CS 416: A Narrative Visualization project by Tirthankar Bhakta

Email: tbhakta2@illinois.edu

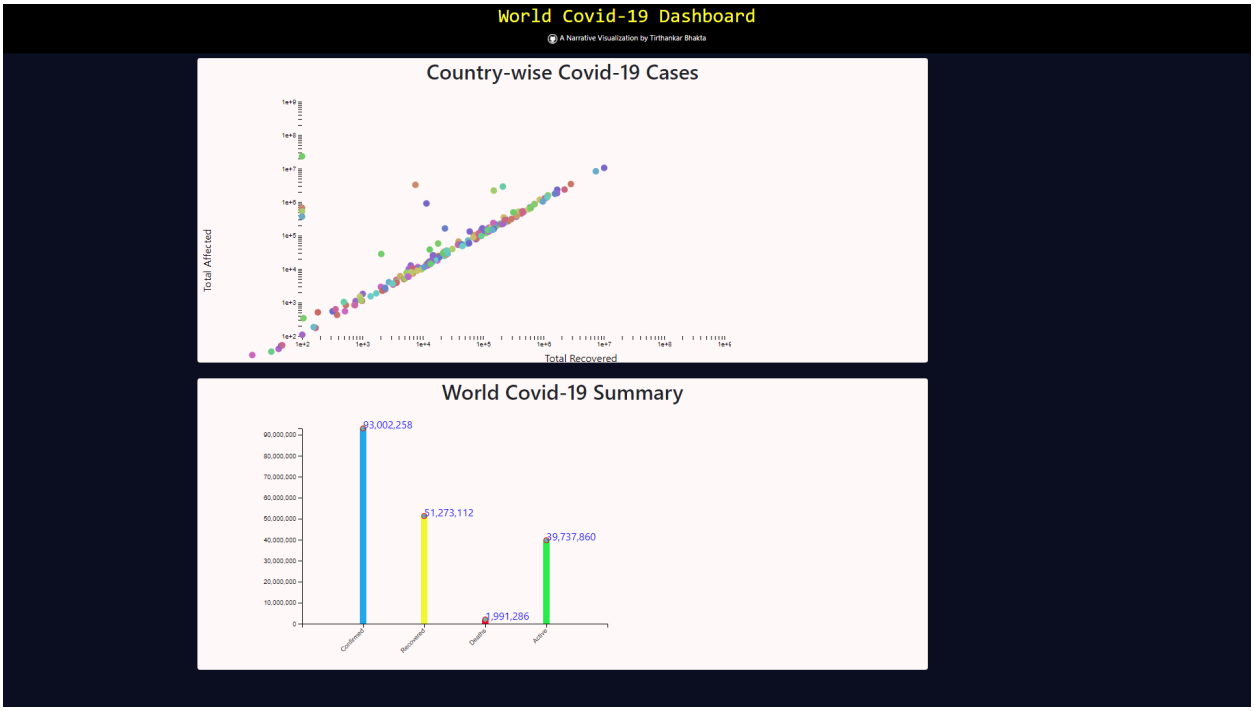
COVID-19 Dashboard Narrative Visualization

Introduction:

Dashboard to display total cases, casualties, recoveries and active cases worldwide. There is a link in the Dashboard that navigates to my Github repository where this page is hosted.

The Dashboard consists of **2 graphs**:

1. Country-wise Covid-19 Statistics Scatterplot (Total Recovered versus Total Confirmed)
2. Worldwide Covid-19 Summarized Line Graph (summarized parameters - Total Confirmed cases, Total Recovered, Total Deaths and Total Active Cases)

