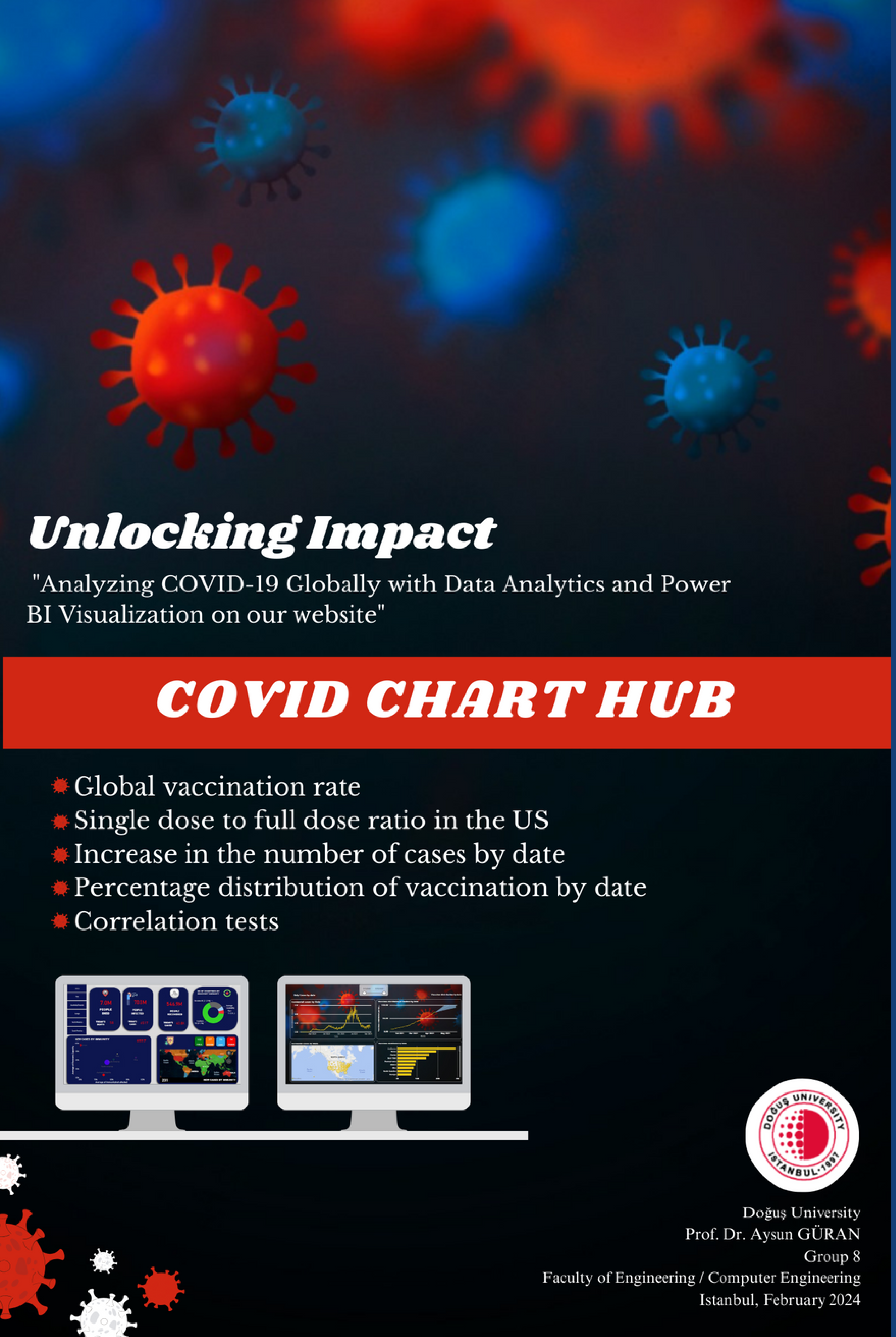


Unlocking Impact

"Analyzing COVID-19 Globally with Data Analytics and Power BI Visualization on our website"

COVID CHART HUB

- Global vaccination rate
- Single dose to full dose ratio in the US
- Increase in the number of cases by date
- Percentage distribution of vaccination by date
- Correlation tests

Doğuş University
Prof. Dr. Aysun GÜRAN
Group 8
Faculty of Engineering / Computer Engineering
Istanbul, February 2024

