

## Matplotlib

- Matplotlib is a low level graph plotting library in python that serves as a visualization utility.
- Matplotlib is open source and we can use it freely.
- Matplotlib is mostly written in python, a few segments are written in C, Objective-C and Javascript for Platform compatibility.
- Matplotlib consists of several plots like line, bar, scatter, histogram, etc.
- For more Information check out: <https://matplotlib.org/stable/>

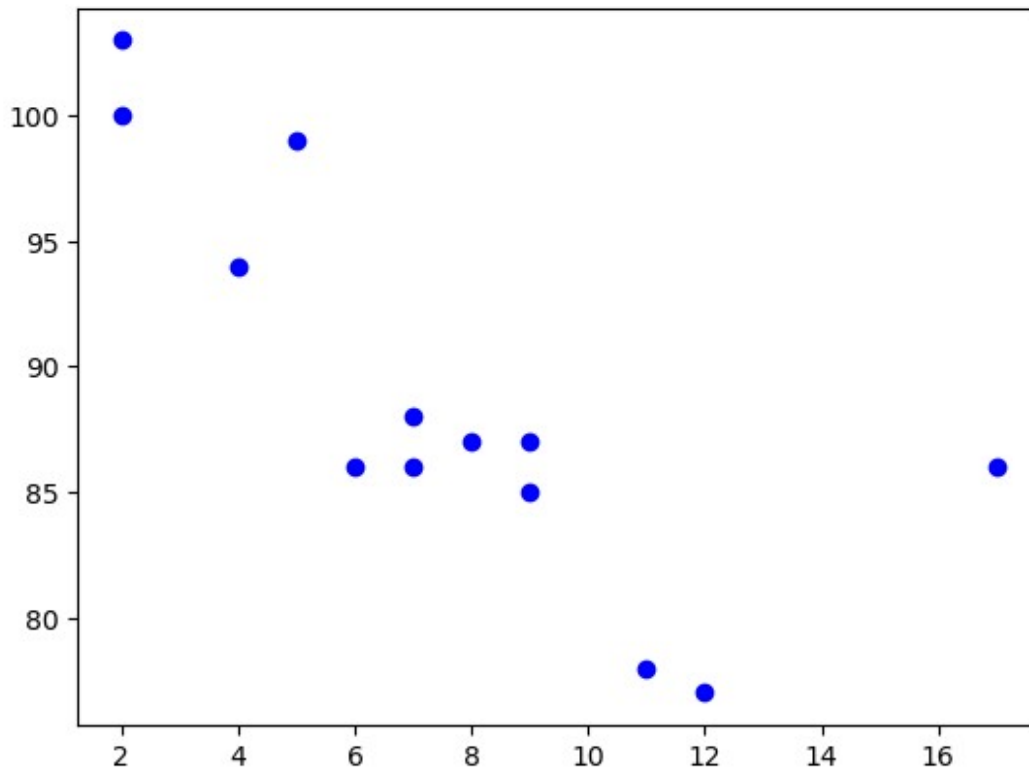
```
import matplotlib
import matplotlib.pyplot as plt
print(matplotlib.__version__)
```

3.7.2

## Scatter plot

- Scatter plot is used to visualize the relationship between 2 continuous numerical variables.
- The scatter() function plots one dot for each observation.
- It needs two arrays of the same length, one for the values of the x-axis, and one for values on the y-axis

