CS732 Data Visualization - Assignment 3
Browser-based Visualizations

Shathin R Rao [**MT2020067**]

Abstract

This report describes the two data sets used for visualization: **Infect Dublin** - A Network Data set and **AH Sickle Cell Disease Deaths** - A Multivariate Data set, the library used to create these visualizations and the visualizations themselves.

1 About the Data

1.1 Network Data

A network dataset is a dataset typically consisting of nodes and edges, with the nodes representing different real world subjects like a person, a company, etc and the edge representing some sort of connection between the nodes.
The network dataset chosen for visualization is the **Infect Dublin** dataset from Network Repository. Infect Dublin represents a human contact network where nodes represent humans and edges between them represent proximity (i.e., contacts in the physical world). It consists of a total of 410 nodes and 2.8K edges with a maximum in-degree of 32, a maximum out-degree of 42 and a minimum in- & out-degree of 0.

1.2 Multivariate Data

A multivariate dataset consists of multiple features per data point. Multivariate datasets allow for finding relations between the different features in the dataset.
The multivariate dataset chosen for visualization is the **AH Sickle Cell Disease Provisional Death Counts 2019-2021** from Data.Gov. The dataset provides the provisional death counts of sickle cell disease and coronavirus disease 2019 (COVID-19), by quarter, age, and race or Hispanic origin from 2019 through Quarter 1, 2021.

2 The visualization library - D3

D3.js is a JavaScript library for manipulating HTML elements based on data. D3 allows the visualization artists to bind arbitrary data to a Document Object Model (DOM), and then apply data-driven

transformations to the document. D3 provides a rich set of functions that leverages the Scalable Vector Graphics (SVG), HTML5, and Cascading Style Sheets standards allowing for the creation of a plethora of beautiful visualizations with smooth transitions and interactivity. With D3, the artist is not limited to a certain set of standard visualization techniques; because D3 manipulates HTML, any type of visualization can be created with the right code. This level of flexibility does make D3 very verbose and hence has a steep learning curve.

3 Assignment Structure & Code Execution

The code for this assignment follows a structure of a website, with one main HTML file - `index.html` which contains buttons that help in navigating to the different visualizations that are described in the next section and multiple sub-folders containing the code files corresponding to a specific visualization. Hence, there is no need to run each visualization individually and instead, only the main `index.html` can be executed.

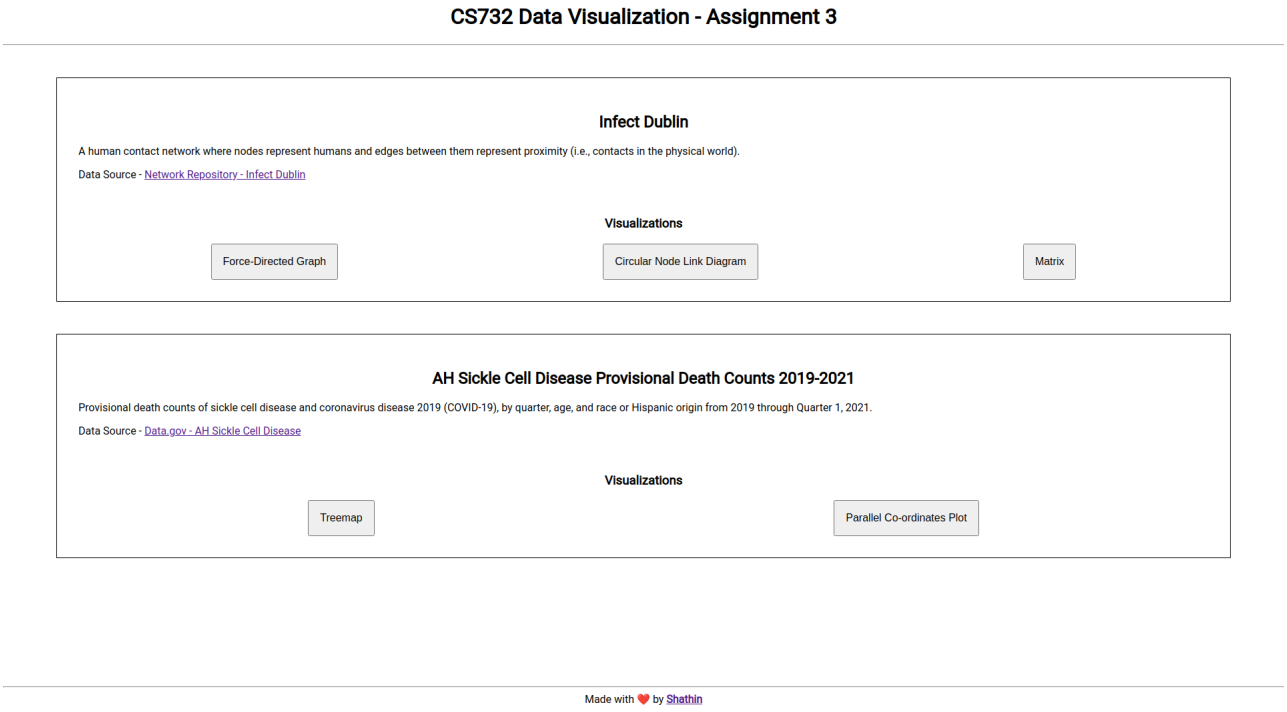

Figure 1: Home Page

To run the website locally, the Node Package Manager (npm) must be installed in the system. Post installation, open the terminal inside the directory containing the website code and execute the command `npx http-server ./ --cors`. The site can be accessed at the URL provided by the `http-server`.

If one wishes to avoid any hassle with the above process of setting up and running the visualization

site locally, the code (and the site) has been setup in CodeSandbox as well. The CodeSandbox version can be accessed here - https://hr7i6.csb.app/. The code can be viewed by the clicking on the "Open Sandbox" button on the bottom right side of the screen.

4 Visualizations

A total of 5 visualization techniques were applied using the JavaScript library D3.js. Each of the visualizations has some form of interactivity.

4.1 Visualizations of the Network Dataset

Three visualization techniques were applied to the network dataset - Force Directed Graph, Circular Node Link Diagram and Matrix Visualization.

4.1.1 Force Directed Graph

A force directed graph is an aesthetically pleasing way of drawing network graphs. Force Directed Graphs position the nodes in a 2D or 3D space and then assign forces to the nodes, like collision forces, gravitational force, etc.

The **D3 Force** package was used for simulating with the forces. This package helps automatically compute the ideal positions of the nodes and how the nodes interact with each other. It also allows the artists to supply arguments that affect the intensity of the force.

There are 4 forces in action -

- Force Link → The link force pushes linked nodes together or apart according to the desired link distance. The strength of the force is proportional to the difference between the linked nodes' distance and the target distance, similar to a spring force.
- Force Many Body → The many-body (or n-body) force applies mutually amongst all nodes. It can be used to simulate an attractive or repulsive force. And unlike link force which affects only two nodes, the charge force is global: every node affects every other node, even if they are on disconnected subgraphs.
- Force Centre → The centering force translates nodes uniformly so that the mean position of all nodes is at the given position (x, y), where x and y are the desired co-ordinates in the 2D space.
- Force Collide → The collision force treats nodes as circles with a given radius and prevents nodes from overlapping.

