Experiment 25

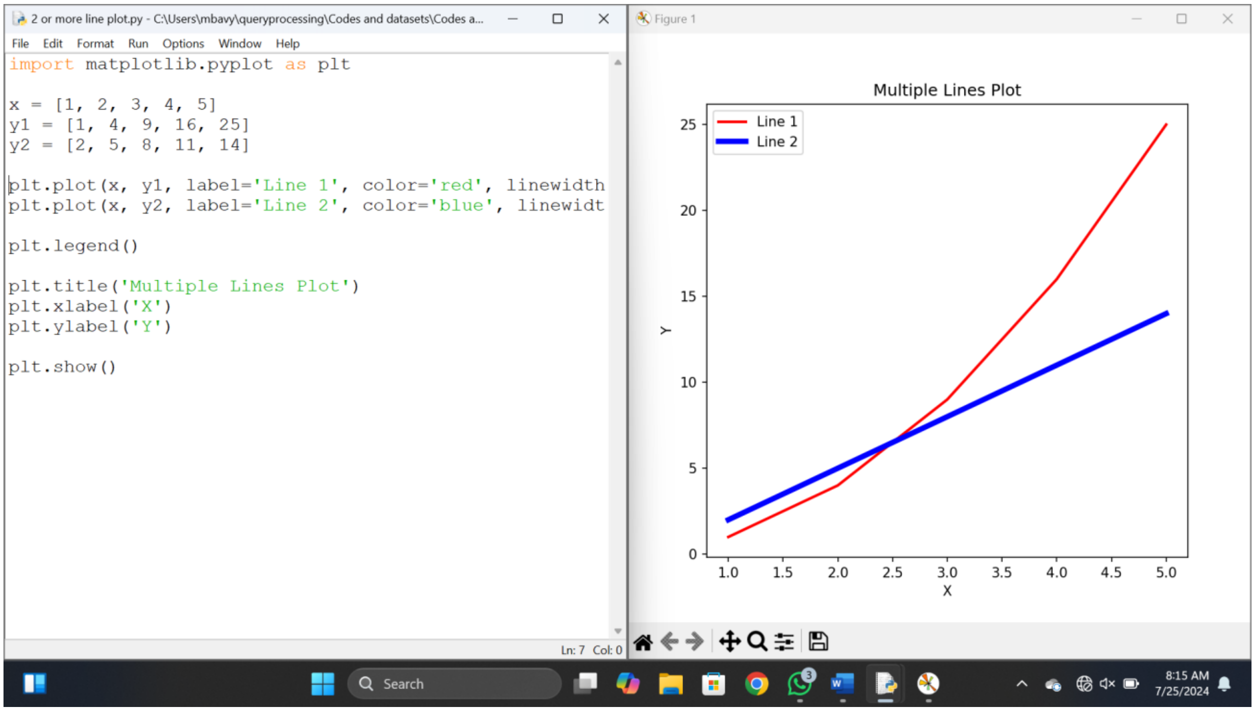
Aim:

To Write a Python program to plot two or more lines with legends, different widths and colors.

Code :

```
import matplotlib.pyplot as plt

x = [1, 2, 3, 4, 5]

y1 = [1, 4, 9, 16, 25]

y2 = [2, 5, 8, 11, 14]

plt.plot(x, y1, label='Line 1', color='red', linewidth=2)

plt.plot(x, y2, label='Line 2', color='blue', linewidth=4)

plt.legend()

plt.title('Multiple Lines Plot')

plt.xlabel('X')

plt.ylabel('Y')

plt.show()
```

Output:



```
import matplotlib.pyplot as plt

x = [1, 2, 3, 4, 5]
y1 = [1, 4, 9, 16, 25]
y2 = [2, 5, 8, 11, 14]

plt.plot(x, y1, label='Line 1', color='red', linewidth
plt.plot(x, y2, label='Line 2', color='blue', linewidt

plt.legend()

plt.title('Multiple Lines Plot')
plt.xlabel('X')
plt.ylabel('Y')

plt.show()
```

Results :

Thus a Python program to plot two or more lines with legends, different widths and colors is done.

