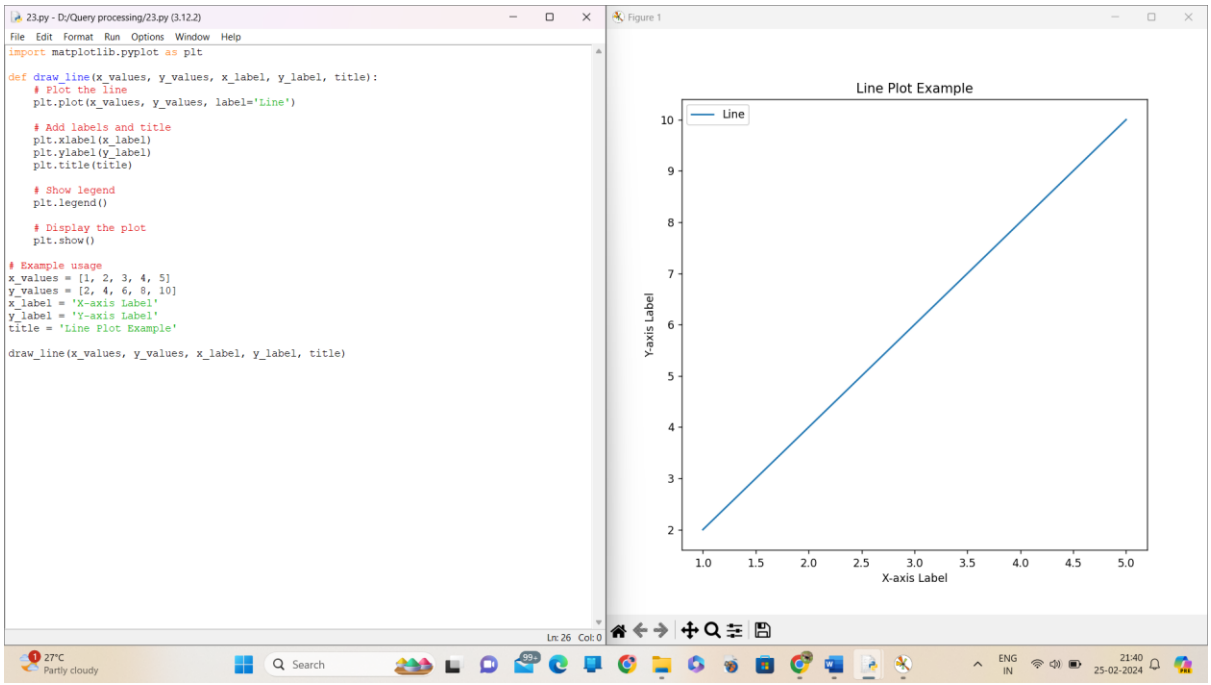
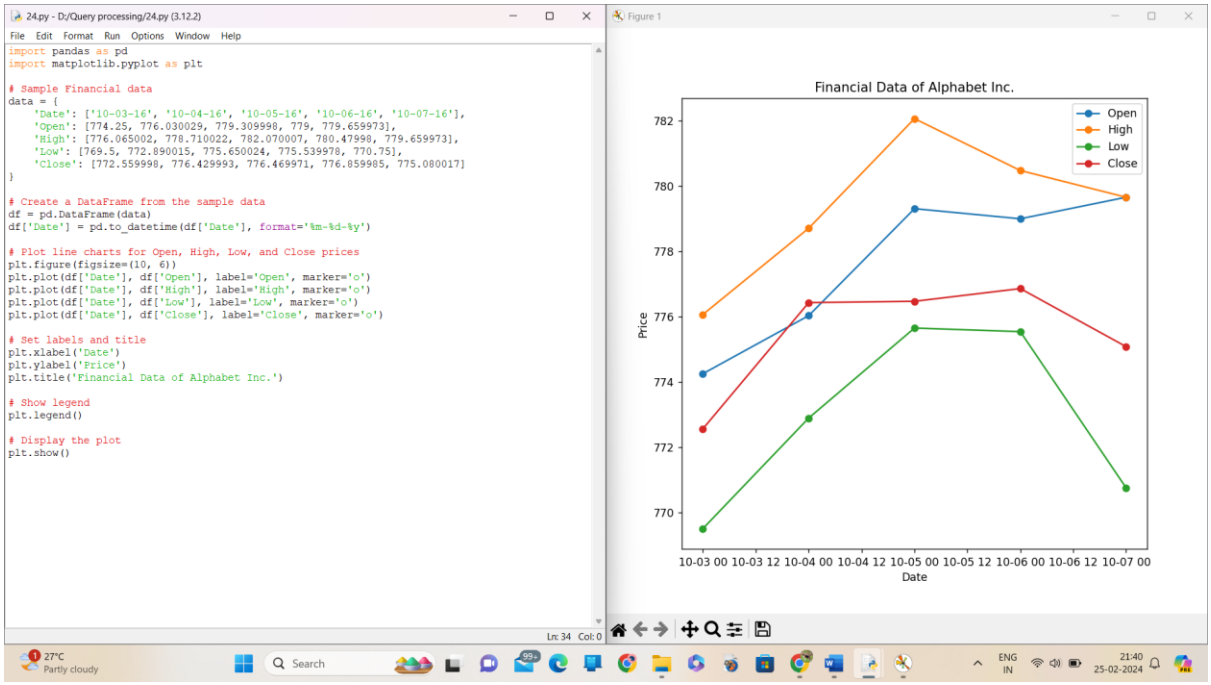
QUERY PROCESSING LAB

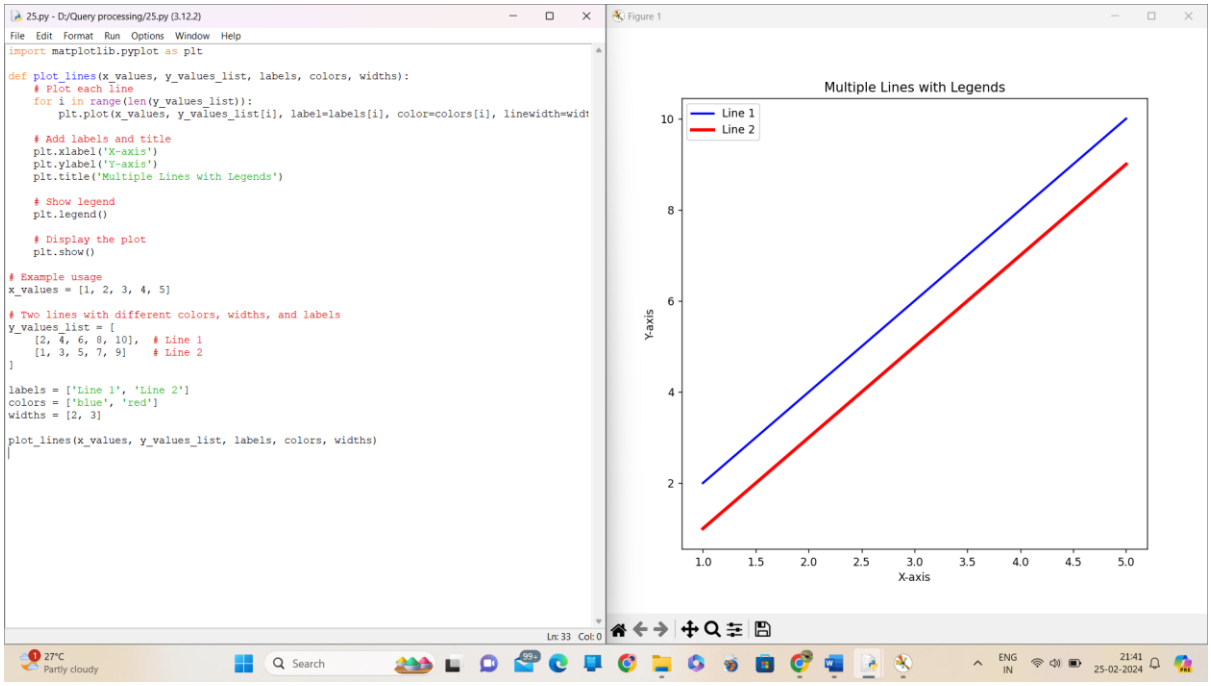
23.



