Team Members:
Team Leader - Priyanka Gururaj – pguruaj1@student.gsu.edu
Ramani Desai – rdesai20@student.gsu.edu
Sowmya Raghavendra – sraghavendra1@student.gsu.edu

Covid 19 Impact: GCP pipeline and analysis

1. Motivation and Problem Statement

Motivation

The motivation for this project stems from the need for data-driven insights to navigate the complexities of a global health crisis.

InDepth Analysis: COVID-19 data is constantly evolving. This project uses data from the Disease.sh API, ensuring up-to-date information for analysis.

Public Health Impact: Understanding recovery rates, case fatality rates, and the correlation between population and COVID-19 outcomes can help governments and organizations better manage resources and policies.

Data Skills Development: This project leverages skills in data extraction, transformation, analysis, and visualization, emphasizing the practical application of BigQuery and visualization tools like Looker Studio.

Global Relevance: The insights derived from this analysis have universal significance, as every country has been affected by the pandemic in unique ways.

By addressing this problem, the project not only explores the technical aspects of data analysis but also contributes to a greater understanding of a critical global issue

Problem Statement

The COVID-19 pandemic has impacted countries worldwide, with varying levels of cases, deaths, recoveries, and population densities contributing to diverse outcomes. Understanding these metrics and their relationships at a global level is crucial for public health strategies, resource allocation, and recovery planning. The goal of this project is to:

Extract COVID-19 data from a reliable public API.

Store the data in a structured format in Google BigQuery.

Perform in-depth analysis of key metrics such as case and recovery rates by population.

Provide actionable insights through visualization to aid in better decision-making through looker studio.

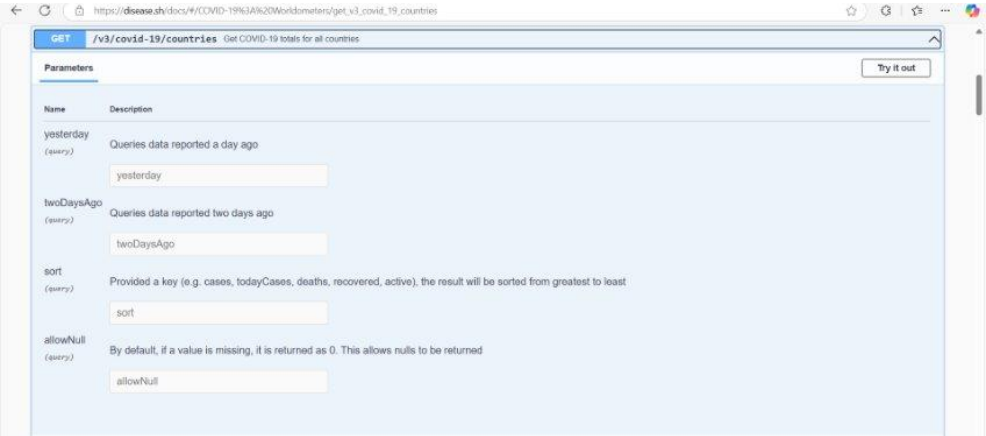
2. Solution Overview

Approach

1. Data Extraction (API Integration)

Extracted COVID-19 data from the Disease.sh API, which provided global statistics on COVID-19 cases, deaths, recoveries, and population.

Why it's important: The API provided accurate and up-to-date data, ensuring that our analysis reflected the current global situation.



2. GCP environment setup

Step 1: Created GCP Project

Logged in to the Google Cloud Console.

Created a new project:

Name: dbms-443800

Linked a billing account.

Noted the Project ID (e.g., dbmsproject-443800).

Step 2: Enabled APIs

Go to APIs & Services > Library.

Enabled:BigQuery API

Cloud Functions API (optional for serverless processing).

Step 3: Set Up a Service Account

Navigated to IAM & Admin > Service Accounts.

Created a service account:

Name: data-pipeline-account

Roles:

Project Owner

BigQuery Admin

Generated and downloaded a JSON key file for authentication.

Step 4: Assigned Permissions

Granted the service account access to the BigQuery dataset.

Role: Editor or Admin.

Verify permissions in IAM > Permissions.

Step 5: Configured Local Authentication

Saved the JSON key file in your project directory.