Experiment 26

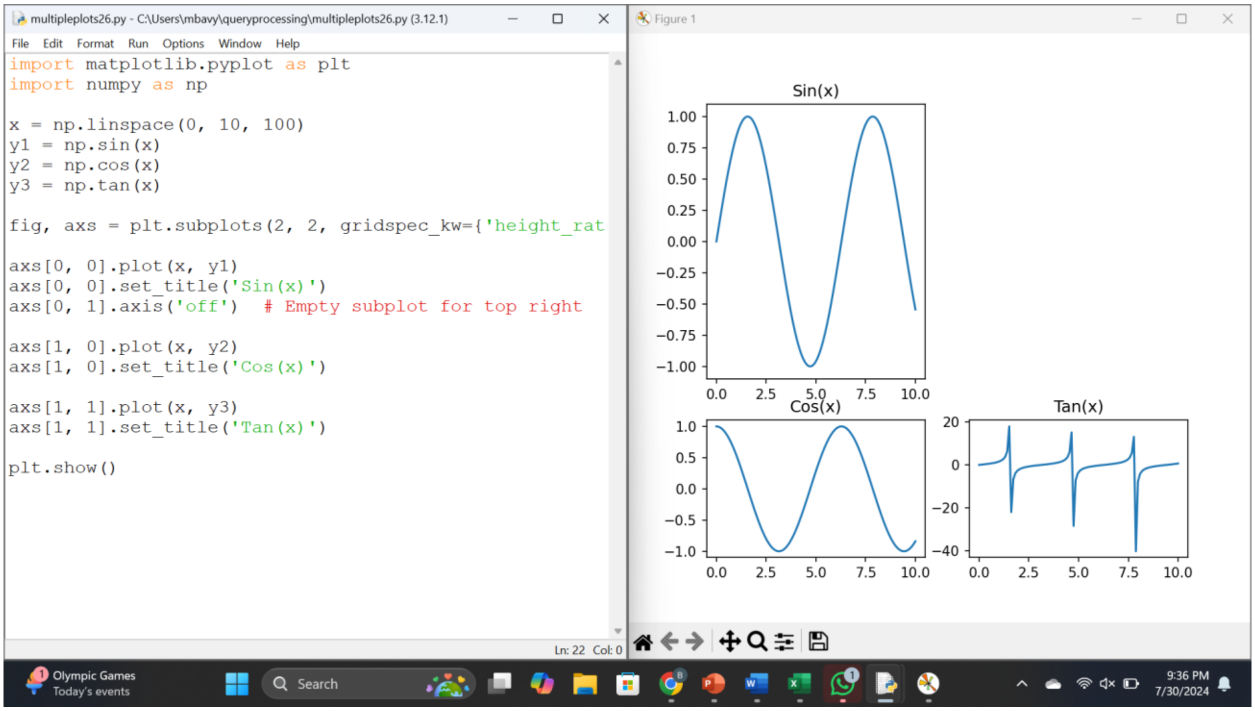
Aim:

To Write a Python program to create multiple plots.

Code :

```
import matplotlib.pyplot as plt

import numpy as np


x = np.linspace(0, 10, 100)

y1 = np.sin(x)

y2 = np.cos(x)

y3 = np.tan(x)


fig, axs = plt.subplots(2, 2, gridspec_kw={'height_ratios': [2, 1], 'width_ratios': [1, 1]})


axs[0, 0].plot(x, y1)

axs[0, 0].set_title('Sin(x)')

axs[0, 1].axis('off')  # Empty subplot for top right


axs[1, 0].plot(x, y2)

axs[1, 0].set_title('Cos(x)')


axs[1, 1].plot(x, y3)

axs[1, 1].set_title('Tan(x)')


plt.show()
```

Output:



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

x = np.linspace(0, 10, 100)
y1 = np.sin(x)
y2 = np.cos(x)
y3 = np.tan(x)

fig, axs = plt.subplots(2, 2, gridspec_kw={'height_rat

axs[0, 0].plot(x, y1)
axs[0, 0].set_title('Sin(x)')
axs[0, 1].axis('off')  # Empty subplot for top right

axs[1, 0].plot(x, y2)
axs[1, 0].set_title('Cos(x)')

axs[1, 1].plot(x, y3)
axs[1, 1].set_title('Tan(x)')

