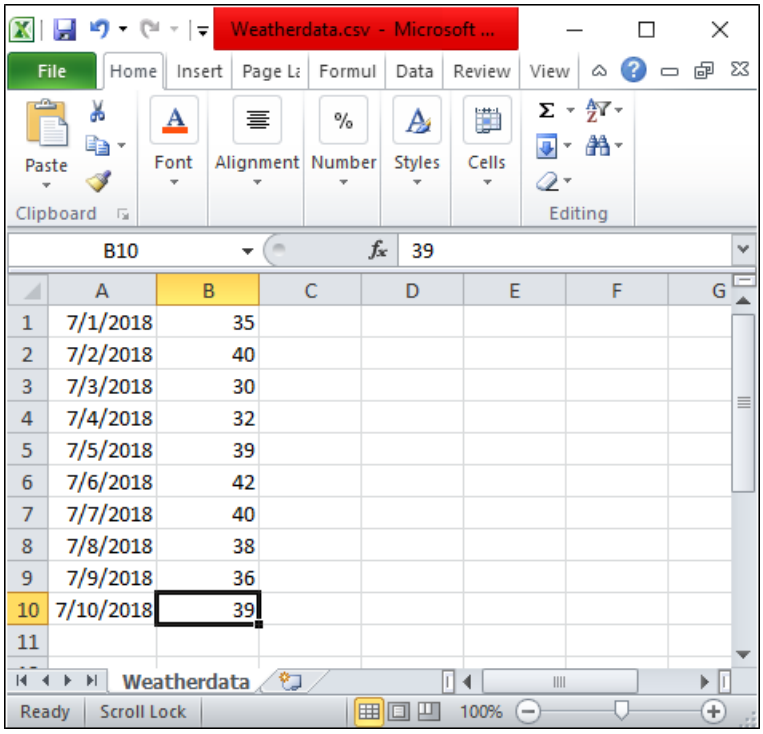
1. **Write a Python program to read data from a CSV file and create a line plot to visualize the trend over time. Customize the plot with appropriate labels, title, and color.**



	A	B
1	7/1/2018	35
2	7/2/2018	40
3	7/3/2018	30
4	7/4/2018	32
5	7/5/2018	39
6	7/6/2018	42
7	7/7/2018	40
8	7/8/2018	38
9	7/9/2018	36
10	7/10/2018	39
11		

Temperature (°C) on different dates is stored in a CSV file as 'Weatherdata.csv'. These two rows 'Dates' and 'Temperature (°C)' are used as X and Y-axis for visualizing weather reports.

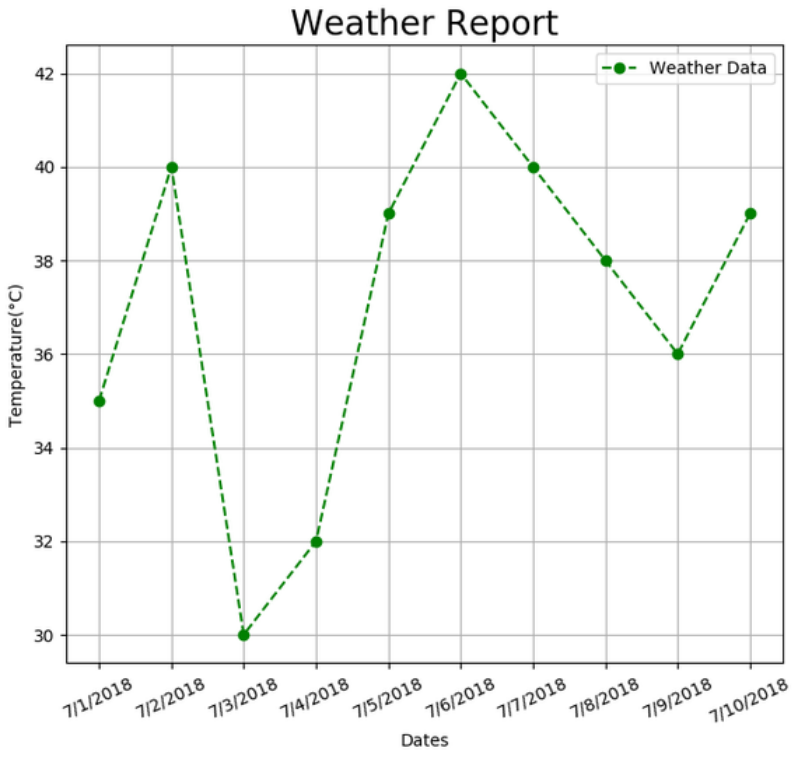
Approach of the program:

1. Import required libraries, matplotlib library for visualizing, and csv library for reading CSV data.
2. Open the file using open( ) function with 'r' mode (read-only) from CSV library and read the file using csv.reader( ) function.
3. Read each line in the file using for loop.
4. Append required columns of the CSV file into a list.
5. After reading the whole CSV file, plot the required data as X and Y axis.
6. In this Example, we are plotting Dates as X-axis and Temperature(°C) as Y-axis.

```
import matplotlib.pyplot as plt
import csv

x = []
y = []

with open('Weatherdata.csv','r') as csvfile:
```

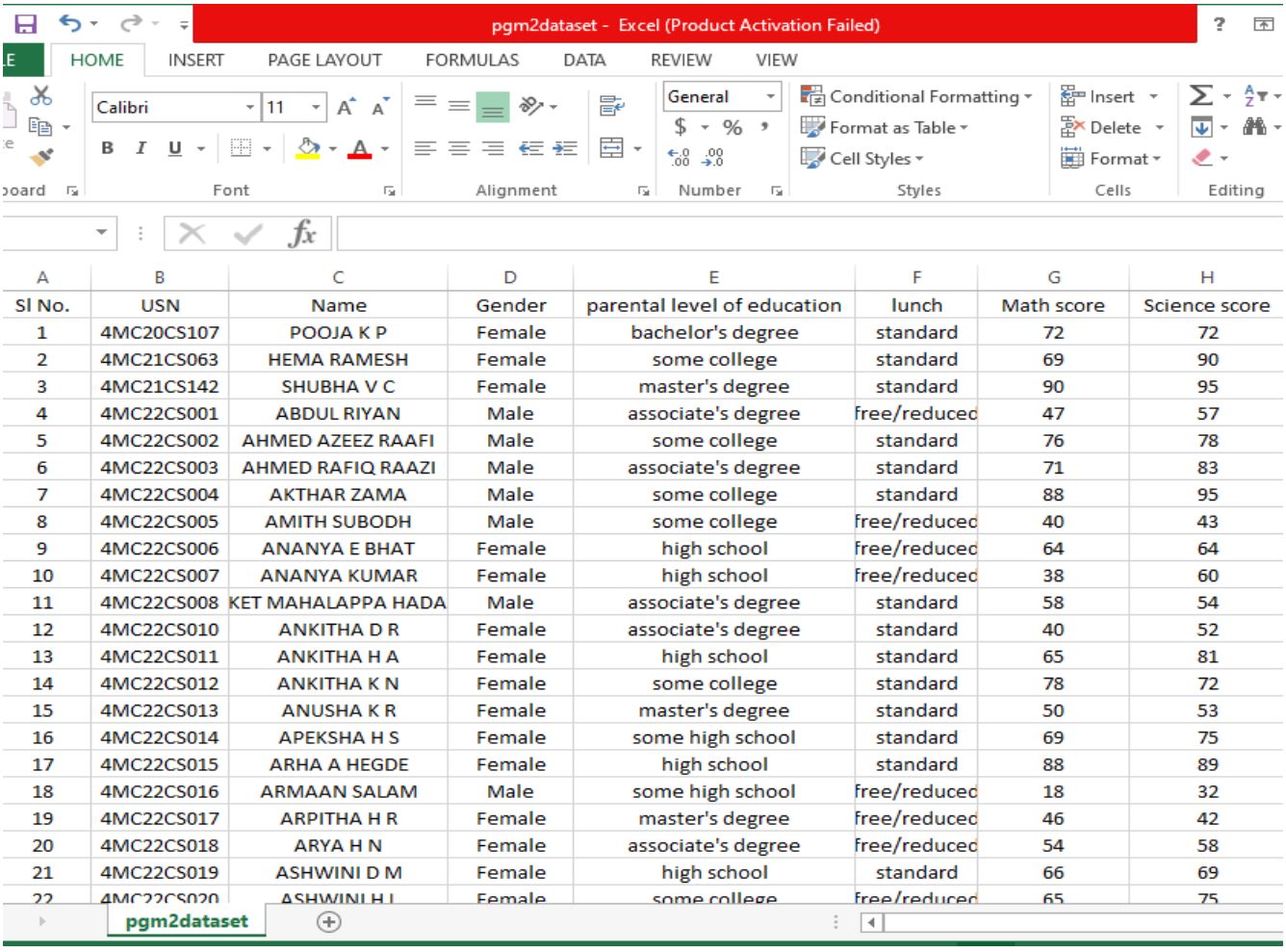
```
    lines = csv.reader(csvfile, delimiter=',')
    for row in lines:
        x.append(row[0])
        y.append(int(row[1]))

plt.plot(x, y, color = 'g', linestyle = 'dashed',
         marker = 'o',label = "Weather Data")

plt.xticks(rotation = 25)
plt.xlabel('Dates')
plt.ylabel('Temperature(°C)')
plt.title('Weather Report', fontsize = 20)
plt.grid()
plt.legend()
plt.show()
```

Output:



2. **Load a dataset containing information about students' scores in different subjects. Create a scatter plot to visualize the relationship between two variables (e.g., math score vs. science score). Use Seaborn to enhance the plot with appropriate styling and add labels.**

Visualizing Relationship between variables with scatter plots in Seaborn
To understand how variables in a dataset are related to one another and how that relationship is dependent on other variables, we perform statistical analysis. This Statistical analysis helps to visualize the trends and identify various patterns in the dataset. One of the functions which can be used to get the relationship between two variables in Seaborn is relplot().
Relplot() combines FacetGrid with either of the two axes-level functions scatterplot() and lineplot(). Scatterplot is default kind of relplot(). Using this we can visualize joint distribution of two variables through a cloud of points. We can draw scatterplot in seaborn using various ways. The most common one is when both the variables are numeric.

