COVID-19 Trends Analysis: Exploring Disease Conditions, Vaccination Rates and Mortality Trends

Anjali Augestin
Master of Science in Data Analytics
National College of Ireland
Dublin, Ireland
x23155086@student.ncirl.ie

Nayana Palathinkal Shibu
Master of Science in Data Analytics
National College of Ireland
Dublin, Ireland
x23174960@student.ncirl.ie

Gayathri Gangadharan
Master of Science in Data Analytics
National College of Ireland
Dublin, Ireland
x22203427@student.ncirl.ie

***Abstract*—The COVID-19 pandemic has been a devastating global event, impacting countless lives and causing significant health challenges, particularly affecting the brain, lungs, heart, kidneys and blood vessels. SARS-CoV-2, the virus that causes COVID-19, has directly contributed to mortality whereas other underlying illnesses have also played a role in the overall mortality rate. To mitigate the impact of COVID-19 on the public health, several vaccines have been developed such as Pfizer-BioNTech, Novavax and Moderna Vaccines. The objective of the study is to analyze COVID-19 trends and explore how various pre-existing disease conditions may influence COVID-19 mortality rates. In addition to that, the impact of vaccination on reducing mortality associated with COVID-19 is also assessed.**

***Index Terms*—Python, Docker, MongoDB, GitHub, PostgreSQL**

I. Introduction

The project aims to analyse the COVID-19 mortality rate, conditions contributing to Covid-19 death and the effect of vaccination among different age group in United States. The semi-structured datasets are programmatically stored, analyzed, transformed to understand the individual datasets. Moreover, the datasets are merged to derive insights and research questions based on the underlying situation. Then visualizations are made on the merged datasets to derive insights from the datasets.

II. Literature Review

Alaa M. O. Abdelsamad and Azza Z. Karrar [1] proposed a paper, in which an interactive dashboard was designed to monitor the status of COVID-19 in Sudan. Their aim was to help the public and health authorities to understand the exact situation of the people of the country and they used Tableau for creating the visualizations. The dashboards and visualizations provided significant insights, which in turn are useful for the authorities to aware about the spreading and the worse condition of the society.

The paper proposed by M R Mufid and et.al [2], proposes a system to provide the latest information about the development of the COVID-19 case in Indonesia and introduces an expert system to identify the COVID-19 cases independently and faster. They used web scraping technique to extract data from the original website to the site they developed for the system. The expert system acts as a early detection system which also provides the information to the web application. The results are given based upon the knowledge gained from the web scrapping and from the expert system. Both the data are combined to fetch results and it is shown in the we application.

B. Sharma and et. al [3], proposed a paper which discusses about the COVID-19, which helps the people to understand about the virus and its severity. Moreover, it gives insights about how to conduct further analysis and visualizations. It also helps us to know which all factors related to and affecting COVID-19, need to be studied and explored more. Also, they strongly promote the need for conducting more complex research on the underlying cause and effects of the virus.

The paper proposed by L. G. Wiseso and et.al [4], examines the performance of PostgreSQL, MongoDB, and Neo4J and then analyze each of its complexity using computational complexity theory with Big O notation as its tool. The study helps us to know the better performing and secured database. The paper gives us insights regarding which database should be used according to the specifications of the datasets. The conclusion of their analysis indicates that MongoDB has excellent performance because it has an O(1) complexity, PostgreSQL has good performance because it has an O(n) and O(1) complexities, and Neo4j has worse performance than MongoDB and PostgreSQL because it has an O(nlog n) complexity.

I. Stančin and A. Jović [5], proposed a paper in which, they describe and compare different data mining and big data analysis libraries in Python. They analysed more than 20 libraries, also classified them into six groups as core libraries, data preparation, data visualization, machine learning, deep learning and big data. The classification helped us to understand the libraries deeply and enabled to comprehend which libraries are perfect for different data analysis and processing. As the conclusion, the paper recommends ‘pandas‘ as the best library for data preparation

Identify applicable funding agency here. If none, delete this.

and transformation. They also conclude ‘matplotlib‘ and ‘plotly‘ for the better visualizations of the data as per their comparison study. Based on these conclusions we selected the particular libraries for our project.

