Student Declaration of Authorship

HERIOT WATT UNIVERSITY
UK | DUBAI | MALAYSIA

Course code and name:	F21DV Data Visualization and Analytics
Type of assessment:	**Individual**
Coursework Title:	Coursework 1: Data Visualization & Analytics
Student Name:	Sasha Fathima Suhel
Student ID Number:	H00410394

Declaration of authorship. By signing this form:

- **I declare** that the work I have submitted for individual assessment OR the work I have contributed to a group assessment, is entirely my own. I have NOT taken the ideas, writings or inventions of another person and used these as if they were my own. My submission or my contribution to a group submission is expressed in my own words. Any uses made within this work of the ideas, writings, or inventions of others, or of any existing sources of information (books, journals, websites, etc.) are properly acknowledged and listed in the references and/or acknowledgements section.
- I confirm that I have read, understood, and followed the University's Regulations on plagiarism as published on the University's website, and that I am aware of the penalties that I will face should I not adhere to the University Regulations.
- I confirm that I have read, understood, and avoided the different types of plagiarism explained in the University guidance on Academic Integrity and Plagiarism

Student Signature *(type your name):* *Sasha Fathima Suhel*

Date: *March 16, 2023*

Copy this page and insert it into your coursework file in front of your title page.

For group assessment each group member must sign a separate form and all forms must be included with the group submission.

Your work will not be marked if a signed copy of this form is not included with your submission.

Heriot-Watt University

Coursework 1: Data Visualization & Analytics

Author:

Sasha Fathima Suhel

Coursework submitted in fulfilment of the requirements for the degree of MSc.

Data Science

of

August 2022

in the

School of Mathematical and Computer Sciences

CONTENTS

1 Introduction 4

1.1 Set up 4

2 Application 6

2.1 Map 6

2.2 Line graphs 7

2.3 Scatter Plot 9

3 Query 10

3.1.1 How did the COVID-19 pandemic grow over countries and continents throughout the world from its start? 10

3.1.2 Choosing some specific countries, how successfully did they manage the outbreak? 10

3.1.3 Choosing some specific countries, is there a relationship between the relative "wealth" (e.g. GDP) of a population and the spread of the pandemic? 11

3.1.4 What effect did vaccinations have on the spread of cases/deaths? Did booster jabs also have an impact on the spread/transmissibility of the virus? 11

3.1.5 How might the geographical position of a country change how the pandemic impacted them? 12

4 Conclusion 12

5 References 13

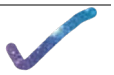
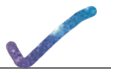
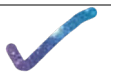
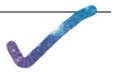
VIDEO LINK

F21DV_CW1_H00410394_SFS.mkv

1 INTRODUCTION

The COVID-19 pandemic has created significant changes worldwide, with numerous countries struggling to contain the virus. The use of data analysis has been instrumental in tracking the spread and impact of the virus. The datasets provided by Our World in Data (OWID) include information on COVID-19 cases, deaths, and vaccinations for countries worldwide. In addition, two additional JSON files provided by you contain information on country names, ISO codes, and latitude-longitude coordinates.

The objective of this report is to describe the design and implementation of an interactive web-based application using the provided datasets.

REQUIREMENT CHECKLIST	
single HTML page called index.html	
Minimum three different visualisation types	
on a data of a single visualisation is updated, all axes of that visualisation also gets update/rescale accordingly	
two visualisations, mousing over a datapoint in one visualisation highlights multiple associated datapoints in another visualisation.	
a map visualisation which must interact with at least one visualisation	
circles of different sizes to indicate scalesn on the map	
cross-visualisation brushing where dragging a rectangle of several datapoints in one visualisation highlights multiple associated datapoints in another visualisation	
faceted selection interaction between two visualisations	

1.1 SET UP

My code is a simple HTML page that installs the libraries and stylesheets required to create a dashboard with D3.js and Tailwind CSS. For data visualization, the code includes D3.js version 7, TopoJSON, and D3-legend. The page also includes Tailwind CSS and Font Awesome for styling. The page's body includes a header with a logo and a navigation bar with my GitHub repository to this project.