plt.show()
```

Results :

Thus a Python program to create multiple plots.

Experiment 27

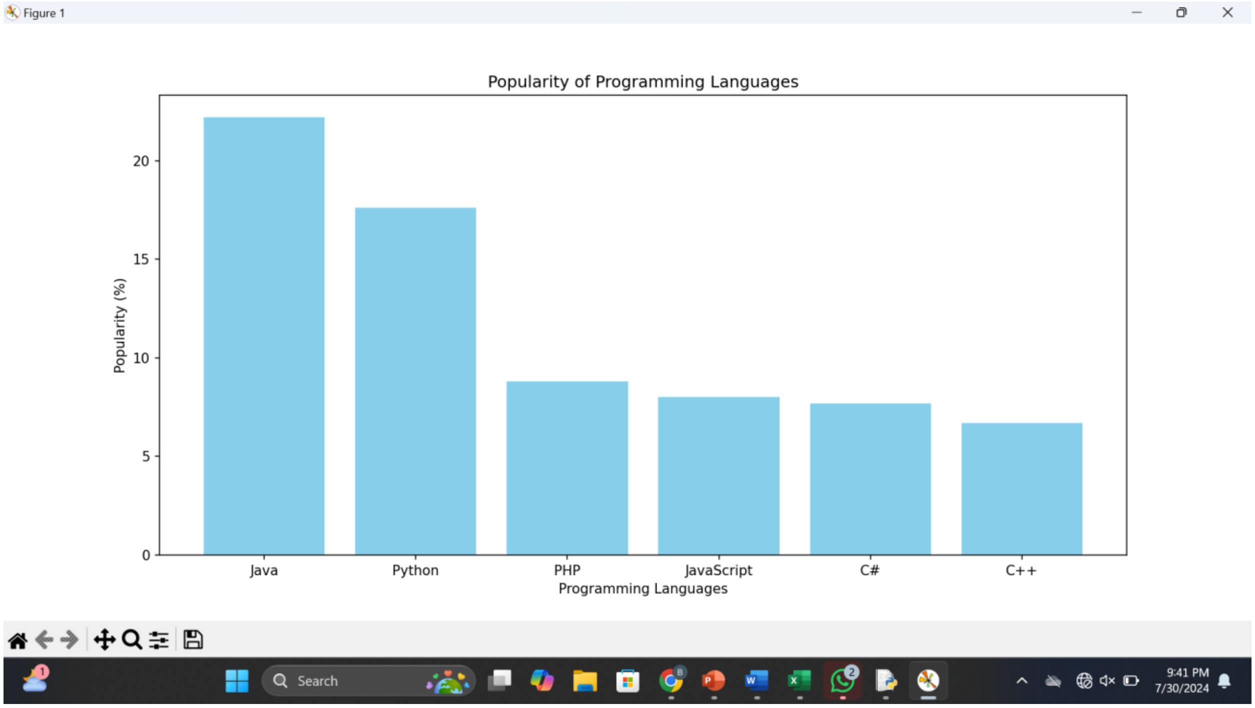
Aim:

To Write a Python programming to display a bar chart of the popularity of programming Languages.

Code :

```
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 titles and labels
plt.title('Popularity of Programming Languages')
plt.xlabel('Programming Languages')
plt.ylabel('Popularity (%)')


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

Output:



Results :

Thus a Python programming to display a bar chart of the popularity of programming Languages.