III. METHODOLOGY

A. Description of Data

The three given datasets were sourced from the DATA.GOV website.

1) AH Monthly Provisional Counts of Deaths for Select Causes of Death by Sex, Age, and Race and Hispanic Origin: : This dataset provides provisional counts of deaths occurred by month, categorized by age group, sex and race, for specific underlying causes of death during the years 2020-2021 along with the finalized data for the year 2019. In addition to that, it also contains monthly provisional counts of deaths due to COVID-19, either as an underlying cause or a contributing factor to the deaths.

2) Conditions Contributing to COVID-19 Deaths, by State and Age, Provisional 2020-2023: : This dataset provides information on health conditions and other contributing factors associated with deaths due to COVID-19, categorized by age group and location of occurrence. The data for the years 2022 and 2023 are provisional, while estimates for 2020 and 2021 are based on finalized data.

3) COVID-19 Vaccination Coverage, ZIP Code: : This dataset provides Chicago residents’ COVID-19 vaccination status by ZIP Code and age group based on home address data from the Illinois Comprehensive Automated Immunization Registry Exchange (I-CARE). The dataset also includes information about the number of individuals vaccinated and the corresponding vaccination coverage percentages.

B. Technologies and Libraries

1) Technologies utilized:

- **Jupyter Notebook**: The Jupyter Notebook is an open-source web application for creating and distributing computational documents. It supports various programming languages, including Python, R and Julia, allowing to write and execute code interactively in a flexible and collaborative environment. This project is implemented using the Python programming language.
- **Docker Desktop**: The Docker Desktop is a containerization software that provides developers with a comprehensive toolkit to easily build, share and run applications across different environments.
- **MongoDB Compass**: The MongoDB Compass is the graphical user interface for MongoDB used to interact with the MongoDB databases. It helps in keeping databases streamlined and offers a simple way to aggregate data.
- **pgAdmin**: pgAdmin is an open-source platform of the PostgreSQL. The PostgreSQL is a powerful open-source Relational Database Management System which supports SQL queries, indexing and transactions.
- **GitHub**: The GitHub is an open-source AI-powered platform for hosting and collaborating on GitHub repositories. The key feature of GitHub is its support for collaborative coding which enables multiple developers to work together on the same project.

2) Libraries utilized:

- **seaborn**: It is a library used for making statistical visuals in Python. It is like a special tool box that works well with pandas data structures and offers advanced connection to matplotlib.
- **pymongo**: This distribution contains different tools for connecting with MongoDB database from Python.
- **requests**: Requests are an Apache2 licensed HTTP library in Python and it contains many features to result in productivity.
- **pandas**: Pandas provides different data structures and functions crafted to make working with structured data fast, simple and expressive.
- **numpy**: Numerical Python or numpy is an essential package for scientific computing in Python.
- **matplotlib**: It is a popular data visualization library in Python. It is often used for designing static, interactive and animated visuals in Python. It can save images in several output formats like PNG, PS and others.
- **plotly**: It is a powerful Python library to create interactive data visualizations. It supports a variety of chart types like simple line plots and scatter plots to more advanced charts namely histogram, box plots, heat map and 3D surface plots.
- **SQLAlchemy**: It is a library used for interacting between Python programs and different databases. It is commonly used as a tool that transforms Python classes into relational database tables and automatically converts function calls to SQL statements.
- **pyscopg2**: psycopg2 is a popular Python library which allows Python applications to connect to PostgreSQL databases, to execute SQL queries and manage database transactions

