

Experiment 31

Aim:

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

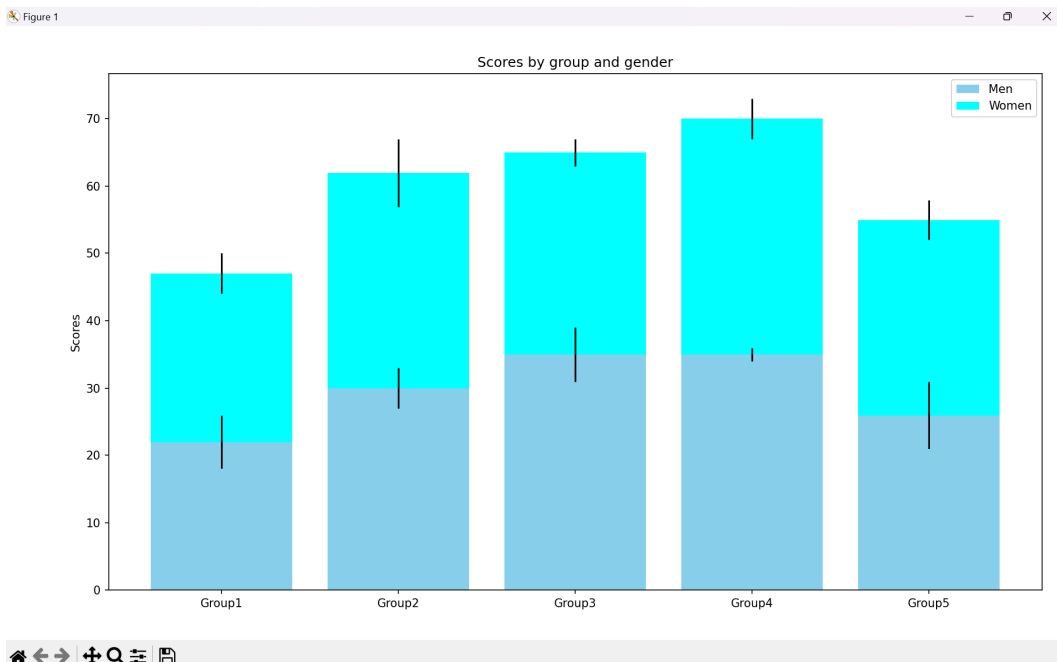
Code:

```
File Edit View Navigate Code Refactor Run Tools VCS Window Help
multipleplots26.py Version control Current File
stackedbarerror31.py
1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 means_men = [22, 30, 35, 35, 26]
5 means_women = [25, 32, 30, 35, 29]
6 std_men = [4, 3, 4, 1, 5]
7 std_women = [3, 5, 2, 3, 3]
8
9 N = len(means_men)
10 ind = np.arange(N)
11
12 plt.bar(ind, means_men, color='skyblue', label='Men', yerr=std_men)
13
14 plt.bar(ind, means_women, bottom=means_men, color='cyan', label='Women', yerr=std_wome
15
16 plt.ylabel('Scores')
17 plt.title('Scores by group and gender')
18 plt.xticks(ind, labels=('Group1', 'Group2', 'Group3', 'Group4', 'Group5'))
19 plt.legend()
20
```

Input:

```
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]
```

Output:



Experiment 32

Aim:

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

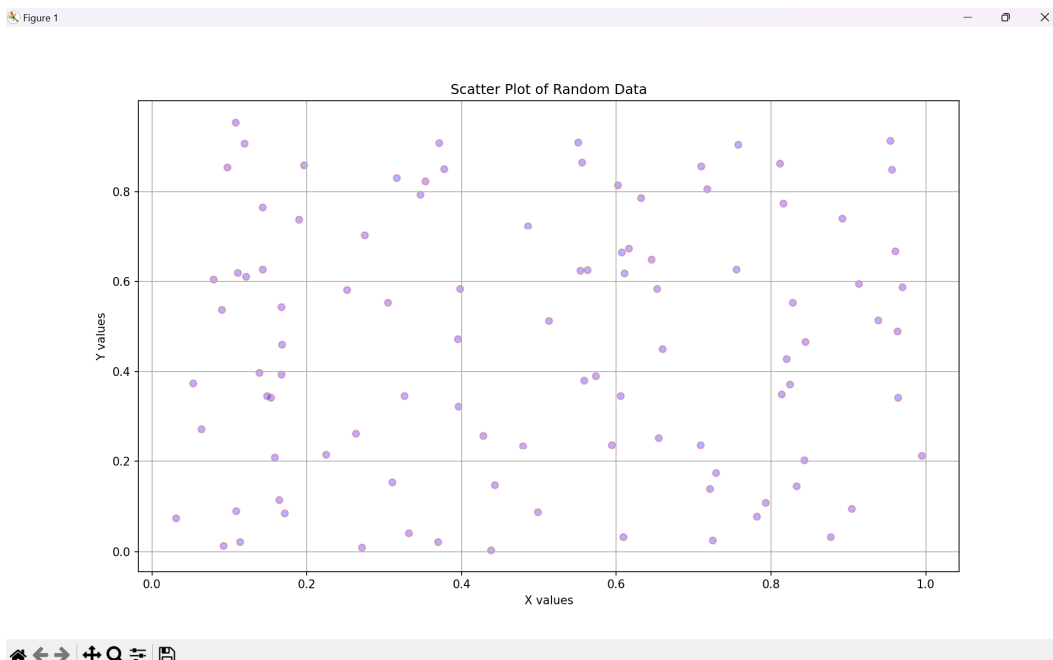
Code:

```
File Edit View Navigate Code Refactor Run Tools VCS Window Help
multipleplots26.py Version control Current File
scatter_random32.py
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 x = np.random.rand(100)
5 y = np.random.rand(100)
6
7 plt.figure(figsize=(8, 6))
8 plt.scatter(x, y, color='blue', alpha=0.3, edgecolors='red')
9 plt.title('Scatter Plot of Random Data')
10 plt.xlabel('X values')
11 plt.ylabel('Y values')
12 plt.grid(True)
13 plt.show()
14
15
16
```

Input:

```
x = np.random.rand(100)
y = np.random.rand(100)
```

Output:



Experiment 33

Aim:

To develop a Python program to draw a scatter plot with empty circles taking a random distribution in X and Y and plotted against each other.

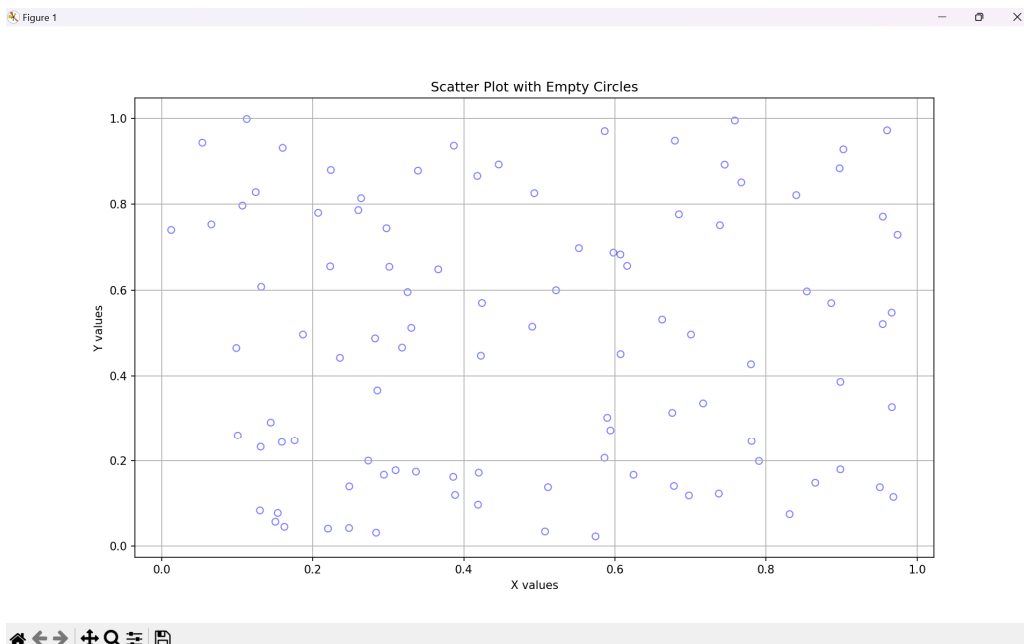
Code:

```
File Edit View Navigate Code Refactor Run Tools VCS Window Help
multipleplots26.py Version control Current File
scatteremptycircle33.py
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 x = np.random.rand(100)
5 y = np.random.rand(100)
6
7 plt.figure(figsize=(8, 6))
8 plt.scatter(x, y, edgecolors='blue', facecolors='none', alpha=0.5)
9 plt.title('Scatter Plot with Empty Circles')
10 plt.xlabel('X values')
11 plt.ylabel('Y values')
12 plt.grid(True)
13 plt.show()
14
```

Input:

```
x = np.random.rand(100)
y = np.random.rand(100)
```

Output:



Experiment 34

Aim:

To develop a Python program to draw a scatter plot using random distributions to generate balls of different sizes.