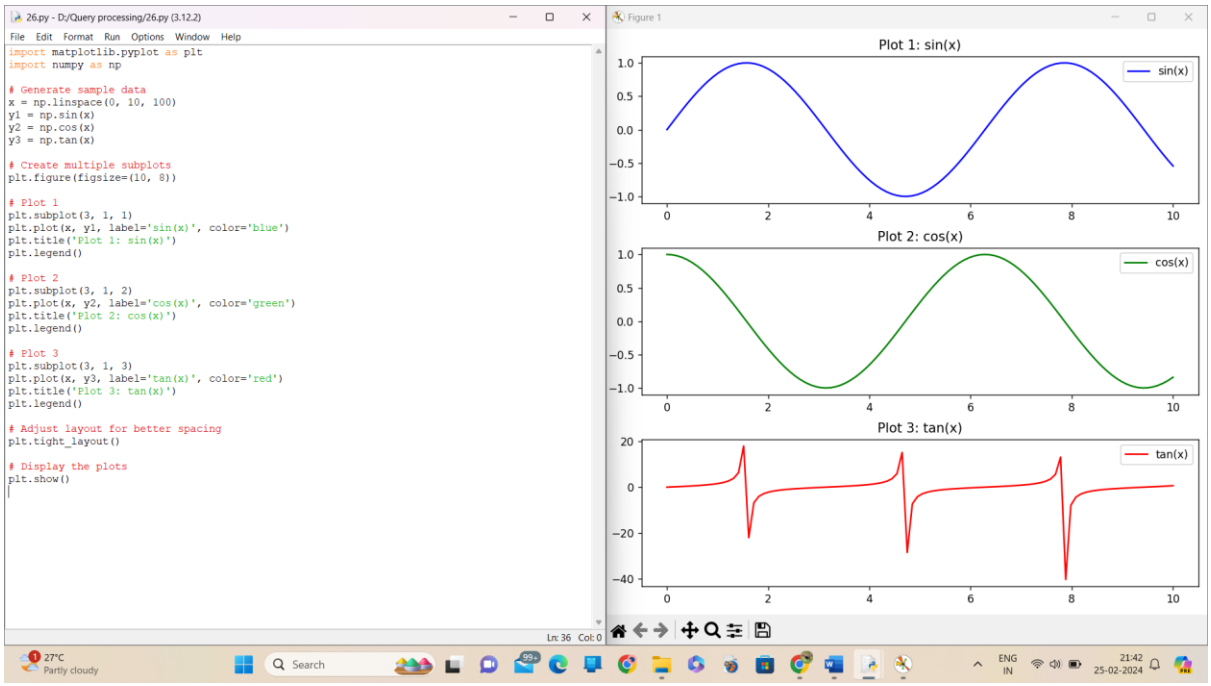
24



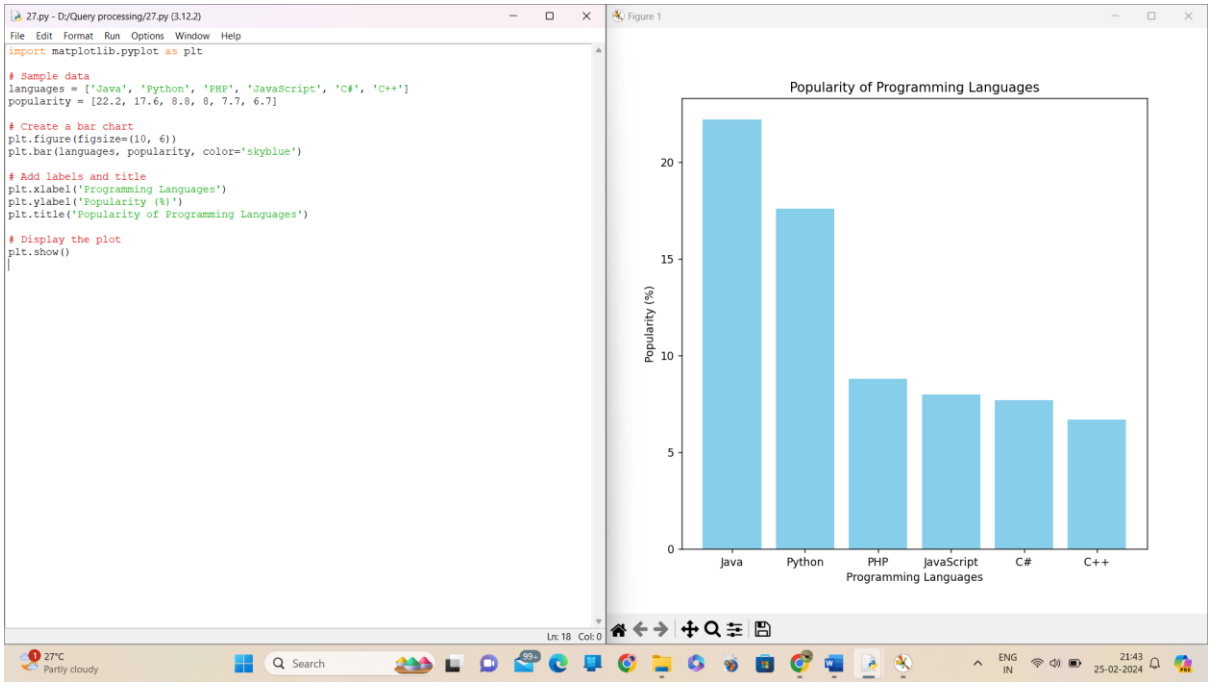
25.


25.py - D:/Query processing/25.py (3.12.2)
File Edit Format Run Options Window Help
import matplotlib.pyplot as plt
def plot_lines(x_values, y_values_list, labels, colors, widths):
# Plot each line
for i in range(len(y_values_list)):
plt.plot(x_values, y_values_list[i], label=labels[i], color=colors[i], linewidth=widt
# Add labels and title
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.title('Multiple Lines with Legends')
# Show legend
plt.legend()
# Display the plot
plt.show()
# Example usage
x_values = [1, 2, 3, 4, 5]
# Two lines with different colors, widths, and labels
y_values_list = [
[2, 4, 6, 8, 10], # Line 1
[1, 3, 5, 7, 9] # Line 2
]
labels = ['Line 1', 'Line 2']
colors = ['blue', 'red']
widths = [2, 3]
plot_lines(x_values, y_values_list, labels, colors, widths)
Ln: 33 Col: 0
Figure 1
Multiple Lines with Legends
Line 1
Line 2
Y-axis
X-axis
10
8
6
4
2
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
27°C
Partly cloudy
Search
ENG
IN
21:41
25-02-2024

