Python Data Visualization Worksheet

Name: _____ **Topic:** Matplotlib (Line, Bar, and Histogram)

Questions Write the Python code to plot a line graph showing the temperature over seven days. Include markers for each day, a title, and axis labels for "Days" and "Temperature (°C)".

Questions Create a bar chart that compares the number of students in four different houses (Red, Blue, Green, Yellow). Ensure each bar is assigned its corresponding color using the `color` attribute.

Questions Generate a histogram for a list of exam scores ranging from 0 to 100. Set the number of bins to 10 and explain how the `edgecolor` parameter helps in visualizing the distribution.

Questions How do you plot two different lines on the same axes representing "Sales" and "Profit"? Include a legend, a dashed line style for profit, and a grid for better readability.

Questions Explain the code structure required to display a Bar Chart and a Histogram in a single window using `plt.subplot`. Specify the arguments used for a 1-row, 2- column layout.

Answer Key

1. Line Plot Implementation

python

```
import matplotlib.pyplot as plt
days = [1, 2, 3, 4, 5, 6, 7]
temp = [22, 24, 19, 21, 25, 23, 20]
plt.plot(days, temp, marker='o', color='blue')
plt.title("Weekly Temperature"); plt.xlabel("Days");
plt.ylabel("Temperature (°C)"); plt.show()
```

Use code with caution.

2. Bar Chart with Specific Colors

python

```
import matplotlib.pyplot as plt
houses = ['Red', 'Blue', 'Green', 'Yellow']
students = [45, 52, 48, 50]
colors = ['red', 'blue', 'green', 'yellow']
plt.bar(houses, students, color=colors)
plt.title("Students per House"); plt.xlabel("House Name");
plt.ylabel("Count"); plt.show()
```

Use code with caution.

3. Histogram and Binning Logic

python

```
import matplotlib.pyplot as plt
scores = [55, 67, 88, 92, 45, 76, 81, 95, 62, 71, 85, 90, 40,
58]
plt.hist(scores, bins=10, edgecolor='black', color='skyblue')
# The 'bins=10' parameter divides the score range into 10
equal intervals.
# The 'edgecolor' draws distinct borders around bars to
separate the frequencies visually.
plt.title("Exam Score Distribution"); plt.show()
```

Use code with caution.

4. Multi-Line Comparison with Legend

python

```
import matplotlib.pyplot as plt
months = ['Jan', 'Feb', 'Mar', 'Apr']; sales = [200, 300, 250,
400]; profit = [50, 80, 60, 110]
plt.plot(months, sales, label='Sales', color='green',
linewidth=2)
plt.plot(months, profit, label='Profit', color='red',
linestyle='--')
plt.legend(); plt.grid(True, linestyle=':', alpha=0.6)
plt.title("Sales vs Profit"); plt.show()
```

Use code with caution.

5. Subplot Layout Logic

python

```
import matplotlib.pyplot as plt
```

```
plt.subplot(1, 2, 1) # Creates the first plot in a 1-row, 2-column grid
plt.bar(['A', 'B'], [10, 20], color='orange')
plt.subplot(1, 2, 2) # Switches focus to the second plot in the same grid
plt.hist([1, 2, 1, 3, 2, 1], bins=3, color='purple')
plt.tight_layout(); plt.show() # tight_layout prevents the labels from overlapping
```

Use code with caution.