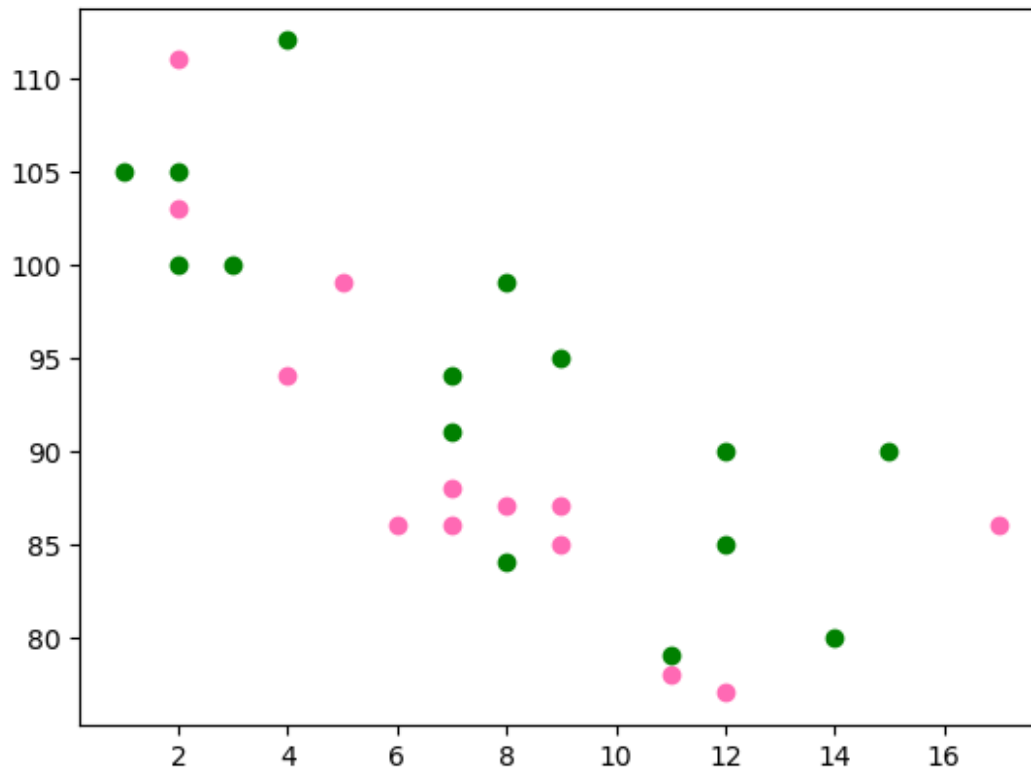
```
x = [5, 7, 8, 7, 2, 17, 2, 9, 4, 11, 12, 9, 6]
y = [99, 86, 87, 88, 100, 86, 103, 87, 94, 78, 77, 85, 86]
plt.scatter(x, y, c = "blue")
plt.show()
```



```
import numpy as np
x = np.array([5,7,8,7,2,17,2,9,4,11,12,9,6])
y = np.array([99,86,87,88,111,86,103,87,94,78,77,85,86])
plt.scatter(x, y, c='hotpink')

x = np.array([2,2,8,1,15,8,12,9,7,3,11,4,7,14,12])
y = np.array([100,105,84,105,90,99,90,95,94,100,79,112,91,80,85])
plt.scatter(x, y, c='green')

plt.show()
```