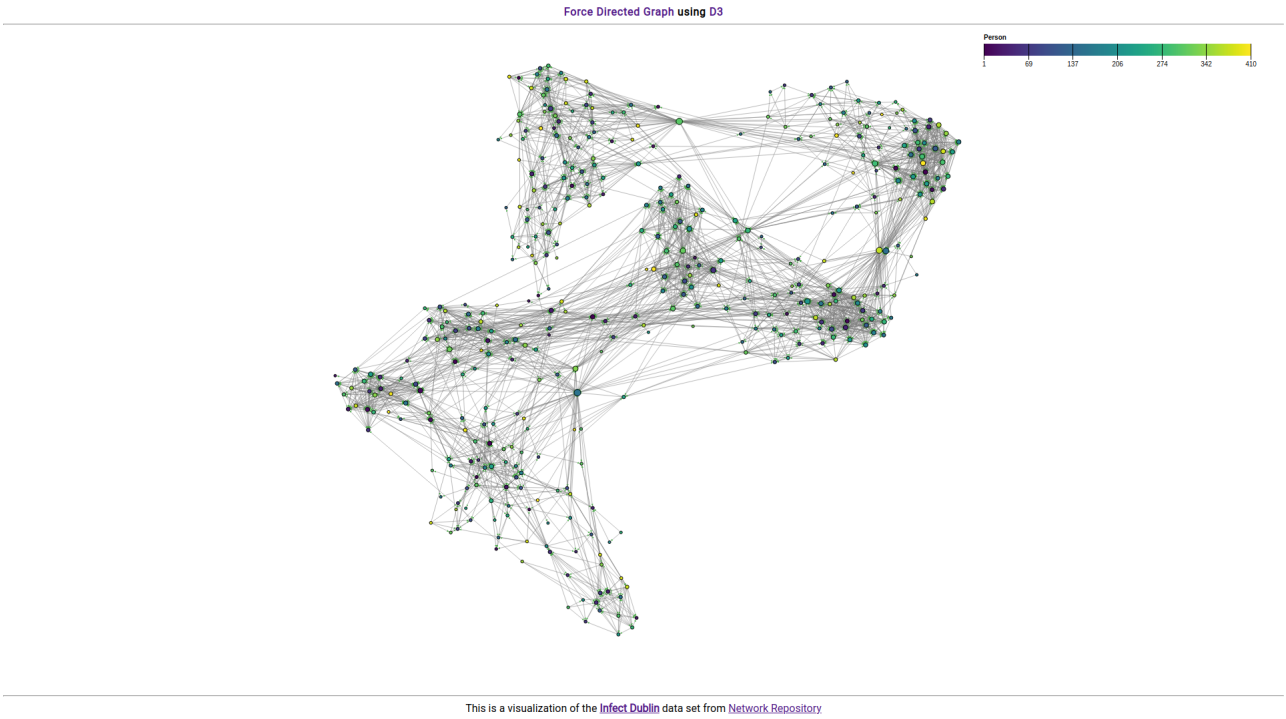

Figure 2: Force Directed Graph

Each person (node) in the graph is represented by a circle and their proximity to other people is represented by lines linking the nodes. The coloring scheme used for coloring the nodes is Viridis. Viridis is a robust color scheme which helps improve graph readability for readers with common forms of color blindness and/or color vision deficiency. The nodes are colored based on the node number i.e., 1, 2, 3,....,410. The size of each node is dependent on the total number of links i.e., the number of incoming and outgoing edges to and from the node - more the number of links, larger is the size.

The visualization created supports the following interactions -

- Hovering over the nodes displays a tooltip with Node (person) number, the number of incoming and outgoing links.
- Hovering over the nodes also focuses that node, along with the connecting edges and the nodes that have direct links to the hovered node. All the other nodes & links are put out of focus.
- Hovering over the edges / links displays a tooltip with the link information showing which two nodes are linked by that edge and in what direction.
- Hovering over the edges also focuses that edge and the two connected nodes while putting all the other nodes & edges out of focus.
- The nodes can be dragged around. This allows for viewing the forces in action.

- The visualization can be zoomed into or out of for better view by using the mouse / trackpad scroll. Furthermore, the visualization, as a whole, can be dragged around.