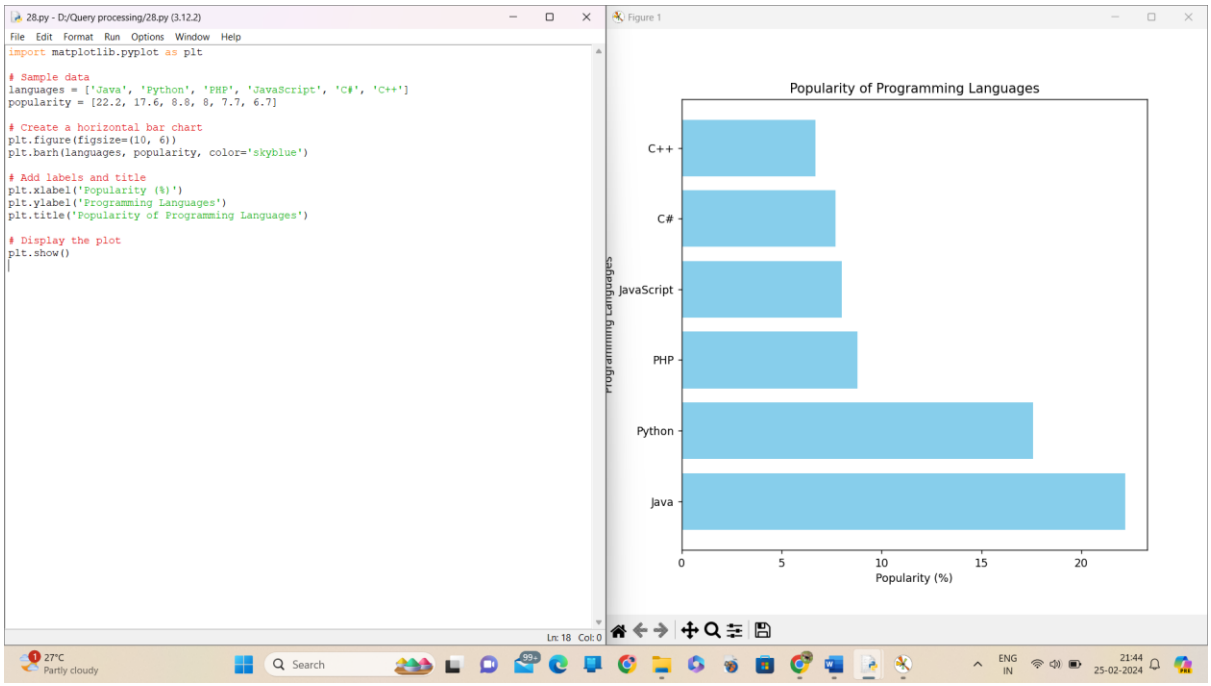
26.


26.py - D:/Query processing/26.py (3.12.2)
File Edit Format Run Options Window Help
import matplotlib.pyplot as plt
import numpy as np
# Generate sample data
x = np.linspace(0, 10, 100)
y1 = np.sin(x)
y2 = np.cos(x)
y3 = np.tan(x)
# Create multiple subplots
plt.figure(figsize=(10, 8))
# Plot 1
plt.subplot(3, 1, 1)
plt.plot(x, y1, label='sin(x)', color='blue')
plt.title('Plot 1: sin(x)')
plt.legend()
# Plot 2
plt.subplot(3, 1, 2)
plt.plot(x, y2, label='cos(x)', color='green')
plt.title('Plot 2: cos(x)')
plt.legend()
# Plot 3
plt.subplot(3, 1, 3)
plt.plot(x, y3, label='tan(x)', color='red')
plt.title('Plot 3: tan(x)')
plt.legend()
# Adjust layout for better spacing
plt.tight_layout()
# Display the plots
plt.show()
Ln: 36 Col: 0
Figure 1
Plot 1: sin(x)
sin(x)
Plot 2: cos(x)
cos(x)
Plot 3: tan(x)
tan(x)
27°C
Partly cloudy
Search
ENG
IN
21:42
25-02-2024

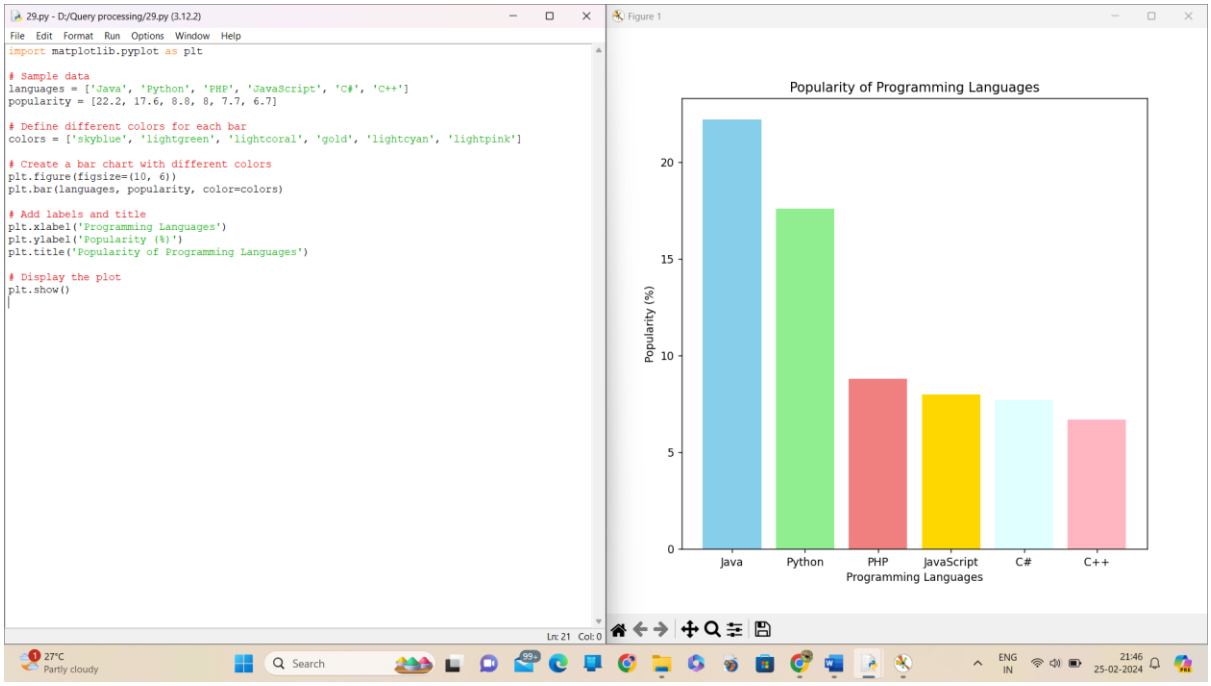
27.


27.py - D:/Query processing/27.py (3.12.2)
File Edit Format Run Options Window Help
import matplotlib.pyplot as plt
# Sample data
languages = ['Java', 'Python', 'PHP', 'JavaScript', 'C#', 'C++']
popularity = [22.2, 17.6, 8.8, 8, 7.7, 6.7]
# Create a bar chart
plt.figure(figsize=(10, 6))
plt.bar(languages, popularity, color='skyblue')
# Add labels and title
plt.xlabel('Programming Languages')
plt.ylabel('Popularity (%)')
plt.title('Popularity of Programming Languages')
# Display the plot
plt.show()
Ln: 18 Col: 0
Figure 1
Popularity of Programming Languages
Popularity (%)
Java
Python
PHP
JavaScript
C#
C++
Programming Languages

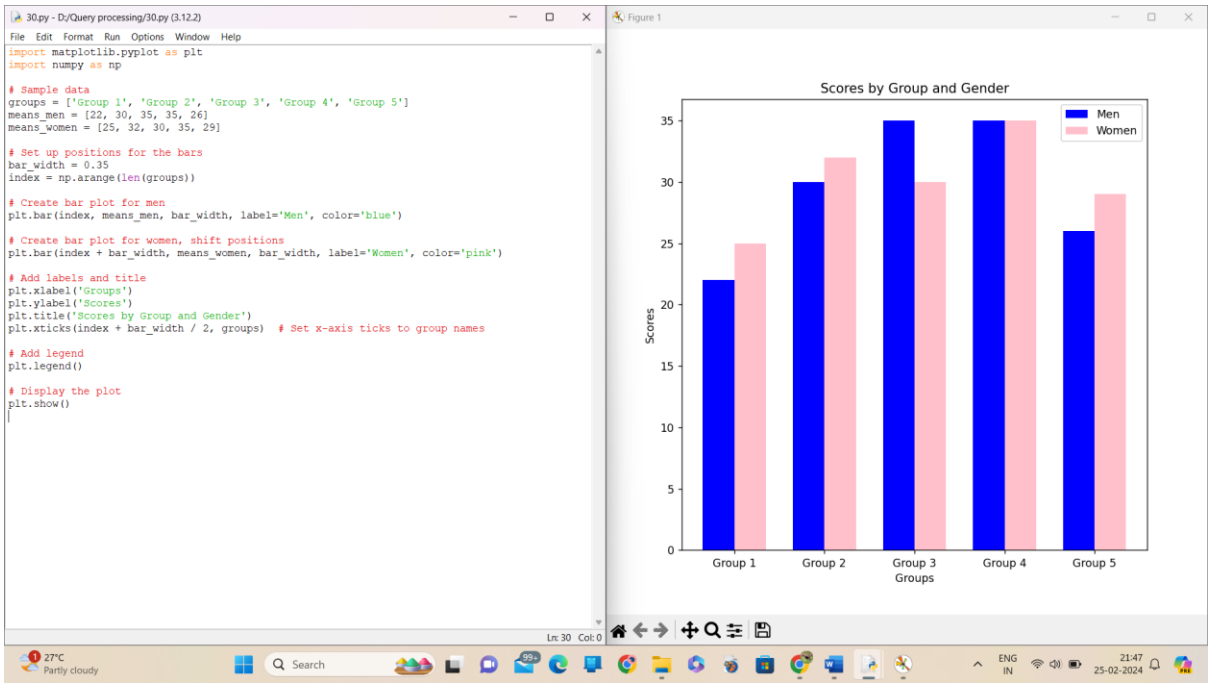
28.


28.py - D:/Query processing/28.py (3.12.2)
File Edit Format Run Options Window Help
import matplotlib.pyplot as plt
# Sample data
languages = ['Java', 'Python', 'PHP', 'JavaScript', 'C#', 'C++']
popularity = [22.2, 17.6, 8.8, 8, 7.7, 6.7]
# Create a horizontal bar chart
plt.figure(figsize=(10, 6))
plt.barh(languages, popularity, color='skyblue')
# Add labels and title
plt.xlabel('Popularity (%)')
plt.ylabel('Programming Languages')
plt.title('Popularity of Programming Languages')
# Display the plot
plt.show()
Ln: 18 Col: 0
Figure 1
Popularity of Programming Languages
C++
C#
JavaScript
PHP
Python
Java
Programming Languages
Popularity (%)

29.

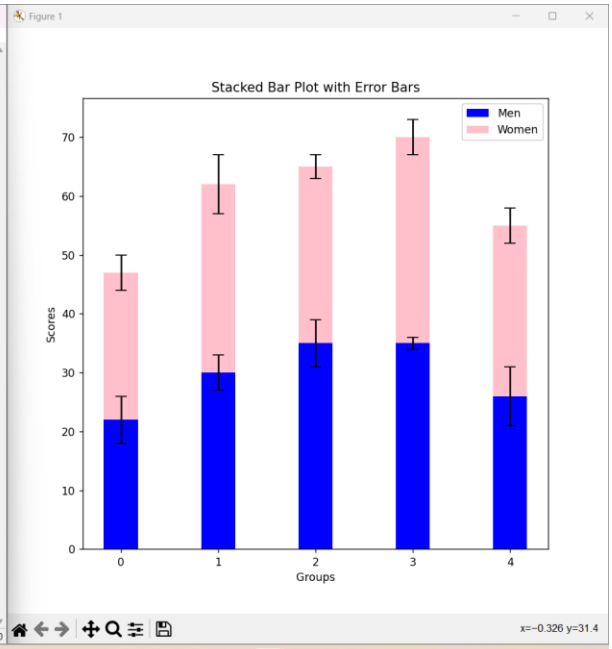

29.py - D:/Query processing/29.py (3.12.2)
File Edit Format Run Options Window Help
import matplotlib.pyplot as plt
# Sample data
languages = ['Java', 'Python', 'PHP', 'JavaScript', 'C#', 'C++']
popularity = [22.2, 17.6, 8.8, 8, 7.7, 6.7]
# Define different colors for each bar
colors = ['skyblue', 'lightgreen', 'lightcoral', 'gold', 'lightcyan', 'lightpink']
# Create a bar chart with different colors
plt.figure(figsize=(10, 6))
plt.bar(languages, popularity, color=colors)
# Add labels and title
plt.xlabel('Programming Languages')
plt.ylabel('Popularity (%)')
plt.title('Popularity of Programming Languages')
# Display the plot
plt.show()
Ln: 21 Col: 0
Figure 1
Popularity of Programming Languages
Popularity (%)
0
5
10
15
20
Java
Python
PHP
JavaScript
C#
C++
Programming Languages
27°C
Partly cloudy
Search
ENG
IN
21:46
25-02-2024

30.


30.py - D:/Query processing/30.py (3.12.2)
File Edit Format Run Options Window Help
import matplotlib.pyplot as plt
import numpy as np
# Sample data
groups = ['Group 1', 'Group 2', 'Group 3', 'Group 4', 'Group 5']
means_men = [22, 30, 35, 35, 26]
means_women = [25, 32, 30, 35, 29]
# Set up positions for the bars
bar_width = 0.35
index = np.arange(len(groups))
# Create bar plot for men
plt.bar(index, means_men, bar_width, label='Men', color='blue')
# Create bar plot for women, shift positions
plt.bar(index + bar_width, means_women, bar_width, label='Women', color='pink')
# Add labels and title
plt.xlabel('Groups')
plt.ylabel('Scores')
plt.title('Scores by Group and Gender')
plt.xticks(index + bar_width / 2, groups) # Set x-axis ticks to group names
# Add legend
plt.legend()
# Display the plot
plt.show()
Ln: 30 Col: 0
Figure 1
Scores by Group and Gender
Men
Women
Scores
0
5
10
15
20
25
30
35
Group 1
Group 2
Group 3
Group 4
Group 5
Groups
27°C
Partly cloudy
Search
ENG
IN
21:47
25-02-2024

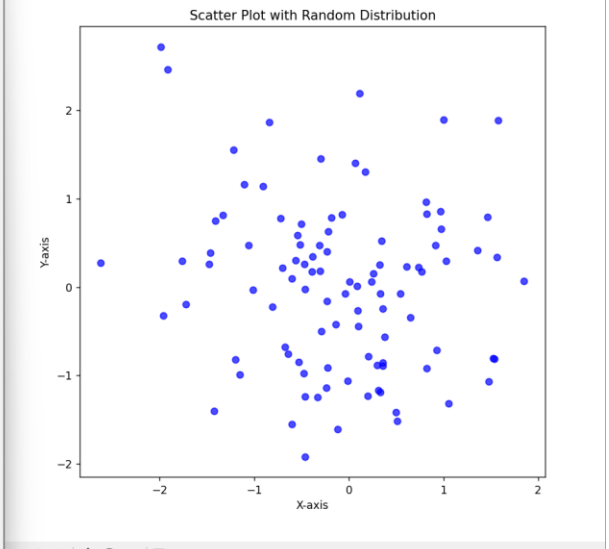
31.