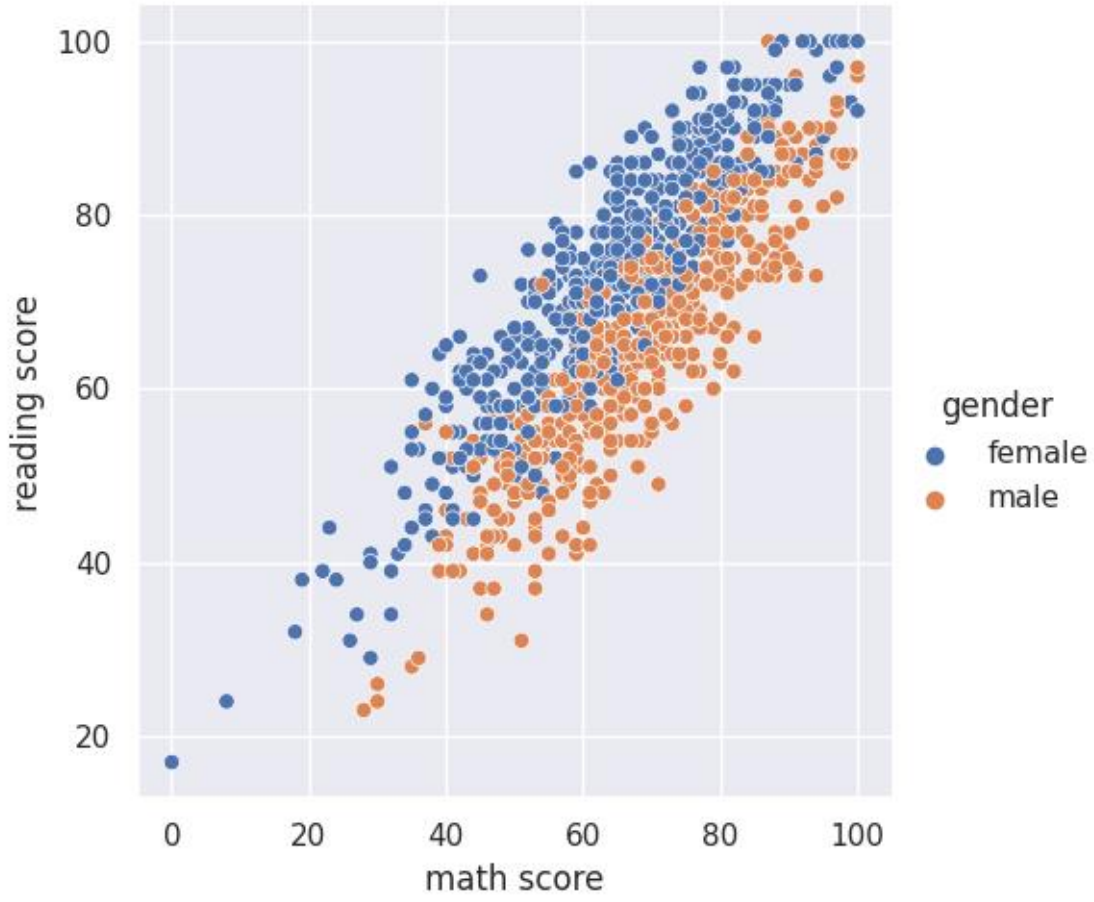


A	B	C	D	E	F	G	H
Sl No.	USN	Name	Gender	parental level of education	lunch	Math score	Science score
1	4MC20CS107	POOJA K P	Female	bachelor's degree	standard	72	72
2	4MC21CS063	HEMA RAMESH	Female	some college	standard	69	90
3	4MC21CS142	SHUBHA V C	Female	master's degree	standard	90	95
4	4MC22CS001	ABDUL RIYAN	Male	associate's degree	free/reduced	47	57
5	4MC22CS002	AHMED AZEEZ RAAFI	Male	some college	standard	76	78
6	4MC22CS003	AHMED RAFIQ RAAZI	Male	associate's degree	standard	71	83
7	4MC22CS004	AKTHAR ZAMA	Male	some college	standard	88	95
8	4MC22CS005	AMITH SUBODH	Male	some college	free/reduced	40	43
9	4MC22CS006	ANANYA E BHAT	Female	high school	free/reduced	64	64
10	4MC22CS007	ANANYA KUMAR	Female	high school	free/reduced	38	60
11	4MC22CS008	KET MAHALAPPA HADA	Male	associate's degree	standard	58	54
12	4MC22CS010	ANKITHA D R	Female	associate's degree	standard	40	52
13	4MC22CS011	ANKITHA H A	Female	high school	standard	65	81
14	4MC22CS012	ANKITHA K N	Female	some college	standard	78	72
15	4MC22CS013	ANUSHA K R	Female	master's degree	standard	50	53
16	4MC22CS014	APEKSHA H S	Female	some high school	standard	69	75
17	4MC22CS015	ARHA A HEGDE	Female	high school	standard	88	89
18	4MC22CS016	ARMAAN SALAM	Male	some high school	free/reduced	18	32
19	4MC22CS017	ARPITHA H R	Female	master's degree	free/reduced	46	42
20	4MC22CS018	ARYA H N	Female	associate's degree	free/reduced	54	58
21	4MC22CS019	ASHWINI D M	Female	high school	standard	66	69
22	4MC22CS020	ASHWINI H L	Female	some college	free/reduced	65	75

pgm2dataset

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# set grid style
sns.set(style ="darkgrid")

# import dataset
dataset = pd.read_csv('pgm2dataset.csv')
dataset.info()
sns.relplot(x ="math score", y ="reading score", hue ="gender", data = dataset);
```

Output:



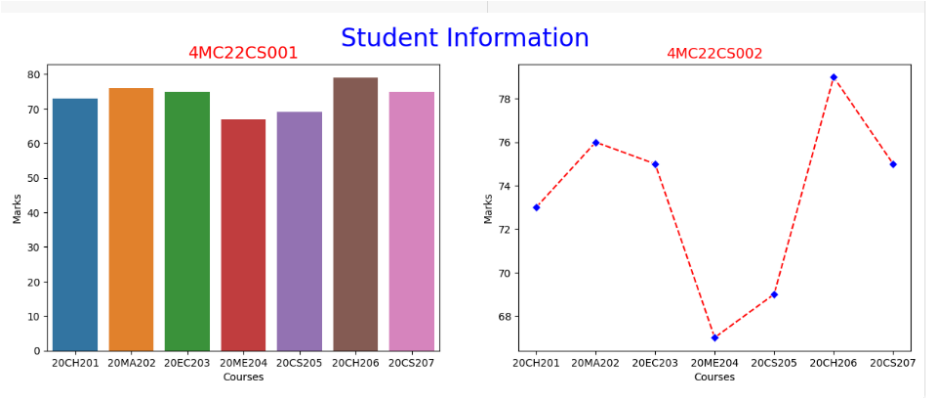
3. Given a dataset with multiple variables, create a figure with two subplots: one displaying a line plot and the other showing a bar chart. Customize the subplots with appropriate titles, legends, and colors.

A	B	C	D	E	F	G	H	I
USN	Name	20CH201	20MA202	20EC203	20ME204	20CS205	20CH206	20CS207
4MC20CS107	POOJA K P	79	75	76	73	72	80	77
4MC21CS063	HEMA RAMESH	75	76	75	69	70	77	73
4MC21CS142	SHUBHA V C	71	68	69	68	63	74	67
4MC22CS001	ABDUL RIYAN	73	76	75	67	69	79	75
4MC22CS002	AHMED AZEEZ RAAFI	85	88	85	81	81	88	82
4MC22CS003	AHMED RAFIQ RAAZI	85	83	83	80	82	86	83
4MC22CS004	AKTHAR ZAMA	95	91	92	91	87	95	94
4MC22CS005	AMITH SUBODH	89	85	88	79	81	90	68
4MC22CS006	ANANYA E BHAT	70	71	74	74	68	76	69
4MC22CS007	ANANYA KUMAR	90	88	87	84	84	90	87

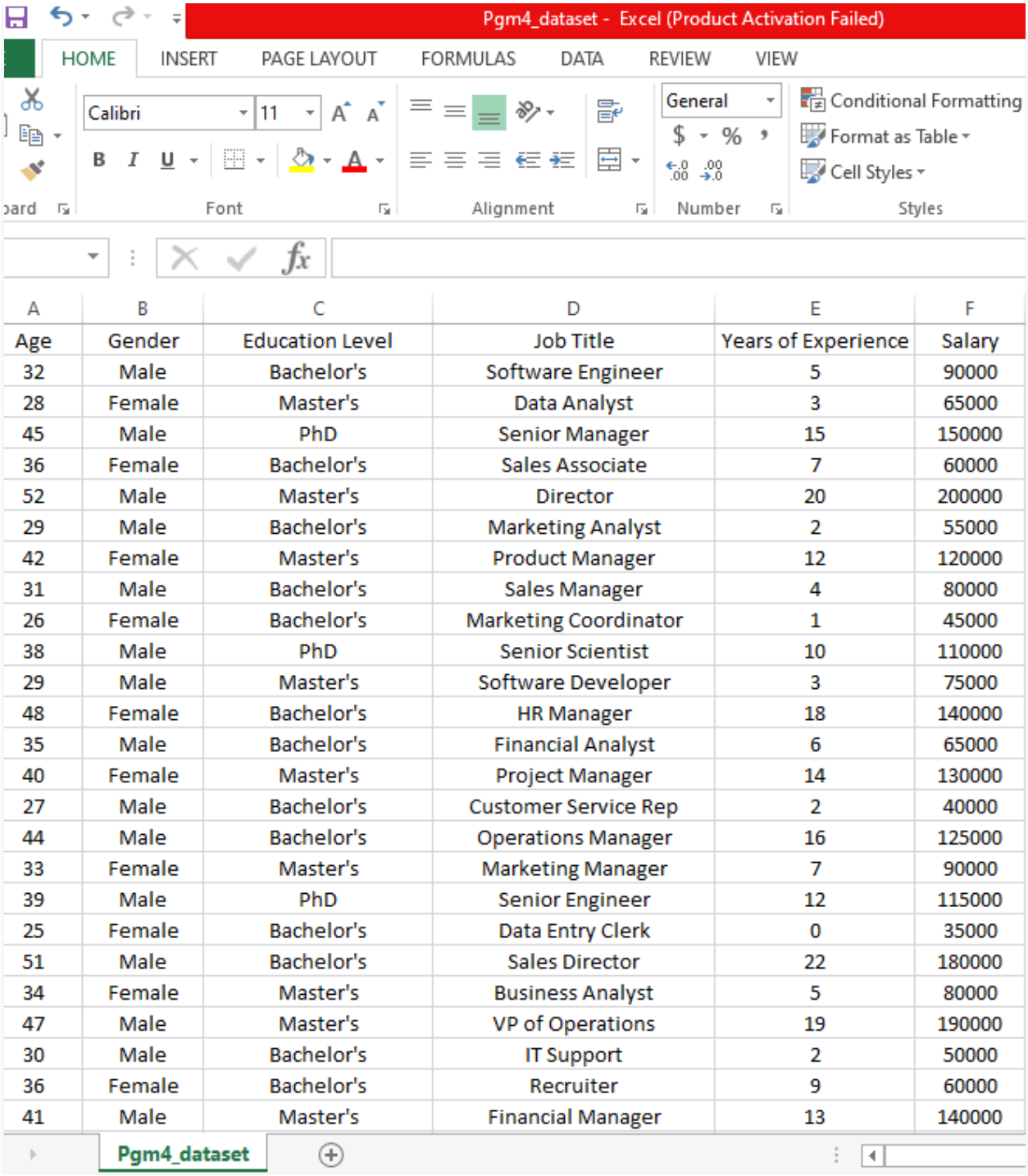
```
import pandas as pd
import seaborn as sns
from matplotlib import pyplot as plt

data = pd.read_csv('Pgm3_dataset.csv')
data.head()
data_num = data[['USN', '20CH201', '20MA202', '20EC203', '20ME204', '20CS205',
'20CH206', '20CS207']].set_index('USN')
Student1 = data_num.loc['4MC22CS001']
Student2 = data_num.loc['4MC22CS001']
fig, axes = plt.subplots(1, 2, figsize=(15, 5))
fig.suptitle('Student Information', size=24, color = 'blue')