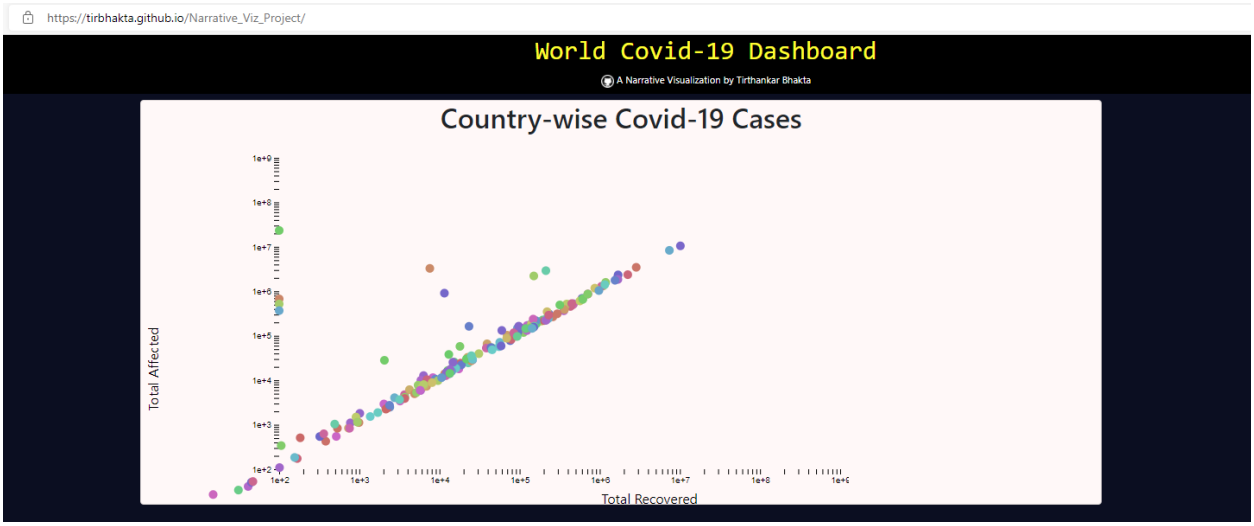


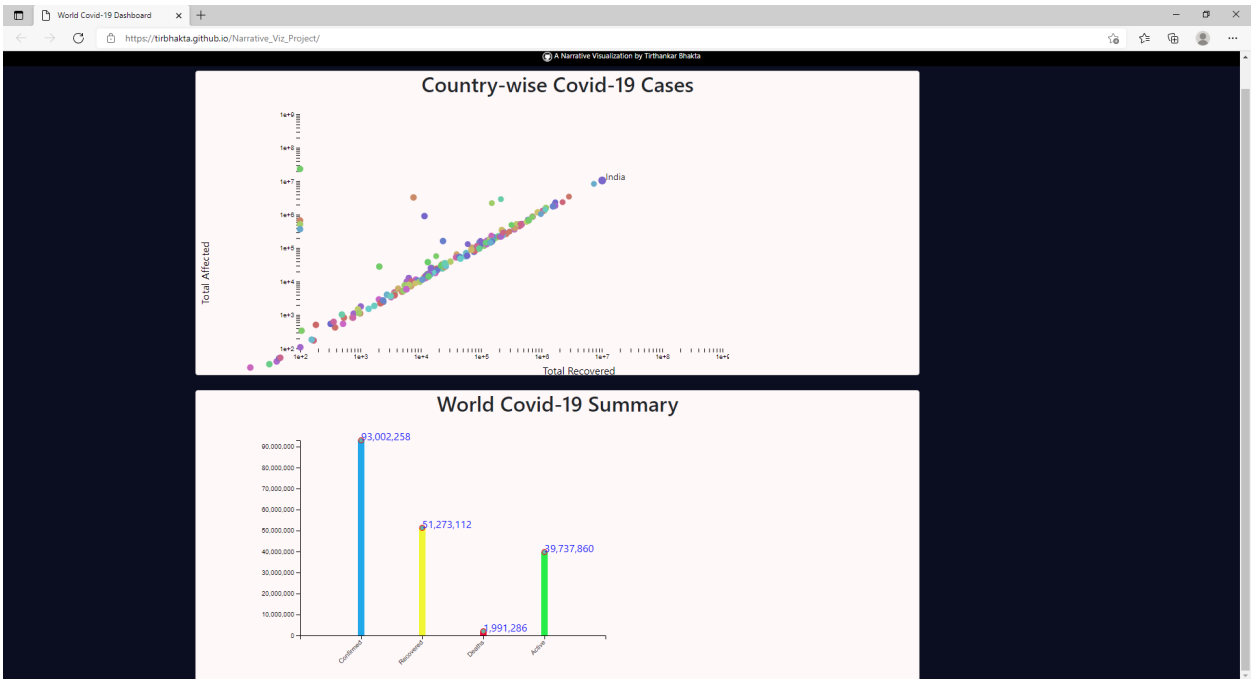
Usage:

The dashboard has some functionalities:

1. Country-wise Covid-19 Statistics Scatterplot –

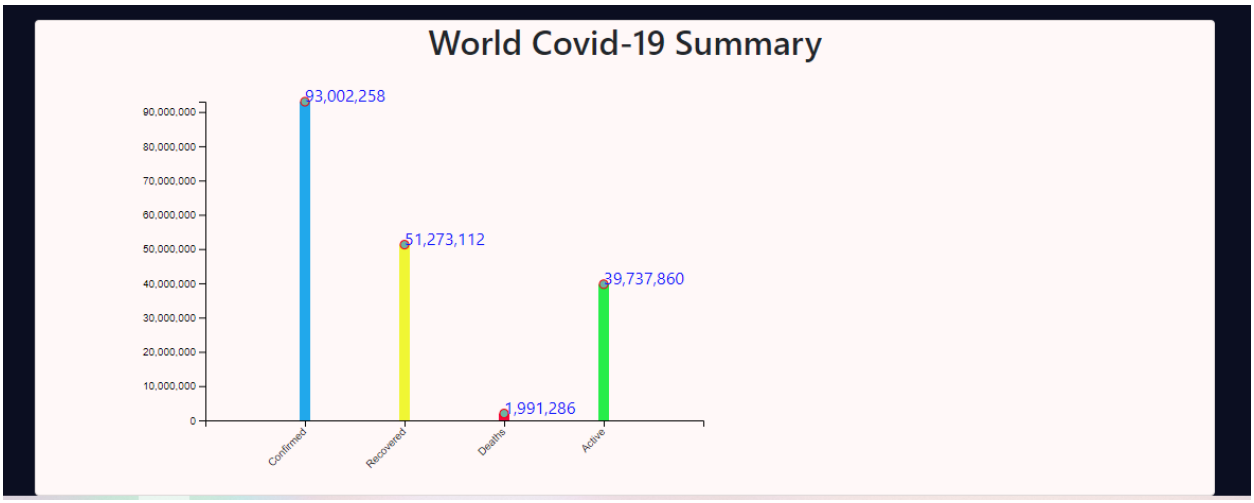
   a) On mouse hover show the Country
   b) X-axis denotes the Total Recovered Cases per country
   c) Y-axis denotes the Total Confirmed Cases per Country
   d) Random colors to distinguish each country





2. Covid-19 Summarized Line Graph -

a) Each parameter is displayed in a separate color
b) Aggregated values of Total Confirmed cases, Recovered, Deaths and Active across all countries
c) Y-axis displays the countries
d) X-axis displays the aggregated parameters in tick format
e) Static tooltip with a circle overlay showing the exact count for each parameter



Technical files:

D3.js files:

The following files are used to visualize the graphs:

1. d3 js files/scatterPlotWorld.js** is for the country-wise statistics ScatterPlot graph.
2. d3 js files/worldSummary.js** renders the Worldwide summarized Line graph.

Data file

World_Covid_Data.csv - this is the main Covid-19 datasheet used. This is referenced from the public site https://ourworldindata.org/coronavirus-source-data

HTML Webpage:

index.html - this is the main HTML file for the Dashboard

CSS file:

bootStrap.css - generic CSS file used for styling. Core CSS properties and values are inline in the HTML file and the D3.js files

LINKS:

Github Repo: https://github.com/tirbhakta/Narrative_Viz_Project

Github Page: https://tirbhakta.github.io/Narrative_Viz_Project/

ESSAY:

Messaging:

I found the World Covid data interesting when I attempted the Tableau Dashboard assignment, and decided to make use of the data in the Narrative Visualization.

The data from the dashboards clearly displays that countries where the population is higher, the impact is higher, in general.

It also depicts an interesting message that the progressive countries recovered better than others and had less casualties even when the cases were higher.

The summarized view in the dashboard displays a clear message that the casualties worldwide is minimal compared to the overall population impacted, although the number of deaths is very alarming and is truly disheartening.

Narrative Structure:

I have used the Martini Glass narrative model. Both the scenes/graphs use the same structure. The graph are center-aligned on a dark background giving a bright visual effect.

Visual Structure:

The structure is a simplified Martini Glass structure where there are very less buttons and navigation, thus removing unnecessary distractions for the user.

All graphs use similar template and distinguishable colors. Also the graph axes are labelled for ease of understanding.

Scenes:

The scenes themselves are constructed via HTML id tags. Each scene has a unique id and is composed of a unique chart that highlights its own distinct message.

Annotations:

I have added tooltips for each graph. For the first representation, the country name is visible on mouse hover. In the second graph, the total counts are shown as static tooltips for each line.

Parameters:

The visualizations have a handful of relevant parameters that help the user better explore the data. By interacting with JavaScript input features, such mouse hover, the user can choose to see data about a specific category.

Triggers:

The triggers are implemented via events and callbacks, such as “mouseover”, “mouseout” and “click”. When specific html elements experience these events, the callback function is invoked, and we can change the visualization with respect to the action indicated by the trigger. The example triggers I have used here are the hovering labels and legend.