The dashboard's main content consists of five elements: a line chart displaying new cases by country, a world map displaying the total number of cases, a line chart for the new cases navigator, a line chart displaying total vaccinations by country and a total case based on income bracket.

Finally, this page includes a footer section with information about me and my social contact details. By using my code, developers can quickly set up a web page and start building a dashboard with D3.js.

```
<!-- set content for dashboard -->
<div class="row content p-3 ml-12 w-4xl inline-flex">
  <div class="cases-linechart bg-red-900 mr-3">
    <row class="cases-lc-row inline-flex">
      <h2 class="text-white text-lg tracking-widest font-regular p-5">New Cases by Country             &
    </row>
  </div>
  <div class="world-map bg-red-900 mr-3">
    <row class="cases-lc-row inline-flex">
      <h2 class="text-white text-lg tracking-widest font-regular p-3">Global Map of Total Cases</h2>
    </row>
  </div>
</div>

<div class="row content2 ml-12 pl-3 w-3xl inline-flex">
  <div class="mini-linechart bg-red-900 mr-3">
    <row class="cases-lc-row inline-flex">
      <h2 class="text-white text-lg tracking-widest font-regular p-5">New Cases Navigator</h2>
    </row>
  </div>
  <div class="vax-linechart bg-red-900 mr-3">
    <row class="cases-lc-row inline-flex">
      <h2 class="text-white text-lg tracking-widest font-regular p-5">Total Vaccinations by Country</h2>
    </row>
  </div>
</div>

<div class="row content3 ml-12 pl-3 w-3xl inline-flex" style="margin-top: 0.7%;">
  <div class="income-linechart bg-red-900 mr-3">
    <row class="cases-lc-row inline-flex">
      <h2 class="text-white text-lg tracking-widest font-regular p-5">Total Cases by Global Income Brackets</h2>
```

Figure 1 Setting up rows.

And an index.css file that is implemented to design my html webpage. It includes several rules and declarations that define the visual aspects of the webpage, such as font, colour, and layout. The file also includes rules for creating a gradient background for the webpage's body, as well as styles for various types of SVG elements, such as lines and paths. The CSS file also defines styles for the webpage's footer and header, such as text size and spacing, the placement of social media icons, and the colour of various elements.

```
body {
  background: linear-gradient(to bottom, rgb(255 228 230), rgb(253 164 175));
}

.axisWhite line {
    stroke-opacity: 0.1;
}

.axisWhite path {
    stroke-opacity: 0.1;
}

.axisWhite text {
    fill: rgb(245, 239, 244);
}

.linePink {
    fill: none;
    stroke: rgb(215, 32, 133);
    stroke-width: 3px;
}

.linePurple {
    fill: none;
    stroke: rgb(32, 2, 32);
    stroke-width: 3px;
}

#xAxis .tick line {
    visibility: hidden;
}
```

Figure 2 Css graph styling.