February 2024

Group No: 8

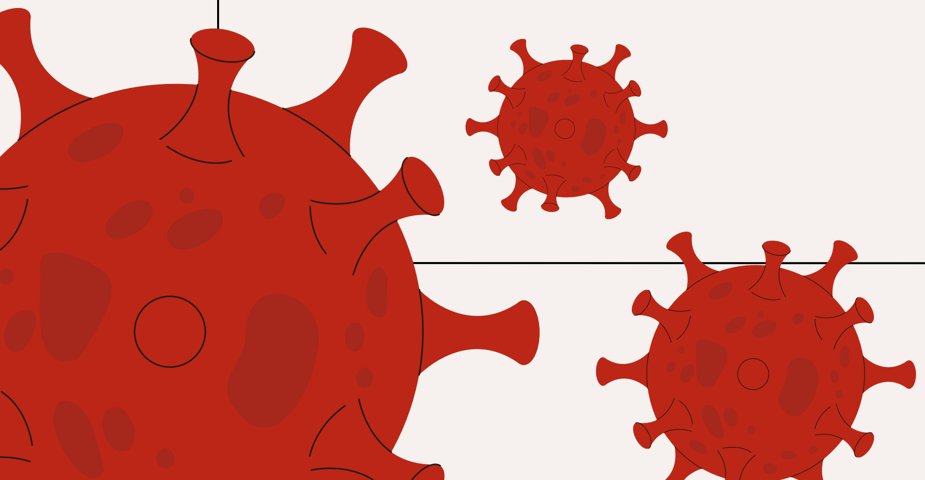
COVID CHART HUB

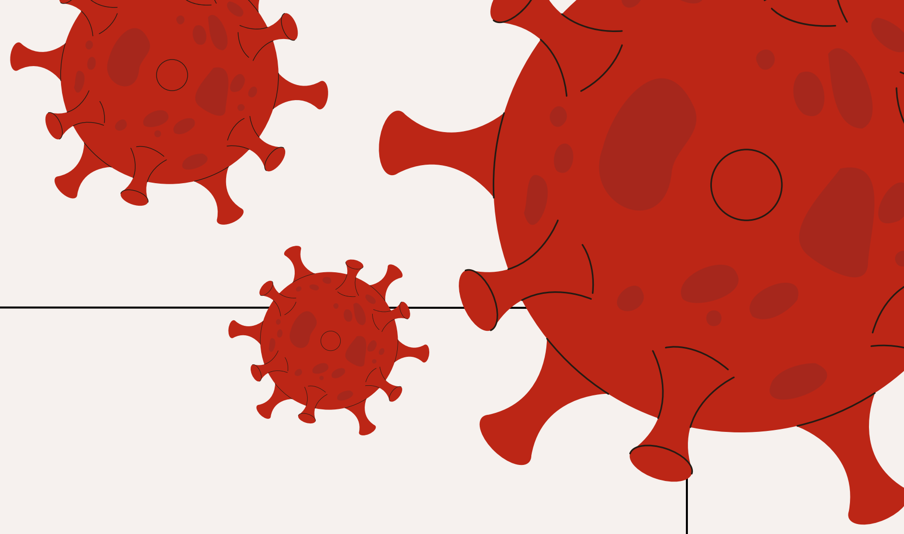
Covid-19 Analysis and Correlation Tests

COME 491 FINAL PRESENTATION

Ayberk Tambay Bekir Enes Şiranlı Şevval Elif Çoban
Zehra Günaydın Nursena Bahadır

CONTENTS



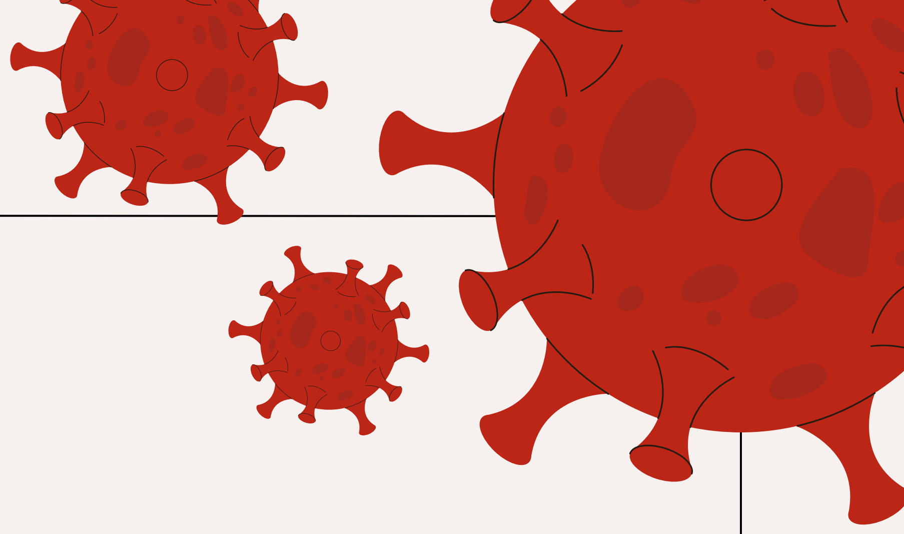


PURPOSE OF THE SYSTEM

The purpose of the system is to analyze the impact of Covid-19 on the world and publish it on a website. Our aim is to conduct a comprehensive analysis of the multifaceted impacts of COVID-19 through data analysis using Python and richly visualize this data using Power Bi. Correlation tests using COVID-19 data aim to contribute to a deeper understanding of the impacts of Covid-19.

GANTT CHART

PERIOD	WEEK 1	WEEK 2	WEEK 3	WEEK 4	WEEK 5	WEEK 6	WEEK 7	WEEK 8	WEEK 9	WEEK 10	WEEK 11	WEEK 12	WEEK 13	WEEK 14
PLANNING	■	■	■	■										
DESIGN					■	■	■	■						
DATA ANALYSIS						■	■	■						
DATA VISUALIZATION						■	■	■	■	■	■	■	■	■
CORRELATİON							■	■	■	■	■	■	■	■
WEBSITE CODING									■	■	■	■	■	■
TEST													■	■



ADVISOR MEETING DATES

November 2023

		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	**16**	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

February 2024

			1	2	3	4
5	**6**	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29			

December 2023

				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

January 2024

1	2	3	**4**	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	**31**				

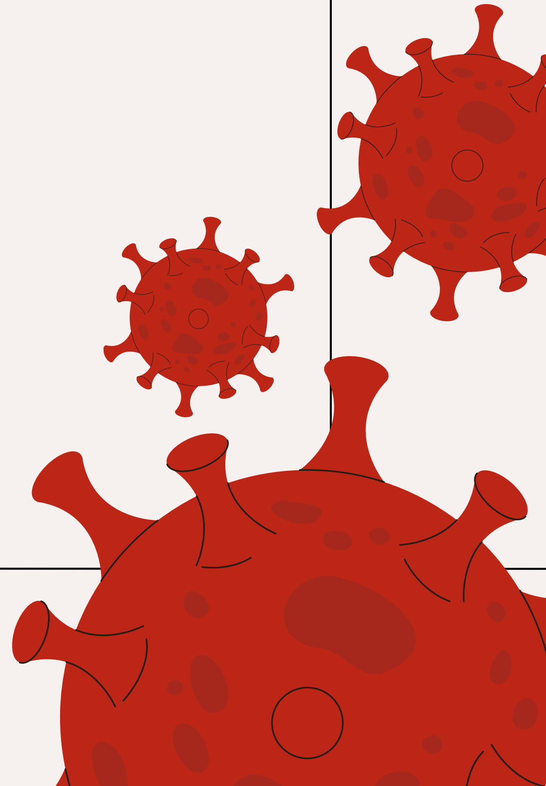
REQUIREMENTS ANALYSIS

SCOPE OF THE SYSTEM

In the data analysis of COVID-19 data using python, we reached the total number of deaths, cases, immunity and vaccination rates of countries and continents and visualized these data through Power BI. Correlation tests showed us the effects of vaccination status on the number of deaths in countries and whether there is a significant result between the increase in deaths between countries.

FUNCTIONAL REQUIREMENTS

The user should be able to access the website without logging in. On our Landing Page, they can browse short summaries and access the analyzes they want with a single click. On the website, the user can view the analysis of the Covid-19 research in the "Data Charts" and "Test Results" sections.



