Lab – 23
Student Name: Navneet P
Student ID: AF0411619

Topic-Some more matplotlib graphs

Graphs used:

1. **Pie Plot**: A pie plot, also known as a pie chart, is a circular data visualization that is commonly used to represent the distribution of a whole into its constituent parts or categories. It is particularly useful for displaying the proportional composition of a dataset, where each part or category is represented by a "slice" of the pie, and the entire pie represents 100% of the whole.

2. **Scatter plot:** A scatter plot is a data visualization technique that is used to display individual data points on a two-dimensional graph. In a scatter plot, each data point represents one observation or data entry and is plotted as a point on the graph. The position of each point is determined by its values on two variables, typically referred to as the x-axis and y-axis.

3. **Subplots:** a subplot, short for "sub-plot," refers to a feature or capability that allows you to create multiple smaller plots within a single, larger plot or figure. Subplots are commonly used when you want to display multiple graphs or charts side by side, making it easier to compare and analyse different aspects of your data within a single visual representation.

Q1. **Market Share Pie Chart In this example.**

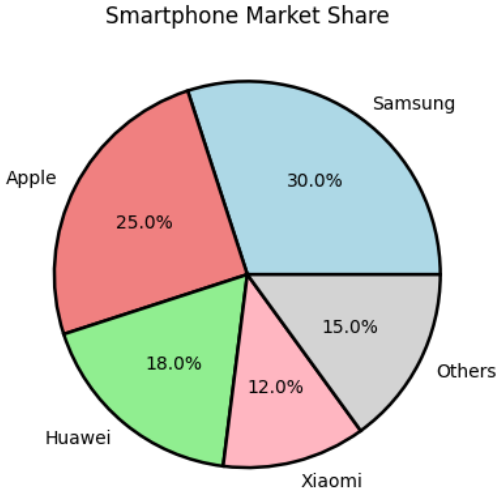
Create a pie chart to illustrate the market share of different smartphone manufacturers in a competitive market. The pie chart will show the proportional distribution of market share among key players.

Solution:

Question 1

```
import matplotlib.pyplot as plt

# Market share data for different smartphone manufacturers
manufacturers = ['Samsung', 'Apple', 'Huawei', 'Xiaomi', 'Others']
market_share = [30, 25, 18, 12, 15]  # Corresponding market share percentages

# Colors for each slice of the pie chart
colors = ['lightblue', 'lightcoral', 'lightgreen', 'lightpink', 'lightgrey']

# Create a pie chart with market share data
plt.pie(market_share,
        labels=manufacturers,  # Labels for each pie slice (manufacturer names)
        colors=colors,  # Custom colors for each slice
        autopct='%1.1f%%',  # Display percentages on each slice, formatted to 1 decimal
place
        wedgeprops = {"edgecolor" : "black",  # Black border around each slice
                      'linewidth': 1.8,  # Thickness of the border
                      'antialiased': True})  # Smooth edges for better appearance

# Add a title to the pie chart
plt.title('Smartphone Market Share')

# Display the pie chart
plt.show()
```

Output:



Q2. **Exam Scores vs. Study Hours**

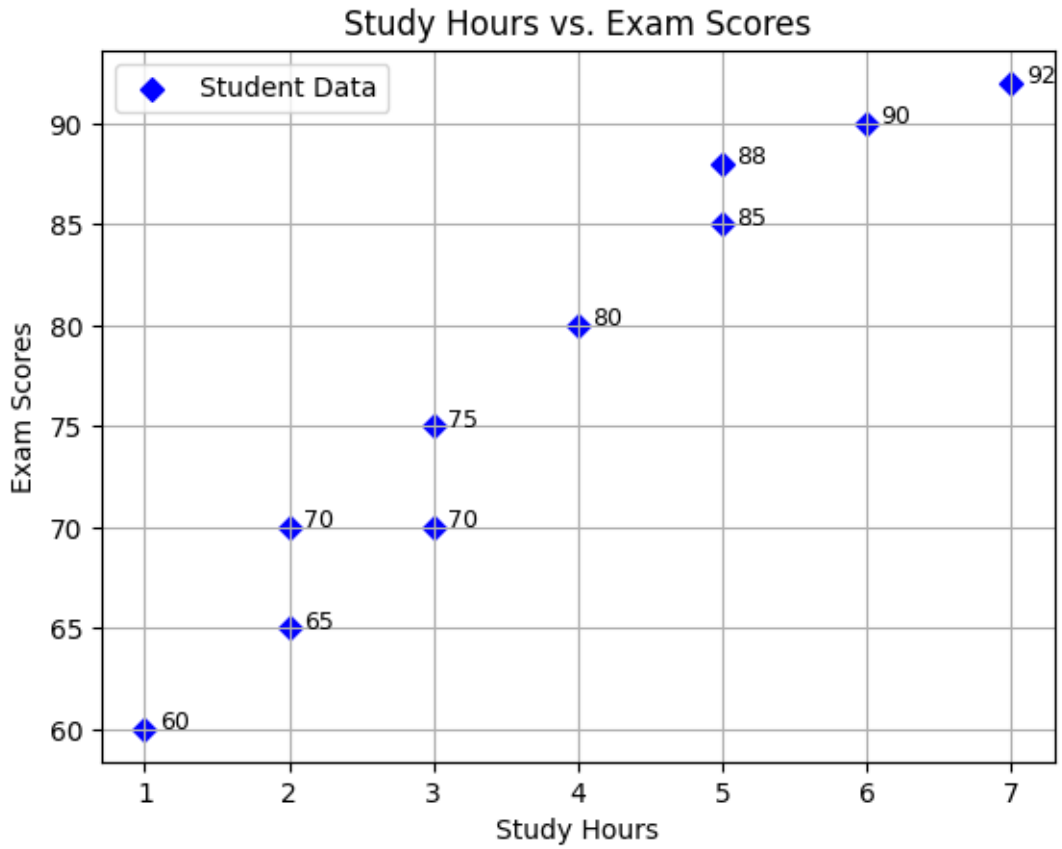
Create a scatter plot to visualize the relationship between the number of study hours and the corresponding exam scores of a group of students.

Solution:

Question 2

```
import matplotlib.pyplot as plt

# Student data representing the number of study hours and corresponding exam scores
study_hours = [2, 3, 1, 4, 3, 5, 2, 6, 5, 7]  # Number of hours studied by students
exam_scores = [65, 75, 60, 80, 70, 85, 70, 90, 88, 92]  # Corresponding exam scores for the
study hours

# Create a scatter plot using study_hours as x-axis and exam_scores as y-axis
scat = plt.scatter(study_hours, exam_scores,
                   c='blue',  # Color of the scatter plot points
                   marker='D',  # Marker shape set to 'D' (diamond)
                   label='Student Data')  # Label for the scatter plot

# Adding text labels to each data point for clarity
for i, sc in enumerate(exam_scores):
    plt.text(study_hours[i] + 0.1, exam_scores[i],  # Slightly offset the text from the
point
             str(sc),  # Display the exam score as the text
             fontsize=9)  # Font size of the text labels

# Label the x-axis and y-axis
plt.xlabel('Study Hours')
plt.ylabel('Exam Scores')

# Add a title to the scatter plot
plt.title('Study Hours vs. Exam Scores')

# Display the legend to label the data points
plt.legend()

# Add a grid for better readability of the plot
plt.grid(True)

# Display the scatter plot
plt.show()
```

Output:


Study Hours vs. Exam Scores
Student Data
Exam Scores
Study Hours
90
85
80
75
70
65
60
1
2
3
4
5
6
7
92
90
88
85
80
75
70
70
65
60