C. High Level Architecture

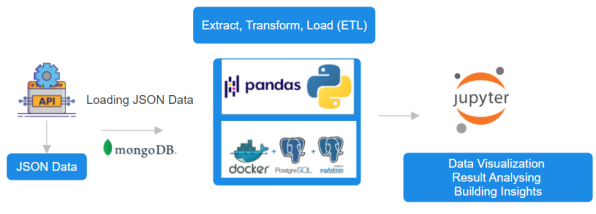

Fig. 1. High Level Block Diagram

The Fig 1 illustrates the high level architecture that was followed in our project. All the three datasets are extracted using API in JSON format. We used the MongoDB Compass to store our semi-structured data and programmatically loaded the data to the database. Next the datasets are Extracted, Transformed and Loaded (ETL) to the appropriate database. The datasets are extracted from from the database, then they are analysed, processed and transformed separately. We used pandas Python library for ETL process.
After that, the transformed datasets are loaded into Pgadmin (postgreSQL). All the processed datasets are stored inside the database. The structured datasets are then extracted for further analysis. They are merged accordingly to derive insights and for visualizations.

D. Data Pre-processing

1) ***Data Loading****:* Initially, a docker instance was created. In order to efficiently store data, a MongoDB server is deployed programmatically with docker instance. We connected to MongoDB server through code and created a database and collection in it programmatically. We used APIs to extract JSON data automatically from the websites and to make the retrieval process easier. Then this extracted JSON data is integrated to MongoDB seamlessly, and organized in specific collections and database structures.

2) ***Data Analyzing****:* We retrieved data from MongoDB collection as a cursor object using PyMongo and converted it into a pandas Dataframe for further analysis in Python. After loading the data, we performed data analysis to understand each of our datasets' structure and qualities. While analyzing, we checked for the existence of missing values in our data set. Then we handled missing values by replacing them with appropriate metrics such as mean and mode. We dropped some of the columns that are not relevant to our analysis.

3) ***Data Transformation****:* In data transformation, we transformed a key column which is common in our 3 datasets using dictionary mapping in Python. We transformed a column which contains the age group of people to a column with the common name 'namesbyAge' with different categories of range such as 'Kids', 'Young', 'Adult', 'Middle Aged', 'Senior', 'Old Age', 'All Ages' for data merging.

E. Data Visualization

The datasets are visualized individually using various charts and plots.

- **AH Monthly Provisional Counts of Deaths for Select Causes of Death by Sex, Age, and Race and Hispanic Origin**:
  Fig 2 shows count of COVID-19 cases as an underlying cause of death by Age and Sex

Fig 3 depicts count of COVID-19 cases as multiple cause of death by Age and Sex.

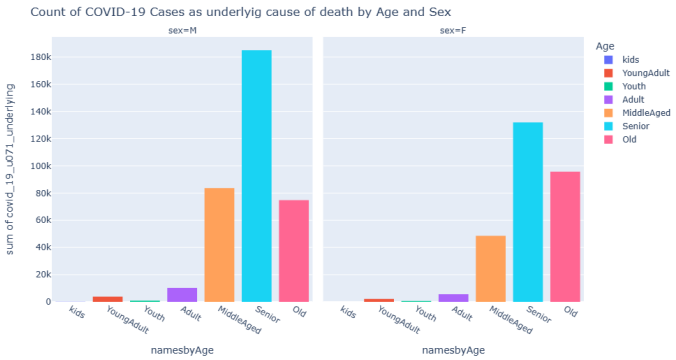

Fig. 2. Count of COVID-19 Cases as underlying cause of death by Age and Sex.

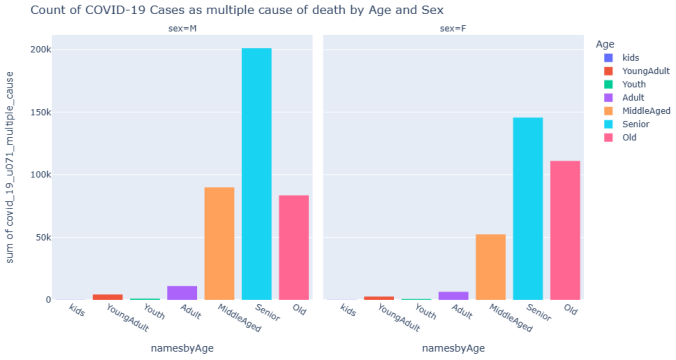

Fig. 3. Count of COVID-19 Cases as multiple cause of death by Age and Sex.