NON-FUNCTIONAL REQUIREMENTS

Availability

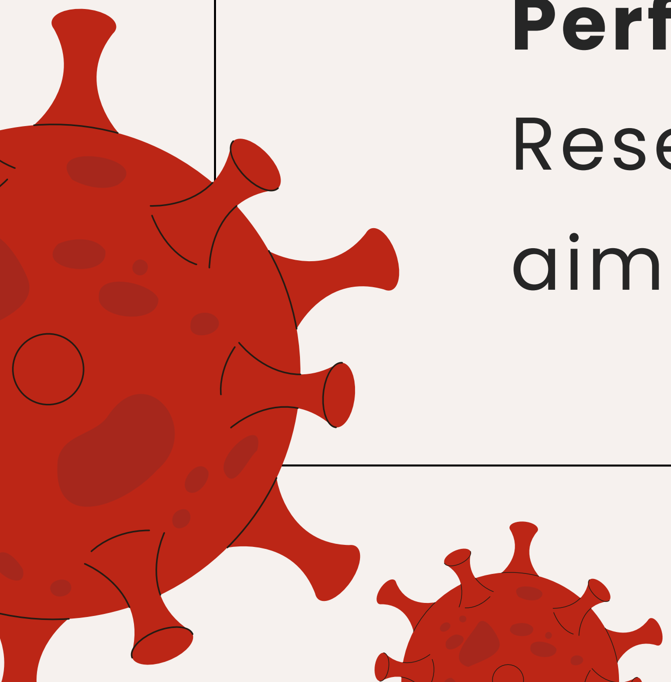
Researchers use the data provided to monitor and analyze the global impacts of Covid-19 in a multidimensional manner. By providing easy access to this data through the website, usability was prioritized.

Reliability

No one other than the software developer can intervene in the resulting analyses.

Performance

Researchers accessing the website can easily access all the data. This aims to provide an effective user experience in terms of performance.



NON-FUNCTIONAL REQUIREMENTS

Supportability

Accessibility of the website through any browser provides support to a wide range of users. Supportability aims to work smoothly on various browsers.

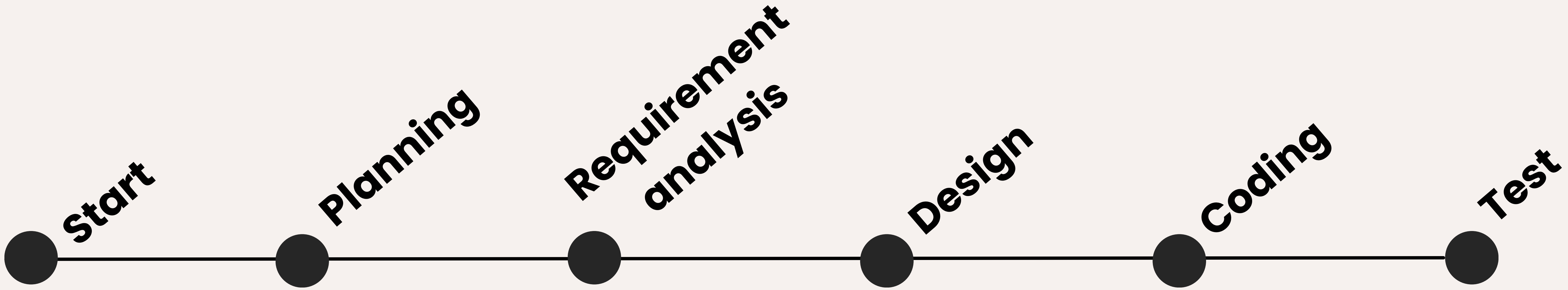
Realization

The analysis will be made available on the website. This platform, which does not include any membership system, allows users to access the data quickly and easily.

Interface

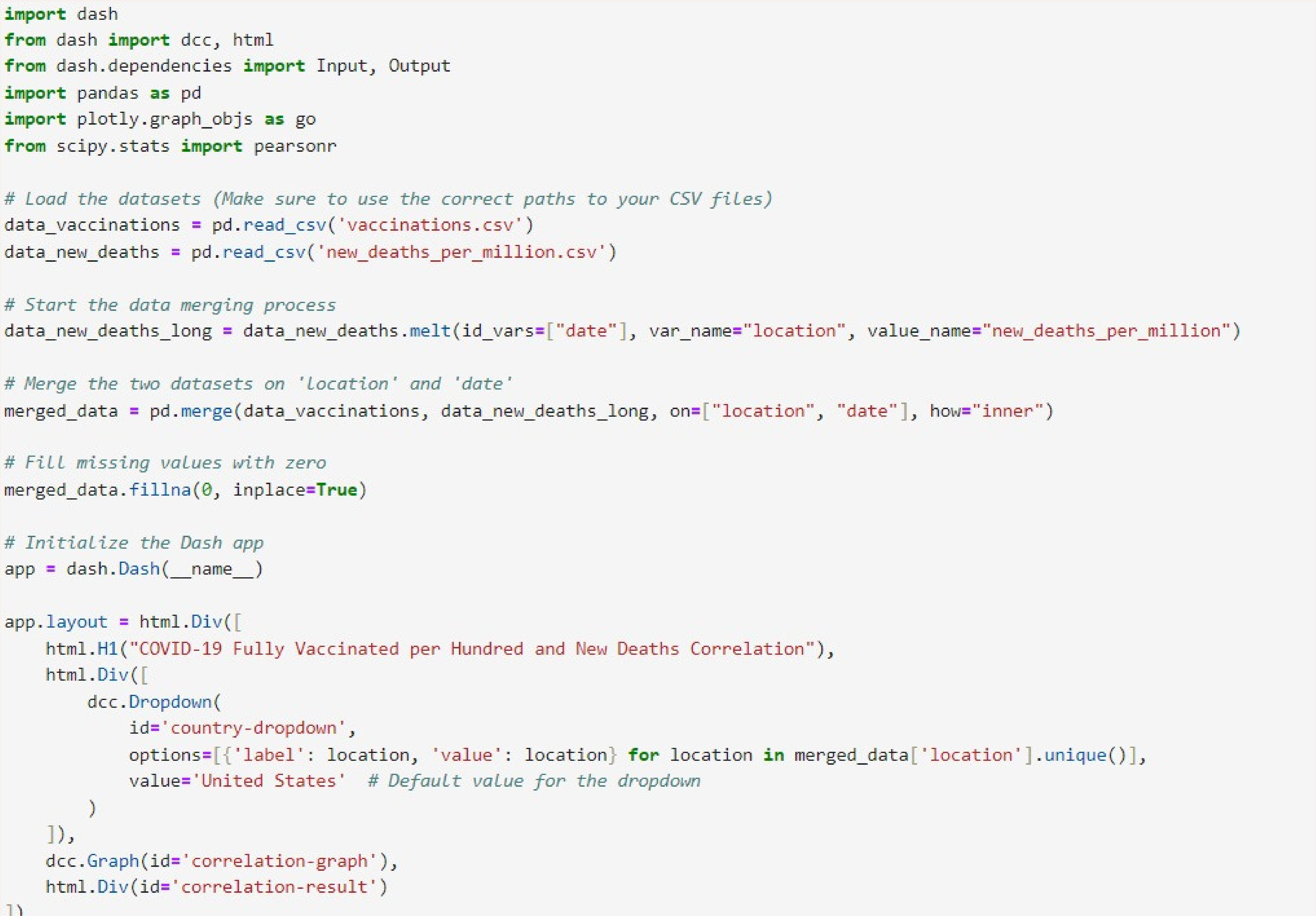
Page designs are designed to be easy for the user to use, are eye-catching and simple to use.