2 APPLICATION

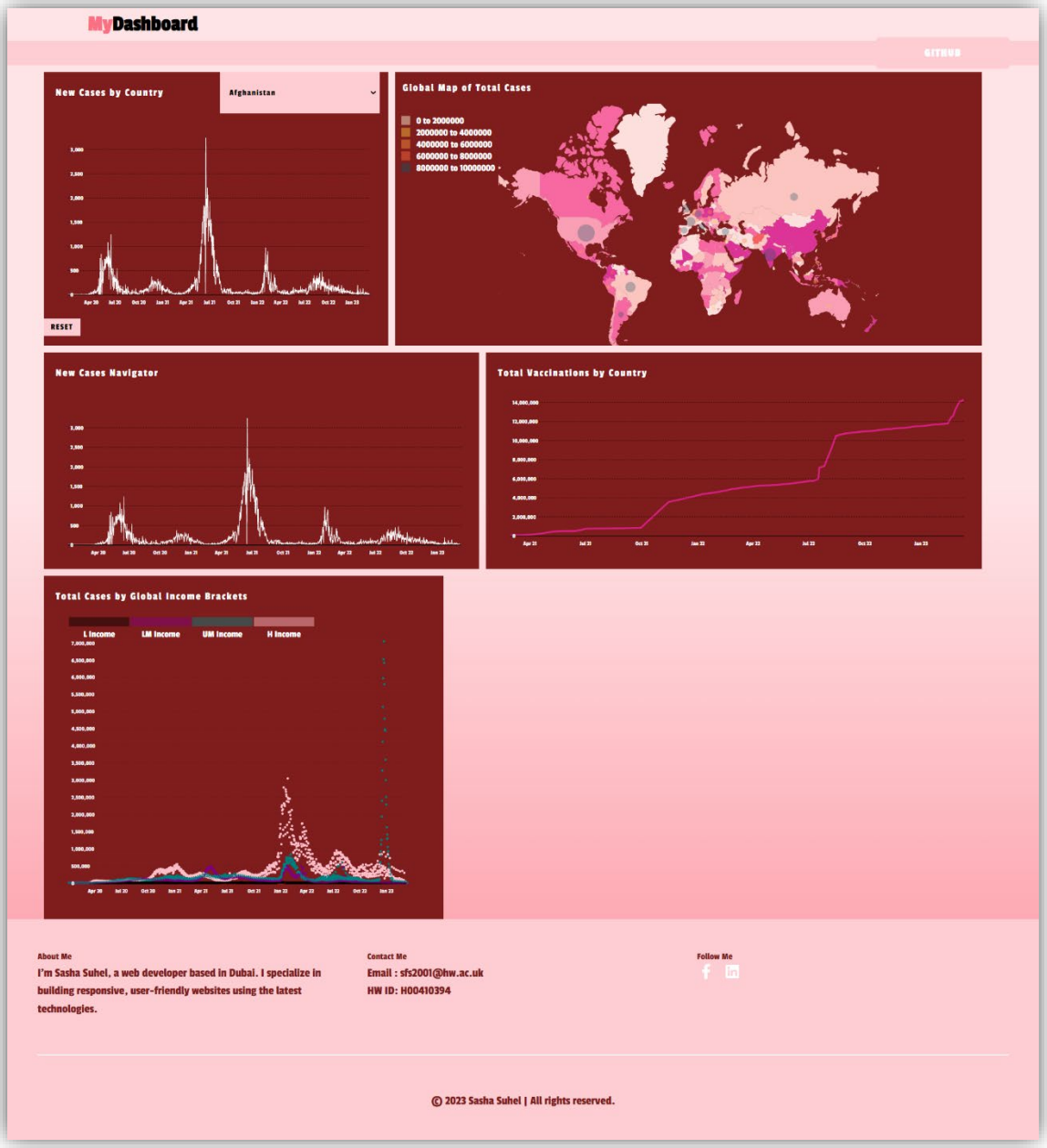


Figure 3 My Dashboard HTML webpage

Shown in Figure 3, is the main view of the My Dashboard, made out of a few different layouts namely the map, the line graphs above, and a scatter plot.

2.1 MAP

If you take a closer look at Figure 4, you'll see a map that's color-coded based on the latest New Cases Per Million Covid-19 numbers. There are also circles on the map that indicate where the Covid hotspots are located all over the world. When you hover over a country, you'll see a legend on the top right that

shows the latest cases numbers for that country. If you don't hover over any country, you'll see the latest global Covid data. The map itself is made up of an svg, which was later filled with shapes and objects.

