

# Assignment # 6

**[Find errors and complete the steps]**

## **[Note]**

Use the PyCharm software if google colab is not working properly

### **1. Stock Market Trend Analysis (Line Plot)**

- **Concepts Used:** Line plots, figure/axes, legends, markers, colors
- **Use Case:** Visualizing stock price trends over time

```
import numpy as np
import matplotlib.pyplot as plt

# Simulated stock prices
days = np.arange(1, 31)
prices = np.cumsum(np.random.randn(30) * 2 + 100) # Randomized stock prices

plt.figure(figsize=(10, 5), dpi=100)
plt.plot(days, prices, linestyle='-', marker='o', color='b', markersize=6,
label="Stock Price")

# Adding labels, title, and legend
plt.xlabel("Day")
plt.ylabel("Price (USD)")
plt.title("Stock Price Trend")
plt.legend()
plt.grid(True)

# Show the plot
plt.show()
```

[Your Task]

1. Install the dependencies
2. Run the above code
3. Show the output

## 2. Customizing Temperature Trends with Line Styles and Markers

- Concepts Used: Line customization, markers, annotations
- Use Case: Analyzing daily temperature variations

```
import matplotlib.pyplot as plt
import numpy as np

days = np.arange(1, 11)
temp_day = np.random.randint(25, 35, size=10)
temp_night = np.random.randint(15, 25, size=10)

plt.figure(figsize=(8, 5))

# Day temperature
plt.plot(days, temp_day, linestyle='--', marker='o', color='r',
markersize=8, label="Day Temperature")

# Night temperature
plt.plot(days, temp_night, linestyle='-', marker='s', color='b',
markersize=8, label="Night Temperature")

# Adding labels, title, and legend
plt.xlabel("Days")
plt.ylabel("Temperature (°C)")
plt.title("Daily Temperature Variation")
plt.legend()
plt.grid()

plt.show()
```

[Your Task]

1. Install the dependencies
2. Compile and run the code
3. Print and show the output

### 3. Population Growth Visualization (Scatter Plot)

- Concepts Used: Scatter plot, color mapping, size variations
- Use Case: Visualizing population growth in different countries

```
import matplotlib.pyplot as plt
import numpy as np

# Sample data
countries = ["USA", "China", "India", "Brazil", "Germany"]
pop_growth = np.array([0.8, 1.2, 1.5, 0.9, 0.5]) # Growth in percentage
population = np.array([331, 1441, 1380, 213, 83]) # Population in millions

# Scatter plot with size variation
plt.figure(figsize=(8, 5))
plt.scatter(pop_growth, population, c=population, cmap='viridis',
            s=population * 2, alpha=0.6)

# Labels
plt.xlabel("Population Growth (%)")
plt.ylabel("Total Population (millions)")
plt.title("Population Growth vs. Total Population")

# Adding annotations
for i, country in enumerate(countries):
    plt.annotate(country, (pop_growth[i], population[i]), fontsize=10,
                 ha='right')

plt.colorbar(label="Population (millions)")
plt.grid()
plt.show()
```

[Your Task]

1. Install the dependencies
2. Compile and run the code
3. Print and show the output

## 4. Customer Segmentation with Scatter Plot (Color Categories)

- Concepts Used: Scatter plot, color categories, transparency
- Use Case: Visualizing customer segments in e-commerce

```
import matplotlib.pyplot as plt
import numpy as np

# Simulated customer data
np.random.seed(42)
age = np.random.randint(18, 60, 100)
spending_score = np.random.randint(20, 100, 100)
categories = np.random.choice([1, 2, 3, 4], 100) # 4 different customer
types

# Scatter plot with category-based colors
plt.figure(figsize=(8, 5))
scatter = plt.scatter(age, spending_score, c=categories, cmap='jet',
alpha=0.7)

plt.xlabel("Age")
plt.ylabel("Spending Score")
plt.title("Customer Segmentation Analysis")
plt.colorbar(label="Customer Category")

plt.grid()
plt.show()
```

[Your Task]

1. Install the dependencies
2. Compile and run the code
3. Print and show the output

## 5. 3D Scatter Plot of Sales Data

- Concepts Used: 3D plotting, scatter plots, colormaps
- Use Case: Visualizing sales performance in different regions

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

# Sample sales data
np.random.seed(42)
region = np.random.randint(1, 6, 50) # 5 regions
sales = np.random.randint(500, 10000, 50)
profit = np.random.randint(50, 1000, 50)

fig = plt.figure(figsize=(10, 6))
ax = fig.add_subplot(111, projection='3d')

# Scatter plot
scatter = ax.scatter(region, sales, profit, c=profit, cmap='coolwarm', s=50,
alpha=0.7)

# Labels
ax.set_xlabel("Region")
ax.set_ylabel("Sales")
ax.set_zlabel("Profit")
ax.set_title("3D Sales Performance Analysis")
fig.colorbar(scatter, label="Profit")

plt.show()
```

[Your Task]

1. Install the dependencies
2. Compile and run the code
3. Print and show the output