plot***

****

The graph above is a box plot and includes a horizontal line between the minimum and maximum value. The box is drawn between the first quartile and the third. A vertical line in the box is drawn at the median.

A Javascript script is provided to you in which you must replace the lines commented ***// à completer*** with Javascript code to create a graph similar to that of the figure above. ***(Do not change the code provided to you other than to replace the previously described comments)***.

* Your javascript code must use the D3 framework to generate the SVG.
* Create the HTML and CSS files to create the web page that will call this script.
* Consider using the ***d3.quantile*** function to determine box and line boundaries.