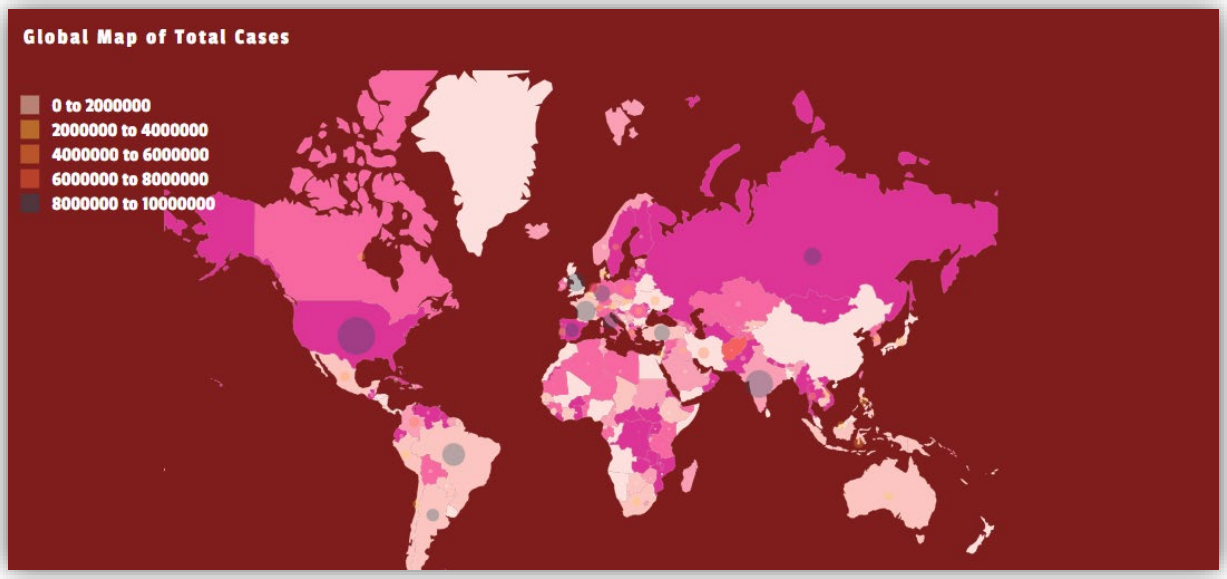


Figure 4 Close view of the map

- The function **selectCountry(country)** is a callback function that is called when a user clicks on a country on the map. This function takes the name of the selected country as an argument and updates various visualizations with data corresponding to that country.
- The code starts by declaring constants for the width and height of the map and creating an SVG element for the map.
- Next, a **mapG** grouping is created to hold the map elements.

```
// callback function when selecting country
function selectCountry(country) {
  d3.select(".selected").classed("selected", false);
  d3.select(`[name=${country.replace(/\s/g, "")}]`).classed(
    "selected",
    true
  );
  updateLC(country, cases, svg, "cases", csv);
  updateVac(country, vacs, svg2);
  updateMini(country);
  d3.select("select").property("value", country);
}
```

Figure 5 Country selection code

- The **createMap()** function is then defined, which loads JSON data for the map and lat-long data for the circles representing COVID-19 cases. The **topojson.feature()** function is used to extract the features from the JSON data and **d3.scaleSqrt()** and **d3.scaleQuantize()** are used to scale the circle size and color, respectively. A legend is also created to indicate the color scale used for the circles. Finally, circles are added to the map using the lat-long data and scaled by the COVID-19 cases.
- The **zoom** variable is then defined to enable zooming and panning on the map.
- Finally, the **createMap()** function is called to create the interactive world map with COVID-19 cases circles.

```
// draw circle which represents cases using lat long data
d3.json(
  "https://raw.githubusercontent.com/sashasuhel/Data-Visualisation-Analytics/main/lat-long.json"
).then((data) => {
  // scale circle size by cases
  var scaleCircle = d3
    .scaleSqrt()
    .domain(
      d3.extent(data, function (d) {
        return d.cases;
      })
    )
    .range([0, 0.5]);

  // divide cases into quantized colour scale
  var scaleColor = d3
    .scaleQuantize()
    .domain([0, 10000000])
    .range(["#feffdf", "#ffc93c", "#ff9a3c", "#ff6f3c", "#155263"]);
```

Figure 6 Circle density on each country

2.2 LINE GRAPHS

In total there are three line graphs implemented in this dashboard that hold their own responsibility in representing a story. All 3 of the line graphs are similar in nature so in this report I will be explaining the "new cases by country". Initially when the dashboards start up "Afghanistan" is loaded up into all the graphs.