Set the environment variable:

export GOOGLE_APPLICATION_CREDENTIALS="path/to/dbms-443800-key.json" into Colab notebook

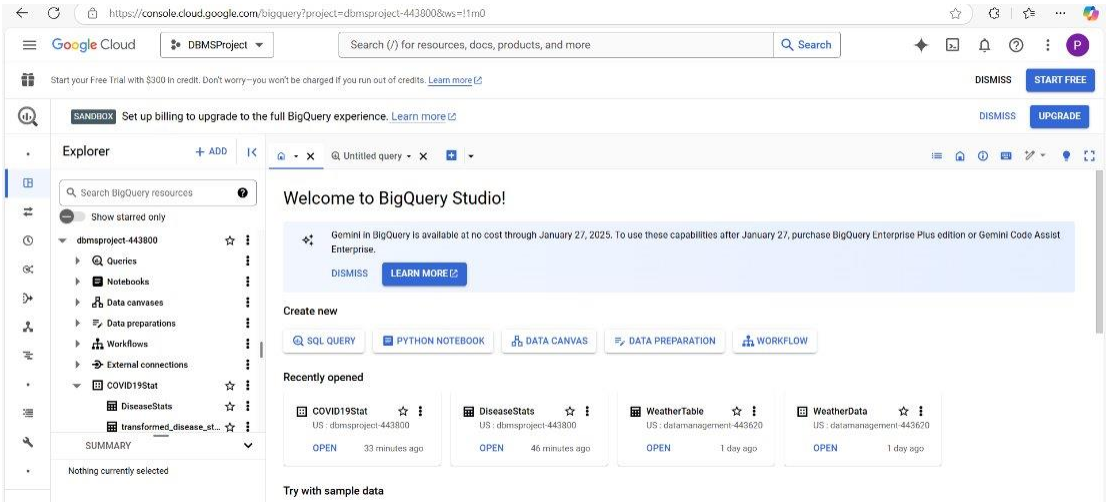
Tested authentication with:

```
21
22 # Initialize BigQuery client
23 client = bigquery.Client()
24
25 # Define BigQuery table schema
26 table_id = "dbmsproject-443800.COVID19Stat.DiseaseStats"
27 job_config = bigquery.LoadJobConfig(
28     schema=[
29         bigquery.SchemaField("country", "STRING"),
30         bigquery.SchemaField("cases", "INTEGER"),
31         bigquery.SchemaField("deaths", "INTEGER"),
32         bigquery.SchemaField("recovered", "INTEGER"),
33         bigquery.SchemaField("population", "INTEGER"),
34     ],
35     write_disposition=bigquery.WriteDisposition.WRITE_TRUNCATE,
36 )
37
38 # Load data into BigQuery
39 job = client.load_table_from_dataframe(df, table_id, job_config=job_config)
40 job.result()
41 print(f"Data loaded into BigQuery table {table_id}")
```

Step 6: Verify Setup

Created a BigQuery dataset (e.g., COVID19Stat) and test table creation.

Execute a sample query in the BigQuery console to confirm setup.



3. Data Transformation (Preprocessing and Structuring)

Transformed the extracted data into a tabular format using Python and Pandas. Relevant columns such as country, cases, deaths, recovered, and population are selected for analysis.

Why it's important: Structuring data ensures it is clean, relevant, and ready for efficient querying and analysis in BigQuery.

