Lab Assignment 23

Student Name: Chauhan Vandana Ramdayal

Student Id: AF0411629

Topic: Matplotlib more plots and formatting

In Matplotlib, we can draw multiple graphs in a single plot in two ways. One is by using subplot() function and other by superimposition of second graph on the first i.e, all graphs will appear on the same plot. We will look into both the ways one by one.
Multiple Plots using subplot () Function
A subplot () function is a wrapper function which allows the programmer to plot more than one graph in a single figure by just calling it once.

Syntax: *matplotlib.pyplot.subplots(nrows=1, ncols=1, sharex=False, sharey=False, squeeze=True, subplot_kw=None, gridspec_kw=None, **fig_kw)*

Parameters:

1. ***nrows, ncols:*** *These gives the number of rows and columns respectively. Also, it must be noted that both these parameters are optional and the default value is 1.*
2. ***sharex, sharey:*** *These parameters specify about the properties that are shared among a and y axis.Possible values for them can be, row, col, none or default value which is False.*
3. ***squeeze:*** *This parameter is a boolean value specified, which asks the programmer whether to squeeze out, meaning remove the extra dimension from the array. It has a default value False.*
4. ***subplot_kw:*** *This parameters allow us to add keywords to each subplot and its default value is None.*
5. ***gridspec_kw:*** *This allows us to add grids on each subplot and has a default value of None.*
6. *****fig_kw:*** *This allows us to pass any other additional keyword argument to the function call and has a default value of None.*

Questions-

1. Pie Chart

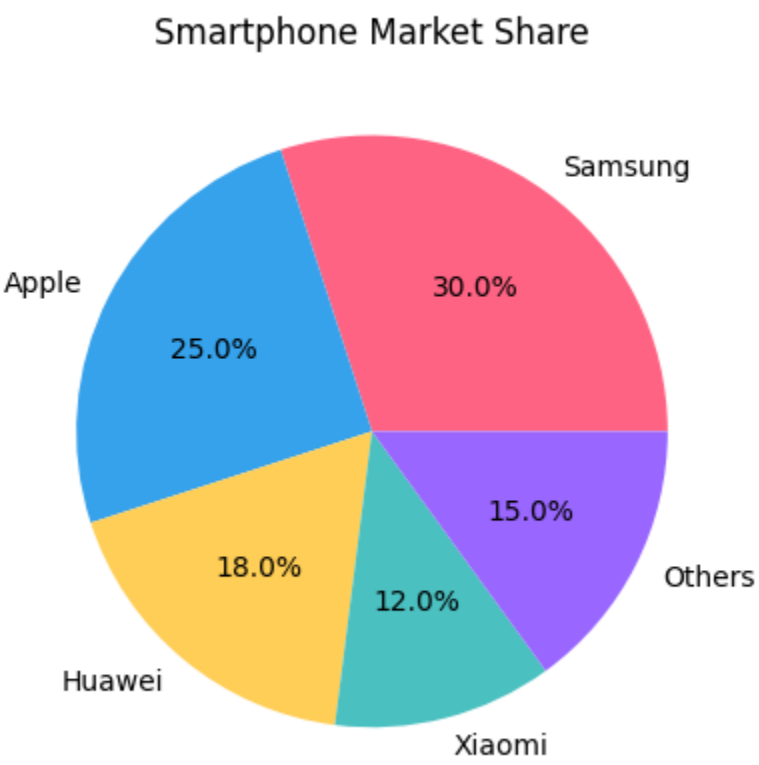
Code:

lab23.py > ...

```
import matplotlib.pyplot as plt
manufacturers = ['Samsung', 'Apple', 'Huawei', 'Xiaomi', 'Others'] # Market share data
market_share = [30, 25, 18, 12, 15]
colors = ['#FF6384', '#36A2EB', '#FFCE56', '#4BC0C0', '#9966FF']
plt.pie(market_share, labels=manufacturers, colors=colors, autopct='%1.1f%%')  # Create a pie chart
# Title
plt.title('Smartphone Market Share')
plt.show()

```

Output:



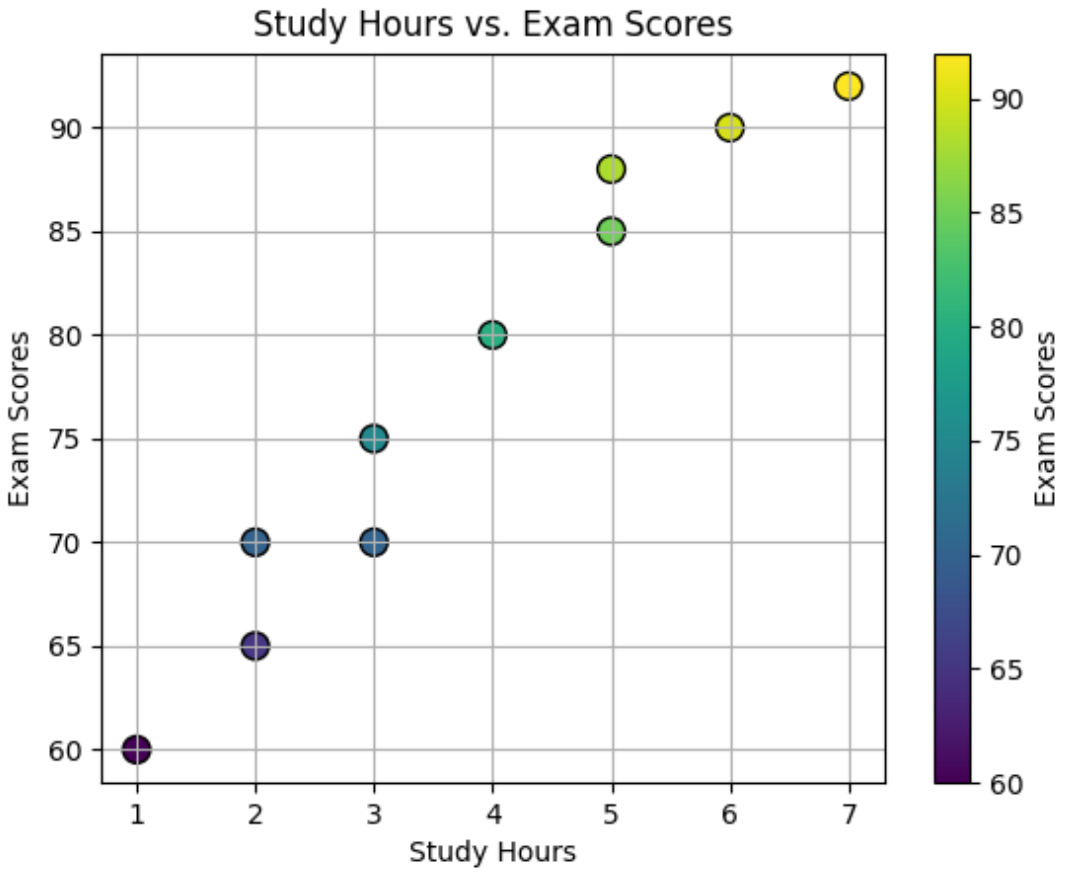
2. Scatter plot Chart

Code:

```
import matplotlib.pyplot as plt
import numpy as np
# Student data (study hours and exam scores)
study_hours = [2, 3, 1, 4, 3, 5, 2, 6, 5, 7]
exam_scores = [65, 75, 60, 80, 70, 85, 70, 90, 88, 92]
# Define a color map to vary colors based on exam scores
colors = np.array(exam_scores)
# Create a scatter plot with varying colors and larger marker size
plt.scatter(study_hours, exam_scores, c=colors, cmap='viridis', s=100, edgecolor='black')
plt.colorbar(label='Exam Scores') # Colorbar for reference

# Labeling and Title
plt.xlabel('Study Hours')
plt.ylabel('Exam Scores')
plt.title('Study Hours vs. Exam Scores')
plt.grid(True)  # Add a grid
plt.show()
```

Output:

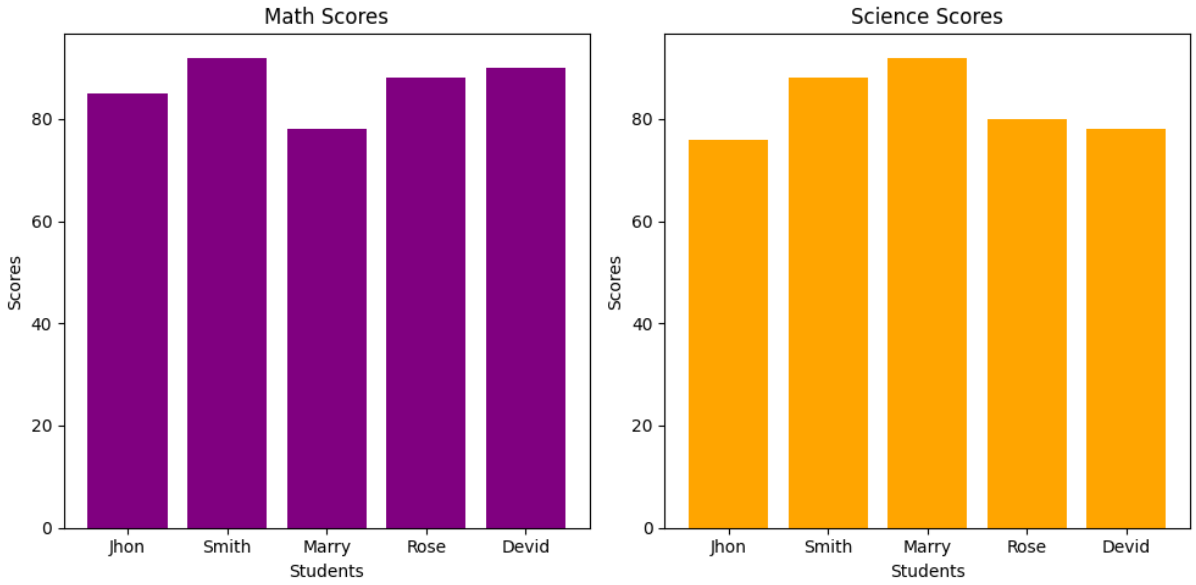


3. Sub-plot chart using axis

Code:

```
import matplotlib.pyplot as plt
# Sample student data
students = ['Jhon', 'Smith', 'Marry', 'Rose', 'Devid']
math_scores = [85, 92, 78, 88, 90]
science_scores = [76, 88, 92, 80, 78]
# Create subplots (1 row, 2 columns)
fig, axs = plt.subplots(1, 2, figsize=(10, 5))
# Plot Math Scores with a new color
axs[0].bar(students, math_scores, color='purple')
axs[0].set_title('Math Scores')
axs[0].set_xlabel('Students')
axs[0].set_ylabel('Scores')
# Plot Science Scores with a new color
axs[1].bar(students, science_scores, color='orange')
axs[1].set_title('Science Scores')
axs[1].set_xlabel('Students')
axs[1].set_ylabel('Scores')
plt.tight_layout() # Adjust layout
plt.show()
```

Output:


Math Scores
Scores
0
20
40
60
80
Jhon
Smith
Marry
Rose
Devid
Students
Science Scores
Scores
0
20
40
60
80
Jhon
Smith
Marry
Rose
Devid
Students