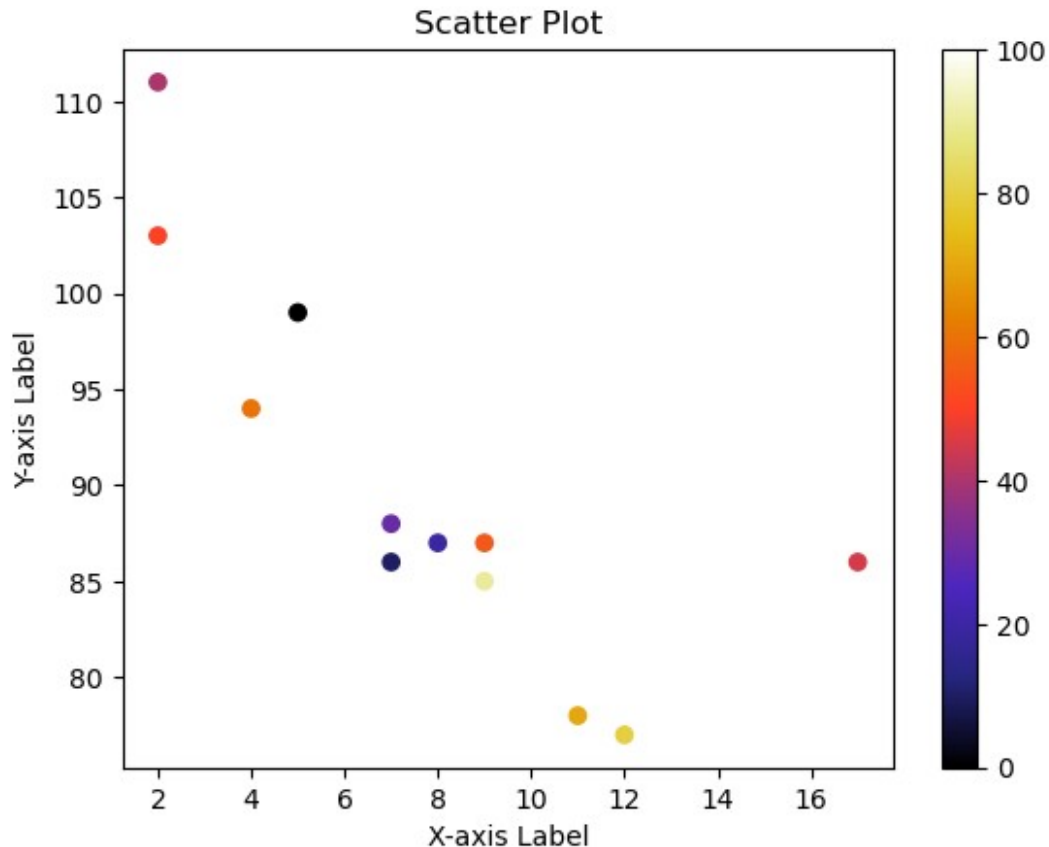


```
x = np.array([5,7,8,7,2,17,2,9,4,11,12,9,6])
y = np.array([99,86,87,88,111,86,103,87,94,78,77,85,86])
colors = np.array([0, 10, 20, 30, 40, 45, 50, 55, 60, 70, 80, 90, 100])

plt.scatter(x, y, c=colors, cmap='CMRmap')

plt.colorbar()
plt.xlabel('X-axis Label')
plt.ylabel('Y-axis Label')
plt.title('Scatter Plot')

plt.show()
```