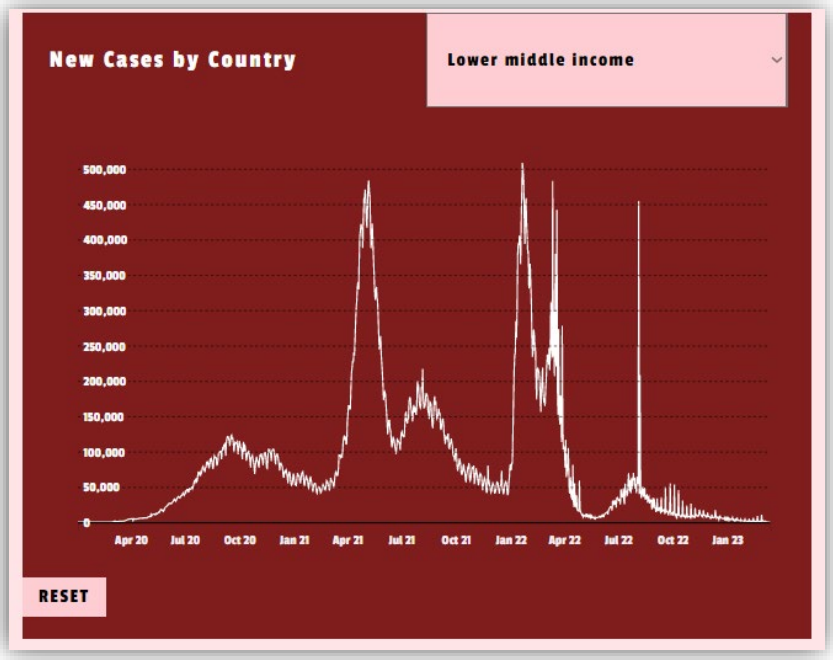


Figure 7 Line graph for new cases

- The code defines constants for the chart's width, height, and margins.
- The code generates a D3 scale for the x- and y-axes (number of cases).
- An SVG element with the defined width and height plus margins is added to the HTML body, and a group element is added to the SVG element and moved to the top left margin.

```
// function which initialises the cases linechart
function initialiseLC(country, arr, svg, data) {
    csv.then((value) => {
        // load csv values into a preprocessing array
        for (var i = 0; i < value.length; i++) {
            if (value[i].location === country) {
                arr.push(value[i]);
            }
        }
```

Figure 8 Pulling data from the dataset

- The initialiseLC function is used to populate the line chart with data from a specific country. The function accepts the country name, an array containing the data, the SVG element, and the data

itself. The CSV data for the selected country is loaded and pushed into the array within the function. The data is then sorted by date and formatted, and the data range for the x and y scales is calculated. The D3 line generator is used to create a value line, and the line path is appended to the SVG element. The x and y axes, as well as a reset button, are also added to the SVG element.

```
// create the interactivity drop down
function createDropdown(append, callback) {
  csv.then((value) => {
    var temp = [];
    // load csv values into a preprocessing array
    for (const element of value) {
      temp.indexOf(element.location) === -1
        ? temp.push(element.location)
        : [];
    }
```

Figure 9 Dropdown function

- When a new country is selected from a dropdown, the updateLC function is called to update the line chart. The function accepts the country of choice, the data array, the SVG element, and the data itself. The data for the selected country is pushed into the array, and if no data for the country is found, a single object is pushed to keep the chart from looking janky. The x and y scales are updated, and the line path is updated to reflect the new information.