SOFTWARE DESIGN

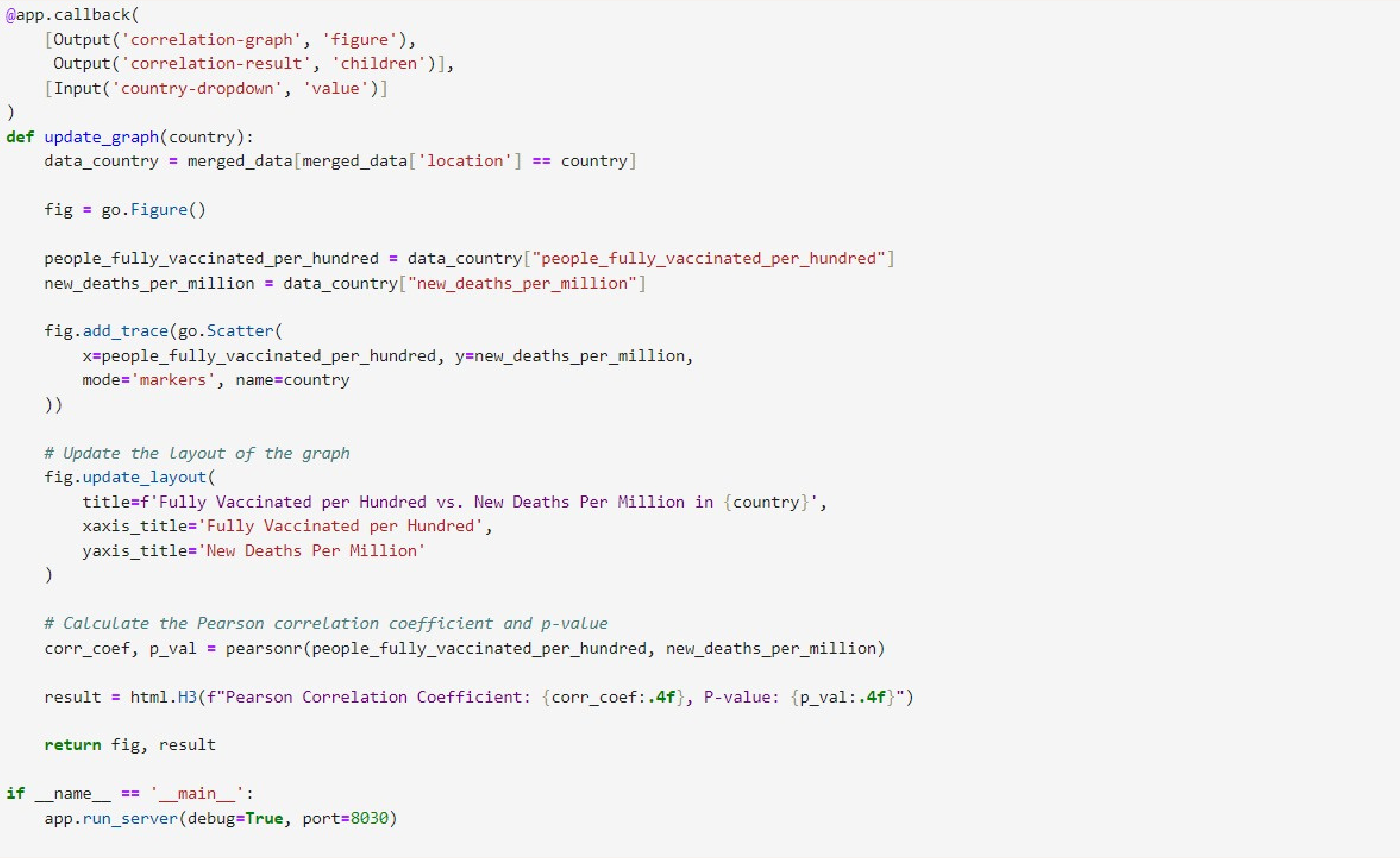


CODE BLOCKS OF THE PROJECT

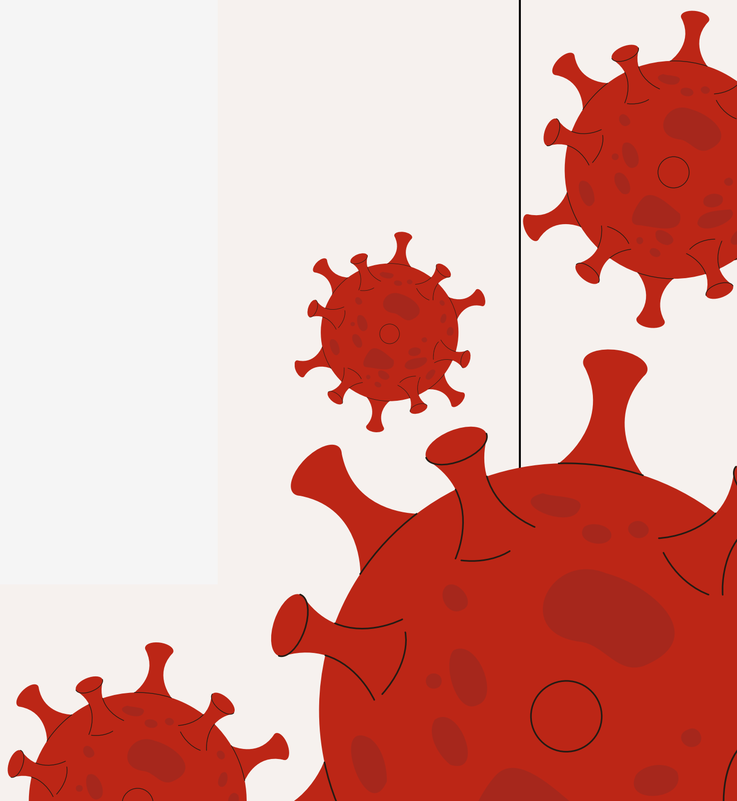
COVID-19 Fully Vaccinated per Hundred and New Deaths Correlation