## Bar Plot

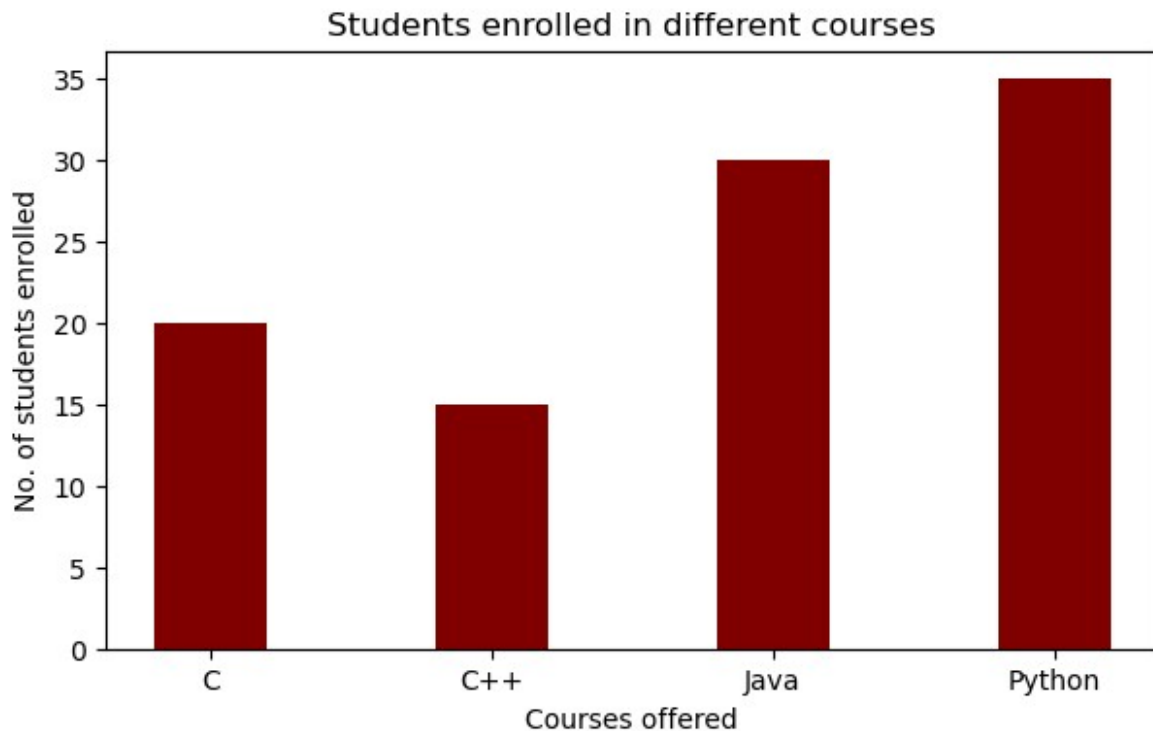
- The `bar()` function takes arguments that describes the layout of the bars.
- The categories and their values represented by the first and second argument as arrays.
- A bar plot or bar chart is a graph that represents the category of data with rectangular bars with lengths and heights that is proportional to the values which they represent.
- The bar plots can be plotted horizontally or vertically.
- If you want the bars to be displayed horizontally instead of vertically, use the `barh()` function
- A bar chart describes the comparisons between the discrete categories.
- One of the axis of the plot represents the specific categories being compared, while the other axis represents the measured values corresponding to those categories.