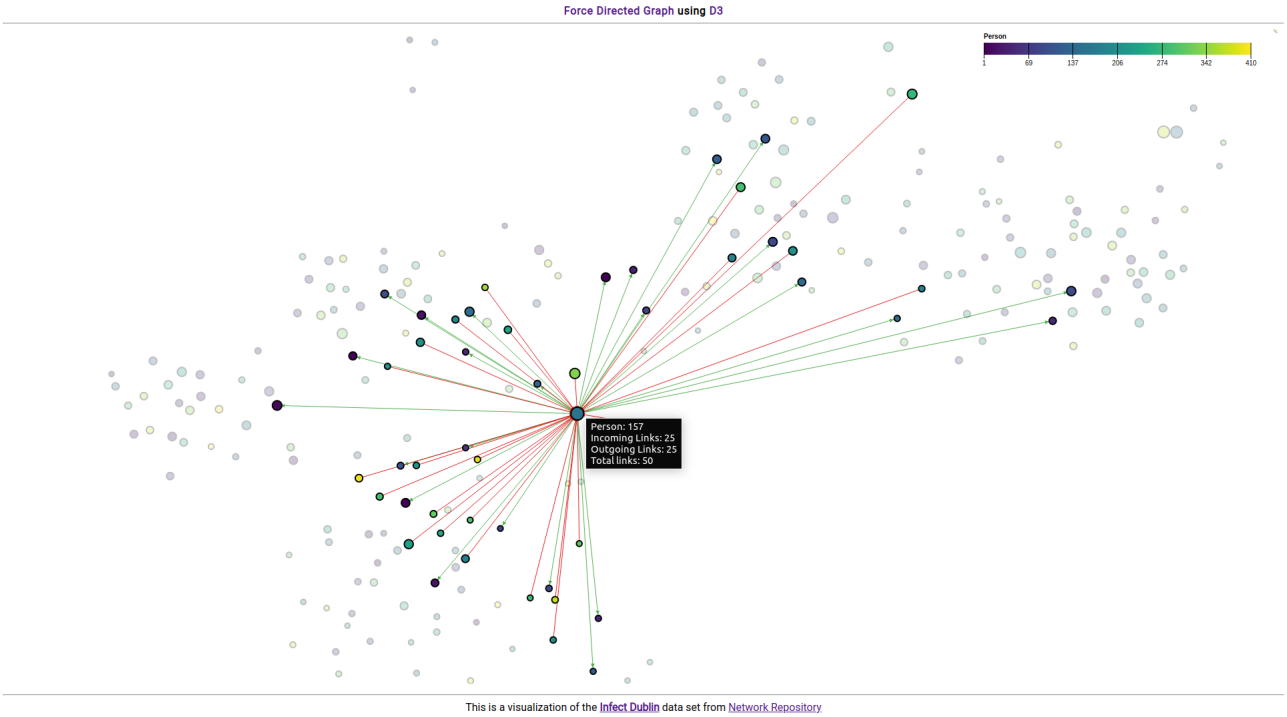

Figure 3: Force Directed Graph - Hovering over a Node

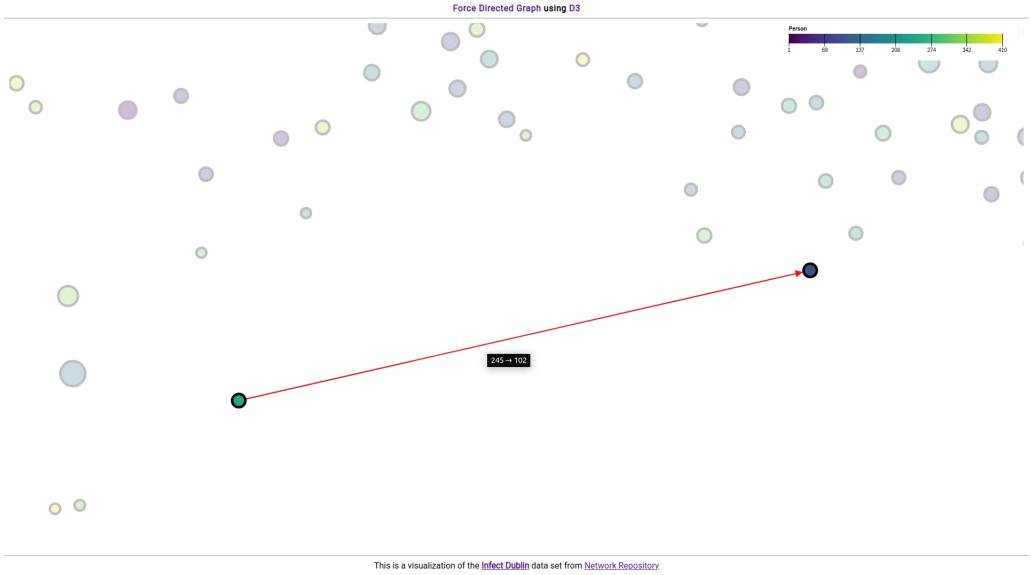

Figure 4: Force Directed Graph - Hovering over an Edge

No much data re-modelling was performed on the data set in order to enable visualizing it as a force-directed graph. The only processing that was done attaching the number of incoming, outgoing & total links to each of the nodes; this was done to provide relevant information when hovering over the nodes.

4.1.2 Circular Node Link Diagram

The circular node link diagram is based on the force-directed graph. The difference being that only the link force is applied and the positions of the nodes are fixed along the circumference of a circle (or ellipse).

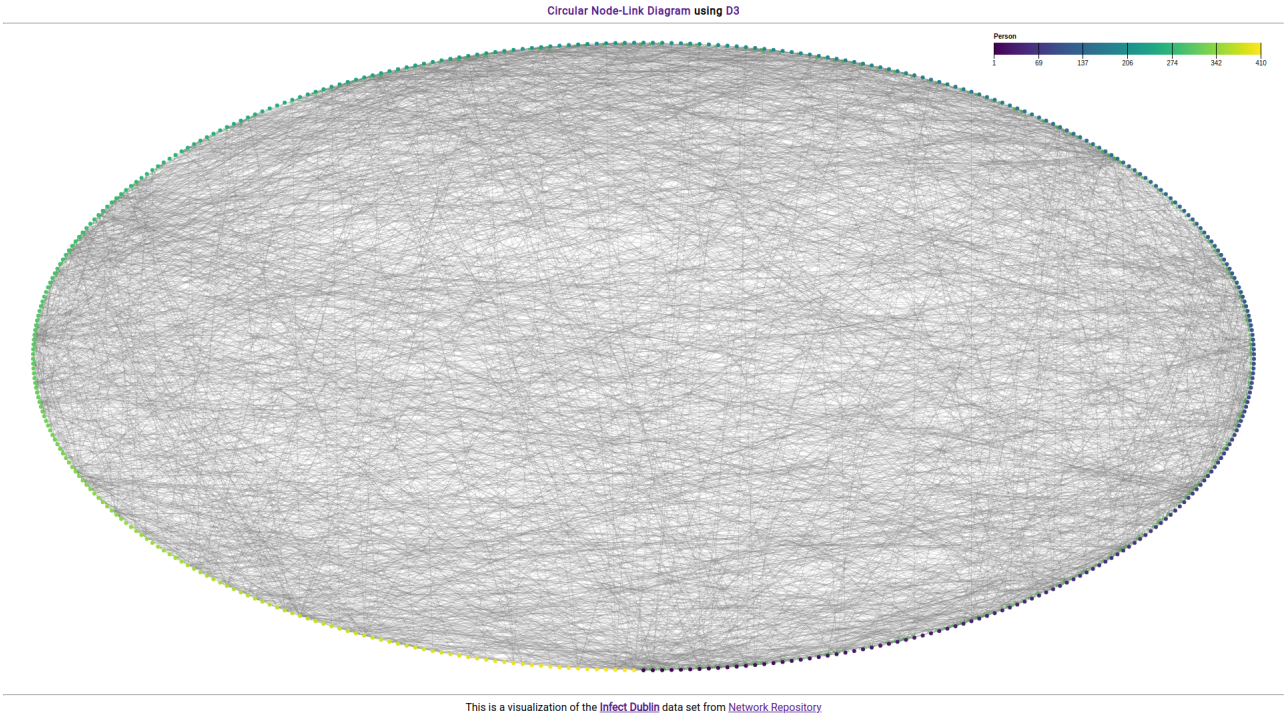

Figure 5: Circular Node Link Diagram

Similar to the force-directed graph visualization, each person (node) in the graph is represented by a circle and their proximity to other people is represented by lines linking the nodes. The color scheme used for coloring the nodes in Viridis. Unlike the force-directed graph, all nodes have the same radius in order to accommodate for a uniform distribution over the circumference of the circle (or ellipse). The visualization created supports the following interactions -

- Hovering over the nodes displays a tooltip with the Node (person) number, the number of incoming and outgoing links.
- Hovering over the nodes also focuses that node, along with the connecting edges and the nodes that have direct links to the hovered node. All the other nodes & links are put out of focus.

- Hovering over the edges / links displays a tooltip with the link information showing which two nodes are linked by that edge and in what direction.
- Hovering over the edges also focuses that edge and the two connected nodes while putting all the other nodes & edges out of focus.
- The visualization can be zoomed into or out of for better view by using the mouse / trackpad scroll. Furthermore, the visualization, as a whole, can be dragged around.