From both the figures, it is evident that seniors have the highest mortality rate, particularly male seniors.

- **Conditions Contributing to COVID-19 Deaths, by State and Age, Provisional 2020-2023** :
  Visualization for covid deaths by age group is seen in Fig 4. In Fig 4, All Ages shows the death count of people across all age groups whereas senior has the highest mortality rate due to COVID-19. This visualization is aligned with the findings from the visualization of the previous dataset that the mortality rate is the highest among seniors.

Fig 5 shows covid death by condition groups which proves that death rate from COVID-19 is higher among individuals with respiratory diseases.

- **COVID-19 Vaccination Coverage, ZIP Code**: Fig 6 depicts the count of individuals who have taken vaccine categorized by age group. In that figure, the number of seniors (64+ yrs) who have received COVID-19 vaccine is relatively low.

The distribution of vaccine counts by zip code is shown in Fig 7.

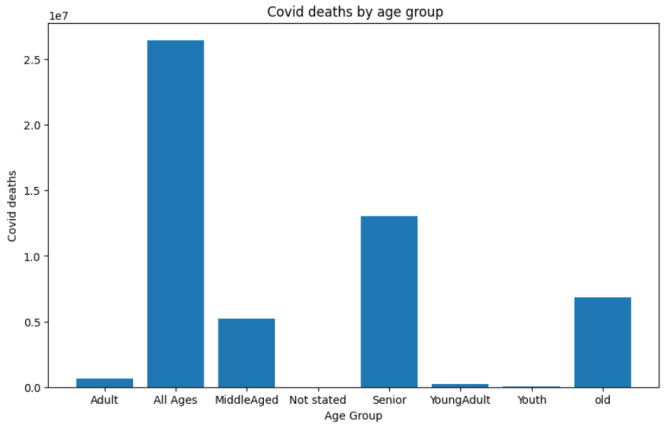

Fig. 4. Covid deaths by age group

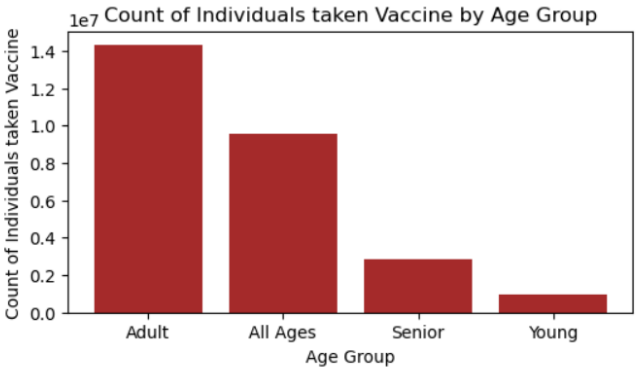

Fig. 6. Count of individuals taken Vaccine by Age Group

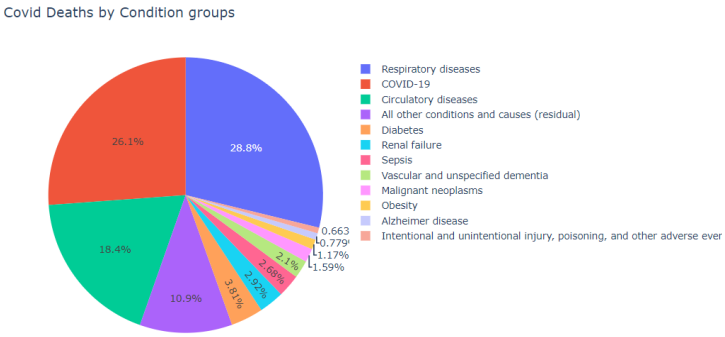

Fig. 5. Covid deaths by condition groups

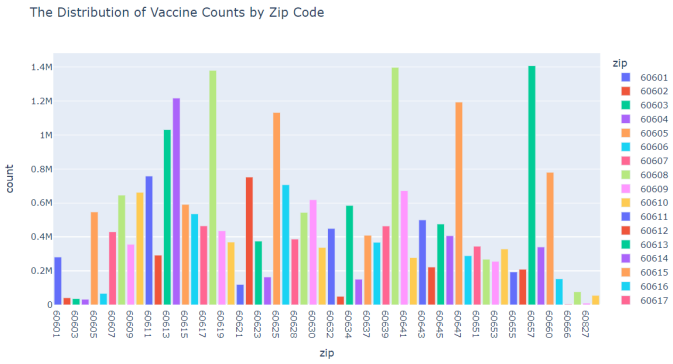

Fig. 7. The Distribution of Vaccine Counts by Zip Code

F. Dataset Loading

All the three cleaned and transformed datasets are loaded using the Pandas library in Python. The first dataset 'updated-setAnjali.csv' now contains information on various conditions and causes of death categorized by age group. After loading the dataset, it is connected to a PostgreSQL database named 'project', where it can be stored for further analysis. This dataset is stored as a table named 'updated_conditions' in the 'project' database.

The second dataset, 'Gayathrig.csv' is loaded and connected to the database in a similar way. This dataset contains information about deaths, including details like age group and causes of death. This dataset is stored as a table 'deathrate' within the same PostgreSQL database.