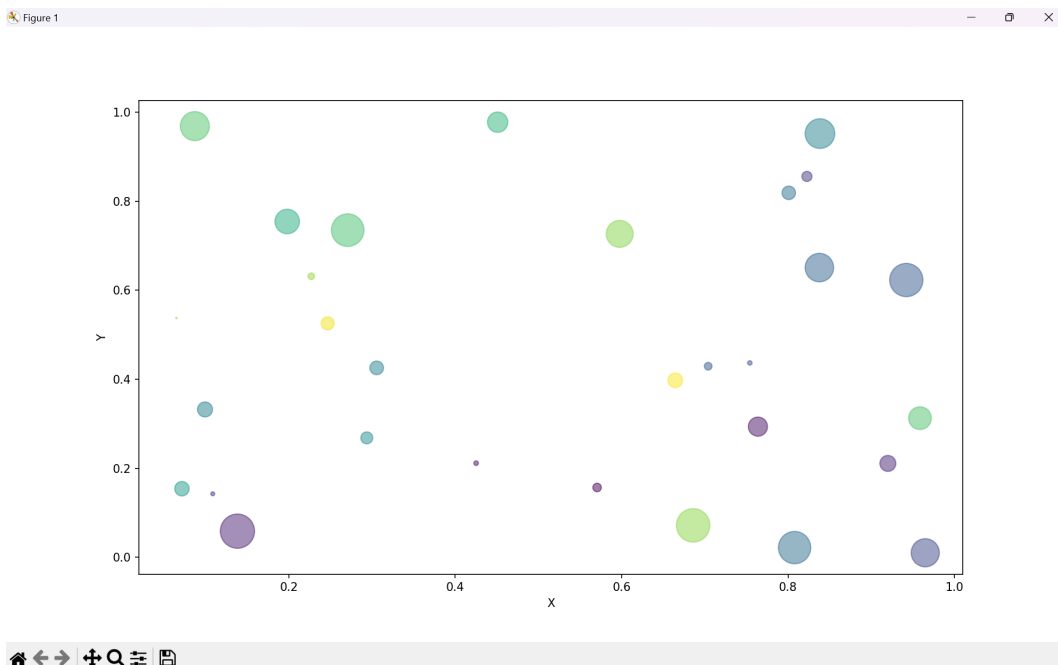
Code:

```
File Edit View Navigate Code Refactor Run Tools VCS Window Help
multipleplots26.py Version control
diffsizescatter34.py
18
19 import matplotlib.pyplot as plt
20 import numpy as np
21 n = 30
22 x = np.random.rand(n)
23 y = np.random.rand(n)
24 colors = np.random.rand(n)
25 area = (30 * np.random.rand(n))**2
26
27 plt.scatter(x, y, s=area, c=colors, alpha=0.5)
28 plt.xlabel('X')
29 plt.ylabel('Y')
30 plt.show()
31
```

Input:

```
n = 30
x = np.random.rand(n)
y = np.random.rand(n)
```

Output:



Experiment 35

Aim:

To develop a Python program to draw a scatter plot comparing two subject marks of Mathematics and Science.

Code:

```
File Edit View Navigate Code Refactor Run Tools VCS Window Help
multipleplots26.py Version control
scattermathsci35.py
1 import matplotlib.pyplot as plt
2
3 math_marks = [88, 92, 80, 89, 100, 80, 60, 100, 80, 34]
4 science_marks = [35, 79, 79, 48, 100, 88, 32, 45, 20, 30]
5 marks_range = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
6
7 plt.scatter(marks_range, math_marks, color='red', alpha=0.5, label='Math marks')
8 plt.scatter(marks_range, science_marks, color='green', alpha=0.5, label='Science marks')
9
10 plt.xlabel('Marks Range')
11 plt.ylabel('Marks Scored')
12 plt.title('Scatter Plot of Math vs. Science Marks')
13
14 plt.legend()
15
16 plt.grid(True)
17 plt.show()
18
```

Input:

```
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]
```

Output:

