NUMPY

NumPy

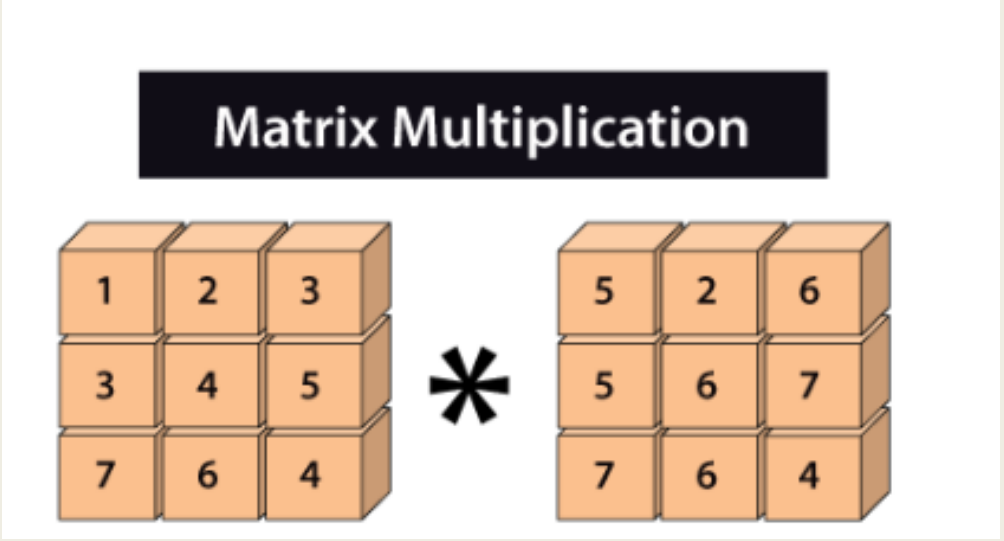
- NumPy is a Python library used for working with arrays.
- It also has functions for working in domain of linear algebra, fourier transform, and matrices.
- NumPy was created in 2005 by Travis Oliphant.
- It is an open source project and you can use it freely.

numpy.array() in Python

- The homogeneous multidimensional array is the main object of **NumPy**.
- It is basically a table of elements which are all of the same type and indexed by a tuple of positive integers.
- The dimensions are called axis in NumPy.

```
import numpy as np
arr=np.array([[1,2.,3.],[4.,5.,7]])
arr
```

```
array([[1., 2., 3.],
       [4., 5., 7.]])
```




```
In :  import numpy as np
      array1=np.array([[1,2,3],[3,4,5],[7,6,4]],ndmin=3)
      array2=np.array([[5,2,6],[5,6,7],[7,6,4]],ndmin=3)
      result=np.matmul(array1,array2)
      result
```

```
Out:  array([[[ 36,  32,  32],
              [ 70,  60,  66],
              [ 93,  74, 100]]])
```

Finding the minimum and maximum elements from the array

```
import numpy as np

a = np.array([[2,10,20],[80,43,31],[22,43,10]])

print("The original array:\n")
print(a)


print("\nThe minimum element among the array:",np.amin(a))
print("The maximum element among the array:",np.amax(a))
```

```
The original array:

[[ 2 10 20]
 [80 43 31]
 [22 43 10]]

The minimum element among the array: 2
The maximum element among the array: 80
```

NumPy Sorting

```
import numpy as np

a = np.array([[10,2,3],[4,5,6],[7,8,9]])

print("The original array:\n")
print(a)

print("Sorting along the columns:")
print(np.sort(a))

print("Sorting along the rows:")
print(np.sort(a, 0))
```

```
The original array:

[[10  2  3]
 [ 4  5  6]
 [ 7  8  9]]
Sorting along the columns:
[[ 2  3 10]
 [ 4  5  6]
 [ 7  8  9]]
Sorting along the rows:
[[ 4  2  3]
 [ 7  5  6]
 [10  8  9]]
```