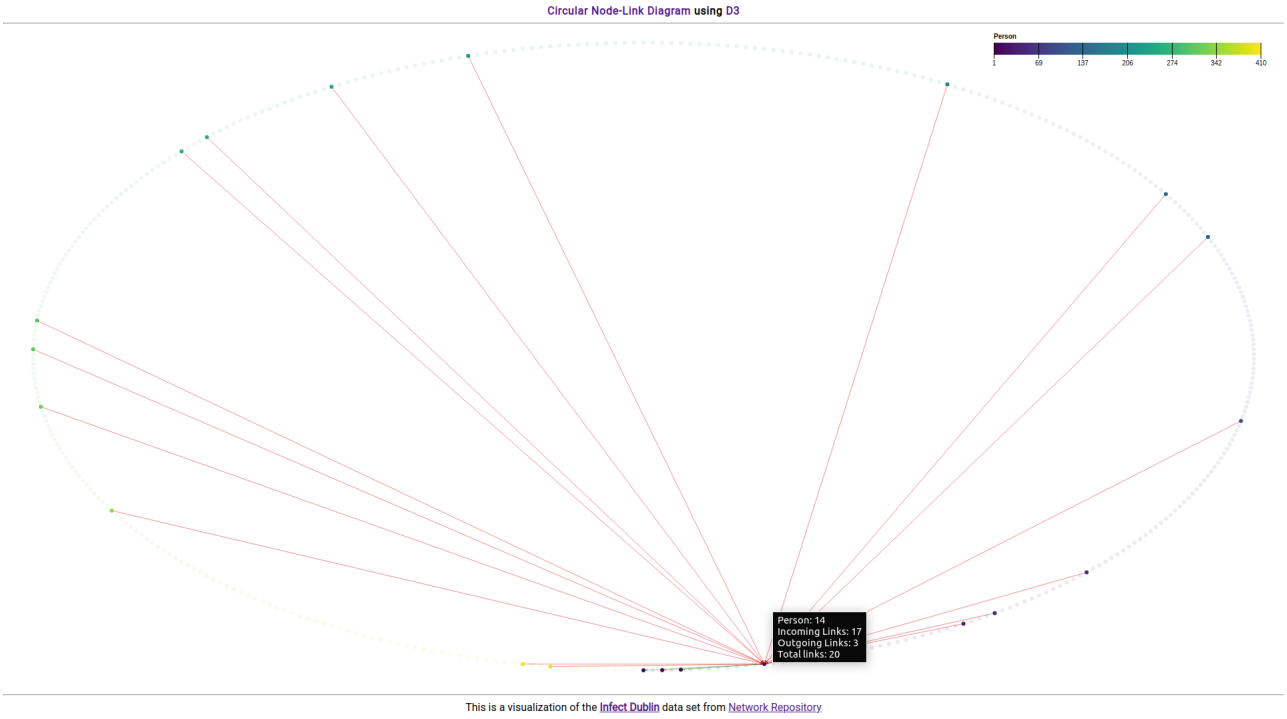

Figure 6: Circular Node Link Diagram - Hovering over a Node

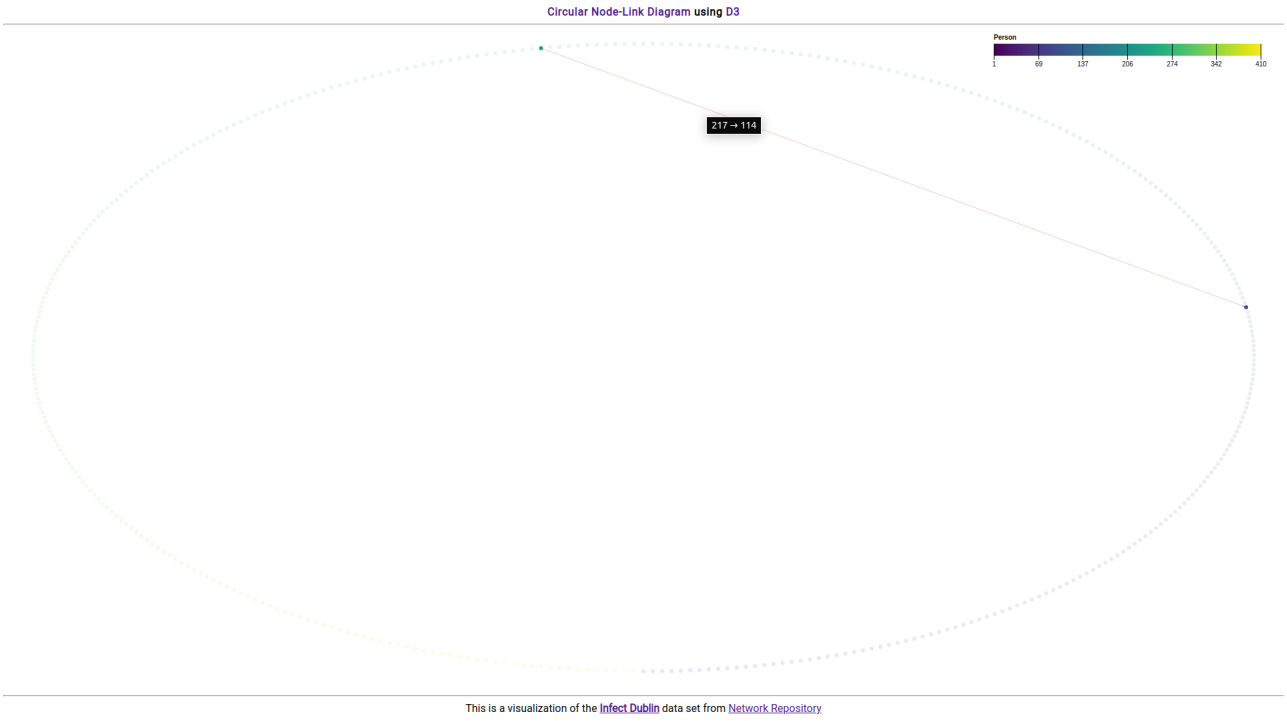

Figure 7: Circular Node Link Diagram - Hovering over an Edge

Similar to the force-directed graph, no much data re-modelling was performed on the data set in order to enable visualizing it as a force-directed graph. The only processing that was done attaching the number of incoming, outgoing & total links to each of the nodes; this was done to provide relevant information when hovering over the nodes.

4.1.3 Matrix Visualization

Matrix visualization is a graphical technique that can simultaneously explore the associations between thousands of subjects, variables, and their interactions, without needing to first reduce the dimensions of the data.