```
data = {'C':20, 'C++':15, 'Java':30, 'Python':35}
courses = list(data.keys())
values = list(data.values())

fig = plt.figure(figsize = (7, 4))

plt.bar(courses, values, color='maroon',
        width = 0.4)
```

```
plt.xlabel("Courses offered")
plt.ylabel("No. of students enrolled")
plt.title("Students enrolled in different courses")
plt.show()
```



```
barWidth = 0.25
fig = plt.subplots(figsize=(12, 8))

IT = [32, 30, 17, 28, 22]
ECE = [28, 6, 16, 5, 10]
CSE = [19, 3, 24, 25, 7]

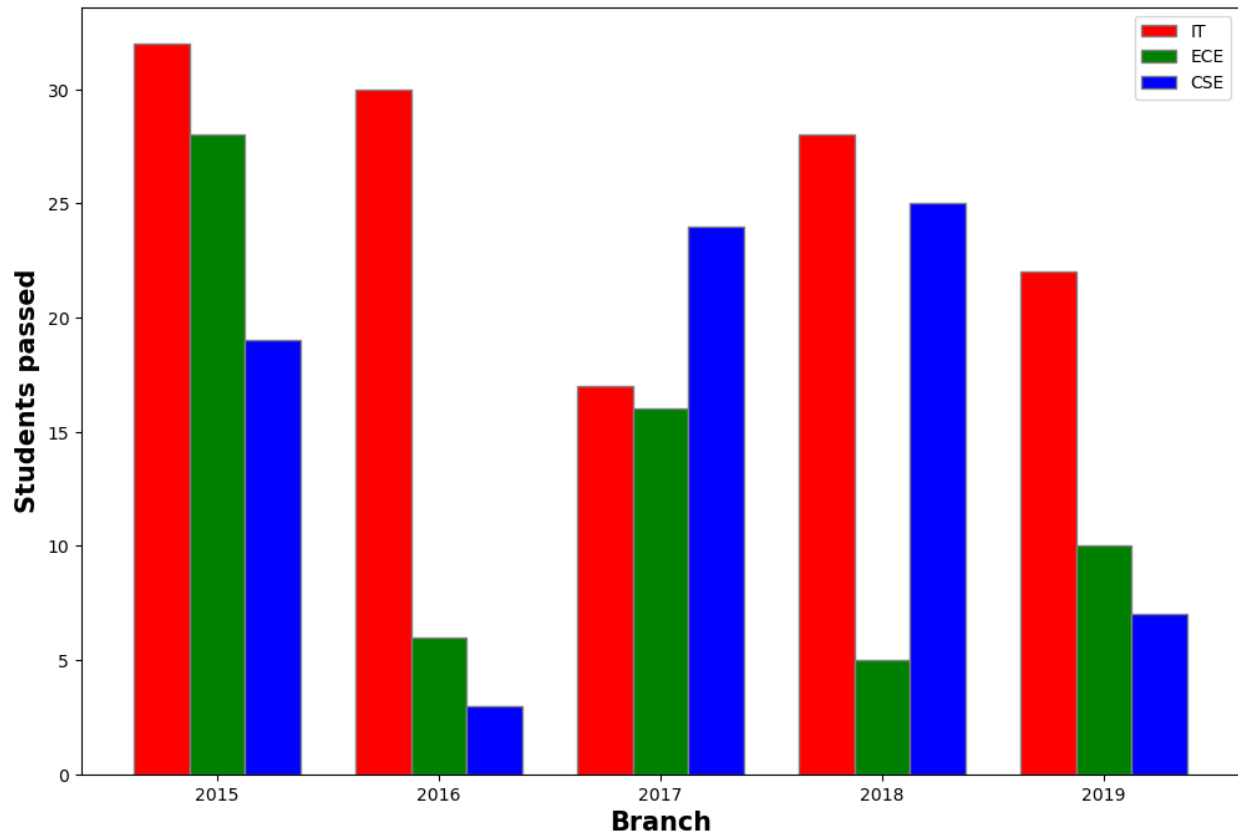
br1 = np.arange(len(IT))
br2 = [x + barWidth for x in br1]
br3 = [x + barWidth for x in br2]

plt.bar(br1, IT, color='r', width = barWidth,