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

# Sample data
groups = ['Group 1', 'Group 2', 'Group 3', 'Group 4', 'Group 5']
means_men = [22, 30, 35, 35, 26]
means_women = [25, 32, 30, 35, 29]
std_dev_men = [4, 3, 4, 1, 5]
std_dev_women = [3, 5, 2, 3, 3]

# Set up positions for the bars
bar_width = 0.35
index = np.arange(len(groups))

# Create stacked bar plot with error bars
plt.bar(index, means_men, bar_width, yerr=std_dev_men, label='Men', color='blue', capsize=5)
plt.bar(index, means_women, bar_width, yerr=std_dev_women, label='Women', color='pink',
        bottom=means_men, capsize=5)

# Add labels and title
plt.xlabel('Groups')
plt.ylabel('Scores')
plt.title('Stacked Bar Plot with Error Bars')

# Add legend
plt.legend()

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

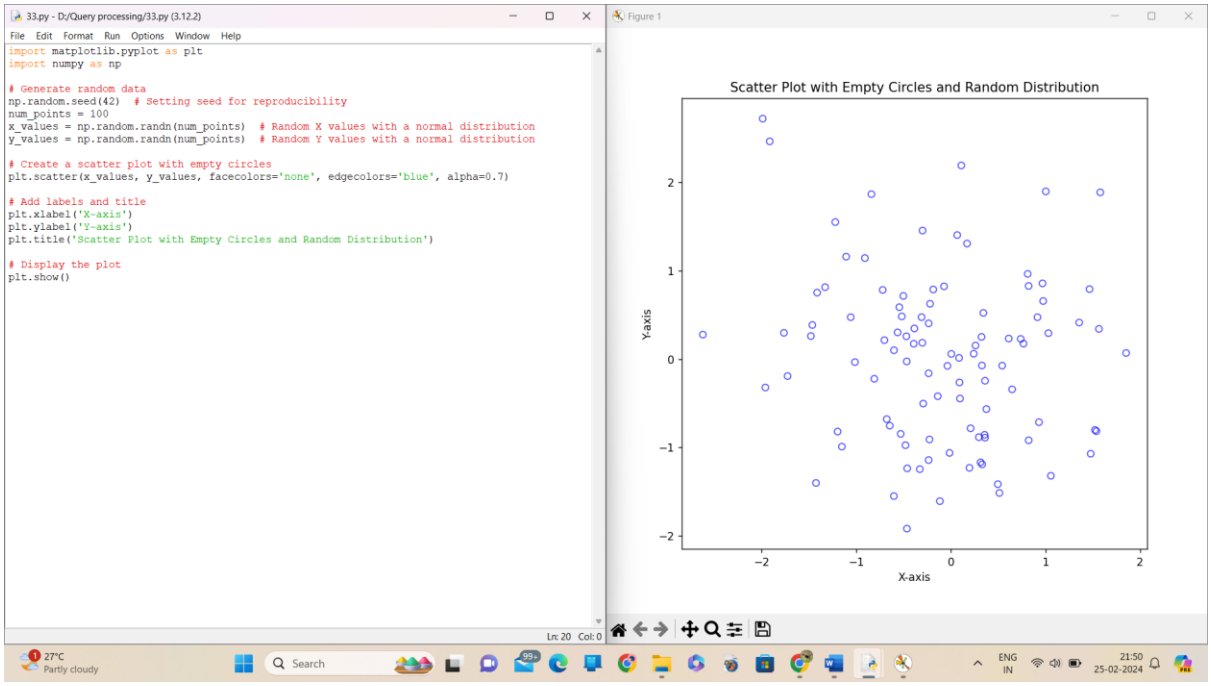


32.