NumPy
Searching

```
import numpy as np

arr = np.array([1, 2, 3, 4, 5, 4, 4])

x = np.where(arr == 4)

print(x)
```

```
(array([3, 5, 6], dtype=int64),)
```

numpy.unique() in Python

```
a=np.array([1,2,2,3,9])
a
b=np.unique(a)
b
```

```
array([1, 2, 3, 9])
```

Array Reshaping

```
import numpy as np

arr = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12])

newarr = arr.reshape(4, 3)

print(newarr)
```

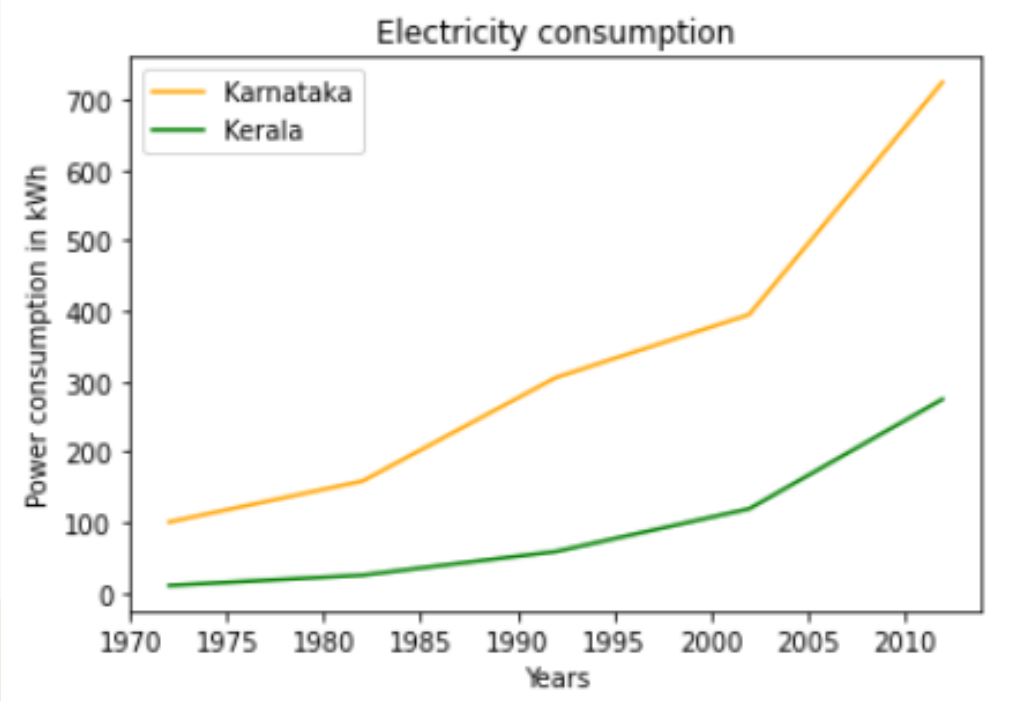
```
[[ 1  2  3]
 [ 4  5  6]
 [ 7  8  9]
 [10 11 12]]
```

```
import numpy as np

num1 = 4
num2 = 6

x = np.lcm(num1, num2)

print(x)
```

```
12
```