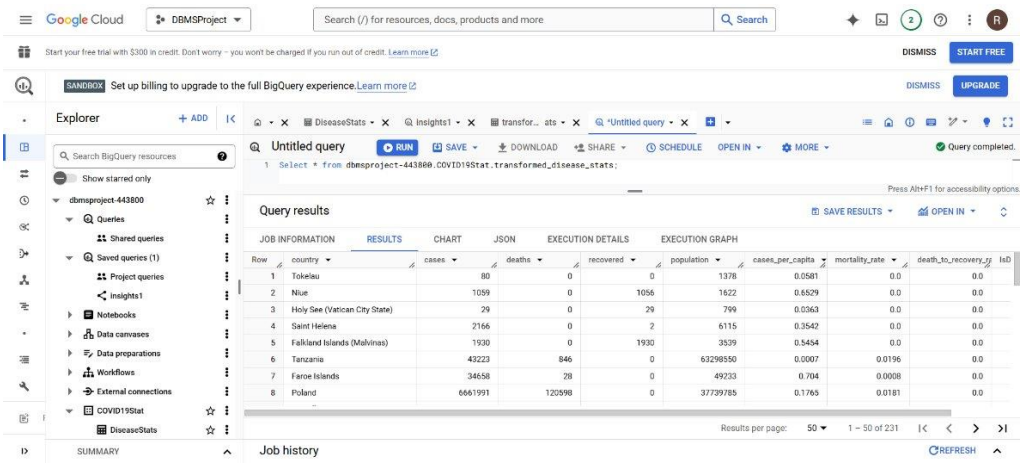
```python
import requests
from google.cloud import bigquery
import pandas as pd

# API Endpoint
url = "https://disease.sh/v3/covid-19/countries"

# Fetch data from the API
response = requests.get(url)
if response.status_code == 200:
    data = response.json()
else:
    print("Failed to fetch data from the API")
    exit()

# Convert to DataFrame
df = pd.json_normalize(data)

# Select relevant columns
df = df[['country', 'cases', 'deaths', 'recovered', 'population']]

# Initialize BigQuery client
client = bigquery.Client()

# Define BigQuery table schema
table_id = "dbmsproject-443800.COVID19Stat.DiseaseStats"
job_config = bigquery.LoadJobConfig(
    schema=[
        bigquery.SchemaField("country", "STRING"),
        bigquery.SchemaField("cases", "INTEGER"),
        bigquery.SchemaField("deaths", "INTEGER"),
        bigquery.SchemaField("recovered", "INTEGER"),
        bigquery.SchemaField("population", "INTEGER"),
    ],
    write_disposition=bigquery.WriteDisposition.WRITE_TRUNCATE,
)

# Load data into BigQuery
job = client.load_table_from_dataframe(df, table_id, job_config=job_config)
job.result()
print(f"Data loaded into BigQuery table {table_id}")

```

```
Data loaded into BigQuery table dbmsproject-443800.COVID19Stat.DiseaseStats
```

4. Data Storage (BigQuery Integration)

Loaded the transformed data into a Google BigQuery table. The schema is explicitly defined to ensure compatibility and data integrity.

Why it's important: Storing the data in BigQuery provides scalability, enabling us to perform advanced analytics with high performance.



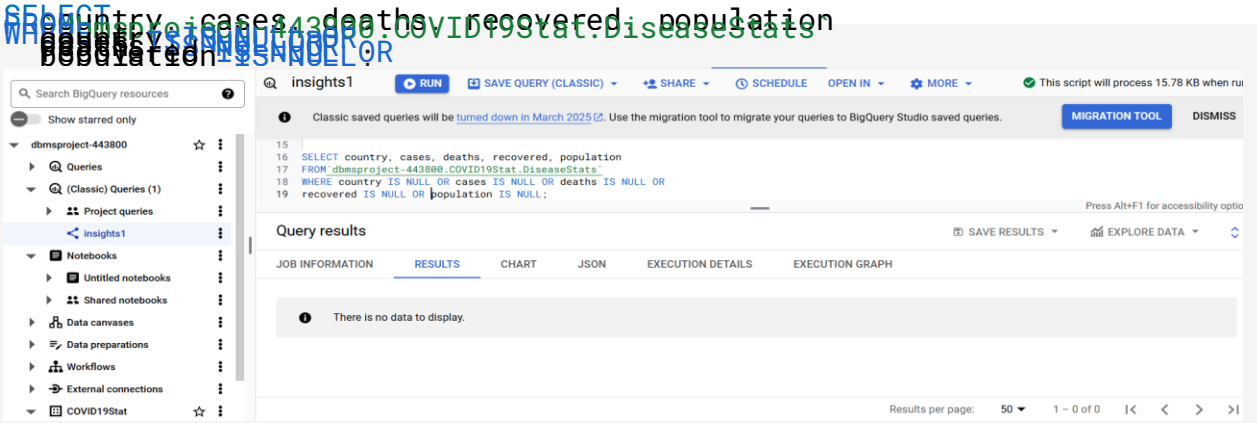
4. Data Validation (Quality Assurance)

Validate the data by checking for missing values, incorrect data types, and ensuring logical constraints (e.g., cases should not exceed population).

Why it's important: Ensures the accuracy and reliability of the data, which is crucial for deriving meaningful insights.