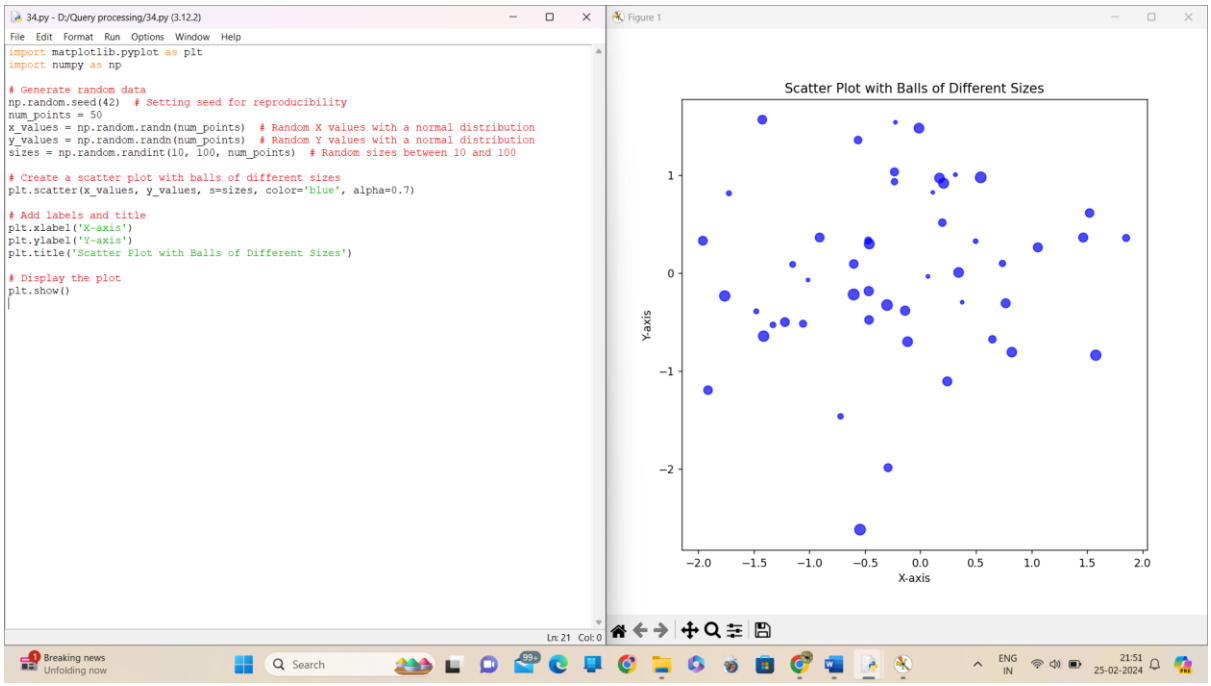
```
import matplotlib.pyplot as plt
import numpy as np

# Generate random data
np.random.seed(42)  # Setting seed for reproducibility
num_points = 100
x_values = np.random.randn(num_points)  # Random X values with a normal distribution
y_values = np.random.randn(num_points)  # Random Y values with a normal distribution

# Create a scatter plot
plt.scatter(x_values, y_values, color='blue', alpha=0.7)  # 'alpha' controls the transparency

# Add labels and title
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.title('Scatter Plot with Random Distribution')

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



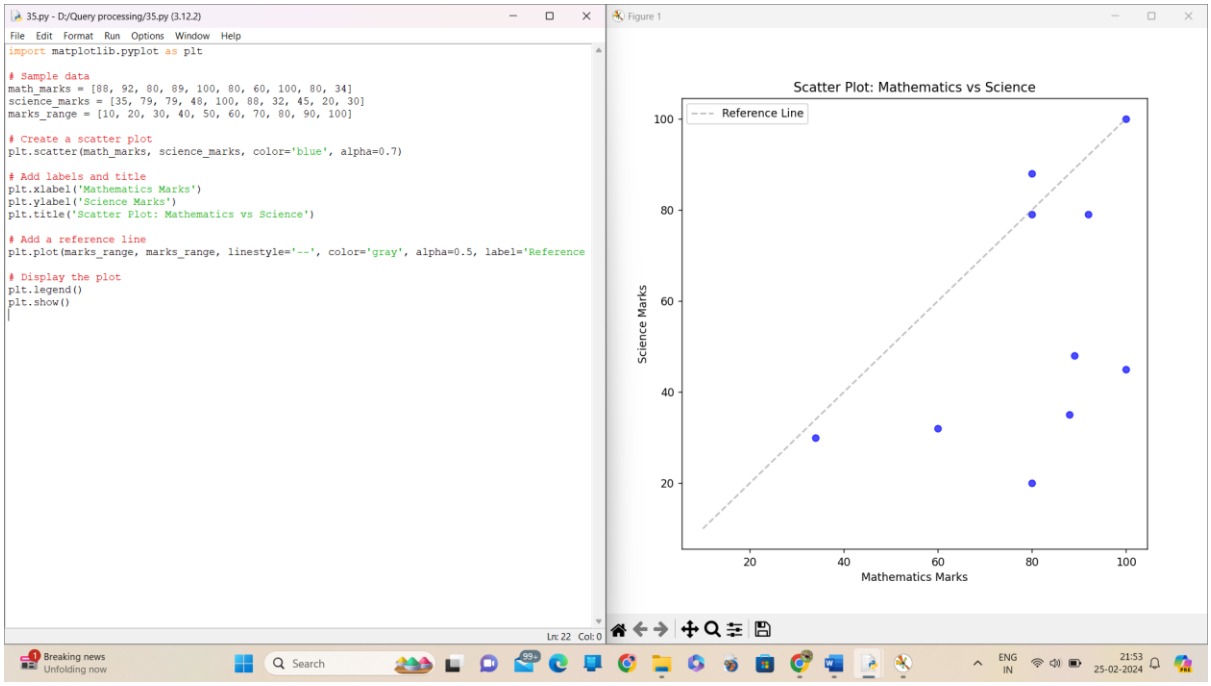
33.


33.py - D:/Query processing/33.py (3.12.2)
File Edit Format Run Options Window Help
import matplotlib.pyplot as plt
import numpy as np
# Generate random data
np.random.seed(42) # Setting seed for reproducibility
num_points = 100
x_values = np.random.randn(num_points) # Random X values with a normal distribution
y_values = np.random.randn(num_points) # Random Y values with a normal distribution
# Create a scatter plot with empty circles
plt.scatter(x_values, y_values, facecolors='none', edgecolors='blue', alpha=0.7)
# Add labels and title
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.title('Scatter Plot with Empty Circles and Random Distribution')
# Display the plot
plt.show()
Ln: 20 Col: 0
Figure 1
Scatter Plot with Empty Circles and Random Distribution
Y-axis
X-axis
27°C
Partly cloudy
Search
ENG
IN
21:50
25-02-2024

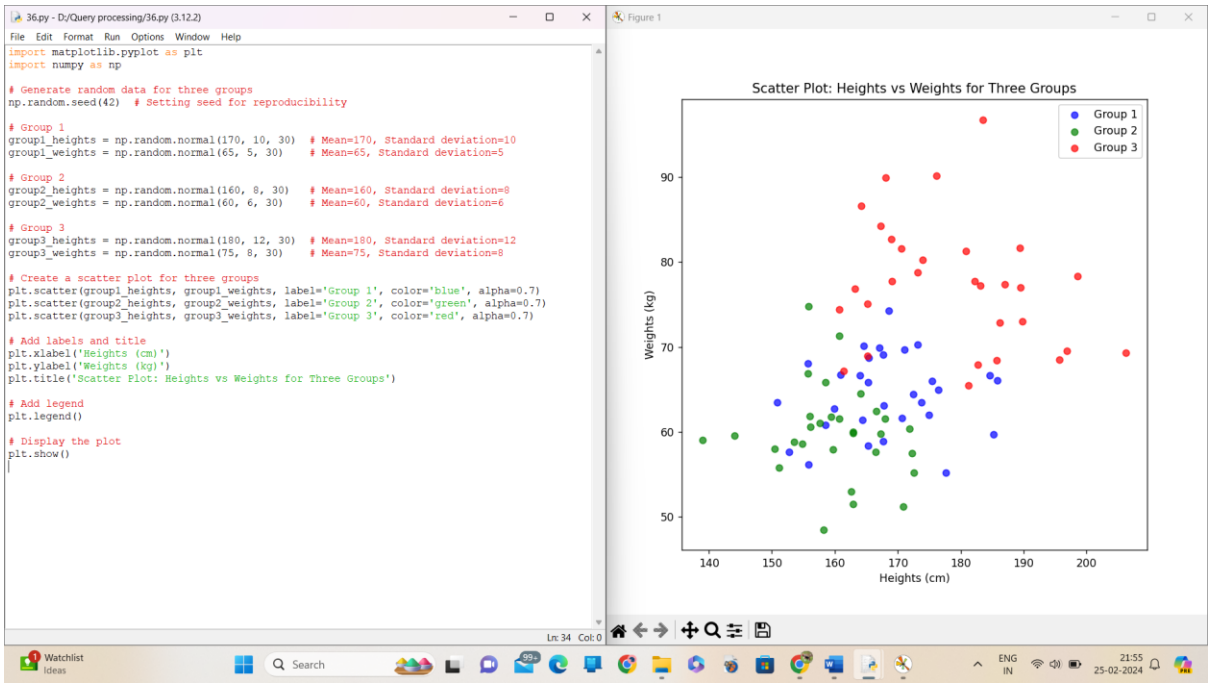
34.


34.py - D:/Query processing/34.py (3.12.2)
File Edit Format Run Options Window Help
import matplotlib.pyplot as plt
import numpy as np
# Generate random data
np.random.seed(42) # Setting seed for reproducibility
num_points = 50
x_values = np.random.randn(num_points) # Random X values with a normal distribution
y_values = np.random.randn(num_points) # Random Y values with a normal distribution
sizes = np.random.randint(10, 100, num_points) # Random sizes between 10 and 100
# Create a scatter plot with balls of different sizes
plt.scatter(x_values, y_values, s=sizes, color='blue', alpha=0.7)
# Add labels and title
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.title('Scatter Plot with Balls of Different Sizes')
# Display the plot
plt.show()
Ln: 21 Col: 0
Figure 1
Scatter Plot with Balls of Different Sizes
Y-axis
X-axis
Breaking news
Unfolding now
Search
ENG
IN
21:51
25-02-2024

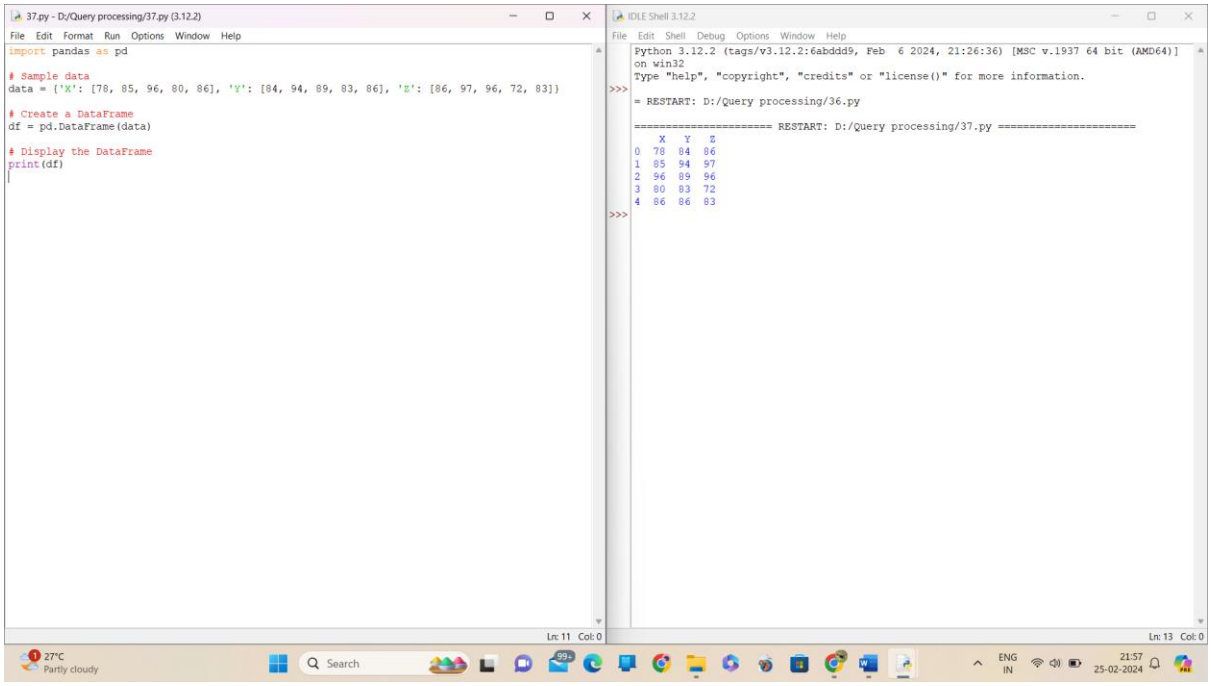
35.


35.py - D:/Query processing/35.py (3.12.2)
File Edit Format Run Options Window Help
import matplotlib.pyplot as plt
# Sample data
math_marks = [88, 92, 80, 89, 100, 80, 60, 100, 80, 34]
science_marks = [35, 79, 79, 48, 100, 88, 32, 45, 20, 30]
marks_range = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