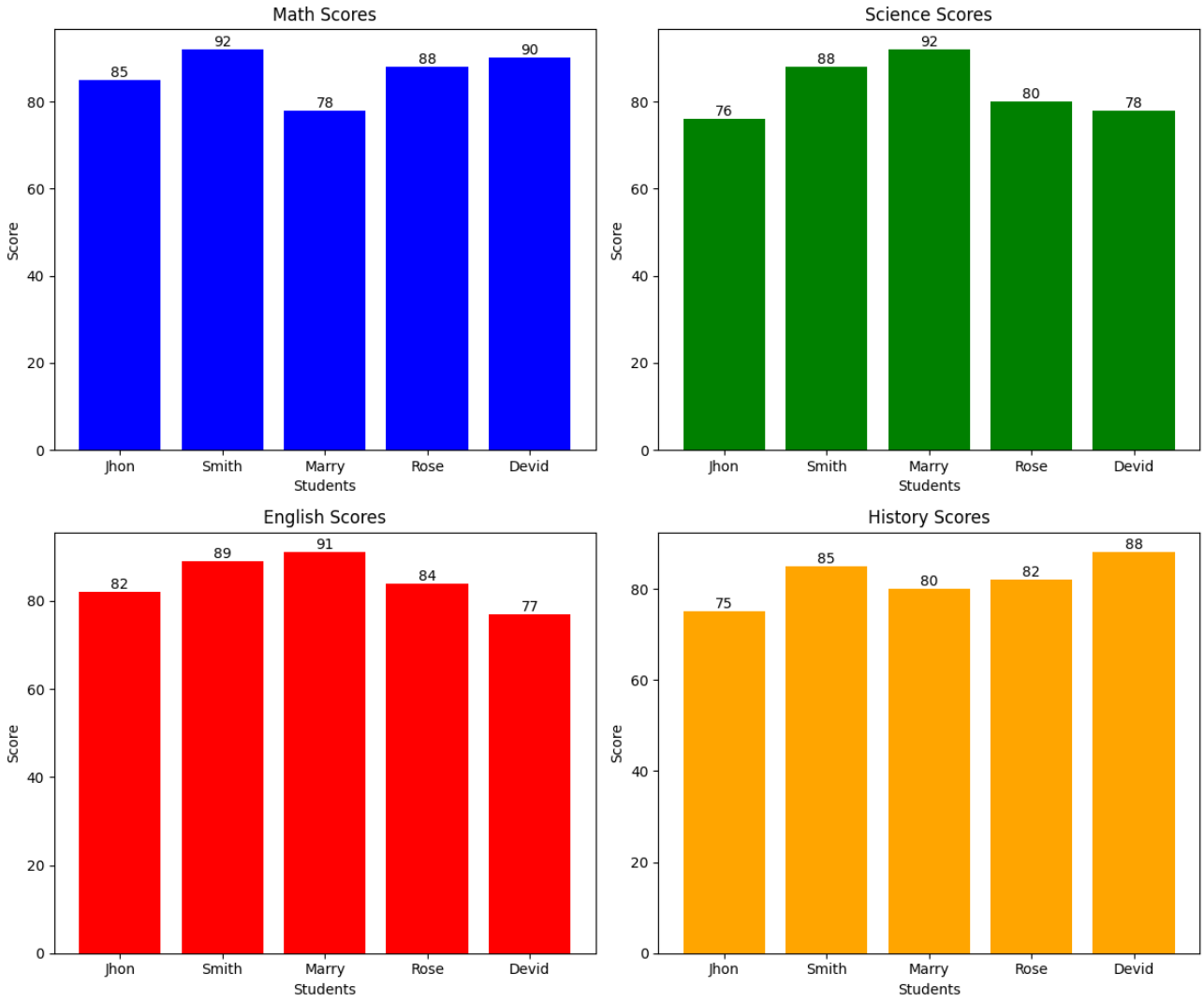
Q3. Using subplot create multiple plots in a single figure.

Solution:

Question 3

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

# Sample data for students and their scores in different subjects
students = ['Jhon', 'Smith', 'Marry', 'Rose', 'Devid']  # List of student names
math_scores = [85, 92, 78, 88, 90]  # Scores in Math
science_scores = [76, 88, 92, 80, 78]  # Scores in Science
english_scores = [82, 89, 91, 84, 77]  # Scores in English
history_scores = [75, 85, 80, 82, 88]  # Scores in History

# Create a figure with 4 subplots arranged in 2 rows and 2 columns, figsize specifies the
dimensions of the plot
fig, axs = plt.subplots(2, 2, figsize=(12, 10))

# Plot Math Scores on the top-left subplot
bar1 = axs[0, 0].bar(students, math_scores, color='b')  # Create a bar chart for Math
scores, with blue bars
axs[0, 0].set_title('Math Scores')  # Set the title for the Math subplot
axs[0, 0].set_xlabel('Students')  # Label x-axis
axs[0, 0].set_ylabel('Score')  # Label y-axis
axs[0,0].bar_label(bar1, labels=math_scores)  # Add the actual scores as labels on the bars


# Plot Science Scores on the top-right subplot
bar2 = axs[0, 1].bar(students, science_scores, color='g')  # Bar chart for Science scores,
with green bars
axs[0, 1].set_title('Science Scores')  # Set the title for the Science subplot
axs[0, 1].set_xlabel('Students')  # Label x-axis
axs[0, 1].set_ylabel('Score')  # Label y-axis
axs[0,1].bar_label(bar2, labels=science_scores)  # Add score labels on the bars

# Plot English Scores on the bottom-left subplot
bar3 = axs[1, 0].bar(students, english_scores, color='r')  # Bar chart for English scores,
with red bars
axs[1, 0].set_title('English Scores')  # Set the title for the English subplot
axs[1, 0].set_xlabel('Students')  # Label x-axis
axs[1, 0].set_ylabel('Score')  # Label y-axis
axs[1,0].bar_label(bar3, labels=english_scores)  # Add score labels on the bars

# Plot History Scores on the bottom-right subplot
bar4 = axs[1, 1].bar(students, history_scores, color='orange')  # Bar chart for History
scores, with orange bars
axs[1, 1].set_title('History Scores')  # Set the title for the History subplot
axs[1, 1].set_xlabel('Students')  # Label x-axis
axs[1, 1].set_ylabel('Score')  # Label y-axis
axs[1,1].bar_label(bar4, labels=history_scores)  # Add score labels on the bars