Experiment 28

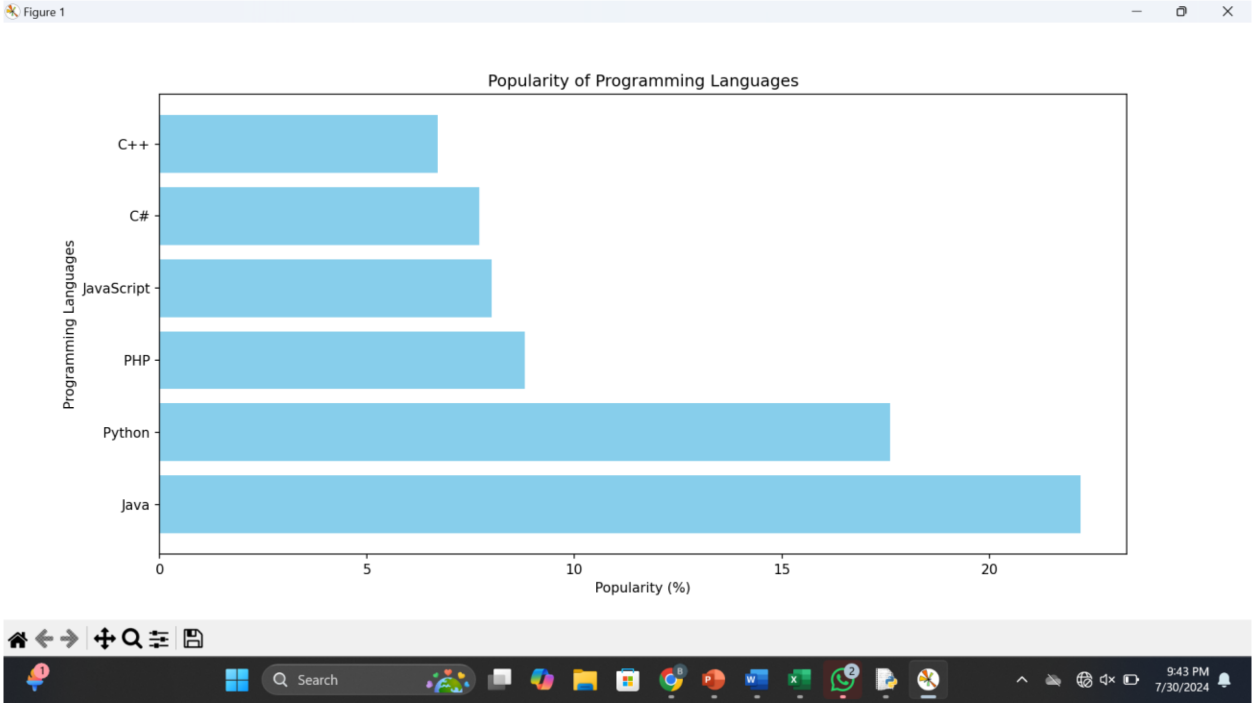
Aim:

To Write a Python programming to display a horizontal bar chart of the popularity of programming Languages.

Code :

```
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 titles and labels
plt.title('Popularity of Programming Languages')
plt.xlabel('Popularity (%)')
plt.ylabel('Programming Languages')

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

Output:



Results :

Thus a Python programming to display a horizontal bar chart of the popularity of programming Languages.

Experiment 29

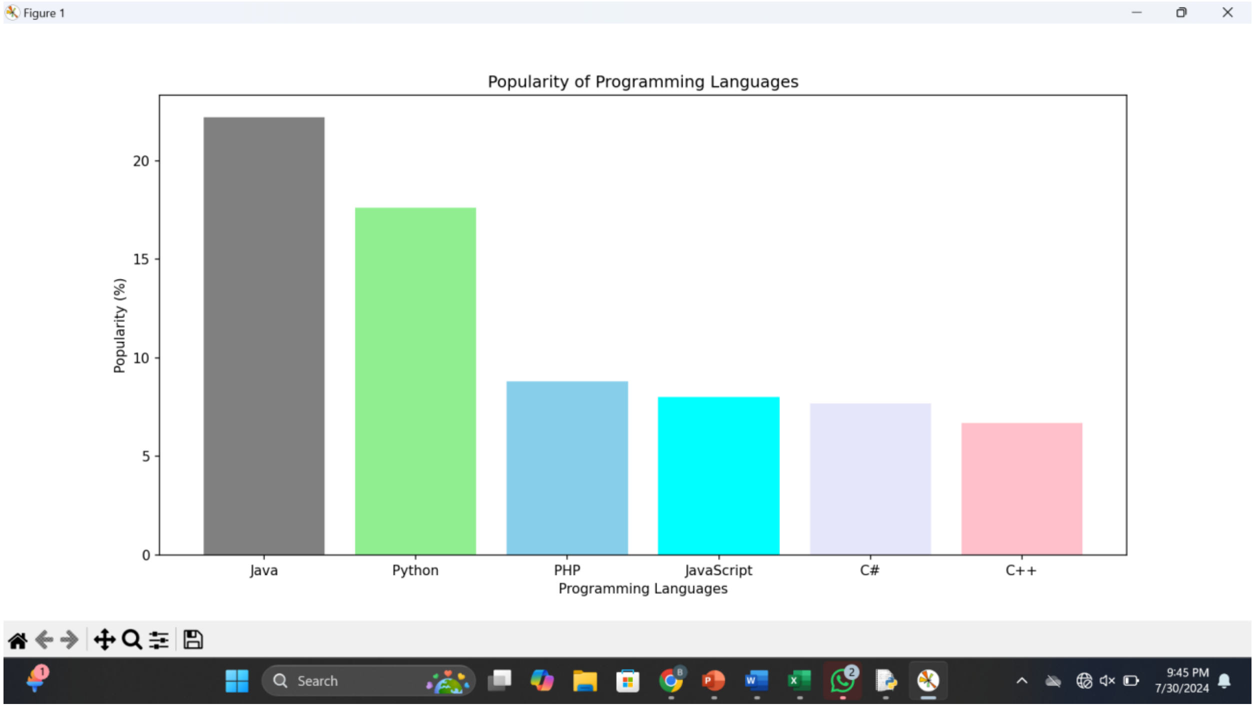
Aim: To Write a Python programming to display a bar chart of the popularity of programming Languages. Use different color for each bar.

Code :