# Create a scatter plot
plt.scatter(math_marks, science_marks, color='blue', alpha=0.7)
# Add labels and title
plt.xlabel('Mathematics Marks')
plt.ylabel('Science Marks')
plt.title('Scatter Plot: Mathematics vs Science')
# Add a reference line
plt.plot(marks_range, marks_range, linestyle='--', color='gray', alpha=0.5, label='Reference
# Display the plot
plt.legend()
plt.show()
Ln: 22 Col: 0
Figure 1
Scatter Plot: Mathematics vs Science
Reference Line
Science Marks
Mathematics Marks
100
80
60
40
20
20
40
60
80
100
Breaking news
Unfolding now
Search
ENG
IN
21:53
25-02-2024

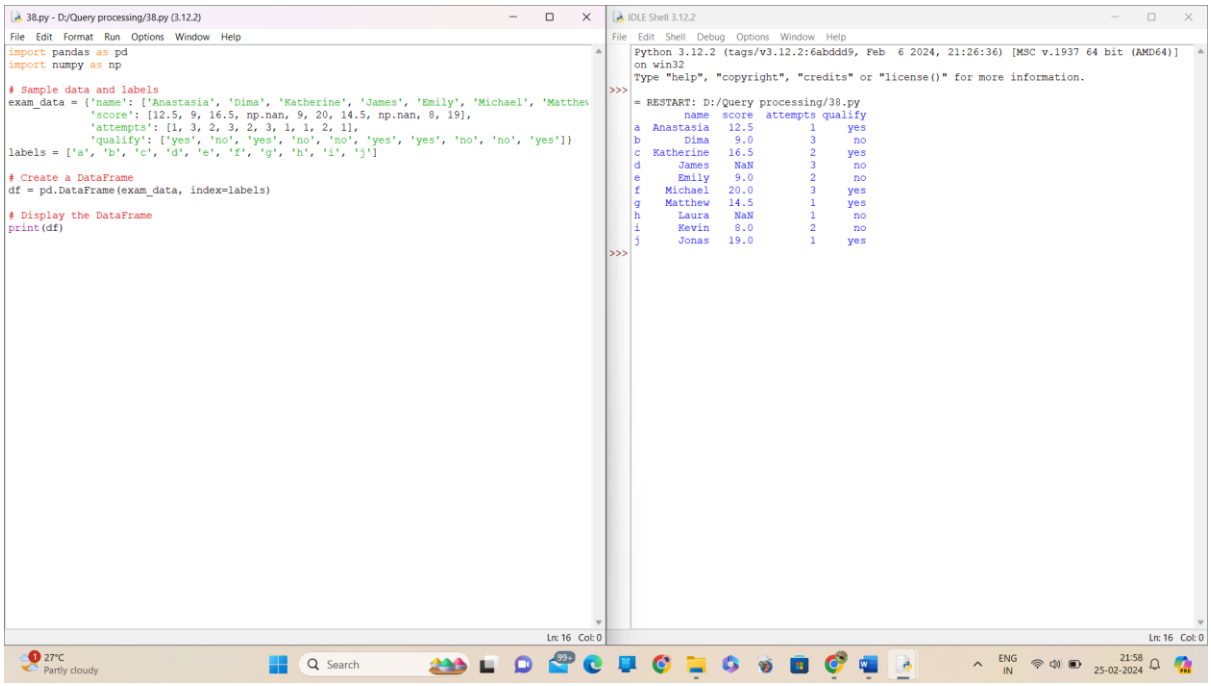
36.


36.py - D:/Query processing/36.py (3.12.2)
File Edit Format Run Options Window Help
import matplotlib.pyplot as plt
import numpy as np
# Generate random data for three groups
np.random.seed(42) # Setting seed for reproducibility
# Group 1
group1_heights = np.random.normal(170, 10, 30) # Mean=170, Standard deviation=10
group1_weights = np.random.normal(65, 5, 30) # Mean=65, Standard deviation=5
# Group 2
group2_heights = np.random.normal(160, 8, 30) # Mean=160, Standard deviation=8
group2_weights = np.random.normal(60, 6, 30) # Mean=60, Standard deviation=6
# Group 3
group3_heights = np.random.normal(180, 12, 30) # Mean=180, Standard deviation=12
group3_weights = np.random.normal(75, 8, 30) # Mean=75, Standard deviation=8
# Create a scatter plot for three groups
plt.scatter(group1_heights, group1_weights, label='Group 1', color='blue', alpha=0.7)
plt.scatter(group2_heights, group2_weights, label='Group 2', color='green', alpha=0.7)
plt.scatter(group3_heights, group3_weights, label='Group 3', color='red', alpha=0.7)
# Add labels and title
plt.xlabel('Heights (cm)')
plt.ylabel('Weights (kg)')
plt.title('Scatter Plot: Heights vs Weights for Three Groups')
# Add legend
plt.legend()
# Display the plot
plt.show()
Ln: 34 Col: 0
Figure 1
Scatter Plot: Heights vs Weights for Three Groups
Group 1
Group 2
Group 3
Weights (kg)
Heights (cm)
90
80
70
60
50
140
150
160
170
180
190
200
Watchlist
Ideas
Search
ENG
IN
21:55
25-02-2024

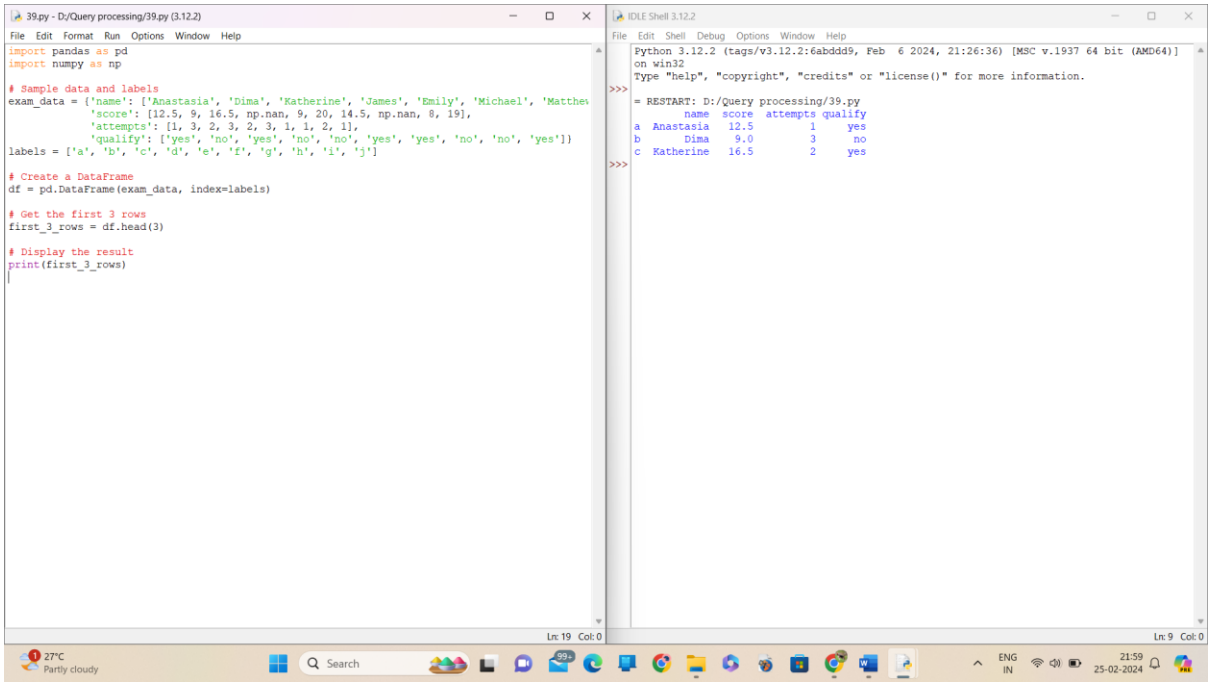
37.


37.py - D:/Query processing/37.py (3.12.2)
File Edit Format Run Options Window Help
import pandas as pd
# Sample data
data = {'X': [78, 85, 96, 80, 86], 'Y': [84, 94, 89, 83, 86], 'Z': [86, 97, 96, 72, 83]}
# Create a DataFrame
df = pd.DataFrame(data)
# Display the DataFrame
print(df)
Ln: 11 Col: 0
IDLE Shell 3.12.2
File Edit Shell Debug Options Window Help
Python 3.12.2 (tags/v3.12.2:6abddd9, Feb 6 2024, 21:26:36) [MSC v.1937 64 bit (AMD64)]
on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: D:/Query processing/36.py
========== RESTART: D:/Query processing/37.py ==========
X Y Z
0 78 84 86
1 85 94 97
2 96 89 96
3 80 83 72
4 86 86 83
>>>
Ln: 13 Col: 0

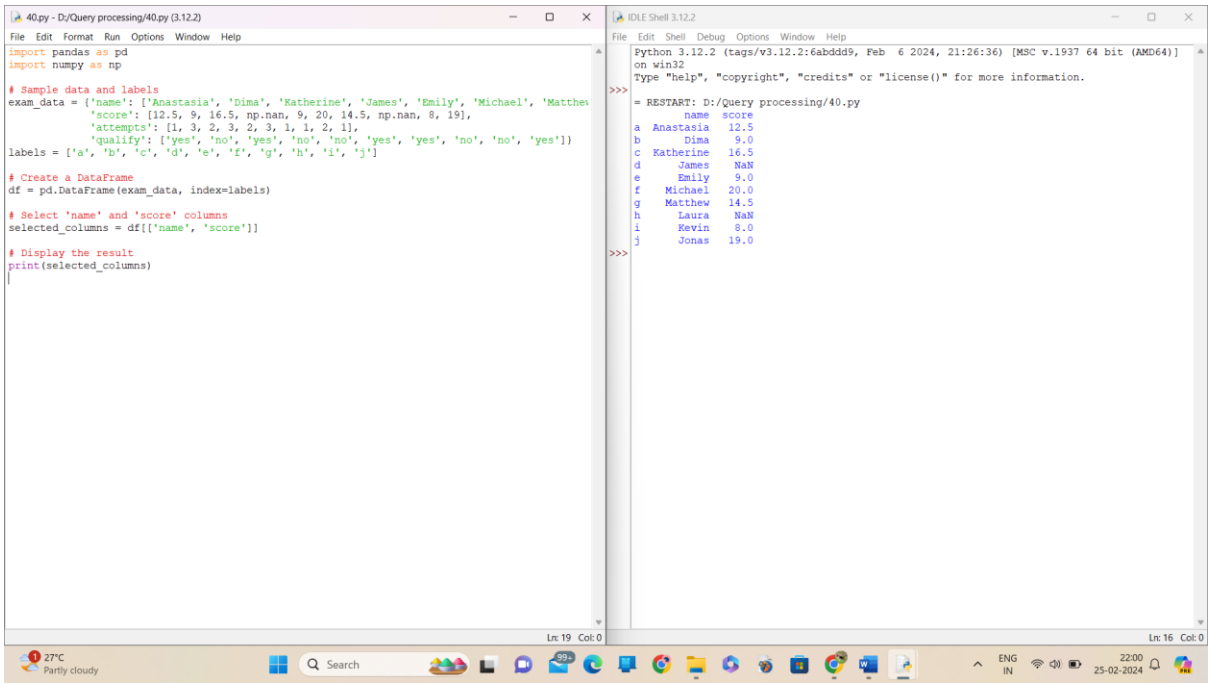
38.


38.py - D:/Query processing/38.py (3.12.2)
