COVID-19 Data Interactive Dashboard Documentation

Objective:

The goal of this project is to create an interactive dashboard for visualizing COVID-19 data by country and date. This will allow users to explore various metrics such as confirmed cases, deaths, and recoveries.

Tools and Libraries Used:

- **pandas**: Data manipulation and processing.
- **plotly.express**: For data visualization.
- **dash**: To build the interactive dashboard.
- **dash_bootstrap_components**: For styling the dashboard.

Data Source:

The project uses COVID-19 data that can be sourced from public datasets like Johns Hopkins University's COVID-19 repository or other similar datasets. For this example, the data was imported using a CSV file.

Step-by-Step Explanation

1. Importing Required Libraries

```
import pandas as pd
import plotly.express as px
from dash import Dash, dcc, html
import dash_bootstrap_components as dbc
```

Explanation:
We begin by importing essential libraries:

- **pandas** for data manipulation.
- **plotly.express** for plotting.
- **dash** and **dash_bootstrap_components** for creating the interactive dashboard.

2. Loading and Viewing the Data

```
df_confirmed = pd.read_csv("time_series_covid19_confirmed_global.csv")
df_deaths = pd.read_csv("time_series_covid19_deaths_global.csv")
df_recovered = pd.read_csv("time_series_covid19_recovered_global.csv")
```

Explanation:

- **pd.read_csv()**: Loads the CSV file into a DataFrame..

Expected Output:

date	country	confirmed	deaths	recovered
2020-01-22	USA	1	0	0
2020-01-22	India	0	0	0

3. Preprocessing the Data

```
python
Copy code
data['date'] = pd.to_datetime(data['date'])
data = data.groupby(['date', 'country']).sum().reset_index()
```

Explanation:

- **pd.to_datetime()**: Converts the date column into a datetime object.
- **groupby()**: Groups the data by date and country, summing the confirmed cases, deaths, and recovered cases.

4. Creating the COVID-19 Cases Line Chart

```
python
Copy code
fig = px.line(data, x='date', y='confirmed', color='country', title='COVID-
19 Cases Over Time')
fig.show()
```

Explanation:

- **px.line()**: Generates a line chart with date on the x-axis, confirmed cases on the y-axis, and different lines colored by country.
- **fig.show()**: Displays the graph.

Expected Output:

A line graph where each line represents the confirmed COVID-19 cases for a country, visualized over time.

---

5. Building the Dash Layout

```
python
Copy code
app = Dash(__name__, external_stylesheets=[dbc.themes.BOOTSTRAP])
app.layout = html.Div([
    html.H1('COVID-19 Dashboard'),
    dcc.Graph(figure=fig)
])
```

Explanation:

- **Dash()**: Initializes the Dash application with Bootstrap for better design.
- **app.layout**: Defines the layout, which includes a title and the previously created figure.

6. Running the Dash Application

```
python
Copy code
if __name__ == '__main__':
    app.run_server(debug=True)
```

Explanation:

- **app.run_server()**: Starts the web server to host the interactive dashboard locally.

Expected Output:

The dashboard will be accessible through a local URL (e.g., `http://127.0.0.1:8050/`). Users will see the COVID-19 line chart and can interact with it, such as zooming into specific regions or hovering over points for details.

Conclusion

This project demonstrates how to build an interactive COVID-19 dashboard using Python libraries like `plotly` and `dash`. The final dashboard allows users to visually explore the COVID-19 data over time and across various countries.