```
import matplotlib.pyplot as plt


# Sample data

languages = ['Java', 'Python', 'PHP', 'JavaScript', 'C#', 'C++']

popularity = [22.2, 17.6, 8.8, 8.0, 7.7, 6.7]


# Colors for each bar

colors = ['grey', 'lightgreen', 'skyblue', 'cyan', 'lavender', 'pink']


# Create the bar chart
```

```
plt.figure(figsize=(10, 6))
plt.bar(languages, popularity, color=colors)

# Add title and labels
plt.title('Popularity of Programming Languages')
plt.xlabel('Programming Languages')
plt.ylabel('Popularity (%)')

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

Output:



Results : Thus a Python programming to display a bar chart of the popularity of programming Languages. Use different color for each bar.

Experiment 30

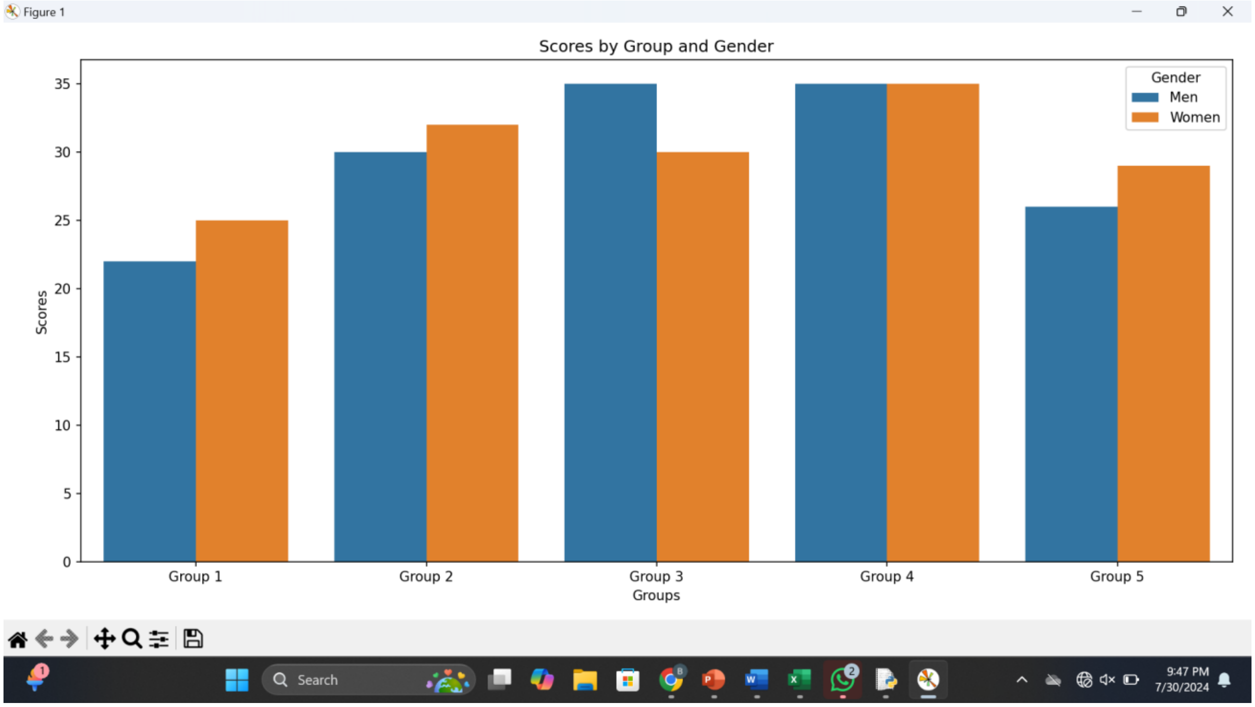
Aim:

To Write a Python program to create bar plot of scores by group and gender. Use multiple X values on the same chart for men and women.