Next, the third dataset, 'DAPNayanaAssignment.csv' is loaded which contains data related to vaccination, including vaccination percentages. This dataset is stored as 'vaccination' table in the database 'project'.

IV. RESULTS AND EVALUATION

A. Datasets Merging

All the data is read from PostegreSQL and converted into Pandas DataFrame. Initially, 'updated_conditions'-which contains the processed and aggrgated data from the dataset2, is merged with 'deathrate'-the table contains the processed data of dataset 1. The merging is done by the common column 'namesbyAge' in the two datasets. We are trying to find which pre-existing condition is more contributing to the COVID-19 death and people with which conditions are more affected by the virus. This merged data includes information about various conditions and the corresponding COVID-19 related deaths and this merged dataframe is saved as 'mergedOne'.

Finally, the previously merged dataframe is combined with the 'vaccination' data based on age group. This final merged data provides a comprehensive view of conditions, count of vaccinations and COVID-19 related deaths based on age group. This merged dataframe is saved as mergedFinal. In this, we got insights about which age group has lowest vaccination rate, which in turn affected their condition and increased chance of death by Covid-19.

B. Loading Results

Fig 8 shows the datasets loaded to the specific database inside the postgreSQL. The table 'deathrate','covid_condition' and 'vaccination' contains the processed, transformed data of Dataset 1, Dataset 2 and Dataset 3 respectively. The Dataset 2 was large in size, so easy comprehension and accessing the transformed dataframe is aggregated and grouped

according to our requirement and it is stored in the table ‘updated_condition‘.
The resultant dataset after the merging all the three datasets are also loaded into the postgreSQL as table ‘result‘.