        edgecolor='grey', label='IT')
plt.bar(br2, ECE, color='g', width = barWidth,
        edgecolor='grey', label='ECE')
plt.bar(br3, CSE, color='b', width = barWidth,
        edgecolor='grey', label='CSE')

plt.xlabel('Branch', fontweight='bold', fontsize = 15)
plt.ylabel('Students passed', fontweight='bold', fontsize = 15)
```

```
plt.xticks([r + barWidth for r in range(len(IT))],
           ['2015', '2016', '2017', '2018', '2019'])

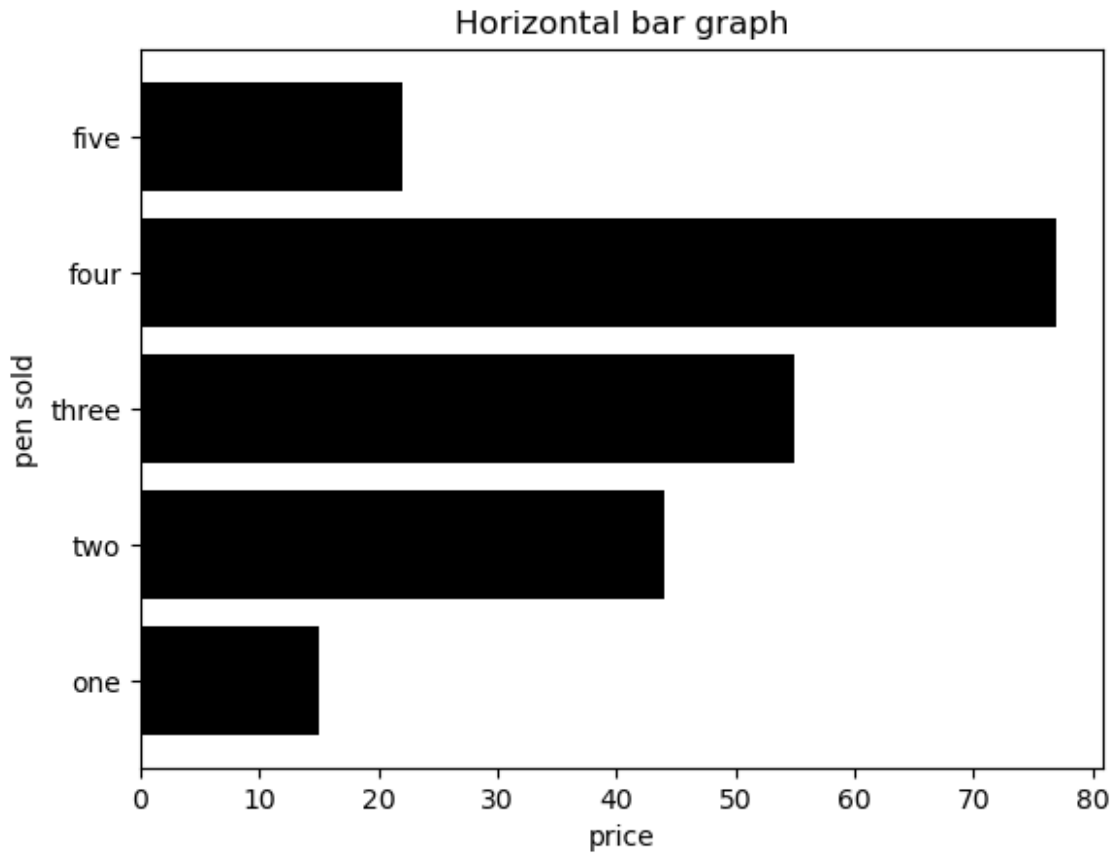
plt.legend()
plt.show()
```



```
y=['one', 'two', 'three', 'four', 'five']
x=[15,44,55,77,22]
plt.barh(y, x, color='black')

plt.ylabel("pen sold")

plt.xlabel("price")
plt.title("Horizontal bar graph")
plt.show()
```