Code

```
import seaborn as sns
t(id_vars='Group', var_name='Gender', value_name='Score')

# Plotting
plt.figure(figsize=(10, 6))
sns.barplot(data=df_melted, x='Group', y='Score', hue='Gender')

# Adding labels and title
plt.xlabel('Groups')
plt.ylabel('Scores')
plt.title('Scores by Group and Gender')

# Show plot
plt.tight_layout()
plt.show()
```

Output:



Results :

Thus a Python program to create bar plot of scores by group and gender. Use multiple X values on the same chart for men and women.

Experiment 31

Aim:

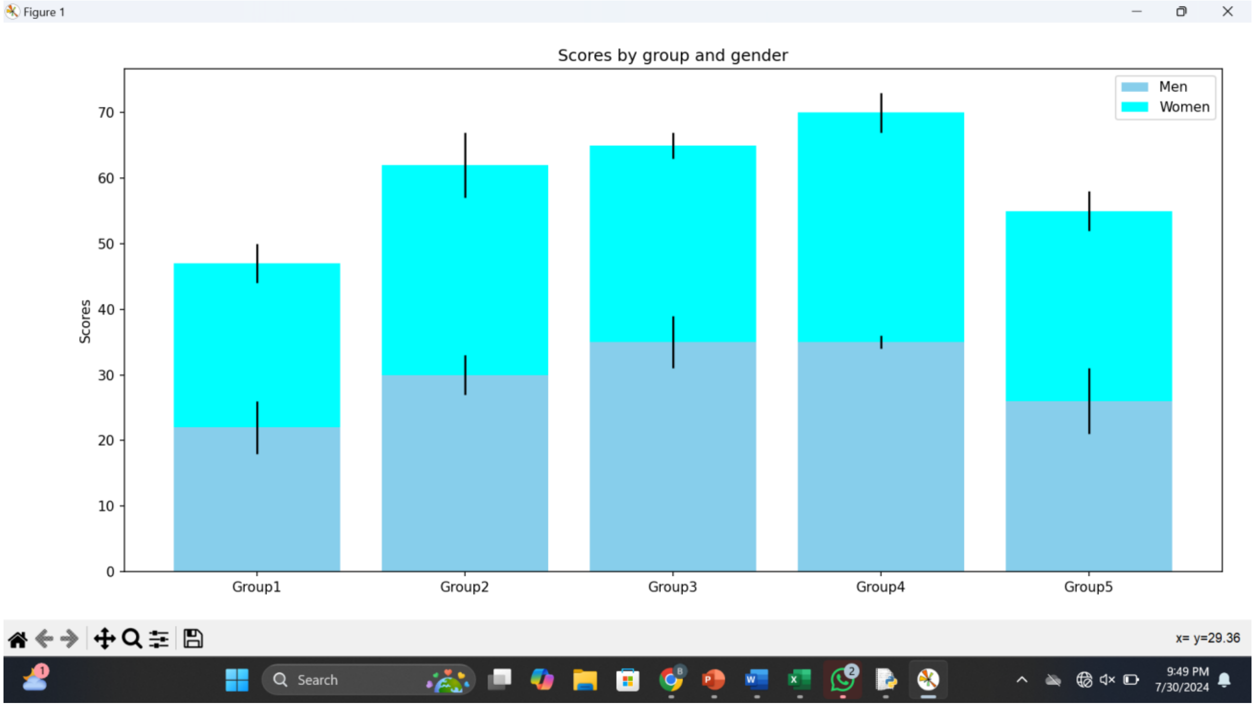
To Write a Python program to create a stacked bar plot with error bars.