# 4MC22CS001
sns.barplot(ax=axes[0], x=Student1.index, y=Student1.values)
axes[0].set(xlabel='Courses', ylabel='Marks')
axes[0].set_title('4MC22CS001', size=16, color = 'Red')

# 4MC22CS001
sns.lineplot(ax=axes[1], x=Student2.index, y=Student2.values, marker='D',
markerfacecolor='blue', color='Red', linestyle='--')
axes[1].set(xlabel='Courses', ylabel='Marks')
```

```
axes[1].set_title('4MC22CS002', size=14, color = 'Red')
plt.show()
```

Output:



4. **Load a dataset containing information about employees' salaries across different departments. Create a box plot and a violin plot to visualize the distribution of salaries by department. Customize the plots and add appropriate labels and titles.**



A	B	C	D	E	F
Age	Gender	Education Level	Job Title	Years of Experience	Salary
32	Male	Bachelor's	Software Engineer	5	90000
28	Female	Master's	Data Analyst	3	65000
45	Male	PhD	Senior Manager	15	150000
36	Female	Bachelor's	Sales Associate	7	60000
52	Male	Master's	Director	20	200000
29	Male	Bachelor's	Marketing Analyst	2	55000
42	Female	Master's	Product Manager	12	120000
31	Male	Bachelor's	Sales Manager	4	80000
26	Female	Bachelor's	Marketing Coordinator	1	45000
38	Male	PhD	Senior Scientist	10	110000
29	Male	Master's	Software Developer	3	75000
48	Female	Bachelor's	HR Manager	18	140000
35	Male	Bachelor's	Financial Analyst	6	65000
40	Female	Master's	Project Manager	14	130000
27	Male	Bachelor's	Customer Service Rep	2	40000
44	Male	Bachelor's	Operations Manager	16	125000
33	Female	Master's	Marketing Manager	7	90000
39	Male	PhD	Senior Engineer	12	115000
25	Female	Bachelor's	Data Entry Clerk	0	35000
51	Male	Bachelor's	Sales Director	22	180000
34	Female	Master's	Business Analyst	5	80000
47	Male	Master's	VP of Operations	19	190000
30	Male	Bachelor's	IT Support	2	50000
36	Female	Bachelor's	Recruiter	9	60000
41	Male	Master's	Financial Manager	13	140000

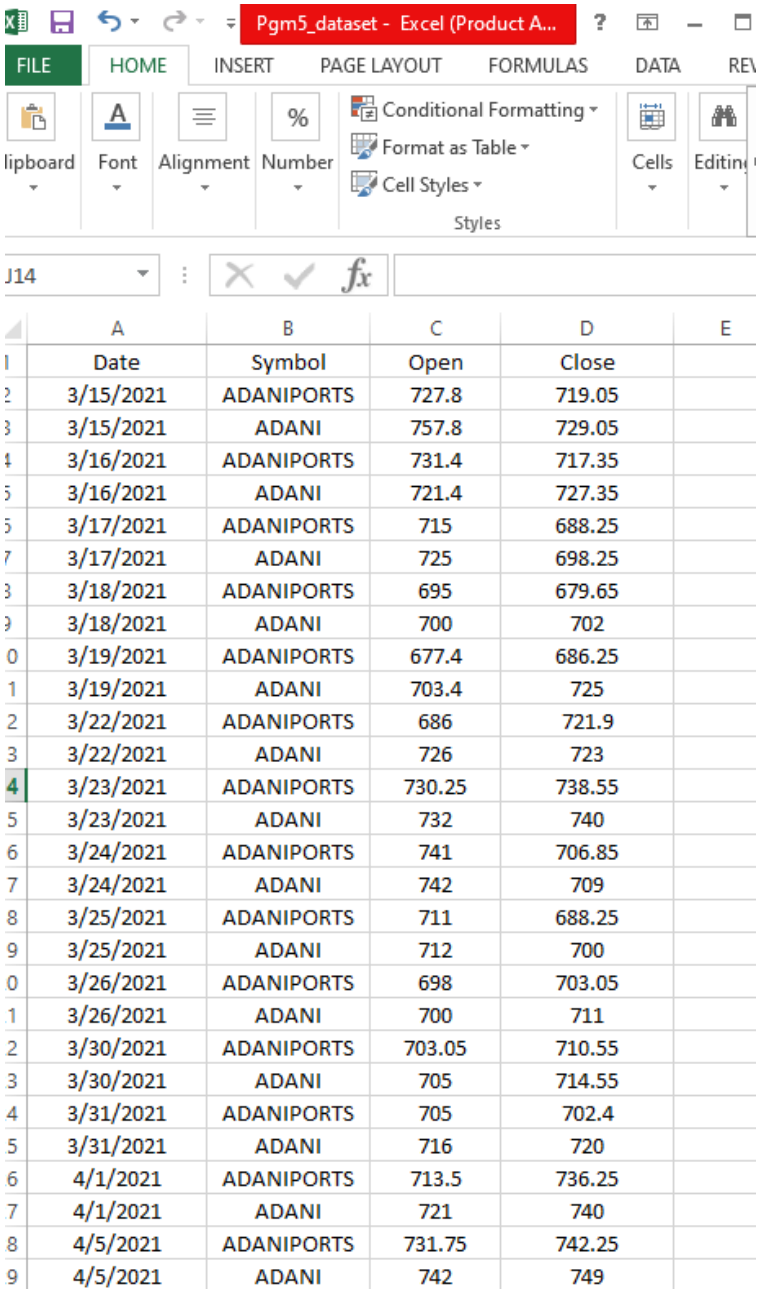
```
import matplotlib.pyplot as plt
import csv
import seaborn as sns
import pandas as pd
df=pd.read_csv('Pgm4_dataset.csv')
df.info()
# Set the plot size
plt.figure(figsize=(20,8))
sns.set_style("whitegrid")

sns.violinplot(data=df, x='Education Level', y='Salary')

# Set the plot title
plt.title('Salary Details', fontsize=16)

# Set the x-axis label
plt.xlabel('Education Level', fontsize=12)

# Set the y-axis label
plt.ylabel('Salary', fontsize=12)

# Display the plot
plt.show()

plt.title('Salary Details', fontsize=16)

sns.boxplot(x = 'Education Level', y = 'Salary', data = df)
plt.xlabel('Education Level', fontsize=12)

# Set the y-axis label
plt.ylabel('Salary', fontsize=12)
plt.show()
```

Output:


Salary Details
250000
200000
150000
100000
50000
0
Salary
Bachelor's
Master's
PhD
Education Level


Salary Details
250000
200000
150000
100000
50000
0
Salary
Bachelor's
Master's
PhD
Education Level

5. Load a dataset containing stock prices over time. Create a line plot to visualize the stock prices and add appropriate labels and titles. Format the x-axis tick labels to display the dates properly.

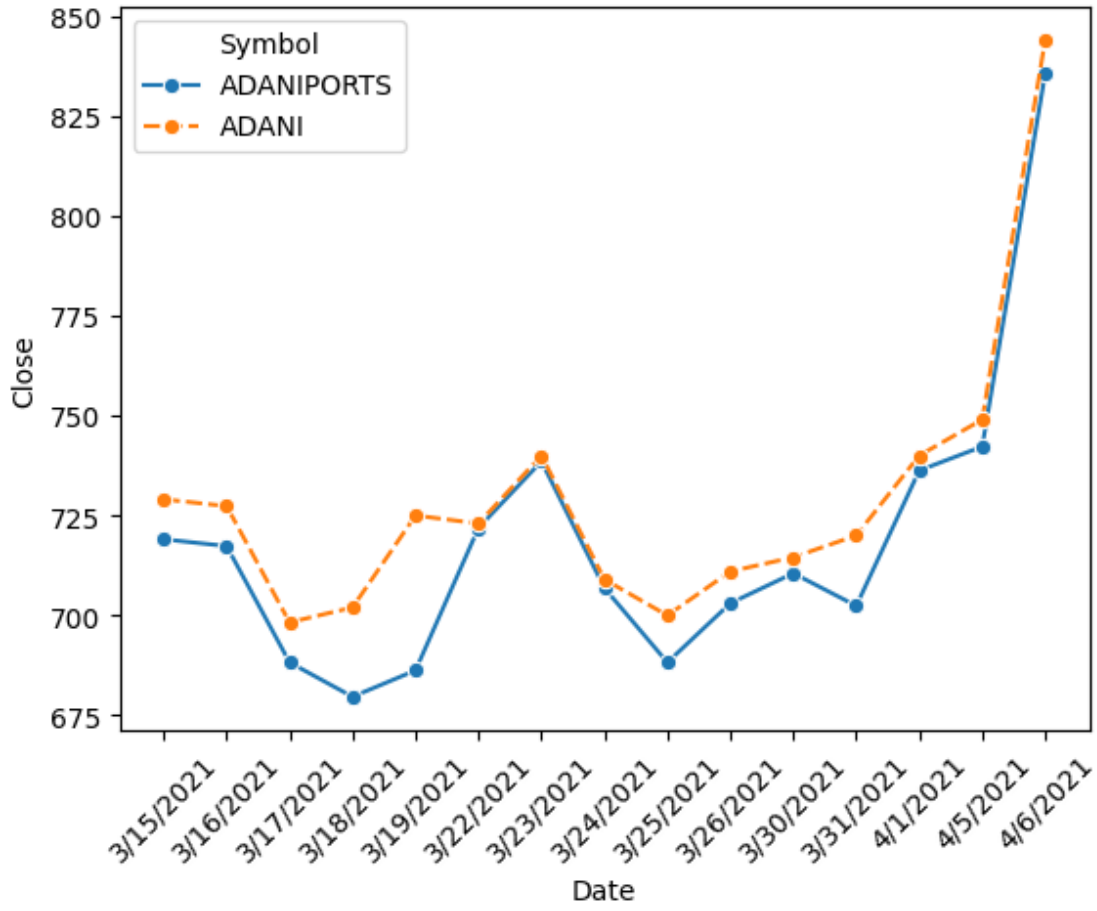


	A	B	C	D	E
1	Date	Symbol	Open	Close	
2	3/15/2021	ADANIPORTS	727.8	719.05	
3	3/15/2021	ADANI	757.8	729.05	
4	3/16/2021	ADANIPORTS	731.4	717.35	
5	3/16/2021	ADANI	721.4	727.35	
6	3/17/2021	ADANIPORTS	715	688.25	
7	3/17/2021	ADANI	725	698.25	
8	3/18/2021	ADANIPORTS	695	679.65	
9	3/18/2021	ADANI	700	702	
10	3/19/2021	ADANIPORTS	677.4	686.25	
11	3/19/2021	ADANI	703.4	725	
12	3/22/2021	ADANIPORTS	686	721.9	
13	3/22/2021	ADANI	726	723	
14	3/23/2021	ADANIPORTS	730.25	738.55	
15	3/23/2021	ADANI	732	740	
16	3/24/2021	ADANIPORTS	741	706.85	
17	3/24/2021	ADANI	742	709	
18	3/25/2021	ADANIPORTS	711	688.25	
19	3/25/2021	ADANI	712	700	
20	3/26/2021	ADANIPORTS	698	703.05	
21	3/26/2021	ADANI	700	711	
22	3/30/2021	ADANIPORTS	703.05	710.55	
23	3/30/2021	ADANI	705	714.55	
24	3/31/2021	ADANIPORTS	705	702.4	
25	3/31/2021	ADANI	716	720	
26	4/1/2021	ADANIPORTS	713.5	736.25	
27	4/1/2021	ADANI	721	740	
28	4/5/2021	ADANIPORTS	731.75	742.25	
29	4/5/2021	ADANI	742	749	

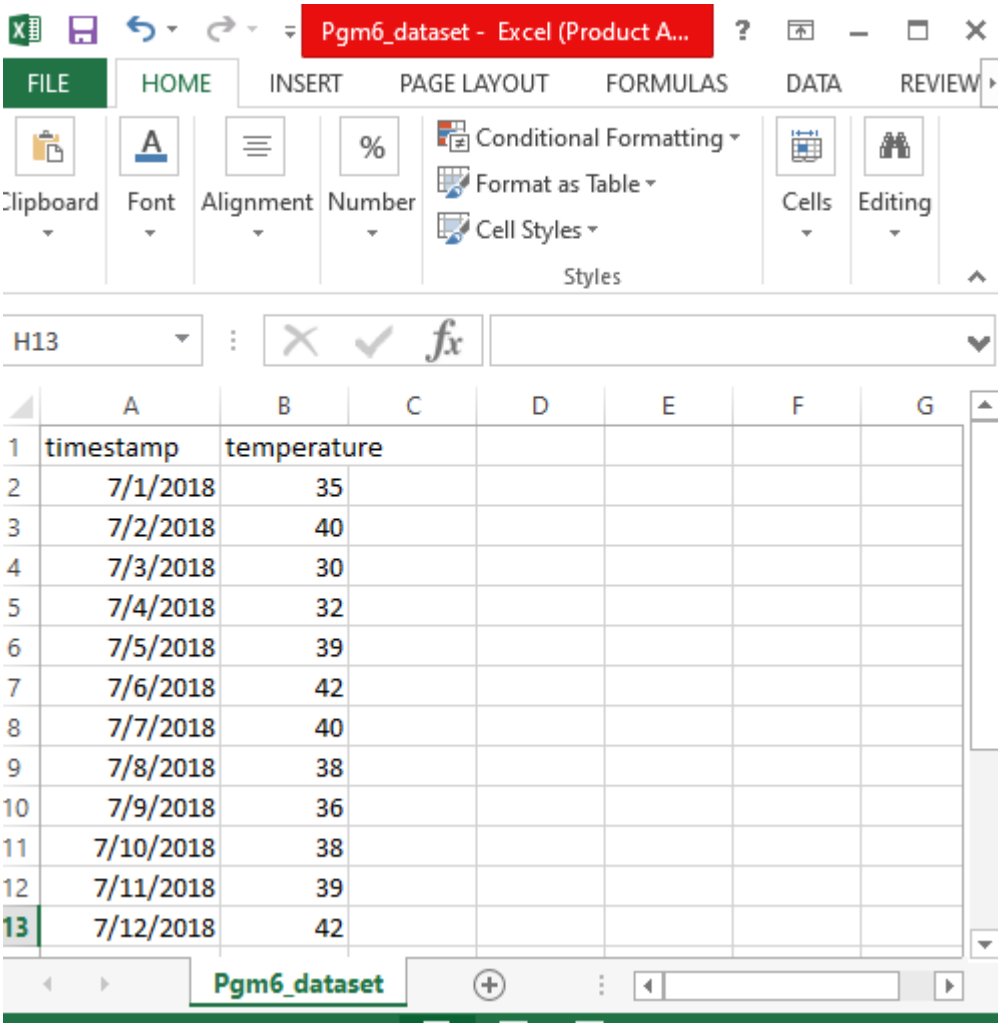
```
# importing packages
    import seaborn as sns
    import pandas as pd
    import csv
    import matplotlib.pyplot as plt
```