# Automatically adjust the spacing between subplots for better layout
plt.tight_layout()

# Display the entire figure with all subplots
plt.show()
```

Output:


Math Scores
Score
Students
Jhon
Smith
Marry
Rose
Devid
85
92
78
88
90
Science Scores
76
88
92
80
78
English Scores
82
89
91
84
77
History Scores
75
85
80
82
88
0
20
40
60
80

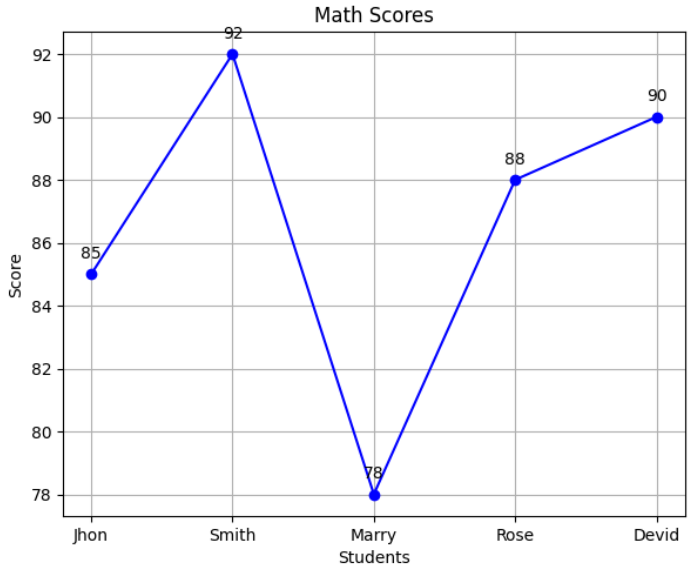
Another subplot example with four line graphs in the same figure.

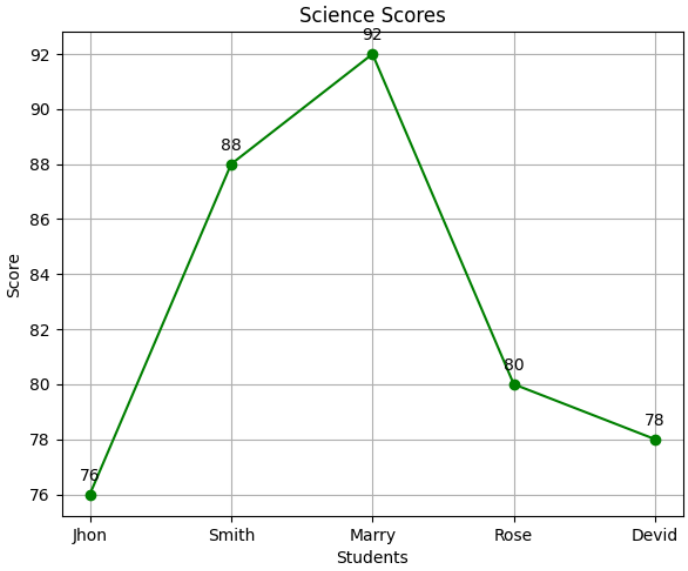
Solution:

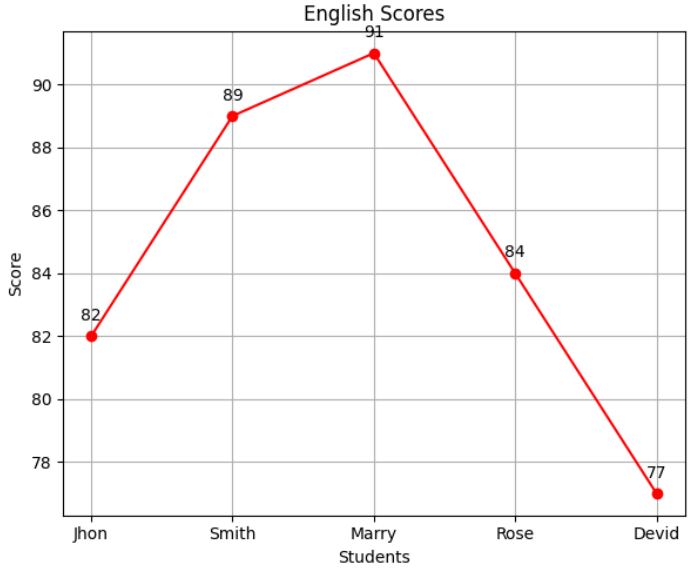
Example

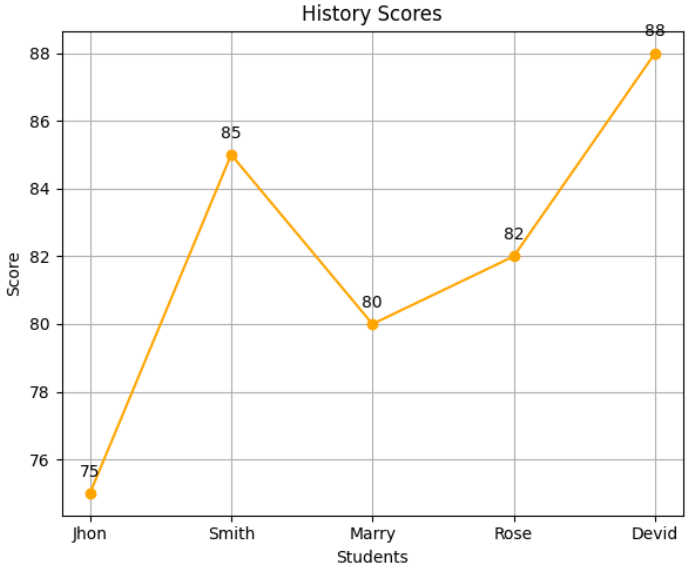
```
import numpy as np
import matplotlib.pyplot as plt