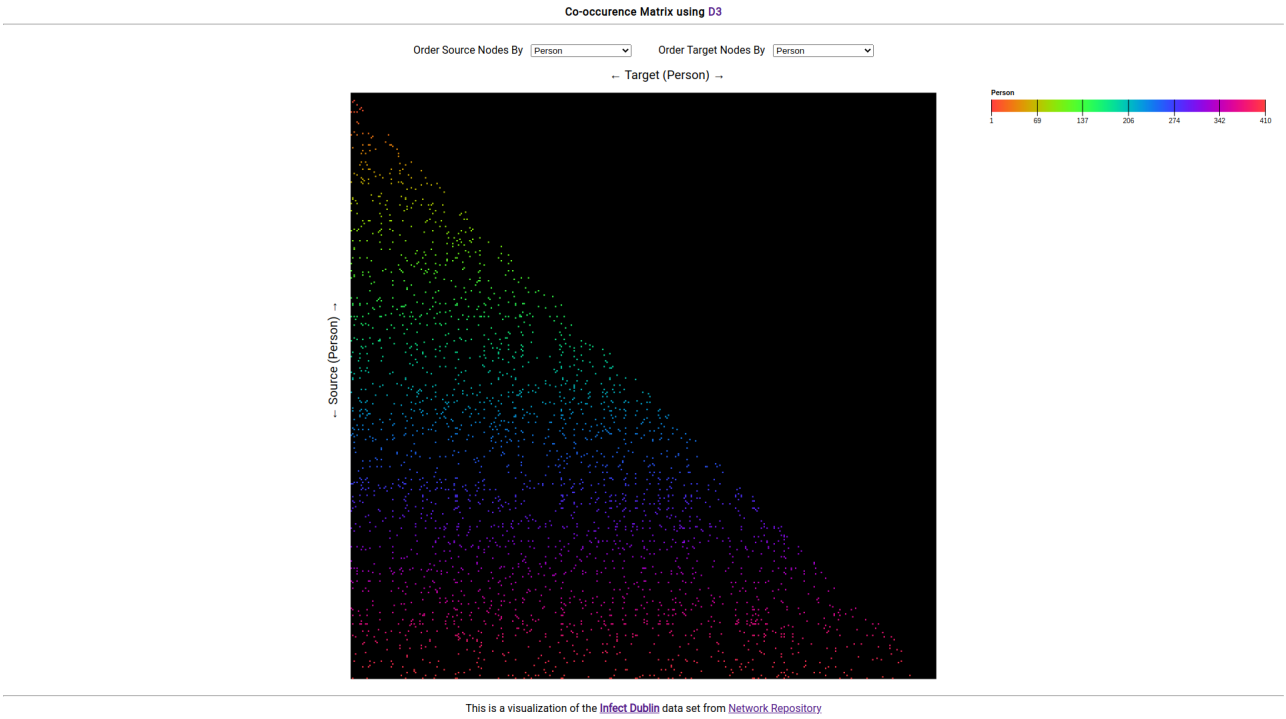

Figure 8: Matrix Visualization

Unlike the previous two visualization, in the matrix visualization, the nodes (source & target) are placed in a adjacency matrix and the same is displayed on the screen with the pixels that represent a connection being painted a particular color while no connection is represented as a black pixel. Seriation techniques help in finding out any sort of patterns in the data. The visualization created supports the following interactions -

- Hovering over a colored pixel displays a tooltip showing the information of both the source and target nodes. This information contains the node number, number of incoming and outgoing links.
- Two drop down menus provide options to sort the source or target nodes by either the node (person) number, number of incoming links, number of outgoing links or total number of links. Each of them can be individually sorted based on a different parameter.
- The label on the left and top of the matrix that represent which type of node lies on that axis i.e., source or target node, can be double clicked; doing so sets the color scheme to that axis.
- The visualization can be zoomed into or out of for better view using the mouse / trackpad scroll. Furthermore, the visualization, as a whole, can be dragged around.

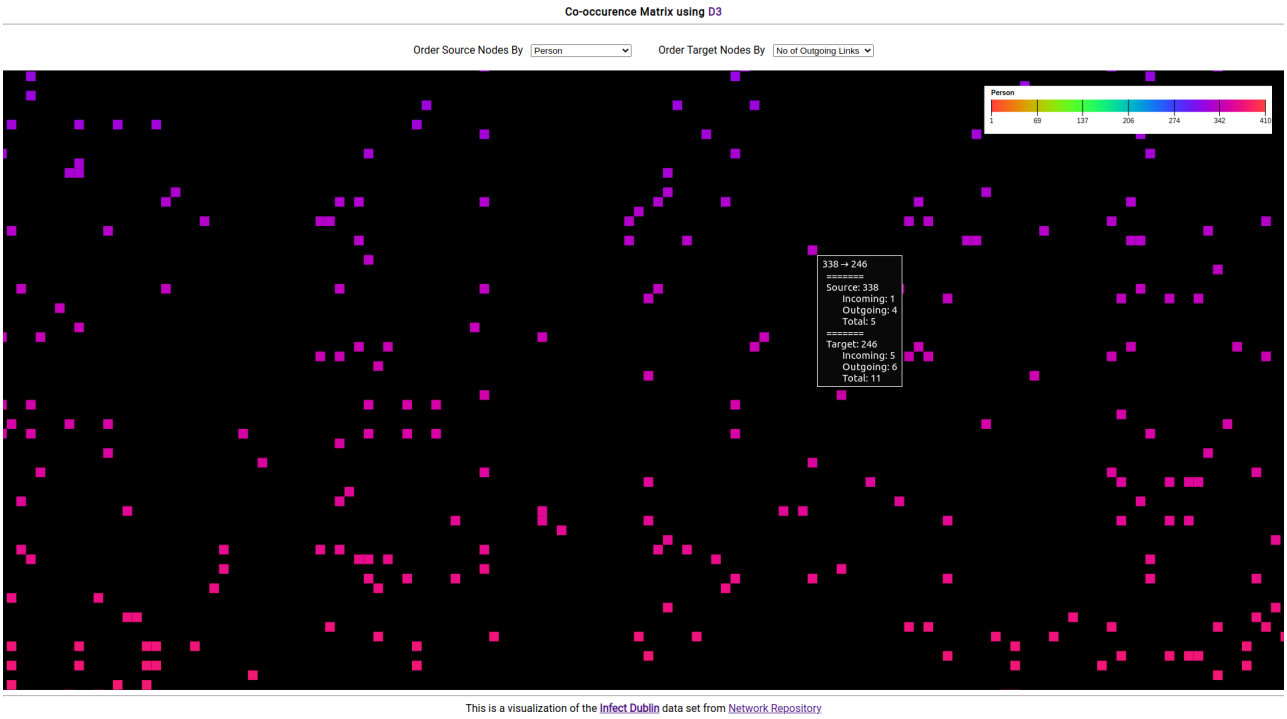

Figure 9: Matrix - Hovering over a Pixel

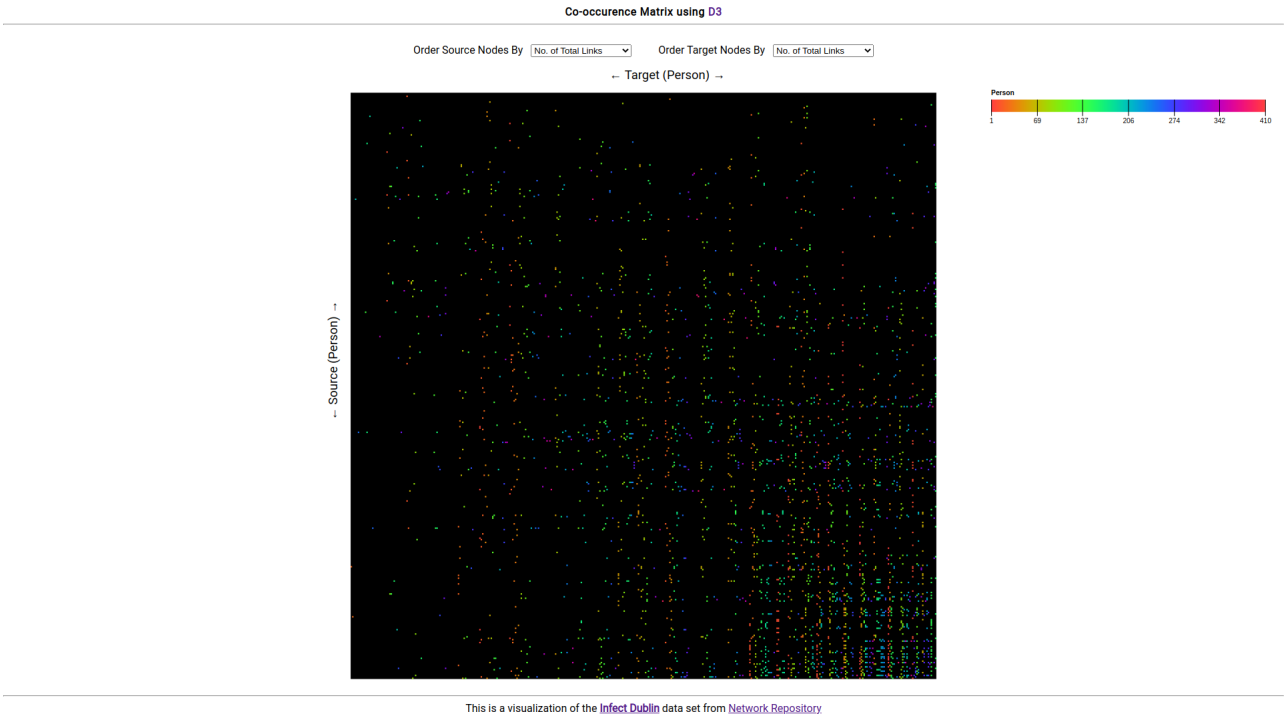

Figure 10: Matrix - Sorting both axis by Total Links

Data was slightly re-modelled to create the matrix data. No seriation algorithms were applied!