```
# loading dataset
    data = pd.read_csv("Pgm5_dataset.csv")

    sns.lineplot(x="Date", y="Close",
                 hue="Symbol", linestyle = 'dashed',style="Symbol", marker = 'o',
    data=data)
    plt.xlabel("Time Line")
    plt.ylabel("Closing Price")
    plt.xticks(rotation = 90, ha ='center', fontsize=20, fontfamily= "serif",
    fontweight='bold')
    plt.show()
```

Output:



6. **Load a dataset containing temperature readings over time. Create an interactive line plot using Plotly, which displays the temperature when hovering over the data points. Add appropriate labels and customize the plot's appearance.**

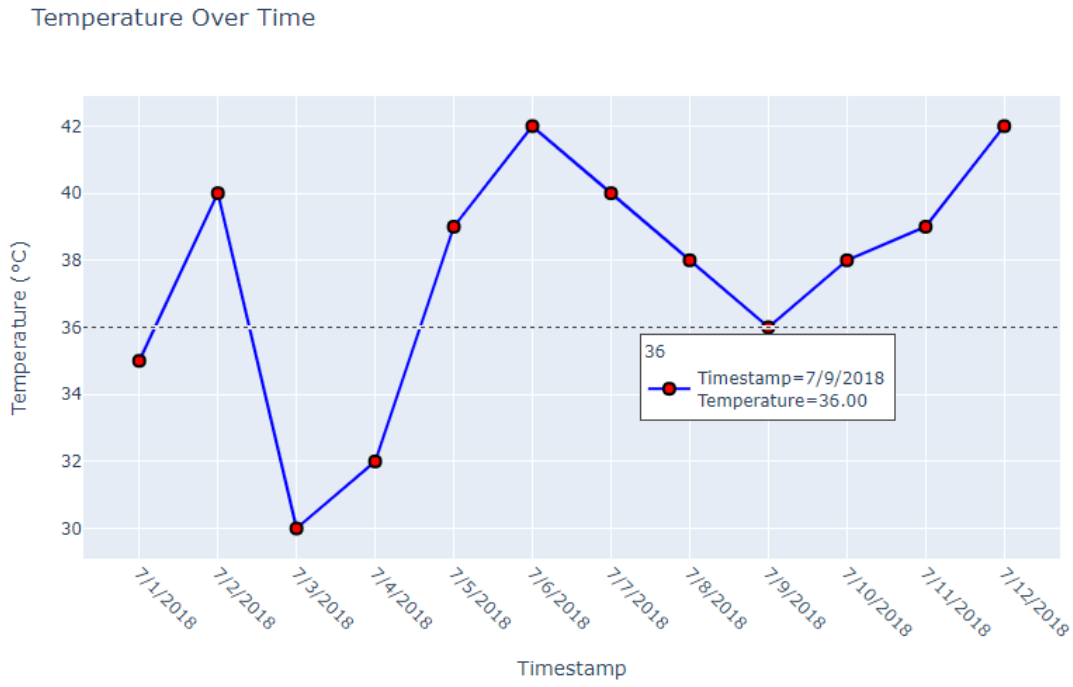


	A	B	C	D	E	F	G
1	timestamp	temperature					
2	7/1/2018	35					
3	7/2/2018	40					
4	7/3/2018	30					
5	7/4/2018	32					
6	7/5/2018	39					
7	7/6/2018	42					
8	7/7/2018	40					
9	7/8/2018	38					
10	7/9/2018	36					
11	7/10/2018	38					
12	7/11/2018	39					
13	7/12/2018	42					

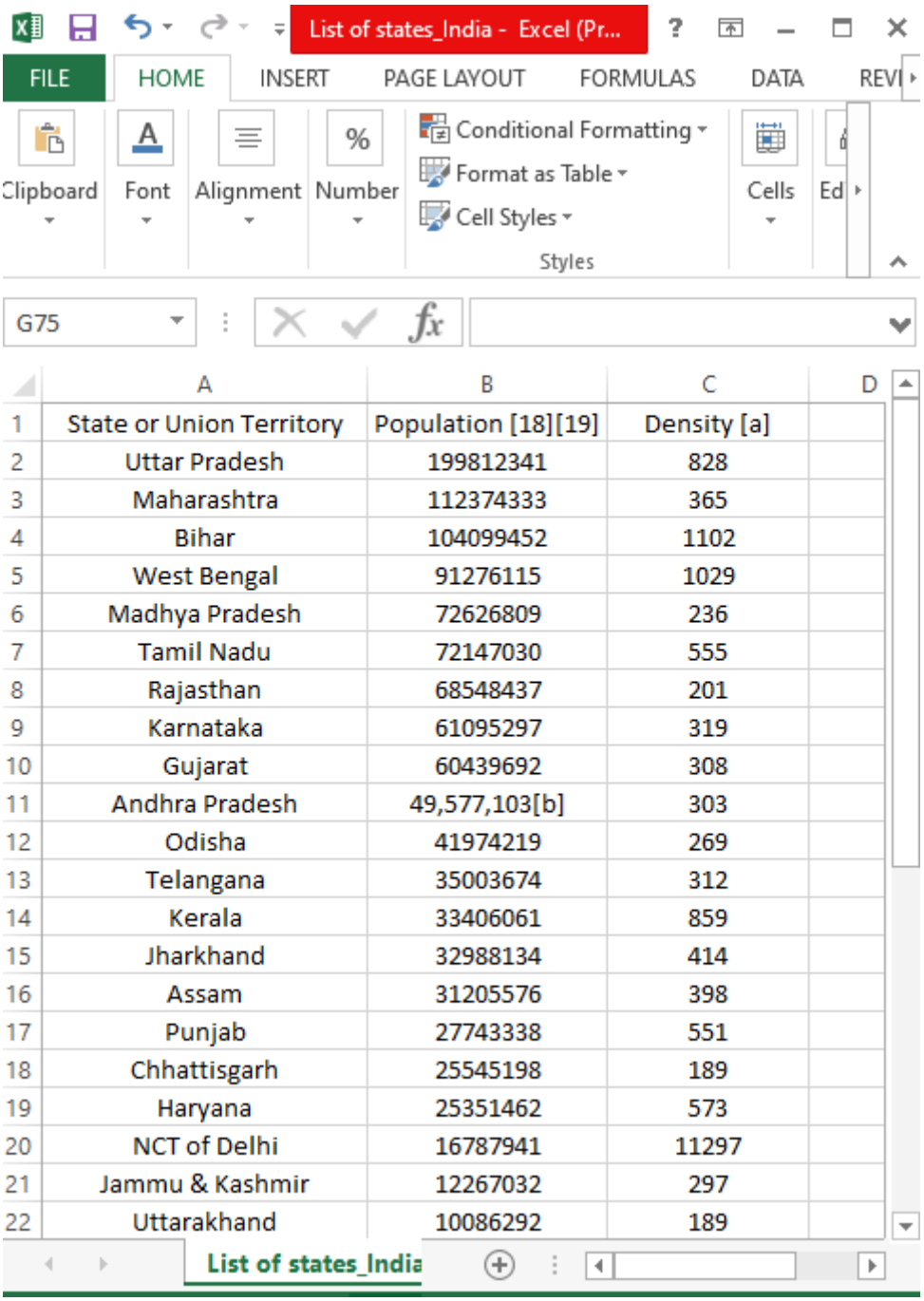
```
import pandas as pd
import plotly.express as px