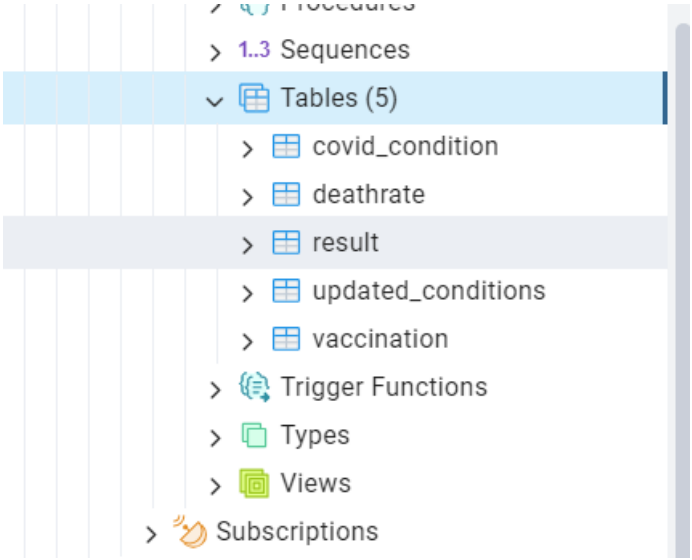

Fig. 8. The Distribution of Vaccine Counts by Zip Code

C. Visualization of the merged Datasets

The processed datasets are extracted from the SQL database for the further analysis. The transformed datasets are merged accordingly to prove the insights that we are trying to find.
The structured data from Dataset 1 and 2 are merged first with the common column ‘namesbyAge‘. Fig 9, 10 and 11 illustrates the visualizations of the dataframe resulted from this merging. The Fig 9 illustrates the diseases that significantly influenced the mortality rate of COVID-19. Analysis of this graph indicates that individuals with respiratory diseases were particularly vulnerable to COVID-19. From the visualization it is evident that, people with pre-existing respiratory diseases are more affected by the Covid-19.

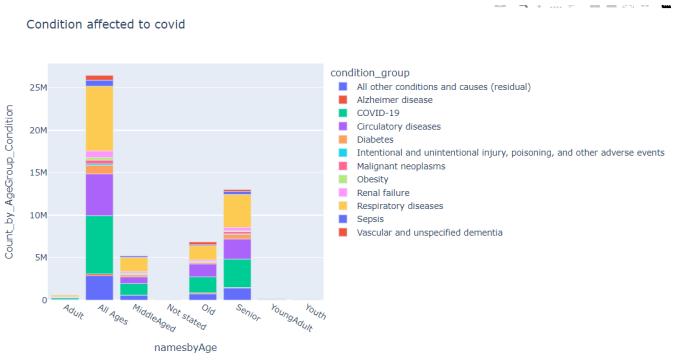

Fig. 9. Condition contributing to covid

The second graph as seen in Fig 10 also demonstrates the insights from the first merging which indicates that seniors with respiratory diseases were most prone to COVID-19. The people between 64-80 years are included in the ‘senior‘ age group , whom with respiratory diseases are more prone to Covid-19. The visualization also supports the first insight we derived.

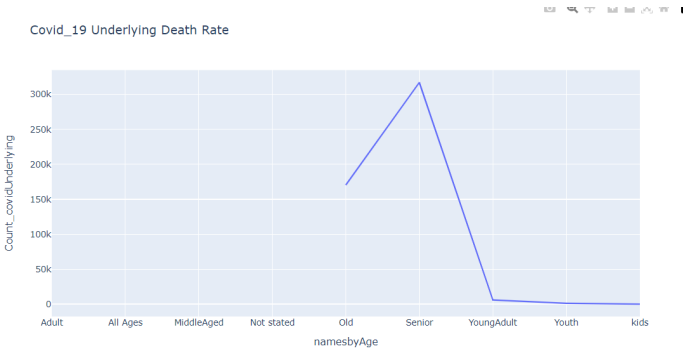

Fig. 10. Covid 19 as an underlying cause of death

The third graph combines the data from the first two visualizations. This combined graph proves that both the above insights is right. It is evident that seniors faced the highest risk of COVID-19, especially those with respiratory conditions which is shown in Fig 11.