4.2 Visualizations of the Multivariate Dataset

Two visualization techniques were applied to the multivariate dataset - Treemap and Parallel Coordinates Plot.

4.2.1 Treemap

Treemaps display hierarchical data as a set of nested rectangles. The **D3 Group(s) method** and **D3 Hierarchy** package was used for converting the given flat CSV data set into a hierarchical data. The package computes the coordinates of the rectangles of the treemap based on the specified hierarchy.

Two types of treemaps were generated - Nested Treemap and Zoomable Treemap; each providing its own set of interactions.The user can can change the treemap type by using the the radio buttons provided on the top of the page.

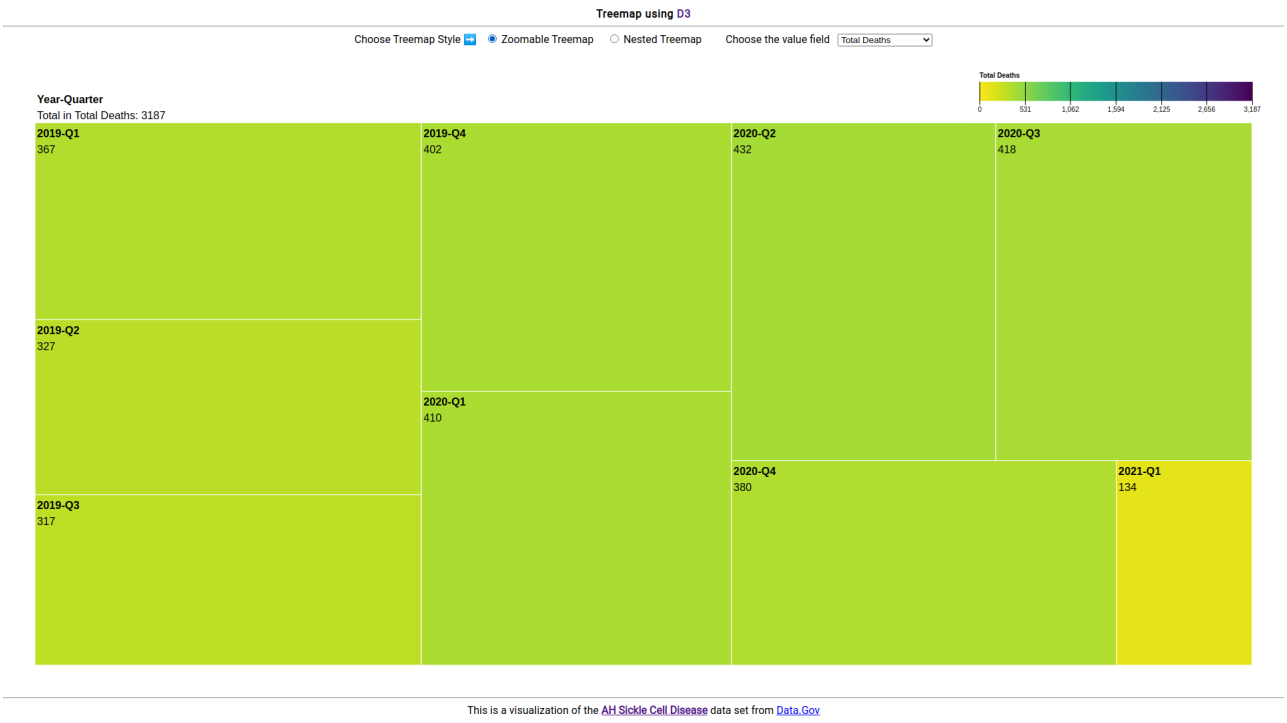

Figure 11: Zoomable Treemap

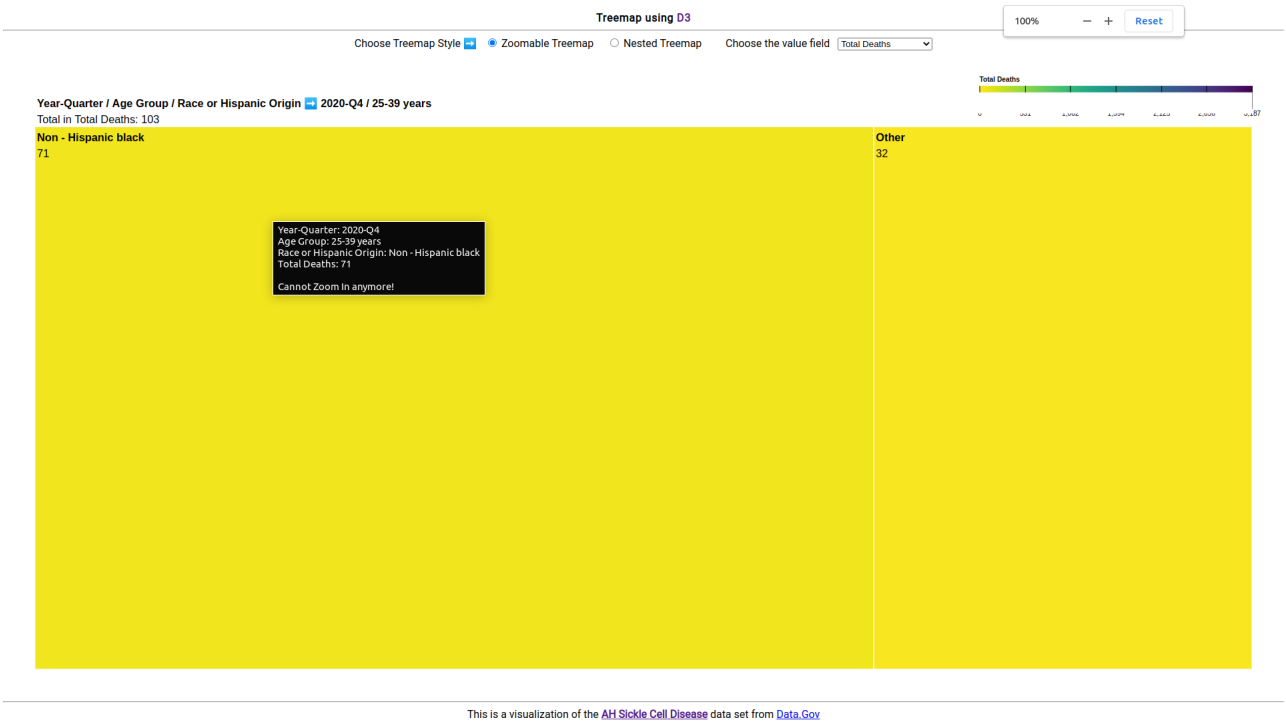

Figure 12: Zoomable Treemap - Hovering over a Rectangle

- **Zoomable Treemap** → The zoomable treemap provides a click to zoom-in & zoom-out functionality. The user can click on any one of the rectangles of the treemap to zoom into that section and traverse the hierarchy. To traverse backwards, the user can click on the breadcrumbs on the top of the treemap.
- **Nested Treemap** → This treemap variant applies padding to label internal nodes, better revealing the hierarchical structure but it is much more congested compared to the zoomable treemap. Furthermore it doesn't support a click to zoom in / out functionality since all the data is displayed at once. It however does provide a regular mouse / trackpad zoom to help the user get a closer look of the treemap.

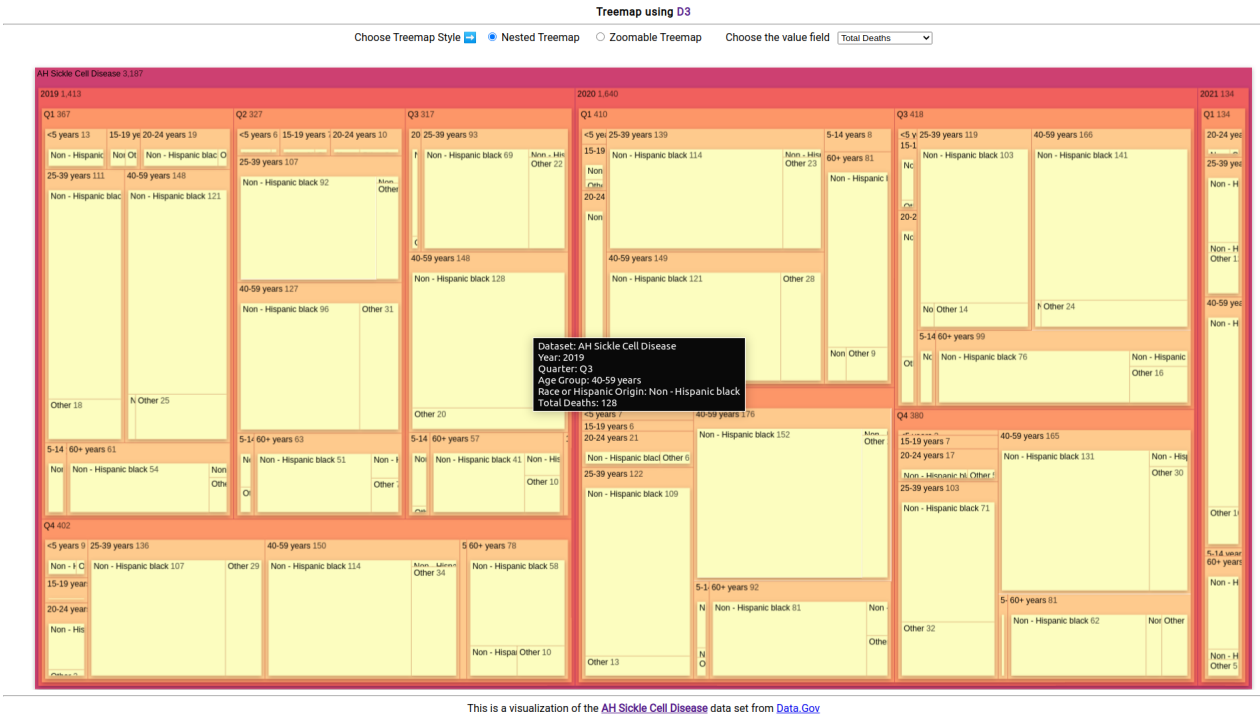

Figure 13: Nested Treemap - Hovering over a Rectangle

Common interactions -

- Hovering over the rectangles reveals the information about the datapoint at that level. Hovering over rectangles at the deeper levels reveals more information.
- The dropdown menu can be used to change the numerical field of the data used to compute the size of the rectangles of the treemap.

For both the types of treemaps, the data was re-modelled into a hierarchical dataset from a flat data set. For the zoomable treemap, two attributes - Year & Quarter were combined into one to reduce the depth of the treemap.

4.2.2 Parallel Co-ordinates Plot

In a Parallel Coordinates Plot, each variable is given its own axis and all the axes are placed in parallel to each other. Each axis can have a different scale, as each variable works off a different unit of measurement. Parallel Coordinates Plots are ideal for comparing many variables together and seeing the relationships between them.

Axis Swapping & Brushing helps in finding the co-relation between the different attributes of the data.

The visualization created supports the following interactions -

- Axis Swapping - The axes can be re-ordered by clicking on the axis name and dragging it.
- Brushing - Only the required the data lines can be selected by click and dragging over the axis line.
- The axis used for the color scheme can be changed via the drop down menu on the top of the page. Any of the columns can be selected.

No much data re-modelling was performed expect that two attributes Year & Quarter were combined into one to reduce the number of axes.