2.3 Scatter Plot

This scatter plot for the world's total income bracket would show the relationship between the total income of various countries and their corresponding income brackets. Each point on the plot represents a different country, with the x-axis representing total income and the y-axis representing income bracket.

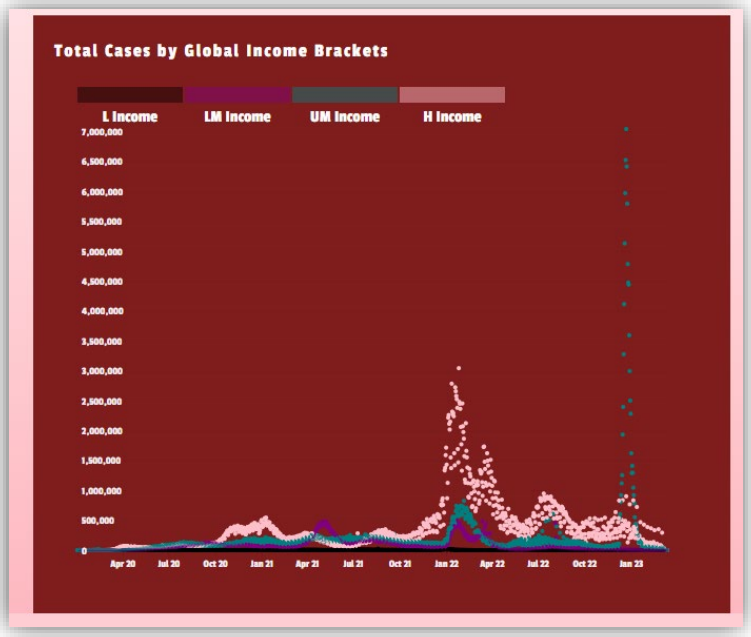


Figure 10 Scatter plot based on total income of the world.

The scatter plot will show you how income brackets are distributed around the world and whether there is a relationship between a country's total income and the income bracket of its citizens.

- Constants for the dimensions of the visualization are declared.
- The d3.timeParse function is used to create a time parser for the date format %b %y.
- Scale functions are created using d3.scaleTime() and d3.scaleLinear() for the x and y axes respectively.
- An SVG element is appended to the .income-linechart element using d3.select() and various attributes are set on it.
- A function createScatter() is defined which takes an array as input and creates the scatter plot using the data in that array.
- The function createScatter() loads data from a CSV file using d3.csv().
- The data is filtered to only include records with certain values for the location field.
- Using forEach() and sort(), the data is formatted and sorted by date ().
- Using.on, mouseover behavior is added to the scatter plot circles ("mouseover").

```
// function which creates scatter plot
function createScatter(arr) {
  csv.then((value) => {
    // add values to array
    for (const element of value) {
      if (
        element.location === "Lower middle income" ||
        element.location === "Upper middle income" ||
        element.location === "High income" ||
        element.location === "Low income"
      ) {
        arr.push(element);
      }
    }
    // format the data
    arr.forEach(function (d) {
      d.date = new Date(d.date);
      d.new_cases = +d.new_cases;
    });

    // sort by date
    arr.sort(function (a, b) {
      // turn strings into dates
      return new Date(b.date) - new Date(a.date);
    });
```

Figure 11 CreateScatter function to section into income groups

3 QUERY

3.1.1 How did the COVID-19 pandemic grow over countries and continents throughout the world from its start?

Ans) The COVID-19 pandemic began in Wuhan, China in December 2019 and quickly spread to other countries and continents worldwide. It started in China and then spread to the European countries and last to USA. By visualizing the map, you can see the circles depicting the level of covid cases based on every country. Countries like India had deadly number of covid cases hence why there is a bigger circle in India. In this dataset there is a missing dataset on USA, which is why you cant see any visualizations on that continent. Hopefully in the future the dataset will be updated and give accurate results.