# Load the dataset
df = pd.read_csv('Pgm6_dataset.csv')

# Create an interactive line plot
fig = px.line(df, x='timestamp', y='temperature', title='Temperature Over Time',
              labels={'timestamp': 'Timestamp', 'temperature': 'Temperature'},
              hover_data={'temperature': ':.2f'})

# Customize the appearance of the plot
```

```
fig.update_traces(mode='lines+markers', line=dict(color='blue', width=2),
                  marker=dict(color='red', size=8, line=dict(color='black', width=2)))

fig.update_layout(title_text='Temperature Over Time',
                  xaxis_title='Timestamp',
                  yaxis_title='Temperature (°C)',
                  hovermode='y unified')
fig.update_xaxes(tickangle = 45)
# Show the plot
fig.show()
import plotly.io as pio
pio.renderers.default = "browser"
```

Output:



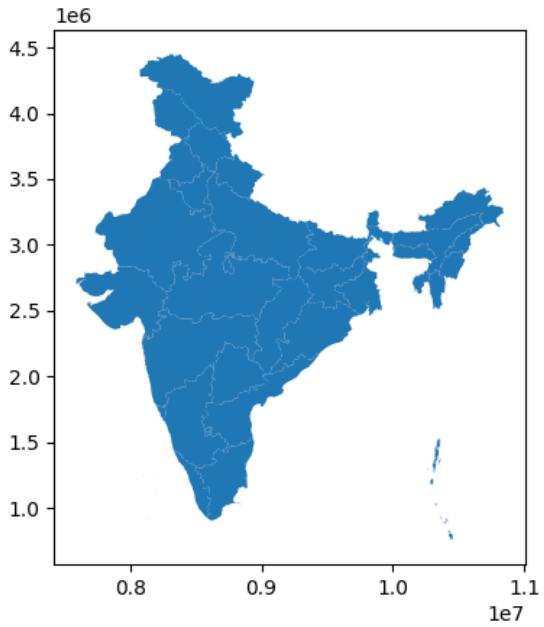
7. Load a dataset with information about population density by country. Create a choropleth map using GeoPandas to visualize the population density. Customize the map's appearance and add a color legend.



	A	B	C	D
1	State or Union Territory	Population [18][19]	Density [a]	
2	Uttar Pradesh	199812341	828	
3	Maharashtra	112374333	365	
4	Bihar	104099452	1102	
5	West Bengal	91276115	1029	
6	Madhya Pradesh	72626809	236	
7	Tamil Nadu	72147030	555	
8	Rajasthan	68548437	201	
9	Karnataka	61095297	319	
10	Gujarat	60439692	308	
11	Andhra Pradesh	49,577,103[b]	303	
12	Odisha	41974219	269	
13	Telangana	35003674	312	
14	Kerala	33406061	859	
15	Jharkhand	32988134	414	
16	Assam	31205576	398	
17	Punjab	27743338	551	
18	Chhattisgarh	25545198	189	
19	Haryana	25351462	573	
20	NCT of Delhi	16787941	11297	
21	Jammu & Kashmir	12267032	297	
22	Uttarakhand	10086292	189	

Sample:

```
import geopandas as gpd
import matplotlib.pyplot as plt
import mapclassify
data = gpd.read_file('India_State_Boundary.shp')
data.plot()
plt.show()
```

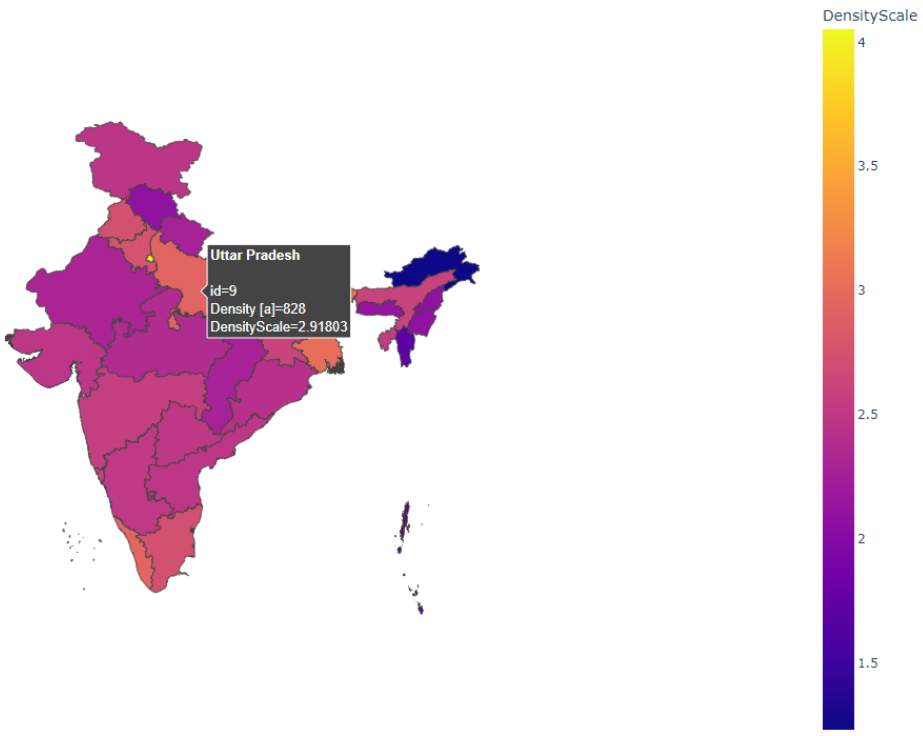


Main program

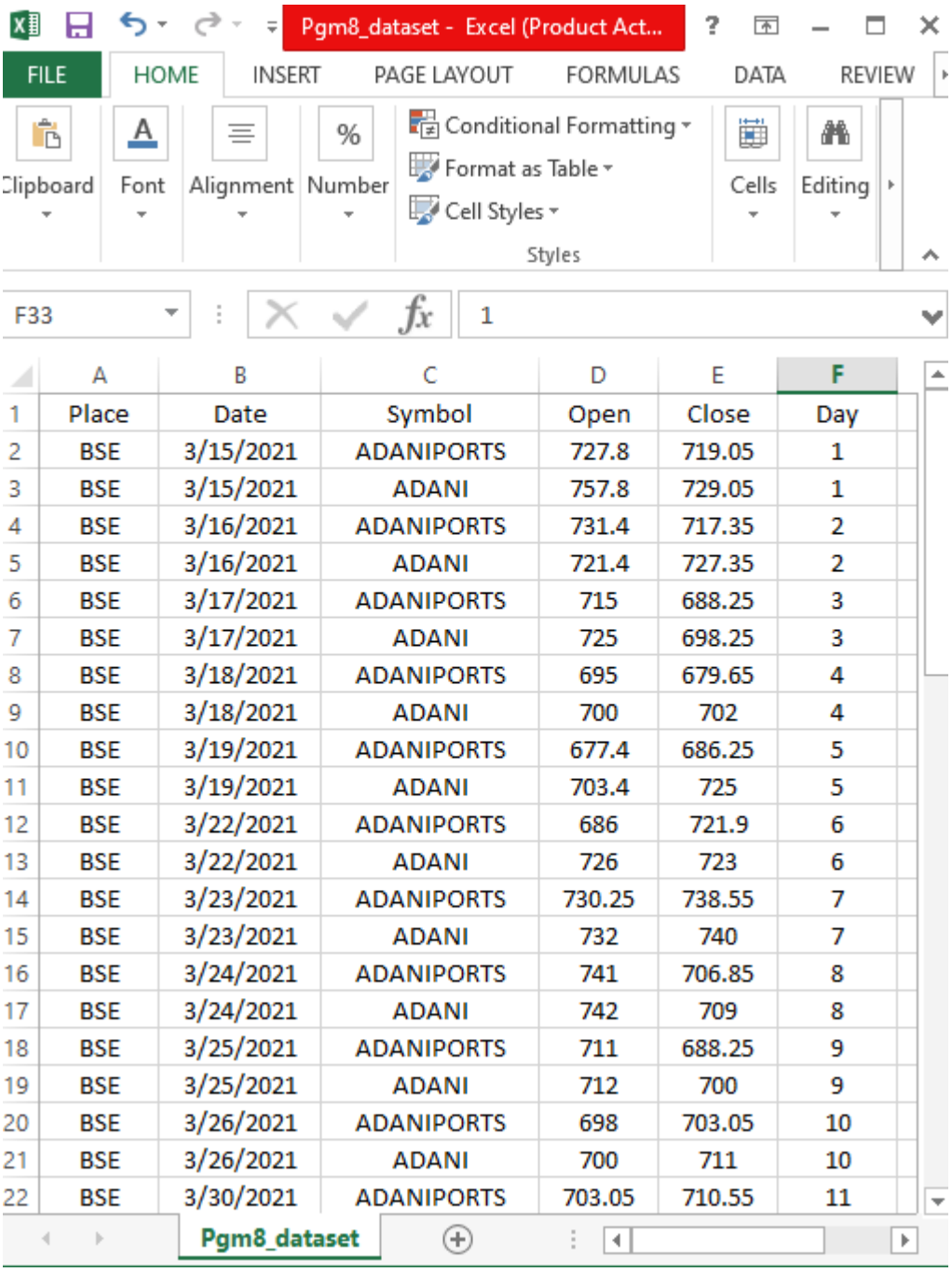
```
import plotly.express as px
import numpy as np
import pandas as pd
import json
import plotly.io as pio
pio.renderers.default = "browser"
india_states = json.load(open('states_india.geojson','r'))
df = pd.read_csv('List of states_India.csv')
state_id_map ={}
for feature in india_states['features']:
    feature['id'] = feature['properties']['state_code']
    state_id_map[feature['properties']['st_nm']] =feature['id']

df['id']=df['State or Union Territory'].apply(lambda x: state_id_map[x])

print(df.head())

print(state_id_map)

df['DensityScale'] = np.log10(df['Density [a]'])
```

```
fig = px.choropleth(df,
                    locations = 'id',
                    geojson=india_states,
                    color= 'DensityScale',
                    hover_name='State or Union Territory',
                    hover_data=['Density [a]'])
fig.update_geos(fitbounds= 'locations', visible=False)

fig.show()
```

Output:



8. Design and implement an interactive dashboard using Dash to display various visualizations. Include at least two interactive controls (e.g., dropdowns, sliders) to update the visualizations dynamically.



	A	B	C	D	E	F
1	Place	Date	Symbol	Open	Close	Day
2	BSE	3/15/2021	ADANIPORTS	727.8	719.05	1
3	BSE	3/15/2021	ADANI	757.8	729.05	1
4	BSE	3/16/2021	ADANIPORTS	731.4	717.35	2
5	BSE	3/16/2021	ADANI	721.4	727.35	2
6	BSE	3/17/2021	ADANIPORTS	715	688.25	3
7	BSE	3/17/2021	ADANI	725	698.25	3
8	BSE	3/18/2021	ADANIPORTS	695	679.65	4
9	BSE	3/18/2021	ADANI	700	702	4
10	BSE	3/19/2021	ADANIPORTS	677.4	686.25	5
11	BSE	3/19/2021	ADANI	703.4	725	5
12	BSE	3/22/2021	ADANIPORTS	686	721.9	6
13	BSE	3/22/2021	ADANI	726	723	6
14	BSE	3/23/2021	ADANIPORTS	730.25	738.55	7
15	BSE	3/23/2021	ADANI	732	740	7
16	BSE	3/24/2021	ADANIPORTS	741	706.85	8
17	BSE	3/24/2021	ADANI	742	709	8
18	BSE	3/25/2021	ADANIPORTS	711	688.25	9
19	BSE	3/25/2021	ADANI	712	700	9
20	BSE	3/26/2021	ADANIPORTS	698	703.05	10
21	BSE	3/26/2021	ADANI	700	711	10
22	BSE	3/30/2021	ADANIPORTS	703.05	710.55	11

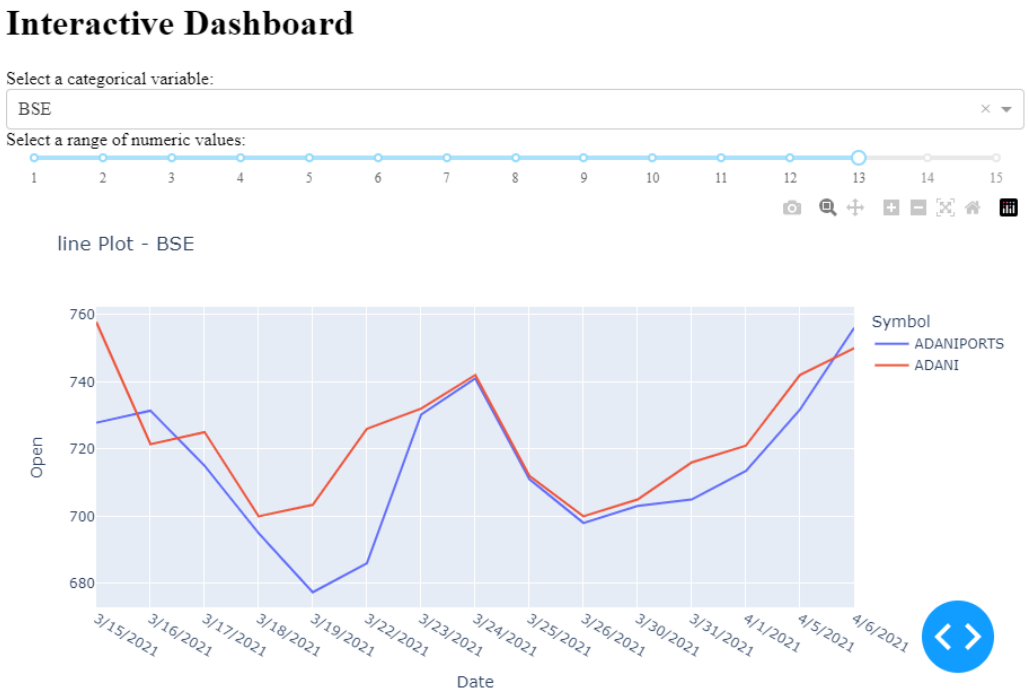
Pgm8_dataset

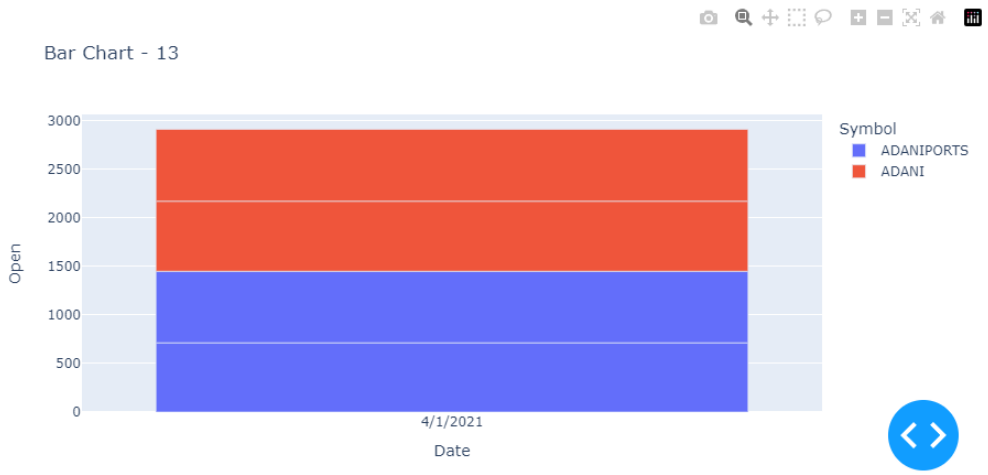
```
import dash
from dash import dcc, html
from dash.dependencies import Input, Output
import pandas as pd
import plotly.express as px
```

```
# Load a sample dataset
df = pd.read_csv('Stock_details_8.csv')  # Replace 'your_data.csv' with the actual file name

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

# Define the layout of the app
app.layout = html.Div([
    html.H1("Interactive Dashboard"),

    # Dropdown for selecting a categorical variable
    html.Label("Select a categorical variable:"),
    dcc.Dropdown(
        id='category-dropdown',
        options= df['Place'].unique(),
        value="BSE",  # Default selected column
        ),

    # Slider for selecting a range of numeric values
    html.Label("Select a range of numeric values:"),
    dcc.Slider(
        id='slider',
        min=df['Day'].min(),
        max=df['Day'].max(),
        step=1,
        value=df['Day'].max()  # Default selected range
    ),

    # Scatter plot for visualization
    dcc.Graph(id='line-plot'),

    # Bar chart for visualization
    dcc.Graph(id='bar-chart'),
])

# Define callback to update scatter plot and bar chart based on dropdown and slider values
@app.callback(
    [Output('line-plot', 'figure'),
     Output('bar-chart', 'figure')],
    [Input('category-dropdown', 'value'),
     Input('slider', 'value')]
)
```

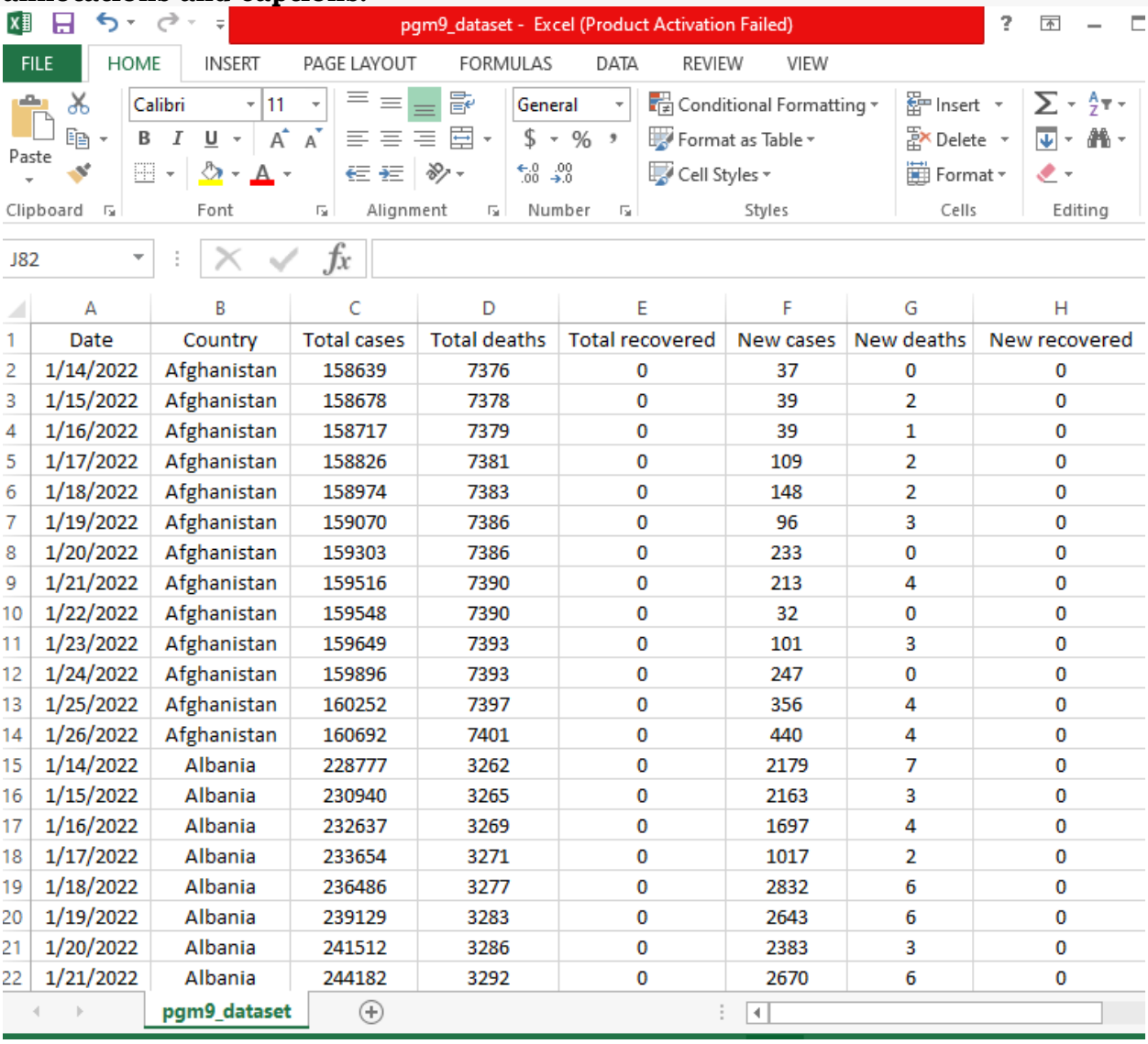
```
def update_visualizations(selected_category, selected_range):
    filtered_df = df[df['Place'] == selected_category]
    filtered_df_bar = df[df['Day']== selected_range]

    # Scatter plot
    line_fig = px.line(filtered_df, x='Date', y='Open', color='Symbol',
                       title=f"line Plot - {selected_category} ")