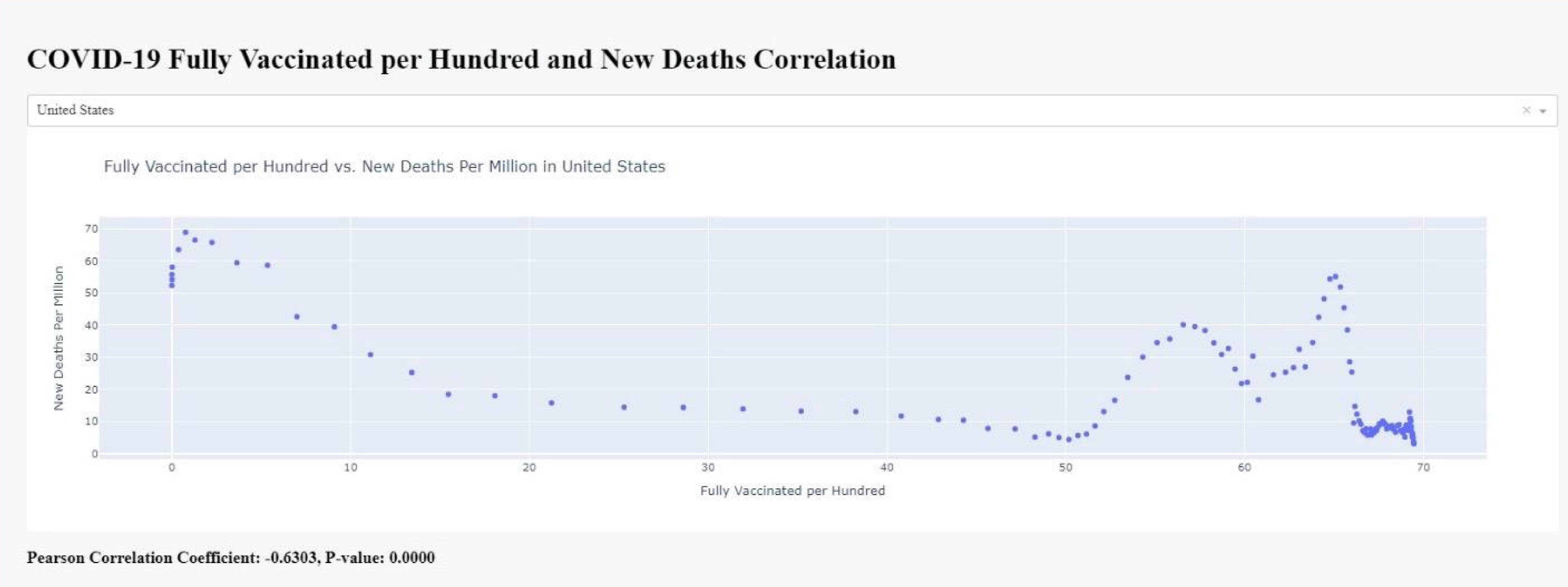


```python
import dash
from dash import dcc, html
from dash.dependencies import Input, Output
import pandas as pd
import plotly.graph_objs as go
from scipy.stats import pearsonr

# Load the datasets (Make sure to use the correct paths to your CSV files)
data_vaccinations = pd.read_csv('vaccinations.csv')
data_new_deaths = pd.read_csv('new_deaths_per_million.csv')

# Start the data merging process
data_new_deaths_long = data_new_deaths.melt(id_vars=["date"], var_name="location", value_name="new_deaths_per_million")

# Merge the two datasets on 'location' and 'date'
merged_data = pd.merge(data_vaccinations, data_new_deaths_long, on=["location", "date"], how="inner")

# Fill missing values with zero
merged_data.fillna(0, inplace=True)

# Initialize the Dash app
app = dash.Dash(__name__)

app.layout = html.Div([
    html.H1("COVID-19 Fully Vaccinated per Hundred and New Deaths Correlation"),
    html.Div([
        dcc.Dropdown(
            id='country-dropdown',
            options=[{'label': location, 'value': location} for location in merged_data['location'].unique()],
            value='United States'  # Default value for the dropdown
        )
    ]),
    dcc.Graph(id='correlation-graph'),
    html.Div(id='correlation-result')
])
```



```python
@app.callback(
    [Output('correlation-graph', 'figure'),
     Output('correlation-result', 'children')],
    [Input('country-dropdown', 'value')]
)
def update_graph(country):
    data_country = merged_data[merged_data['location'] == country]

    fig = go.Figure()

    people_fully_vaccinated_per_hundred = data_country["people_fully_vaccinated_per_hundred"]
    new_deaths_per_million = data_country["new_deaths_per_million"]

    fig.add_trace(go.Scatter(
        x=people_fully_vaccinated_per_hundred, y=new_deaths_per_million,
        mode='markers', name=country
    ))

    # Update the layout of the graph
    fig.update_layout(
        title=f'Fully Vaccinated per Hundred vs. New Deaths Per Million in {country}',
        xaxis_title='Fully Vaccinated per Hundred',
        yaxis_title='New Deaths Per Million'
    )

    # Calculate the Pearson correlation coefficient and p-value
    corr_coef, p_val = pearsonr(people_fully_vaccinated_per_hundred, new_deaths_per_million)

    result = html.H3(f"Pearson Correlation Coefficient: {corr_coef:.4f}, P-value: {p_val:.4f}")

    return fig, result

if __name__ == '__main__':
    app.run_server(debug=True, port=8030)
```