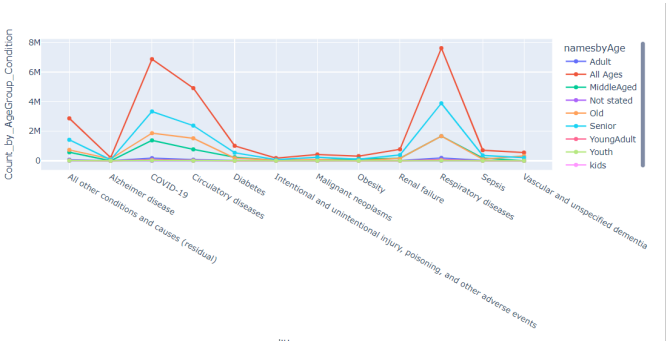

Fig. 11. Death count by Age group

Finally, the merged dataframe is combined with the Dataset3, which contains the vaccination rate. The final graph as depicted in Fig 12 was plotted using final merged data. This graph indicates that seniors have the lowest vaccination rates compared to adults because of which adults’ experience lower mortality rates from COVID-19, while seniors face the highest. The graph clearly demonstrates that seniors age group have the lowest vaccination rate, which adversely affected this age group. We can conclude that the mortality rate is the highest because the people in senior age group took less vaccinations. Also adults took the vaccination most, so that it saved most of their lives from COVID-19 death.

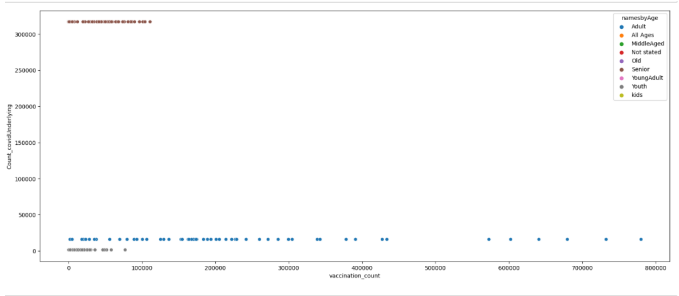

Fig. 12. Vaccination Count by Age group

V. Conclusion

In this project, we analysed the COVID-19 conditions, the diseases contributing to COVID-19 death rate and the effect of vaccination. First, the datasets are individually analyzed, processed and transformed. Then the transformed datasets are merged and aggregated to derive conclusions. After the analysis, we found that the three interlinked insights: 'The senior(65-80 years) age group is more affected by COVID-19, as they have the highest covid mortality rate', 'The seniors with pre-existing respiratory diseases are more prone to the COVID-19 death' and 'Seniors are the age group with the lowest vaccination rate, who have the highest death rate'. These insights are proved and illustrated using strong visualizations and graphs.

References

[1] A. M. O. Abdelsamad and A. Z. Karrar, *"An Interactive Dashboard for Monitoring the Spread of COVID-19 in Sudan*," 2020 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE), Khartoum, Sudan, 2021, pp. 1-6, doi: 10.1109/ICCCEEE49695.2021.9429561.

[2] AM. R. Mufid, A. Basofi, S. Mawaddah, K. Khotimah and N. Fuad, *"Risk Diagnosis and Mitigation System of COVID-19 Using Expert System and Web Scraping*," 2020 International Electronics Symposium (IES), Surabaya, Indonesia, 2020, pp. 577-583, doi: 10.1109/IES50839.2020.9231619.

[3] B. Sharma, K. Gupta, G. Bhardwaj and S. Kumar, *"A Comprehensive Study on Impact, Analysis and Complications of COVID-19*," 2022 3rd International Conference on Intelligent Engineering and Management (ICIEM), London, United Kingdom, 2022, pp. 102-106, doi: 10.1109/ICIEM54221.2022.9853169.

[4] L. G. Wiseso, M. Imrona and A. Alamsyah, *"Performance Analysis of Neo4j, MongoDB, and PostgreSQL on 2019 National Election Big Data Management Database*," 2020 6th International Conference on Science in Information Technology (ICSITech), Palu, Indonesia, 2020, pp. 91-96, doi:10.1109/ICSITech49800.2020.9392041.

[5] I. Stančin and A. Jović, *"An overview and comparison of free Python libraries for data mining and big data analysis*," 2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, 2019, pp. 977-982, doi: 10.23919/MIPRO.2019.8757088.

[6] Avinash Navlani; Armando Fandango; Ivan Idris, *Python Data Analysis: Perform data collection, data processing, wrangling, visualization, and model building using Python* , Packt Publishing, 2021.

[7] Jonathan Rioux, *Data Analysis with Python and PySpark* , Manning, 2022.