3.1.2 Choosing some specific countries, how successfully did they manage the outbreak?

Ans) These are what I understood from the visualization on how the countries managed the outbreak –

1. Germany: Germany has been commended for its pandemic response. The country instituted early and extensive testing, as well as contact tracing and quarantine measures for those who tested positive or had contact with infected people. In comparison to other European countries, these measures helped to keep the number of cases and deaths relatively low.

2. Saudi Arabia: To combat the spread of COVID-19, Saudi Arabia implemented several measures, including the closure of schools, mosques, and other public places. The country also imposed a curfew and required citizens and residents to wear masks in public. These measures have helped to keep the number of cases and deaths in Saudi Arabia relatively low.

3. Russia has had a mixed record in dealing with the pandemic. The country initially downplayed the virus's severity and was chastised for failing to implement strict measures early on. The government, on the other hand, eventually implemented a nationwide lockdown and increased testing and contact tracing efforts.

3.1.3 Choosing some specific countries, is there a relationship between the relative "wealth" (e.g. GDP) of a population and the spread of the pandemic?

Ans) From the scatter plot it is understandable that when you hover over a higher income wealth bracket the rate at which new cases are present in the past few months are comparatively lower. Whereas the income brackets with lower income when the mouse is hovered on those brackets it is visually visible that there is a higher rate of new covid cases in the past same few months. This lets us know that the overall GDP per Capita has an inverse relationship with the evolution of the pandemic.

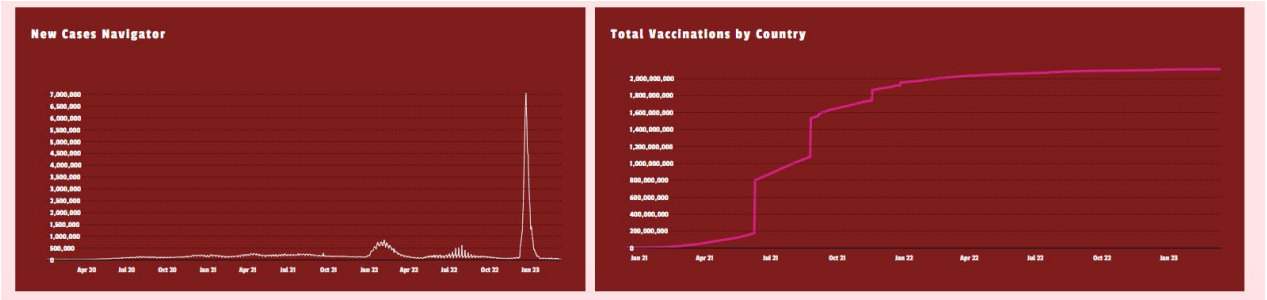


Figure 13 Higher income bracket groups

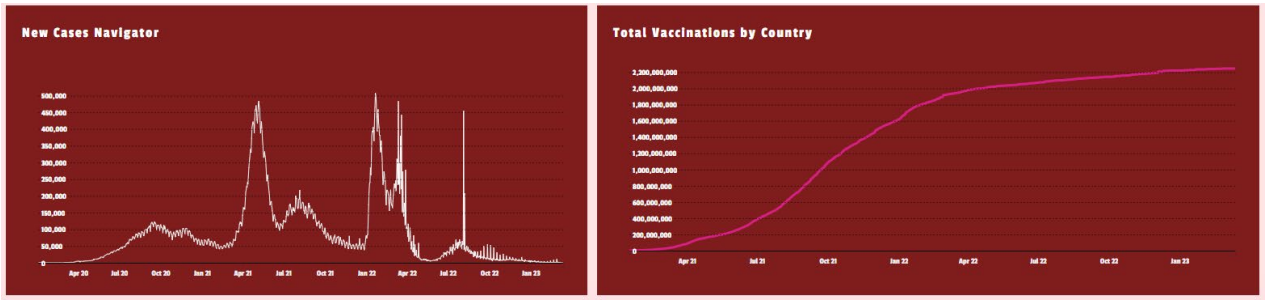


Figure 12 Lower Income bracket groups

However, there are also examples of countries with relatively low GDPs, such as Vietnam and Cuba, that were able to effectively manage the outbreak through proactive public health measures and community engagement.