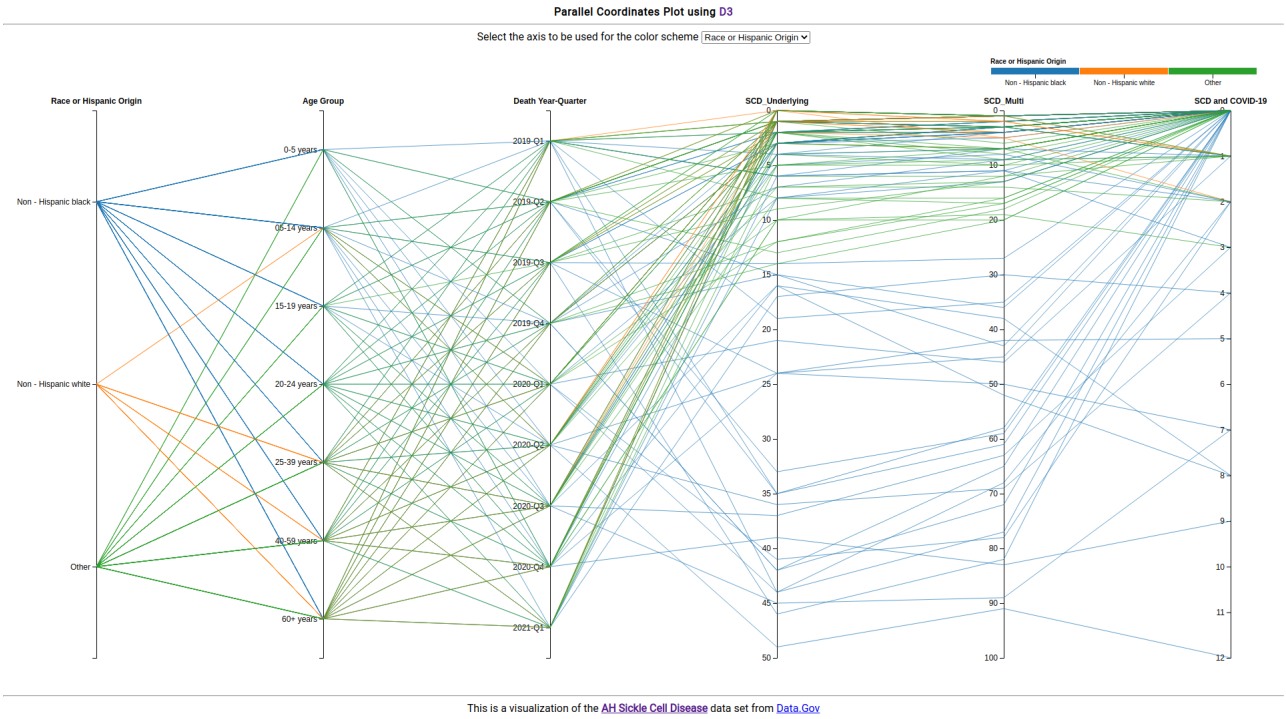

Figure 14: Parallel Co-ordinates Plot

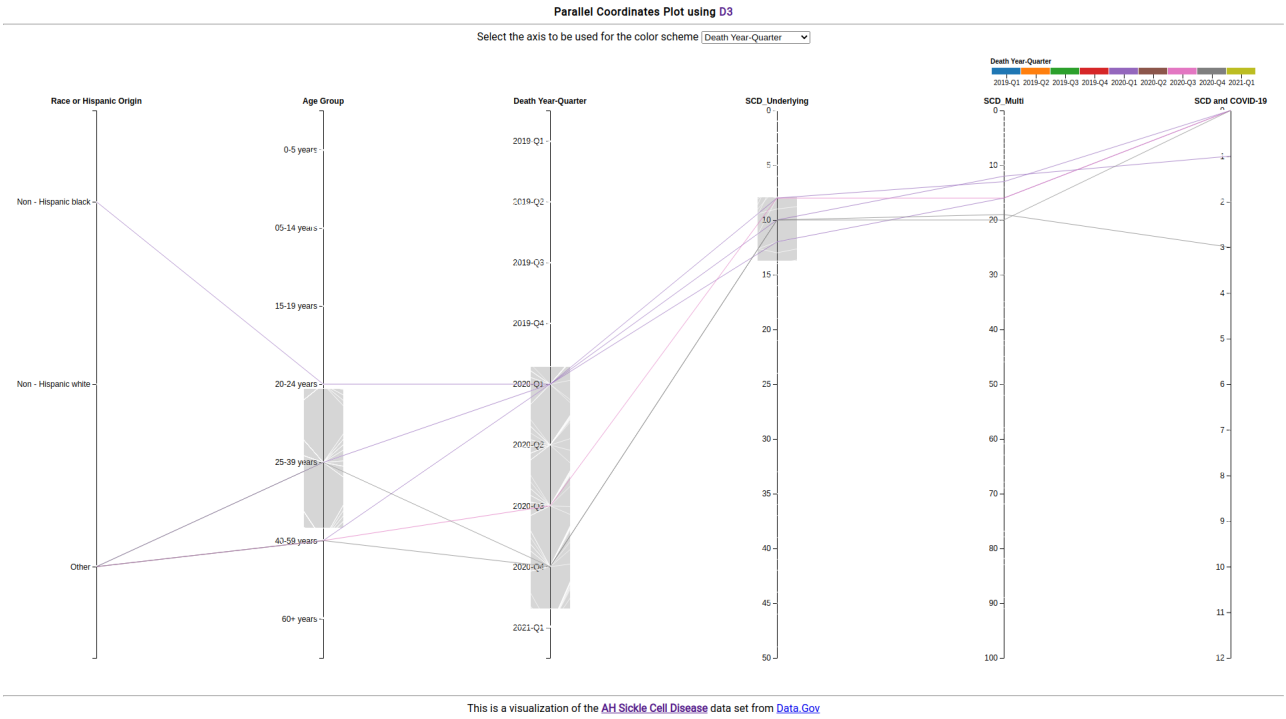

Figure 15: Parallel Co-ordinates Plot - Brushing

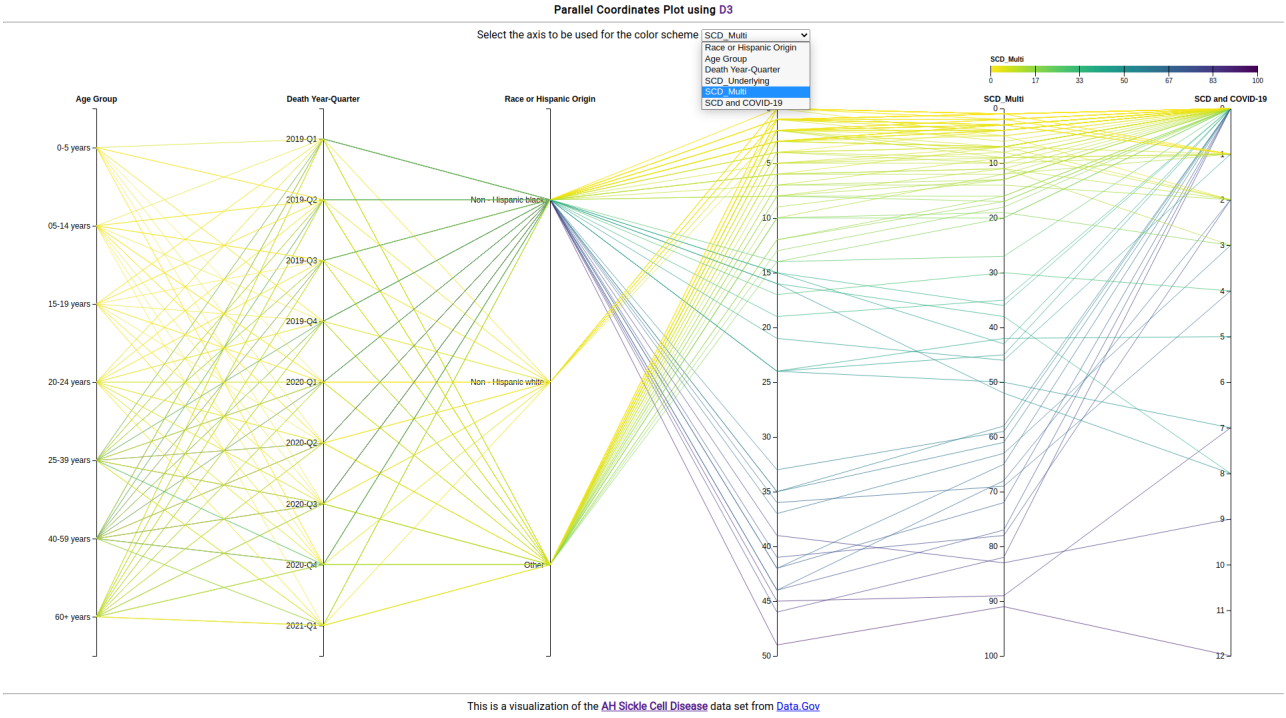

Figure 16: Parallel Co-ordinates Plot - Changing Line Color

5 Conclusion

In conclusion, the report has successfully elaborated on the visualizations performed and the tools & methodology used to perform the required tasks.

6 References

1. Data Driven Documents - D3.js (https://d3js.org)
2. D3 Example Visualizations on Observable (https://observablehq.com/@d3/gallery)
3. D3 Repository & Documentation (https://github.com/d3)