#5.1) Create a series of plots to analyze a given dataset.

import matplotlib.pyplot as plt

import numpy as np

# Sample data

x = np.linspace(0, 2 \* np.pi, 100)

y = np.sin(x)

y2 = np.cos(x)

y3 = np.sin(x) \* np.cos(x)

# Create a figure with a series of plots

plt.figure(figsize=(12, 4))

plt.subplot(1, 3, 1)

plt.plot(x, y, color='blue')

plt.title('Sine Wave')

plt.xlabel('X-axis')

plt.ylabel('Y-axis')

plt.subplot(1, 3, 2)

plt.plot(x, y2, color='green')

plt.title('Cosine Wave')

plt.xlabel('X-axis')

plt.ylabel('Y-axis')

plt.subplot(1, 3, 3)

plt.plot(x, y3, color='red')

plt.title('Product of Sine and Cosine')

plt.xlabel('X-axis')

plt.ylabel('Y-axis')

plt.tight\_layout()

# Adjusts subplot params for a tight layout

plt.show()

#5.2) Generate a subplot layout with various plot types (scatter, line, bar).

import matplotlib.pyplot as plt

import numpy as np

import pandas as pd

# Sample data

data = {'Category': ['A', 'B', 'C', 'D'],

'Value1': [10, 25, 15, 30],

'Value2': [15, 20, 25, 10]}

df = pd.DataFrame(data)

x\_scatter = np.random.rand(50)

y\_scatter = np.random.rand(50)

x\_line = np.arange(10)

y\_line = np.random.randint(1, 10, size=10)

fig, axes = plt.subplots(2, 2, figsize=(10, 8))

# Scatter Plot

axes[0, 0].scatter(x\_scatter, y\_scatter)

axes[0, 0].set\_title('Scatter Plot')

axes[0, 0].set\_xlabel('X-axis')

axes[0, 0].set\_ylabel('Y-axis')

# Line Plot

axes[0, 1].plot(x\_line, y\_line, marker='o')

axes[0, 1].set\_title('Line Plot')

axes[0, 1].set\_xlabel('X-axis')

axes[0, 1].set\_ylabel('Y-axis')

# Bar Chart

axes[1, 0].bar(df['Category'], df['Value1'])

axes[1, 0].set\_title('Bar Chart')

axes[1, 0].set\_xlabel('Category')

axes[1, 0].set\_ylabel('Value1')

# This subplot is intentionally left empty to match the assignment's implied 2x2 grid

axes[1, 1].set\_visible(False)

plt.tight\_layout()

plt.show()

5.3) Visualize time-series data and customize axis labels and date formats.

import matplotlib.pyplot as plt

import pandas as pd

import numpy as np

# Sample time-series data

dates = pd.to\_datetime(pd.date\_range(start='2024-01-01', periods=100, freq='D'))

sales\_data = np.random.randint(100, 500, size=100) + np.sin(np.arange(100) \* 0.2) \* 50

df = pd.DataFrame({'Date': dates, 'Sales': sales\_data})

df = df.set\_index('Date')

plt.figure(figsize=(12, 6))

plt.plot(df.index, df['Sales'])

# Customize axis labels and date formats

plt.xlabel('Date')

plt.ylabel('Sales')

plt.title('Daily Sales Over Time')

# Use pandas' built-in plotting for automatic date formatting

# This is a good practice for clean time-series plots

df.plot(y='Sales', figsize=(12, 6), title='Daily Sales Over Time')

plt.show()

#5.4) Create a 3D plot.

import matplotlib.pyplot as plt

import numpy as np

from mpl\_toolkits.mplot3d import Axes3D

# Sample 3D data

np.random.seed(42)

x = np.random.rand(50)

y = np.random.rand(50)

z = np.random.rand(50)

# Create a 3D figure

fig = plt.figure(figsize=(8, 6))

ax = fig.add\_subplot(111, projection='3d')

# Plot the 3D data

ax.scatter(x, y, z)

# Set labels for the axes

ax.set\_xlabel('X-axis')

ax.set\_ylabel('Y-axis')

ax.set\_zlabel('Z-axis')

ax.set\_title('3D Scatter Plot')

plt.show()