## Histogram

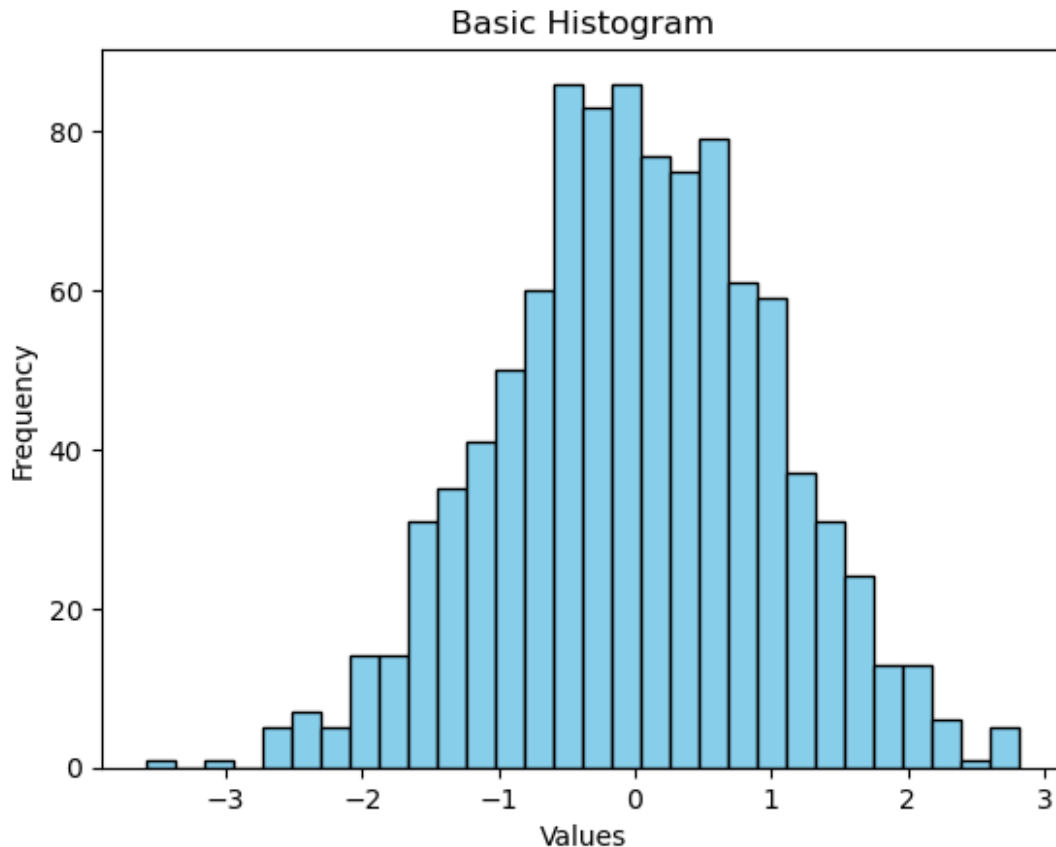
- A Histogram represents data provided in the form of some groups.
- It is an accurate method for the graphical representation of numerical data distribution.
- It is a type of bar plot where the X-axis represents the bin ranges while the Y-axis gives information about frequency.
- In Matplotlib, we use the `hist()` function to create histograms.
- The `hist()` function will use an array of numbers to create a histogram, the array is sent into the function as an argument.

```
data = np.random.randn(1000)

plt.hist(data, bins=30, color='skyblue', edgecolor='black')

plt.xlabel('Values')
plt.ylabel('Frequency')
plt.title('Basic Histogram')

plt.show()
```



```
data1 = np.random.randn(1000)
data2 = np.random.normal(loc=3, scale=1, size=1000)

fig, axes = plt.subplots(nrows=1, ncols=2, figsize=(12, 4))

axes[0].hist(data1, bins=30, color='DarkGreen', edgecolor='black')
axes[0].set_title('Histogram 1')

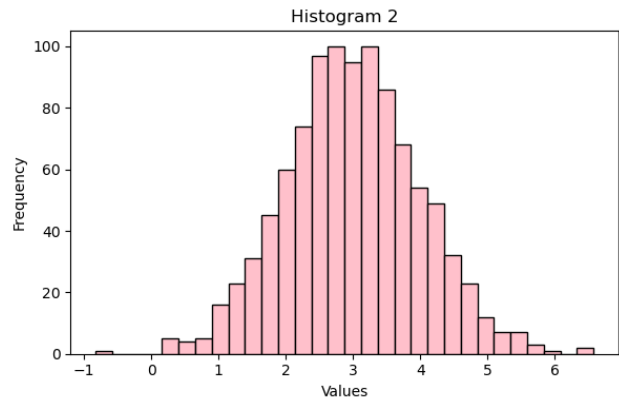
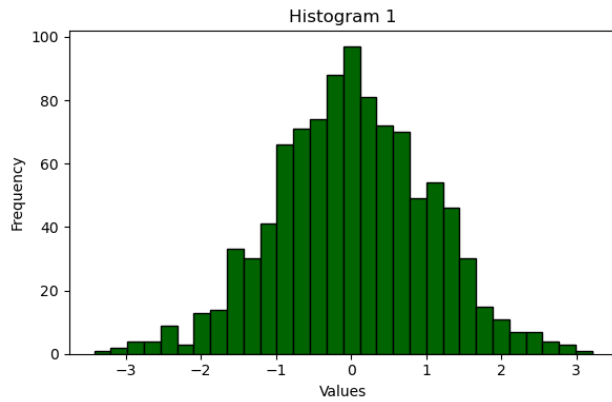
axes[1].hist(data2, bins=30, color='Pink', edgecolor='black')
axes[1].set_title('Histogram 2')

for ax in axes:
    ax.set_xlabel('Values')
    ax.set_ylabel('Frequency')

plt.tight_layout()

plt.show()
```