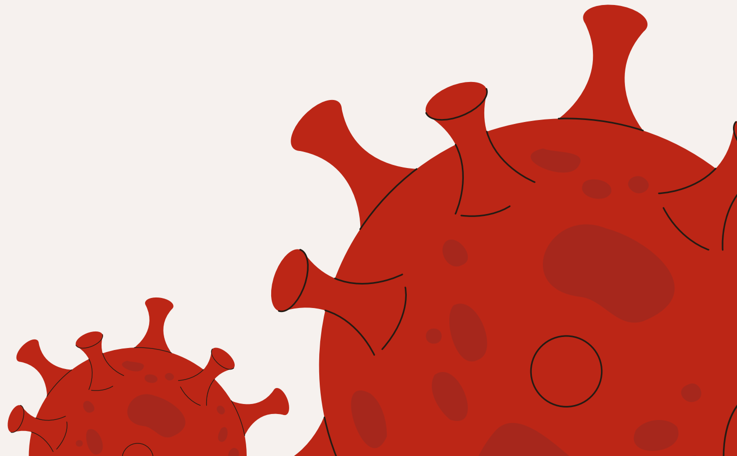


COVID-19 Fully Vaccinated per Hundred and New Deaths Correlation

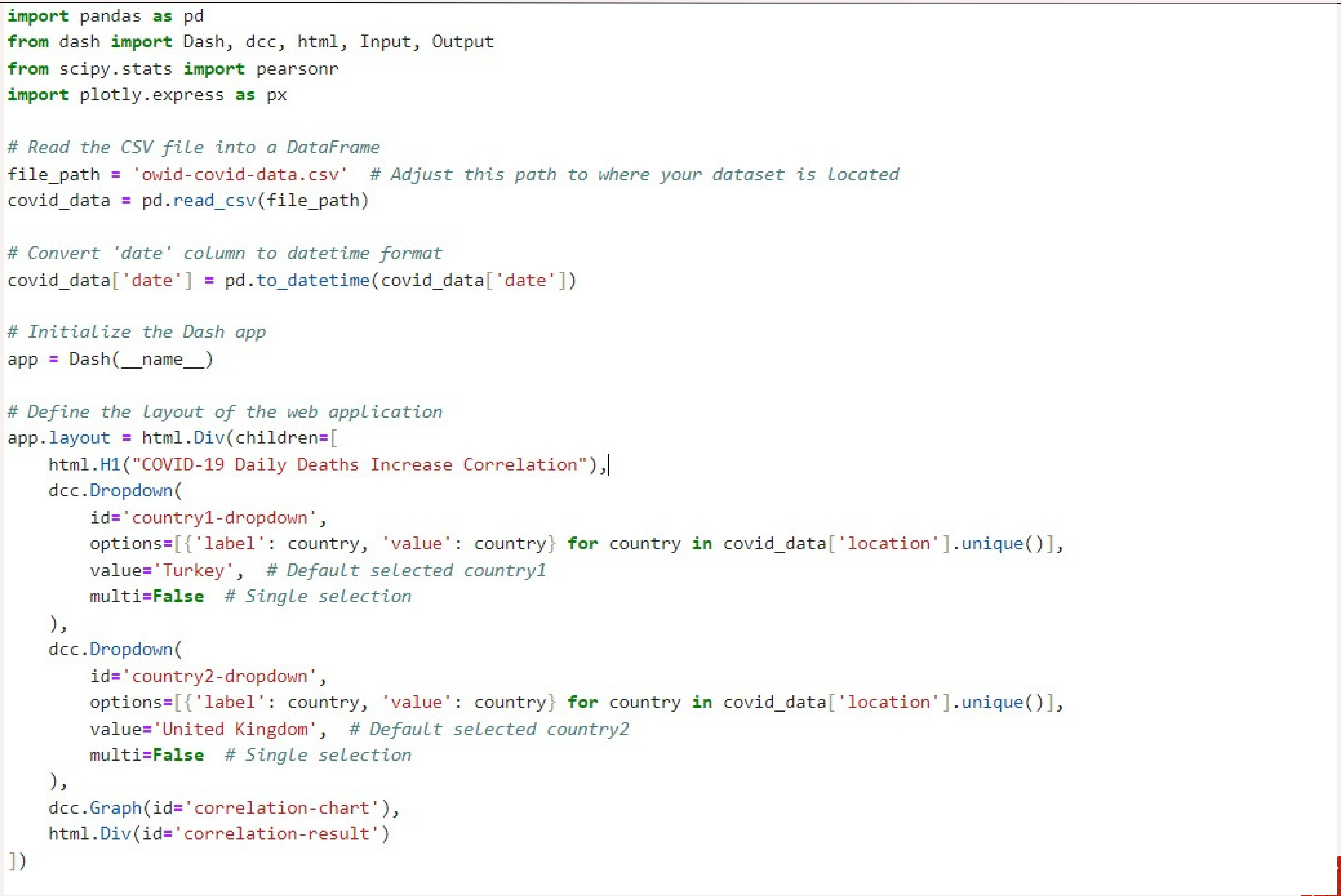
United States

×
Fully Vaccinated per Hundred vs. New Deaths Per Million in United States
New Deaths Per Million
70
60
50
40
30
20
10
0
0
10
20
30
40
50
60
70
Fully Vaccinated per Hundred

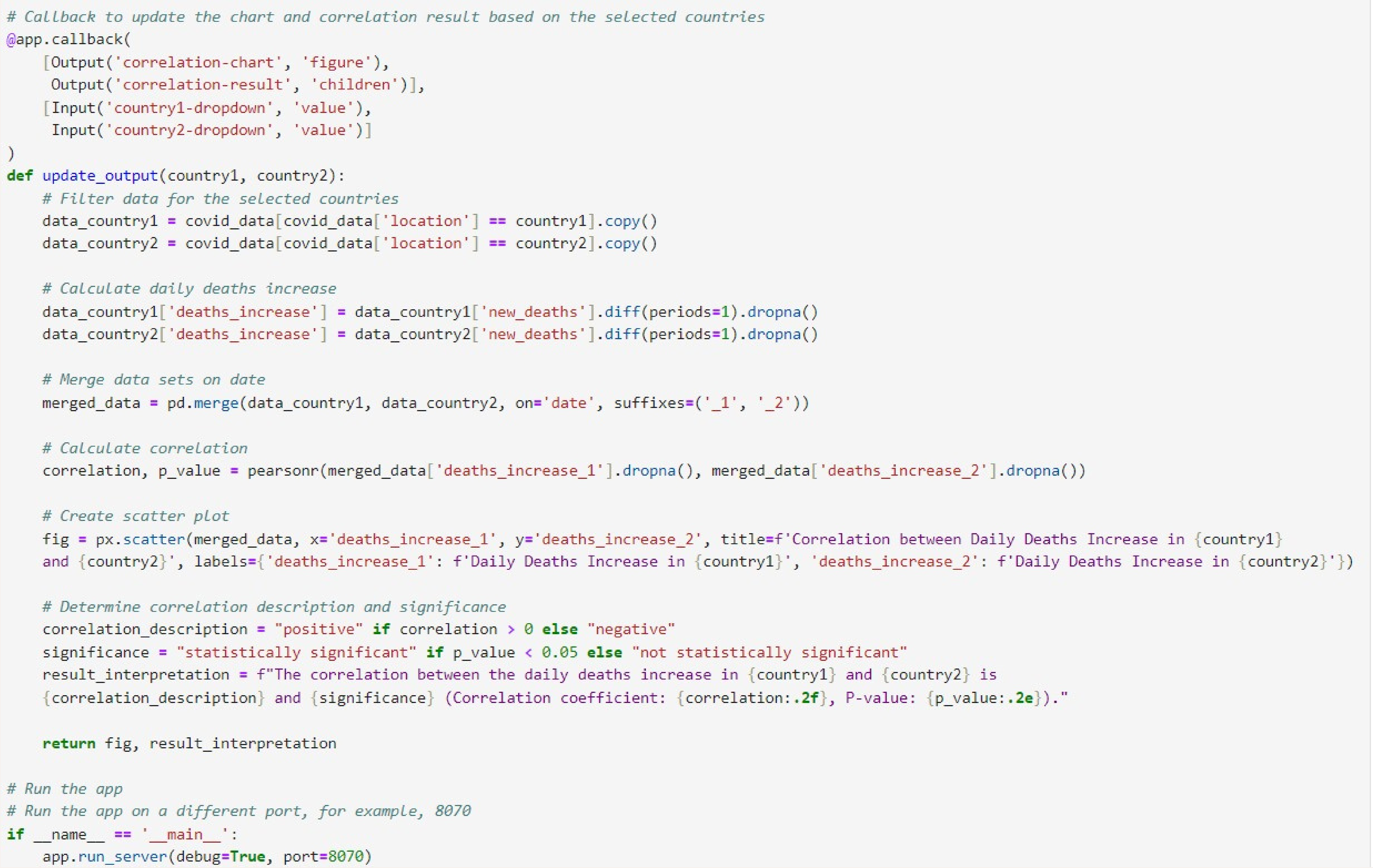
Pearson Correlation Coefficient: -0.6303, P-value: 0.0000



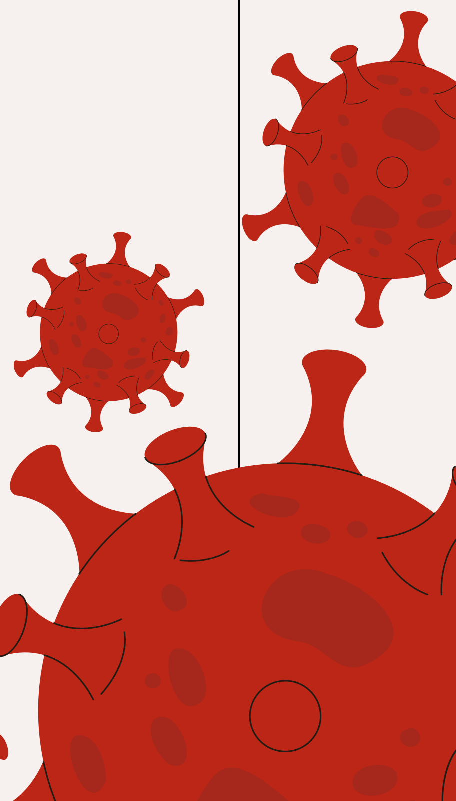
COVID-19 Daily Deaths Increase Correlation Code

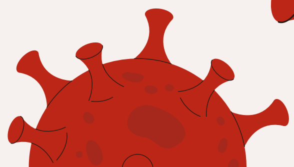


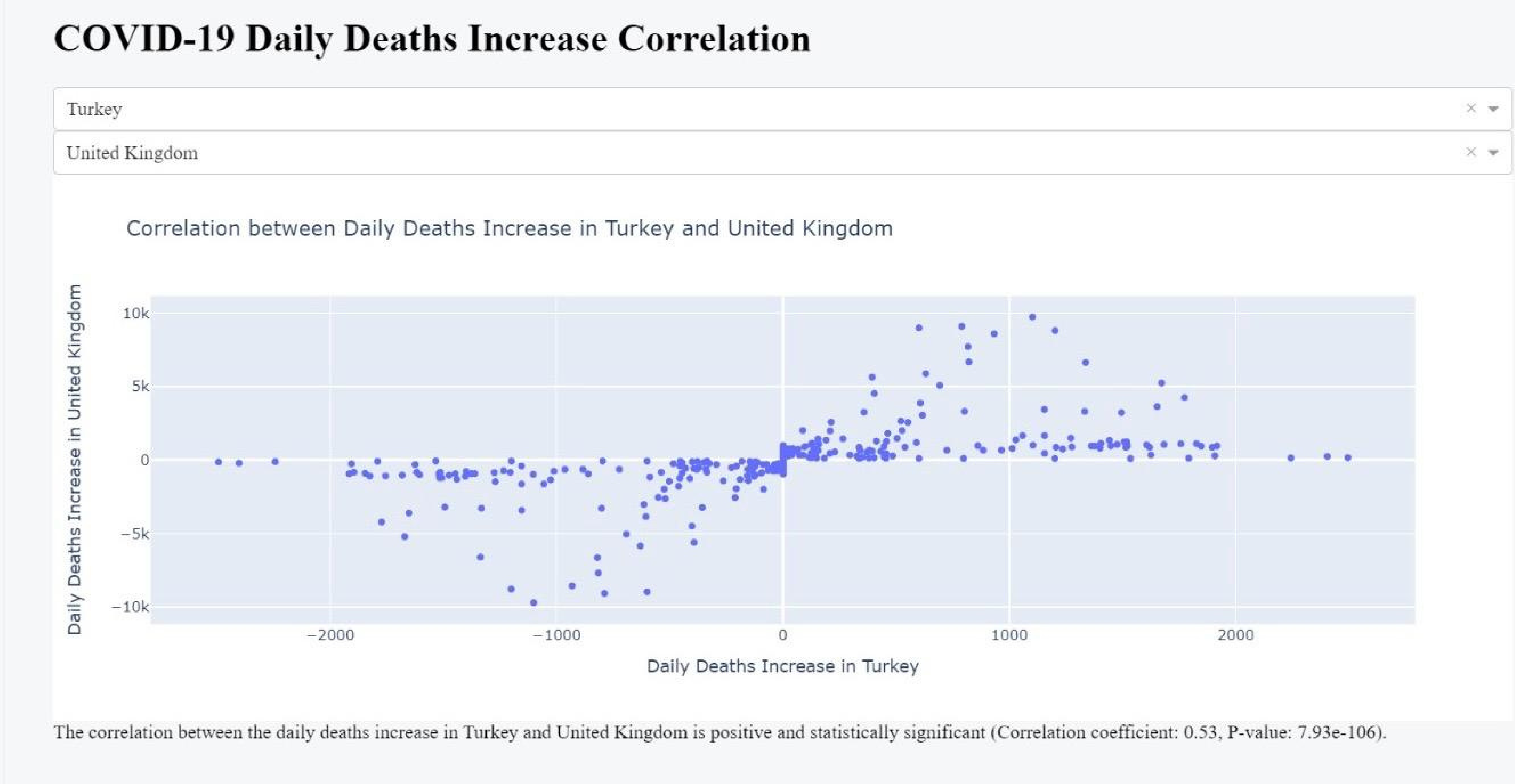
```python
import pandas as pd
from dash import Dash, dcc, html, Input, Output
from scipy.stats import pearsonr
import plotly.express as px

# Read the CSV file into a DataFrame
file_path = 'owid-covid-data.csv'  # Adjust this path to where your dataset is located
covid_data = pd.read_csv(file_path)

# Convert 'date' column to datetime format
covid_data['date'] = pd.to_datetime(covid_data['date'])

# Initialize the Dash app
app = Dash(__name__)

# Define the layout of the web application
app.layout = html.Div(children=[
    html.H1("COVID-19 Daily Deaths Increase Correlation"),
    dcc.Dropdown(
        id='country1-dropdown',
        options=[{'label': country, 'value': country} for country in covid_data['location'].unique()],
        value='Turkey',  # Default selected country1
        multi=False  # Single selection
    ),
    dcc.Dropdown(
        id='country2-dropdown',
        options=[{'label': country, 'value': country} for country in covid_data['location'].unique()],
        value='United Kingdom',  # Default selected country2
        multi=False  # Single selection
    ),
    dcc.Graph(id='correlation-chart'),
    html.Div(id='correlation-result')
])
```



```python
# Callback to update the chart and correlation result based on the selected countries
@app.callback(
    [Output('correlation-chart', 'figure'),
     Output('correlation-result', 'children')],
    [Input('country1-dropdown', 'value'),
     Input('country2-dropdown', 'value')]
)
def update_output(country1, country2):
    # Filter data for the selected countries
    data_country1 = covid_data[covid_data['location'] == country1].copy()
    data_country2 = covid_data[covid_data['location'] == country2].copy()

    # Calculate daily deaths increase
    data_country1['deaths_increase'] = data_country1['new_deaths'].diff(periods=1).dropna()
    data_country2['deaths_increase'] = data_country2['new_deaths'].diff(periods=1).dropna()

    # Merge data sets on date
    merged_data = pd.merge(data_country1, data_country2, on='date', suffixes=('_1', '_2'))

    # Calculate correlation
    correlation, p_value = pearsonr(merged_data['deaths_increase_1'].dropna(), merged_data['deaths_increase_2'].dropna())

    # Create scatter plot
    fig = px.scatter(merged_data, x='deaths_increase_1', y='deaths_increase_2', title=f'Correlation between Daily Deaths Increase in {country1}
    and {country2}', labels={'deaths_increase_1': f'Daily Deaths Increase in {country1}', 'deaths_increase_2': f'Daily Deaths Increase in {country2}'})

    # Determine correlation description and significance
    correlation_description = "positive" if correlation > 0 else "negative"
    significance = "statistically significant" if p_value < 0.05 else "not statistically significant"
    result_interpretation = f"The correlation between the daily deaths increase in {country1} and {country2} is
    {correlation_description} and {significance} (Correlation coefficient: {correlation:.2f}, P-value: {p_value:.2e})."

    return fig, result_interpretation

# Run the app
# Run the app on a different port, for example, 8070
if __name__ == '__main__':
    app.run_server(debug=True, port=8070)
```







COVID-19 Daily Deaths Increase Correlation

Turkey

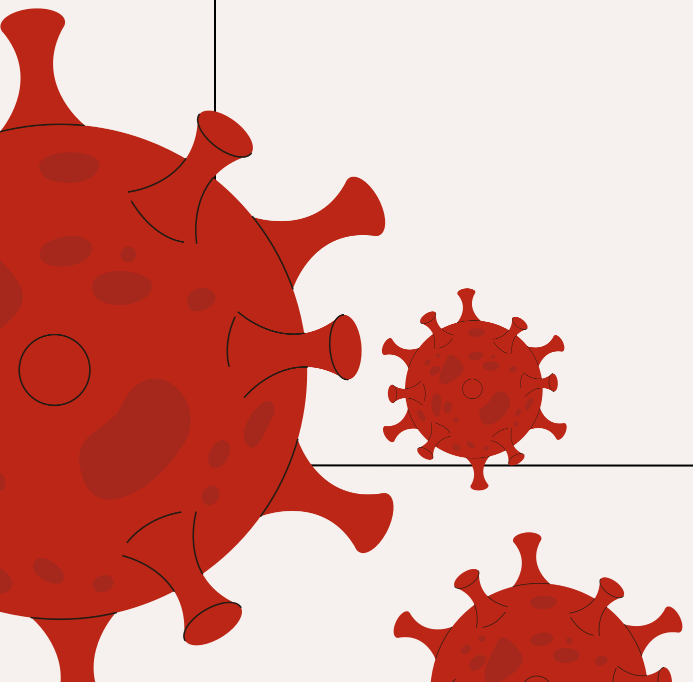
United Kingdom

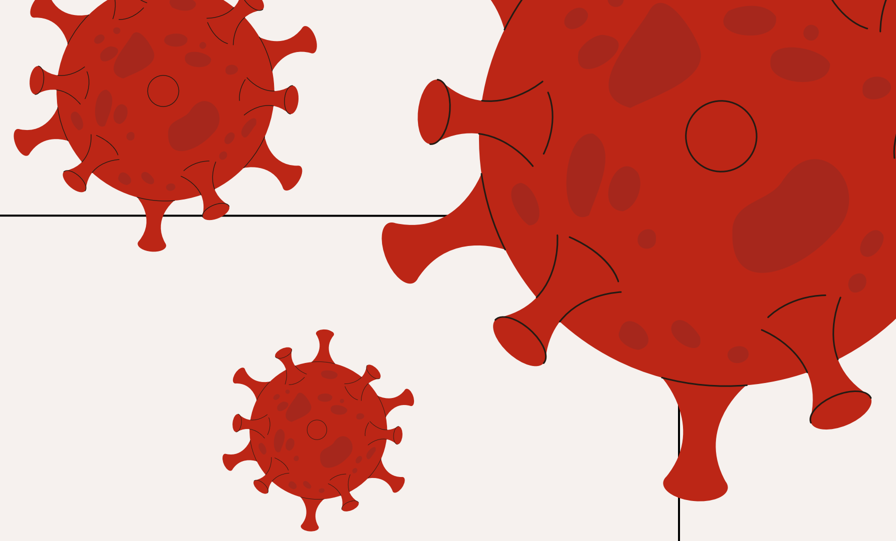
×
Correlation between Daily Deaths Increase in Turkey and United Kingdom
Daily Deaths Increase in United Kingdom
10k
5k
0
−5k
−10k
−2000
−1000
0
1000
2000
Daily Deaths Increase in Turkey

The correlation between the daily deaths increase in Turkey and United Kingdom is positive and statistically significant (Correlation coefficient: 0.53, P-value: 7.93e-106).

TEST PLAN

- Checking the results of data analyzes and correlation tests.
- Testing the accessibility features of the website.
- Testing the usability features of the website.
- Checking the reliability of the project.





THANK YOU FOR LISTENING