    # Bar chart
    bar_fig = px.bar(filtered_df_bar, x='Date', y='Open', color='Symbol', title=f"Bar Chart
- {selected_range}")

    return line_fig, bar_fig

# Run the app
if __name__ == '__main__':
    app.run_server(debug=True)
```

Output:




Bar Chart - 13
3000
2500
2000
1500
1000
500
0
Open
4/1/2021
Date
Symbol
ADANIPORTS
ADANI

9. Select a dataset related to a specific topic of interest (e.g., climate change, COVID-19). Design a series of visualizations that tell a compelling data story, highlighting key insights and trends. Present the visualizations with appropriate annotations and captions.



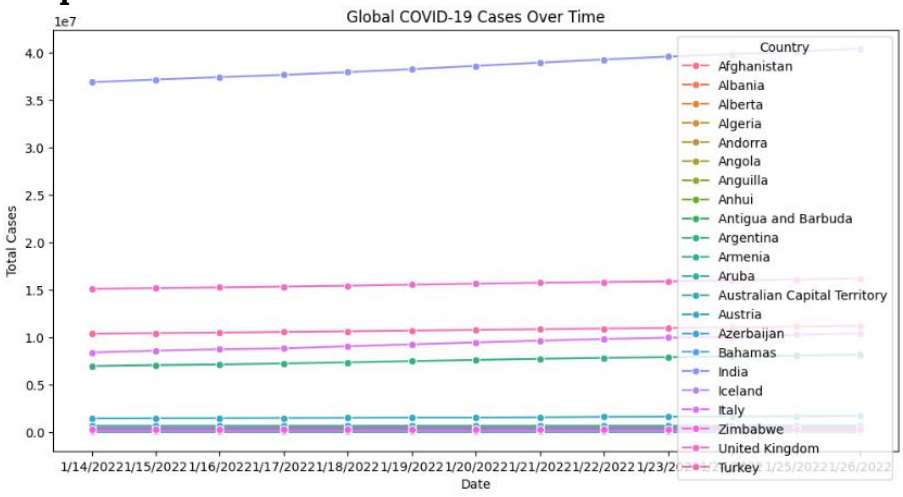
	A	B	C	D	E	F	G	H
1	Date	Country	Total cases	Total deaths	Total recovered	New cases	New deaths	New recovered
2	1/14/2022	Afghanistan	158639	7376	0	37	0	0
3	1/15/2022	Afghanistan	158678	7378	0	39	2	0
4	1/16/2022	Afghanistan	158717	7379	0	39	1	0
5	1/17/2022	Afghanistan	158826	7381	0	109	2	0
6	1/18/2022	Afghanistan	158974	7383	0	148	2	0
7	1/19/2022	Afghanistan	159070	7386	0	96	3	0
8	1/20/2022	Afghanistan	159303	7386	0	233	0	0
9	1/21/2022	Afghanistan	159516	7390	0	213	4	0
10	1/22/2022	Afghanistan	159548	7390	0	32	0	0
11	1/23/2022	Afghanistan	159649	7393	0	101	3	0
12	1/24/2022	Afghanistan	159896	7393	0	247	0	0
13	1/25/2022	Afghanistan	160252	7397	0	356	4	0
14	1/26/2022	Afghanistan	160692	7401	0	440	4	0
15	1/14/2022	Albania	228777	3262	0	2179	7	0
16	1/15/2022	Albania	230940	3265	0	2163	3	0
17	1/16/2022	Albania	232637	3269	0	1697	4	0
18	1/17/2022	Albania	233654	3271	0	1017	2	0
19	1/18/2022	Albania	236486	3277	0	2832	6	0
20	1/19/2022	Albania	239129	3283	0	2643	6	0
21	1/20/2022	Albania	241512	3286	0	2383	3	0
22	1/21/2022	Albania	244182	3292	0	2670	6	0

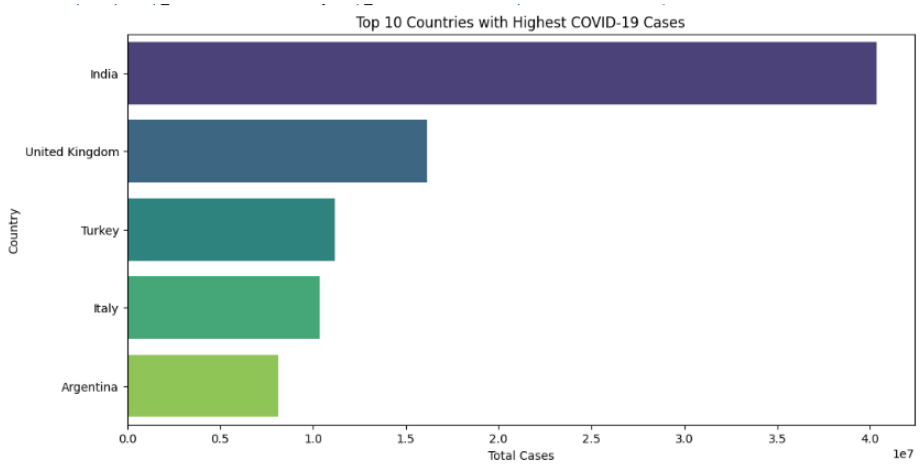
pgm9_dataset

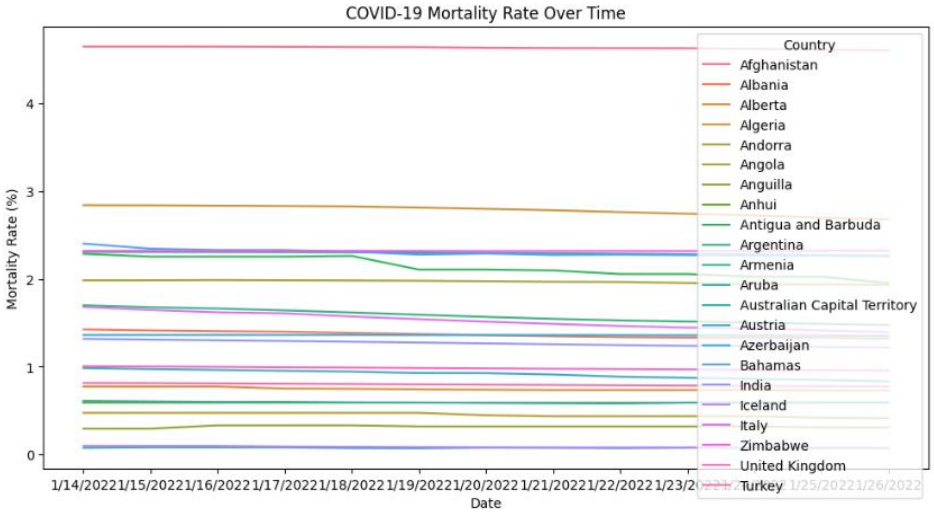
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load COVID-19 dataset (replace 'your_dataset.csv' with the actual dataset)
covid_data = pd.read_csv('pgm9_dataset.csv')
plt.figure(figsize=(12, 6))
sns.lineplot(x='Date', y='Total cases', hue = 'Country', marker = 'o', data=covid_data)
plt.title('Global COVID-19 Cases Over Time')
plt.xlabel('Date')
```

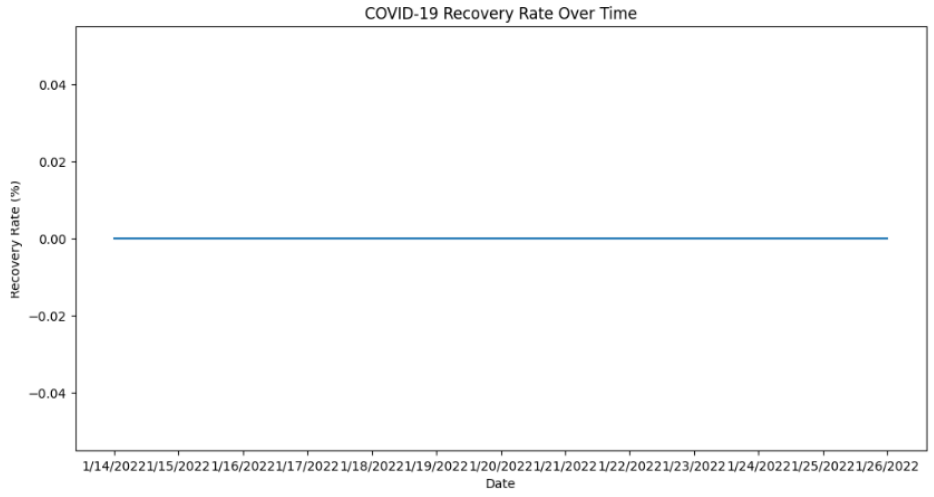
```
plt.ylabel('Total Cases')
plt.show()
top_countries = covid_data.groupby('Country')['Total
cases'].max().sort_values(ascending=False).head(5)
plt.figure(figsize=(12, 6))
sns.barplot(x=top_countries.values, y=top_countries.index, palette='viridis')
plt.title('Top 10 Countries with Highest COVID-19 Cases')
plt.xlabel('Total Cases')
plt.ylabel('Country')
plt.show()
covid_data['MortalityRate'] = (covid_data['Total deaths'] / covid_data['Total cases']) *
100
plt.figure(figsize=(12, 6))
sns.lineplot(x='Date', y='MortalityRate', hue = 'Country', data=covid_data)
plt.title('COVID-19 Mortality Rate Over Time')
plt.xlabel('Date')
plt.ylabel('Mortality Rate (%)')
plt.show()
covid_data['RecoveryRate'] = (covid_data['Total recovered'] / covid_data['Total cases']) *
100
plt.figure(figsize=(12, 6))
sns.lineplot(x='Date', y='RecoveryRate', data=covid_data)
plt.title('COVID-19 Recovery Rate Over Time')
plt.xlabel('Date')
plt.ylabel('Recovery Rate (%)')
plt.show()
```

Output:

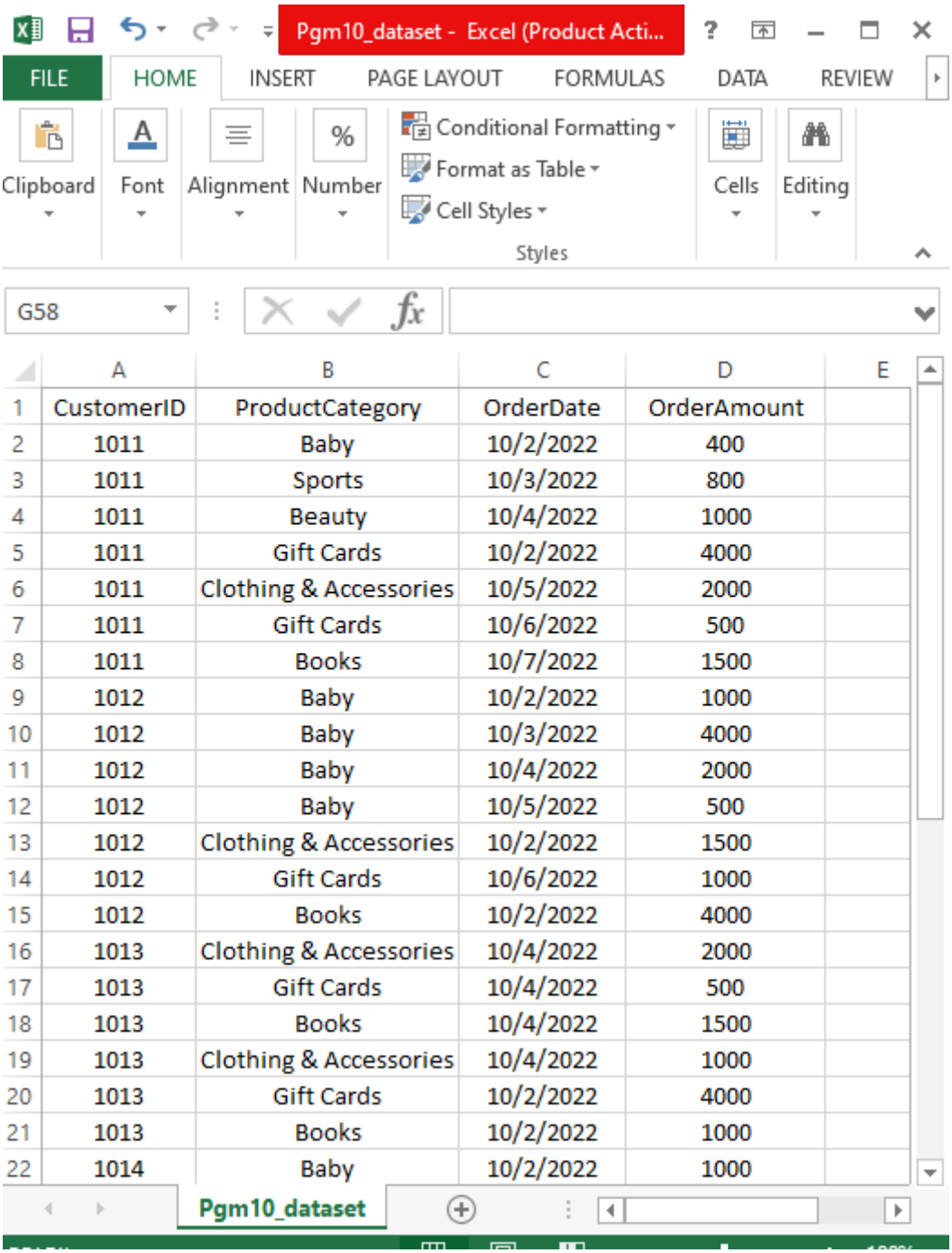



Top 10 Countries with Highest COVID-19 Cases
India
United Kingdom
Turkey
Italy
Argentina
Country
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
1e7
Total Cases


COVID-19 Mortality Rate Over Time
Country
Afghanistan
Albania
Alberta
Algeria
Andorra
Angola
Anguilla
Anhui
Antigua and Barbuda
Argentina
Armenia
Aruba
Australian Capital Territory
Austria
Azerbaijan
Bahamas
India
Iceland
Italy
Zimbabwe
United Kingdom
Turkey
Mortality Rate (%)
0
1
2
3
4
1/14/2022
1/15/2022
1/16/2022
1/17/2022
1/18/2022
1/19/2022
1/20/2022
1/21/2022
1/22/2022
Date


COVID-19 Recovery Rate Over Time
Recovery Rate (%)
0.04
0.02
0.00
−0.02
−0.04
1/14/2022
1/15/2022
1/16/2022
1/17/2022
1/18/2022
1/19/2022
1/20/2022
1/21/2022
1/22/2022
1/23/2022
1/24/2022
1/25/2022
1/26/2022
Date

10. Choose a dataset related to a real-world problem (e.g., retail sales, customer behavior). Explore the dataset, identify interesting patterns, and design a set of visualizations to present the findings effectively. Present the visualizations along with a brief explanation of the insights gained.

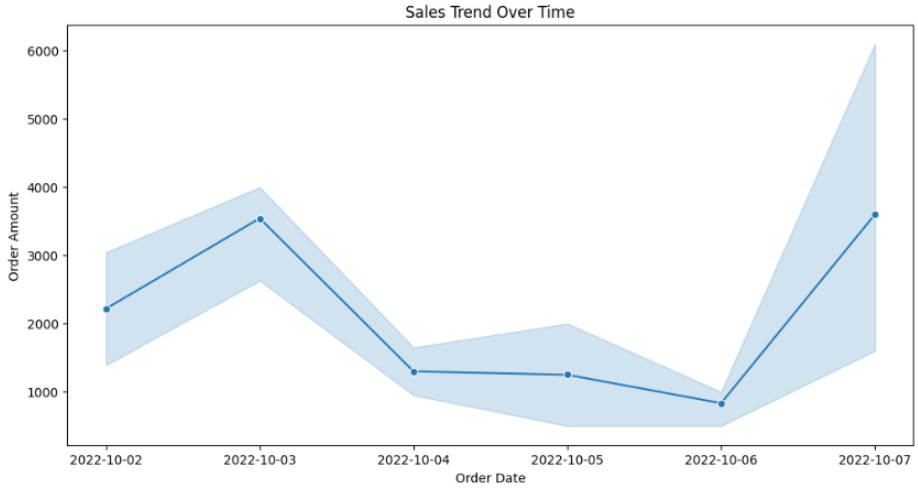


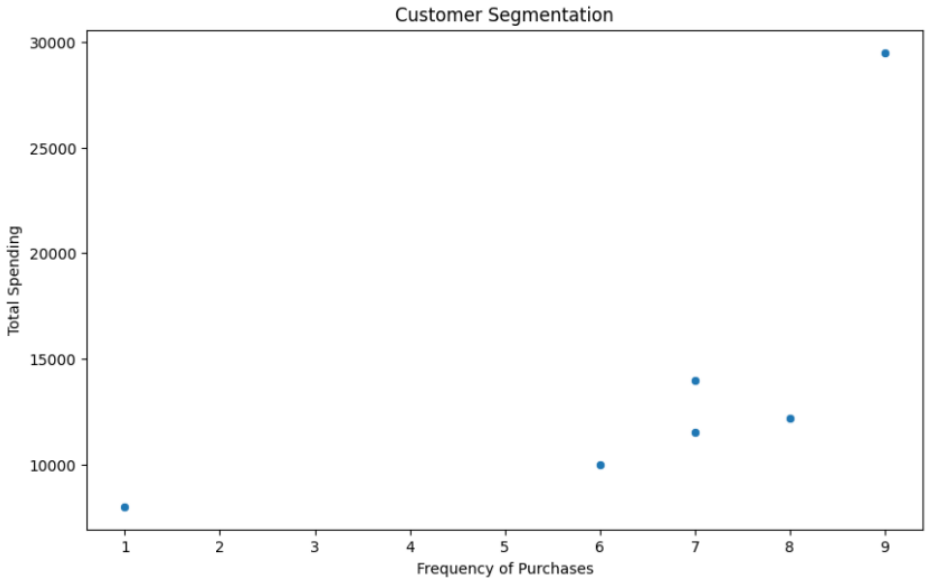
	A	B	C	D	E
1	CustomerID	ProductCategory	OrderDate	OrderAmount	
2	1011	Baby	10/2/2022	400	
3	1011	Sports	10/3/2022	800	
4	1011	Beauty	10/4/2022	1000	
5	1011	Gift Cards	10/2/2022	4000	
6	1011	Clothing & Accessories	10/5/2022	2000	
7	1011	Gift Cards	10/6/2022	500	
8	1011	Books	10/7/2022	1500	
9	1012	Baby	10/2/2022	1000	
10	1012	Baby	10/3/2022	4000	
11	1012	Baby	10/4/2022	2000	
12	1012	Baby	10/5/2022	500	
13	1012	Clothing & Accessories	10/2/2022	1500	
14	1012	Gift Cards	10/6/2022	1000	
15	1012	Books	10/2/2022	4000	
16	1013	Clothing & Accessories	10/4/2022	2000	
17	1013	Gift Cards	10/4/2022	500	
18	1013	Books	10/4/2022	1500	
19	1013	Clothing & Accessories	10/4/2022	1000	
20	1013	Gift Cards	10/2/2022	4000	
21	1013	Books	10/2/2022	1000	
22	1014	Baby	10/2/2022	1000	

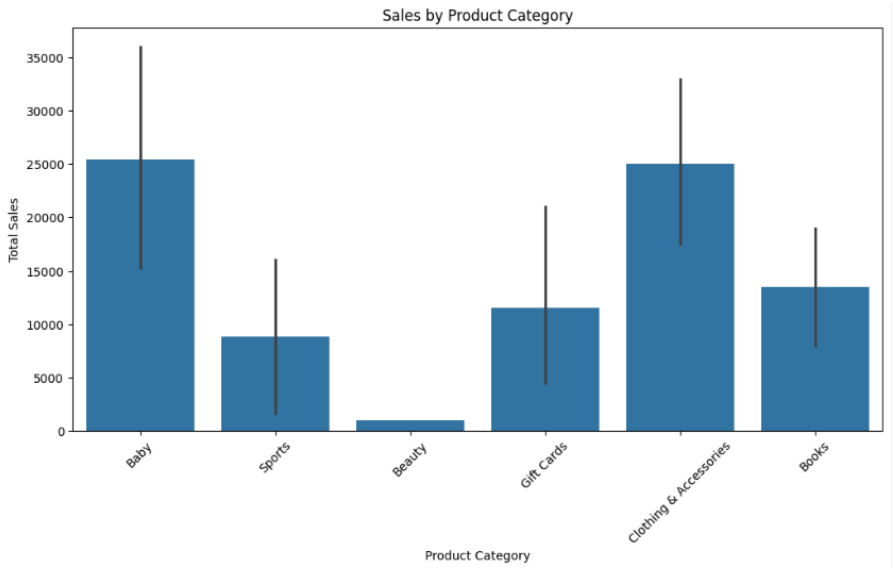
```
import pandas as pd
```

```
import matplotlib.pyplot as plt
import seaborn as sns

retail_data = pd.read_csv('Pgm10_dataset.csv')

# Display the first few rows of the dataset
print(retail_data.head())

# Get summary statistics
print(retail_data.describe())

# Check for missing values
print(retail_data.isnull().sum())
retail_data['OrderDate'] = pd.to_datetime(retail_data['OrderDate'])
plt.figure(figsize=(12, 6))
sns.lineplot(x='OrderDate', y='OrderAmount',marker = 'o', data=retail_data)
plt.title('Sales Trend Over Time')
plt.xlabel('Order Date')
plt.ylabel('Order Amount')
plt.show()
customer_segmentation = retail_data.groupby('CustomerID').agg({'OrderAmount':
'sum', 'OrderDate': 'count'}).reset_index()
customer_segmentation.rename(columns={'OrderDate': 'Frequency', 'OrderAmount':
'TotalSpending'}, inplace=True)

# Plot
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Frequency', y='TotalSpending', data=customer_segmentation)
plt.title('Customer Segmentation')
plt.xlabel('Frequency of Purchases')
plt.ylabel('Total Spending')
plt.show()
plt.figure(figsize=(12, 6))
sns.barplot(x='ProductCategory', y='OrderAmount', data=retail_data, estimator=sum)
plt.title('Sales by Product Category')
plt.xlabel('Product Category')
plt.ylabel('Total Sales')
plt.xticks(rotation=45)
plt.show()
```

Output:


Sales Trend Over Time
Order Amount
6000
5000
4000
3000
2000
1000
2022-10-02
2022-10-03
2022-10-04
2022-10-05
2022-10-06
2022-10-07
Order Date


Customer Segmentation
Total Spending
30000
25000
20000
15000
10000
1
2
3
4
5
6
7
8
9
Frequency of Purchases


Sales by Product Category
Total Sales
35000
30000
25000
20000
15000
10000
5000
0
Baby
Sports
Beauty
Gift Cards
Clothing & Accessories
Books
Product Category