# Sample student data: names of students and their corresponding scores in different
subjects
students = ['Jhon', 'Smith', 'Marry', 'Rose', 'Devid']  # List of student names
math_scores = [85, 92, 78, 88, 90]  # Math scores for the students
science_scores = [76, 88, 92, 80, 78]  # Science scores for the students
english_scores = [82, 89, 91, 84, 77]  # English scores for the students
history_scores = [75, 85, 80, 82, 88]  # History scores for the students

# Create a figure with 4 subplots (2 rows, 2 columns), each subplot for a different subject

fig, axs = plt.subplots(2, 2, figsize=(12, 10))  # 'figsize' controls the overall size of
the figure

# Plot Math Scores in the top-left subplot
axs[0, 0].plot(students, math_scores, color='b', marker='o', linestyle='-')  # Line plot
for Math scores
axs[0, 0].set_title('Math Scores')  # Set the title for this subplot
axs[0, 0].set_xlabel('Students')  # Label x-axis as "Students"
axs[0, 0].set_ylabel('Score')  # Label y-axis as "Score"
# Add text labels for each data point (showing the score values above the corresponding
points)
for i, score in enumerate(math_scores):
    axs[0, 0].text(students[i], math_scores[i] + 0.5, str(score), ha='center')  # Text
label above each point

# Plot Science Scores in the top-right subplot
axs[0, 1].plot(students, science_scores, color='g', marker='o', linestyle='-')  # Line plot
for Science scores
axs[0, 1].set_title('Science Scores')  # Set the title for this subplot
axs[0, 1].set_xlabel('Students')  # Label x-axis
axs[0, 1].set_ylabel('Score')  # Label y-axis
# Add text labels for each data point (showing the score values)
for i, score in enumerate(science_scores):
    axs[0, 1].text(students[i], science_scores[i] + 0.5, str(score), ha='center')

# Plot English Scores in the bottom-left subplot
axs[1, 0].plot(students, english_scores, color='r', marker='o', linestyle='-')  # Line plot
for English scores
axs[1, 0].set_title('English Scores')  # Set the title for this subplot
axs[1, 0].set_xlabel('Students')  # Label x-axis
axs[1, 0].set_ylabel('Score')  # Label y-axis
# Add text labels for each data point (showing the score values)
for i, score in enumerate(english_scores):
    axs[1, 0].text(students[i], english_scores[i] + 0.5, str(score), ha='center')

# Plot History Scores in the bottom-right subplot
axs[1, 1].plot(students, history_scores, color='orange', marker='o', linestyle='-')  # Line
plot for History scores
axs[1, 1].set_title('History Scores')  # Set the title for this subplot
axs[1, 1].set_xlabel('Students')  # Label x-axis
axs[1, 1].set_ylabel('Score')  # Label y-axis
# Add text labels for each data point (showing the score values)
for i, score in enumerate(history_scores):
    axs[1, 1].text(students[i], history_scores[i] + 0.5, str(score), ha='center')

# Automatically adjust the spacing between subplots to prevent overlap
plt.tight_layout()

# Display the figure with all four subplots
plt.show()
```

Output:


Math Scores
Score
Students
Jhon
Smith
Marry
Rose
Devid
85
92
78
88
90


Science Scores
Score
Students
Jhon
Smith
Marry
Rose
Devid
76
88
92
80
78


English Scores
Score
Students
Jhon
Smith
Marry
Rose
Devid
82
89
91
84
77


History Scores
Score
Students
Jhon
Smith
Marry
Rose
Devid
75
85
80
82
88