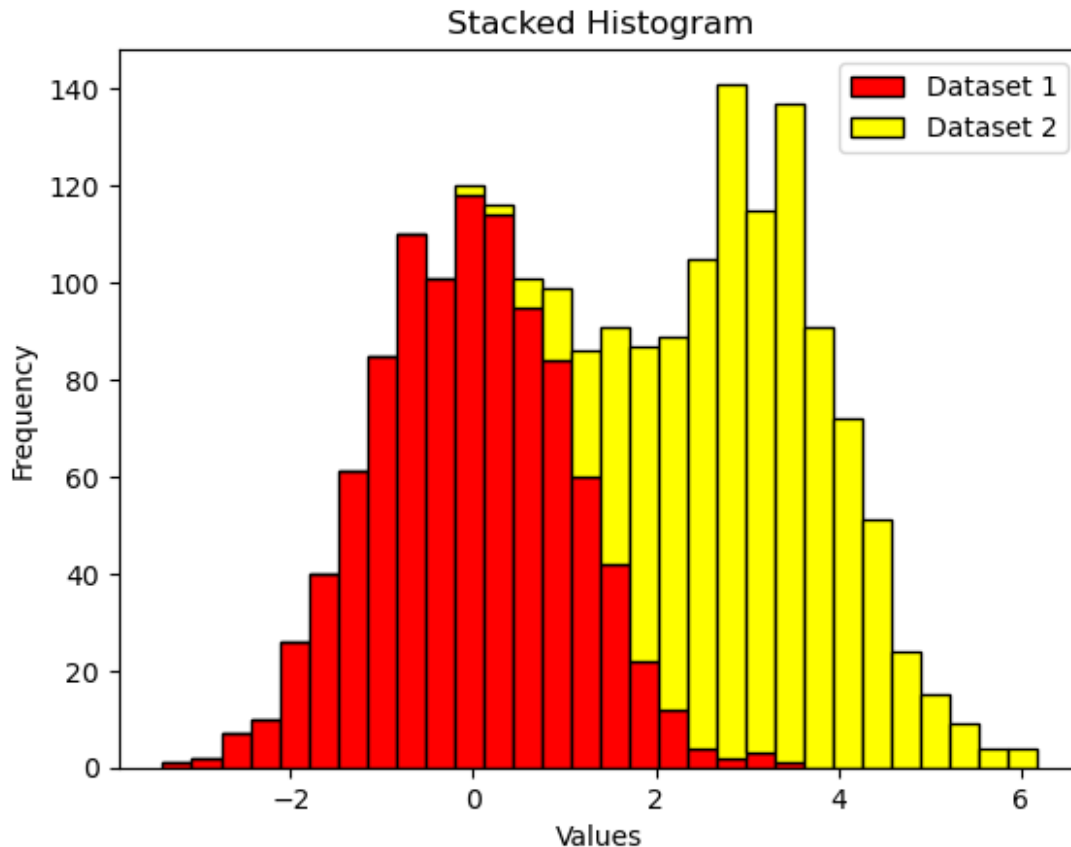
```
data1 = np.random.randn(1000)
data2 = np.random.normal(loc=3, scale=1, size=1000)

plt.hist([data1, data2], bins=30, stacked=True, color=['red',
'yellow'], edgecolor='black')

plt.xlabel('Values')
plt.ylabel('Frequency')
plt.title('Stacked Histogram')

plt.legend(['Dataset 1', 'Dataset 2'])

plt.show()
```



## Hexbin Plot

- a 2D hexbin plot using Matplotlib in Python, provides a visual representation of the 2D data distribution, where hexagons convey the density of data points. The colorbar helps interpret the density of points in different regions of the plot.

```
x = np.random.randn(1000)
y = 2 * x + np.random.normal(size=1000)

plt.hexbin(x, y, gridsize=30, cmap='Greens')

plt.xlabel('X values')
plt.ylabel('Y values')
plt.title('2D Histogram (Hexbin Plot)')

plt.colorbar()

plt.show()
```