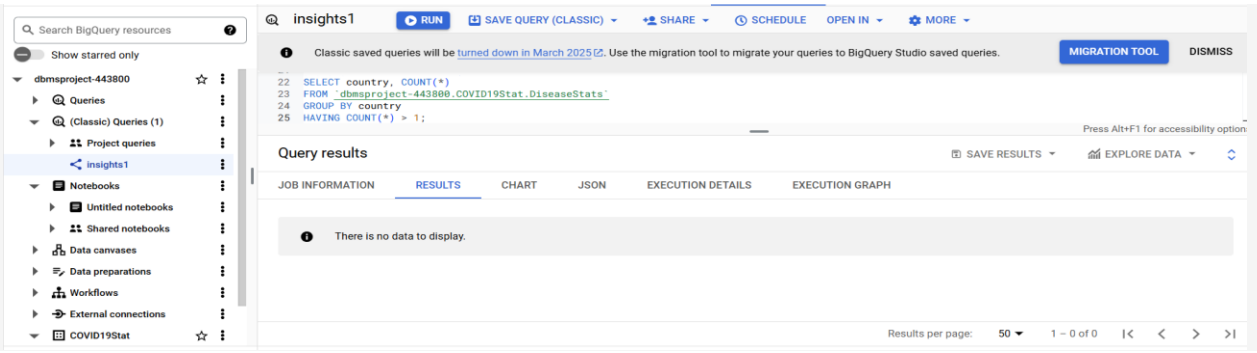
For Null Values:

```
SELECT country, cases, deaths, recovered, population
FROM `dbmsproject-443800.COVID19Stat.DiseaseStats`
WHERE country IS NULL OR cases IS NULL OR deaths IS NULL OR
recovered IS NULL OR population IS NULL;
```



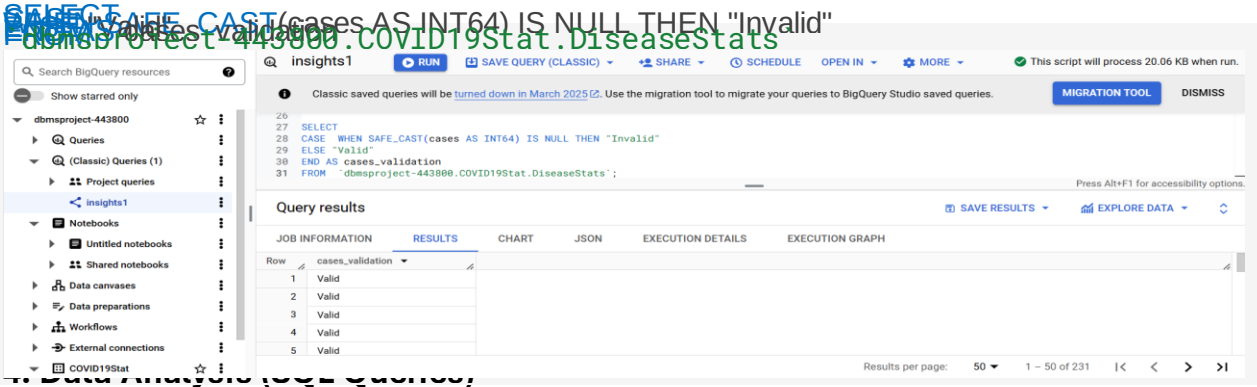
For Duplicate Values:

```
SELECT country, COUNT(*)
FROM `dbmsproject-443800.COVID19Stat.DiseaseStats`
GROUP BY country
HAVING COUNT(*) > 1;
```



To validate Values:

```
SELECT
CASE WHEN SAFE_CAST(cases AS INT64) IS NULL THEN "Invalid"
ELSE "Valid"
END AS cases_validation
FROM `dbmsproject-443800.COVID19Stat.DiseaseStats`;
```

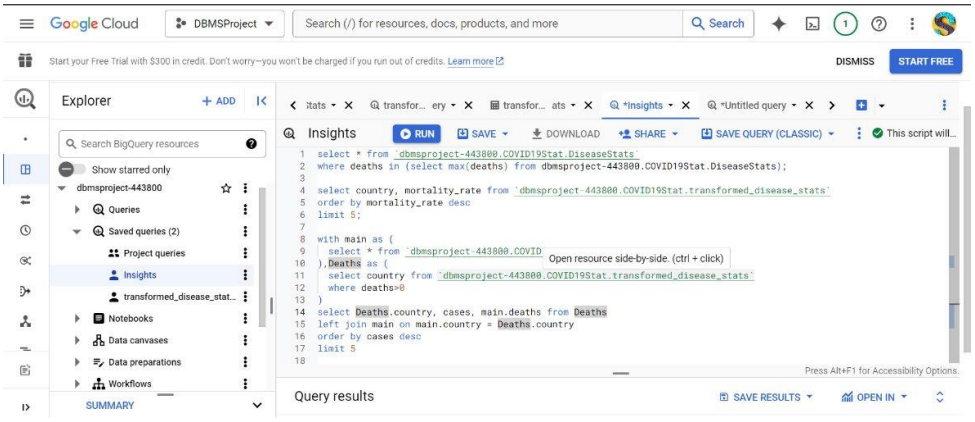


4. Data Analysis (SQL Queries)

Performed in-depth analysis using SQL queries on BigQuery. Key metrics like recovery rates, case fatality rates, and comparisons based on population are calculated.

Why it's important: SQL queries allow for complex analytics, including aggregations, filtering, and deriving relationships between different metrics.

All queries code:



Query 1: Maximum Deaths

Purpose: Identify the country with the highest number of deaths.

Output:

Row	country	cases	deaths	recovered	population
1	USA	111820082	1219487	109814428	334805269

Key Insight: The dataset highlights the country with the most significant COVID-19 fatalities, helping analyze the pandemic's impact.

Query 2: Top 5 Countries by Mortality Rate

Purpose: Rank countries by mortality rate to assess the deadliest regions during the pandemic.

Row	country	mortality_rate
1	MS Zaandam	0.2222
2	Yemen	0.1807
3	Western Sahara	0.1
4	Sudan	0.0789
5	Syrian Arab Republic	0.0548

Key Insight: Provides a perspective on which countries struggled the most relative to their cases.

Query 3: Top 10 Countries by Cases with Death Data

Purpose: Combine case counts and death data to identify countries with high case volumes and mortality.

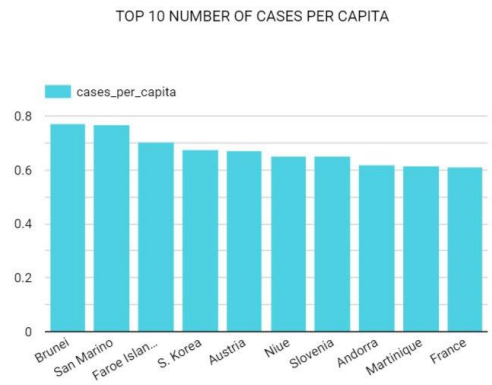
Row	country	cases	deaths
1	USA	111820082	1219487
2	India	45035393	533570
3	France	40138560	167642
4	Germany	38828995	183027
5	Brazil	38743918	711380

Key Insight: Highlights countries with high cases but varying death rates, enabling correlation analysis.

6. Data Visualization (Looker Studio Dashboards)

Created intuitive and informative visualizations in Looker Studio, such as:

Data Visualization 1:



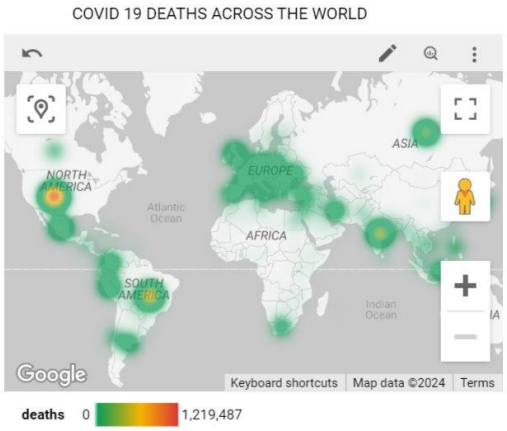
Insights:

The chart highlights the top 10 countries with the highest number of COVID-19 cases per capita, calculated as the ratio of total cases to the population.

Countries such as Brunei, San Marino, and Faroe Islands show significant cases relative to their population sizes.

This metric is critical for understanding the impact of the pandemic in proportion to the population, offering insights into healthcare system strain and the relative spread of the virus in smaller nations.

Data Visualization 2:



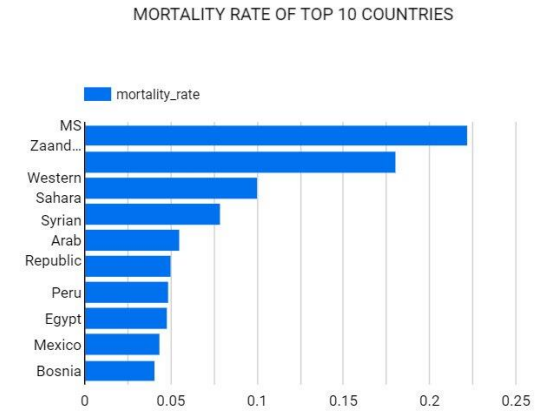
Insights:

The heatmap illustrates the distribution of COVID-19 deaths across the world, with regions in North America, Europe, and South America showing the highest intensity of fatalities.

The color gradient from green to red represents the range of deaths, with red indicating areas of the highest mortality.

This visualization provides a global perspective on the pandemic's impact, highlighting regions most severely affected by COVID-19.

Data Visualization 3:



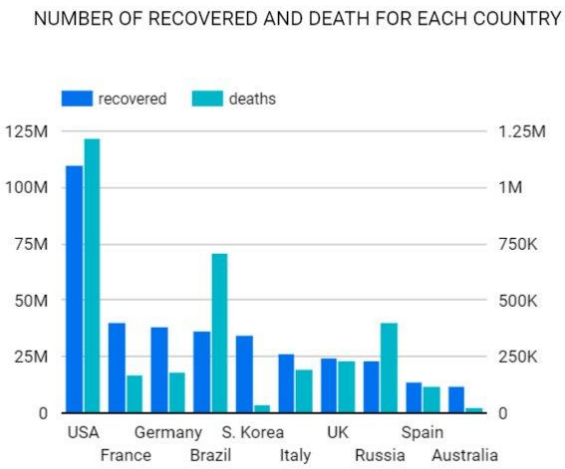
Insights:

The bar chart showcases the mortality rates of the top 10 countries, where mortality rate is calculated as the ratio of deaths to cases.

Countries like MS Zaandam and Yemen exhibit the highest mortality rates, exceeding 0.15, indicating significant fatality proportions relative to the number of confirmed cases.

This visualization underscores the severity of the pandemic's impact in these regions and provides valuable insights for targeted interventions.

Data Visualization 4:



Insights:

This bar chart compares the number of recoveries and deaths across various countries during the COVID-19 pandemic. It highlights the disparity in outcomes based on regional factors, population size, and pandemic response measures.

USA has the highest numbers, leading in both recoveries and deaths, reflecting its large population and COVID-19 impact. South Korea and Australia have fewer deaths and recoveries, indicating effective management or smaller affected populations.

This chart provides a comparative view of the global pandemic response and its outcomes. It helps policymakers and healthcare professionals identify successful strategies in low-impact regions and areas requiring further improvement.

Why it's important: Visualizations simplify complex data, making insights accessible and actionable for decision-makers.

7.Limitations

- **API Rate Limits**:
  The API used may impose rate limits or data restrictions, which can hinder real-time data extraction or large-scale queries.
- **Scalability Constraints**:
  Handling a significant increase in data volume might require advanced optimization in GCP services or infrastructure upgrades.
- **GCP Costs**:
  Extensive use of GCP services like BigQuery can incur costs, especially for large datasets or frequent queries.

8. Future Scope:

Include Additional Data Sources: Integrate vaccination rates, healthcare infrastructure, or mobility data for a more comprehensive analysis.

Enhance Data Validation: Automate integrity checks and anomaly detection to ensure higher accuracy and reliability.

Real-Time Dashboard: Upgrade the pipeline to support real-time updates for dynamic insights using live API data.

Predictive Modeling: Implement machine learning models to forecast future trends and potential outbreaks.

5. Conclusion

The project effectively demonstrates the power of leveraging modern cloud-based data pipelines to process and analyze real-world data. By extracting COVID-19 statistics via an API, storing them in BigQuery, and analyzing them with SQL queries, we uncover actionable insights about global recovery trends and the pandemic's impact.

The use of Looker Studio for visualization enables clear and intuitive communication of findings, helping decision-makers understand key metrics like recovery rates and population impacts. The integration of GCP services ensures scalability, flexibility, and reliability, making this approach applicable to various datasets and industries.

This project not only highlights technical proficiency in GCP and data analytics tools but also emphasizes the importance of storytelling with data, fostering impactful and data-driven decisions.