3.1.4 What effect did vaccinations have on the spread of cases/deaths? Did booster jabs also have an impact on the spread/transmissibility of the virus?

Ans) For example taking Kuwait as a country to understand the spread of outbreak after the booster shots. Around Jan 22 is when you can see the graph becomes stable showing that most people got their booster shots, which shows that after that time period the new cases where people got covid drastically reduced showing huge benefit in getting vaccinations.

Other countries could be taken as examples which will show how after getting booster shots, the affect it had on new covid cases. Some countries reduced their new cases whereas some remained the same or more which results back to how the income bracket plays an important role in determining the rate of new covid cases in the country.

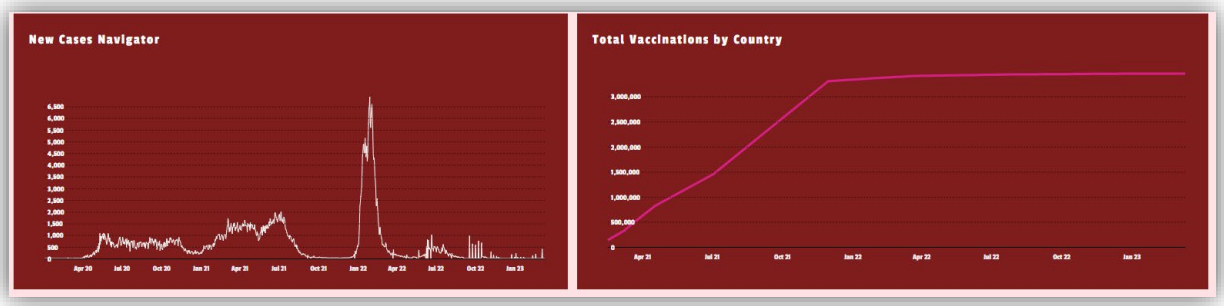


Figure 14 Kuwait's status

3.1.5 How might the geographical position of a country change how the pandemic impacted them?

Ans) Proximity to other countries: Countries located close to other countries with high levels of COVID-19 cases may be more vulnerable to outbreaks. This is due to the ease with which people can cross borders, potentially bringing the virus with them. Many European countries near Italy, for example, which had a high number of cases early in the pandemic, saw an increase in cases as well.

Climate: There is evidence that the virus spreads more easily in colder climates. Countries with colder climates may be more vulnerable to outbreaks. Furthermore, countries with warmer climates may be more vulnerable to outbreaks during certain seasons, such as summer, when people tend to congregate in large numbers.

Population density: Because the virus spreads more easily in crowded areas, countries with higher population densities may be more vulnerable to outbreaks. This is due to the increased likelihood of people coming into close contact with one another, which increases the likelihood of the virus spreading.

4 CONCLUSION

Using HTML and d3.js to create a COVID-19 dashboard is an excellent way to visualize data related to the pandemic. It can aid in identifying trends, patterns, and changes in the virus's spread over time.

The dashboard can track the number of cases, deaths, and recoveries in various countries or regions. It can also be used to visualize other pandemic-related data, such as vaccination rates, hospitalizations, and testing results. Creating a COVID-19 dashboard teaches us that data visualization is an important tool for understanding complex information. The dashboard can assist people in better understanding the pandemic's impact and making informed decisions about how to respond to it.

Overall, creating a COVID-19 dashboard with HTML and d3.js is a valuable exercise that can provide insight into the virus's spread and impact on communities worldwide.

5 REFERENCES

- https://www.w3schools.com/html/html_images_background.asp
- https://tailwindcss.com/docs/background-color
- https://d3-graph-gallery.com/
- https://observablehq.com/@d3/color-schemes
- https://github.com/d3/d3-scale-chromatic
- https://github.com/sashasuhel/Data-Visualisation-Analytics