File Edit Format Run Options Window Help
import pandas as pd
import numpy as np
# Sample data and labels
exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew
'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],
'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],
'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
# Create a DataFrame
df = pd.DataFrame(exam_data, index=labels)
# Display the DataFrame
print(df)
Ln: 16 Col: 0
IDLE Shell 3.12.2
File Edit Shell Debug Options Window Help
Python 3.12.2 (tags/v3.12.2:6abddd9, Feb 6 2024, 21:26:36) [MSC v.1937 64 bit (AMD64)]
on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: D:/Query processing/38.py
name score attempts qualify
a Anastasia 12.5 1 yes
b Dima 9.0 3 no
c Katherine 16.5 2 yes
d James NaN 3 no
e Emily 9.0 2 no
f Michael 20.0 3 yes
g Matthew 14.5 1 yes
h Laura NaN 1 no
i Kevin 8.0 2 no
j Jonas 19.0 1 yes
>>>
Ln: 16 Col: 0

39.


39.py - D:/Query processing/39.py (3.12.2)
File Edit Format Run Options Window Help
import pandas as pd
import numpy as np
# Sample data and labels
exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthe
'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],
'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],
'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
# Create a DataFrame
df = pd.DataFrame(exam_data, index=labels)
# Get the first 3 rows
first_3_rows = df.head(3)
# Display the result
print(first_3_rows)
Ln: 19 Col: 0
IDLE Shell 3.12.2
File Edit Shell Debug Options Window Help
Python 3.12.2 (tags/v3.12.2:6abddd9, Feb 6 2024, 21:26:36) [MSC v.1937 64 bit (AMD64)]
on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: D:/Query processing/39.py
name score attempts qualify
a Anastasia 12.5 1 yes
b Dima 9.0 3 no
c Katherine 16.5 2 yes
>>>
Ln: 9 Col: 0
27°C
Partly cloudy
Search
ENG
IN
21:59
25-02-2024

40.


40.py - D:/Query processing/40.py (3.12.2)
File Edit Format Run Options Window Help
import pandas as pd
import numpy as np
# Sample data and labels
exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthe
'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],
'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],
'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
# Create a DataFrame
df = pd.DataFrame(exam_data, index=labels)
# Select 'name' and 'score' columns
selected_columns = df[['name', 'score']]
# Display the result
print(selected_columns)
Ln: 19 Col: 0
IDLE Shell 3.12.2
File Edit Shell Debug Options Window Help
Python 3.12.2 (tags/v3.12.2:6abddd9, Feb 6 2024, 21:26:36) [MSC v.1937 64 bit (AMD64)]
on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: D:/Query processing/40.py
name score
a Anastasia 12.5
b Dima 9.0
c Katherine 16.5
d James NaN
e Emily 9.0
f Michael 20.0
g Matthew 14.5
h Laura NaN
i Kevin 8.0
j Jonas 19.0
>>>
Ln: 16 Col: 0
27°C
Partly cloudy
Search
ENG
IN
22:00
25-02-2024