Code :

```
import matplotlib.pyplot as plt

import numpy as np


# Sample data

means_men = [22, 30, 35, 35, 26]

means_women = [25, 32, 30, 35, 29]

std_men = [4, 3, 4, 1, 5]

std_women = [3, 5, 2, 3, 3]
```

```
# Number of groups
N = len(means_men)
ind = np.arange(N)  # the x locations for the groups
# Plot bars for men
plt.bar(ind, means_men,  color='skyblue', label='Men', yerr=std_men)
# Plot bars for women, stacked on top of men
plt.bar(ind, means_women,  bottom=means_men, color='cyan', label='Women',
yerr=std_women)

# Add some text for labels, title and axes ticks
plt.ylabel('Scores')
plt.title('Scores by group and gender')
plt.xticks(ind, ('Group1', 'Group2', 'Group3', 'Group4', 'Group5'))
plt.legend()

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

Output:



Results :

Thus a Python program to create a stacked bar plot with error bars

Experiment 32

Aim:

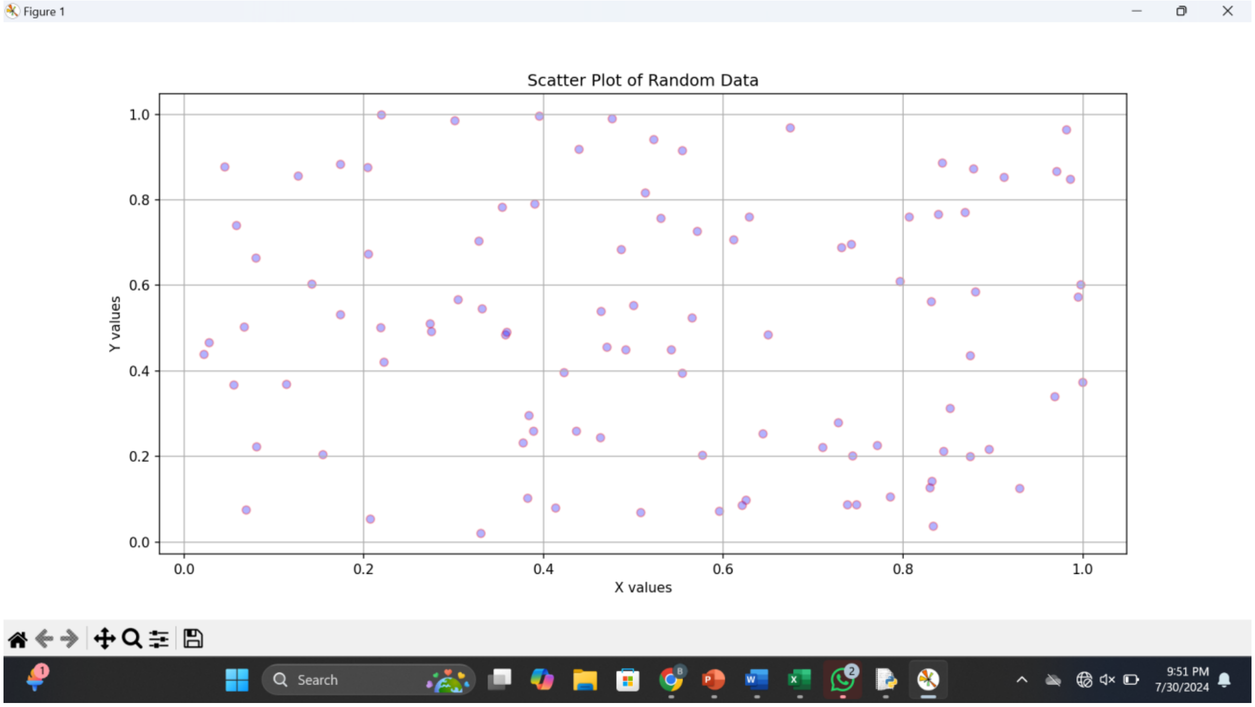
To Write a Python program to draw a scatter graph taking a random distribution in X and Y and plotted against each other.

Code

```
import numpy as np

import matplotlib.pyplot as plt


# Generate random data

  # For reproducibility

x = np.random.rand(100)  # 100 random values for X

y = np.random.rand(100)  # 100 random values for Y


# Create a scatter plot
```

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

plt.scatter(x, y, color='blue', alpha=0.3, edgecolors='red')

plt.title('Scatter Plot of Random Data')

plt.xlabel('X values')

plt.ylabel('Y values')

plt.grid(True)

plt.show()
```

Output:



Results :

Thus a Python program to draw a scatter graph taking a random distribution in X and Y and plotted against each other.