Linear Plot using matplotlib.pyplot

```
# Python Program to illustrate Linear Plotting
import matplotlib.pyplot as plt

year = [1972, 1982, 1992, 2002, 2012]
e_Karnataka = [100.6, 158.61, 305.54, 394.96, 724.79]
e_Kerala = [10.5, 25.21, 58.65, 119.27, 274.87]

# plotting of x-axis(year) and
# y-axis(power consumption)

plt.plot(year, e_Karnataka, color ='orange', label ='Karnataka')

plt.plot(year, e_Kerala, color ='g', label ='Kerala')

# naming of x-axis and y-axis
plt.xlabel('Years')
plt.ylabel('Power consumption in kWh')

# naming the title of the plot
plt.title('Electricity consumption')

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

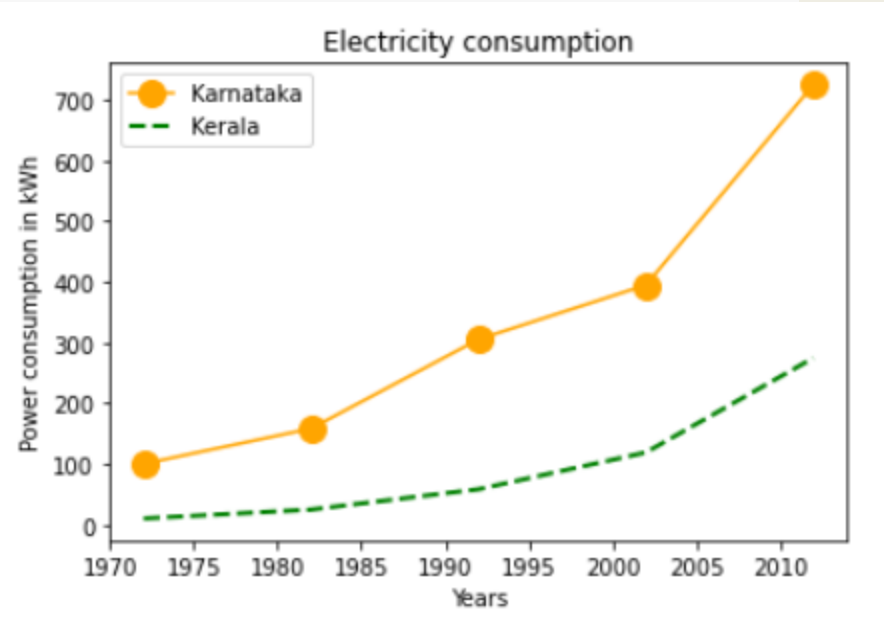


```
import numpy as np

arr = np.array([41, 42, 43, 44])

x = [True, False, True, False]

newarr = arr[x]

print(newarr)
```

```
import numpy as np

arr = np.array([41, 42, 43, 44])

x = [True, False, True, False]

newarr = arr[x]

print(newarr)
```

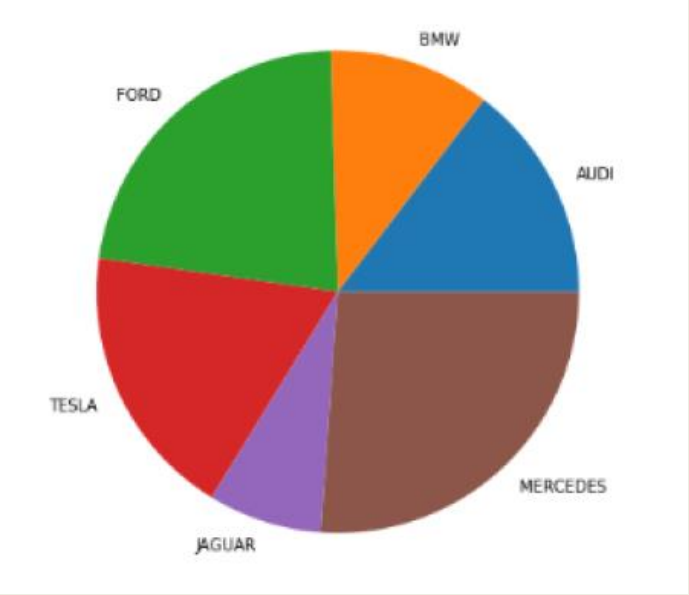
```
[41 43]
```

```python
# Python Program to illustrate Linear Plotting
import matplotlib.pyplot as plt

year = [1972, 1982, 1992, 2002, 2012]
e_Karnataka = [100.6, 158.61, 305.54, 394.96, 724.79]
e_Kerala = [10.5, 25.21, 58.65, 119.27, 274.87]

# plotting of x-axis(year) and
# y-axis(power consumption)

plt.plot(year, e_Karnataka, color ='orange',  marker ='o', markersize = 12, label ='Karnataka')

plt.plot(year, e_Kerala, color ='g',linestyle ='dashed', linewidth = 2, label ='Kerala')

# naming of x-axis and y-axis
plt.xlabel('Years')
plt.ylabel('Power consumption in kWh')

# naming the title of the plot
plt.title('Electricity consumption')

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



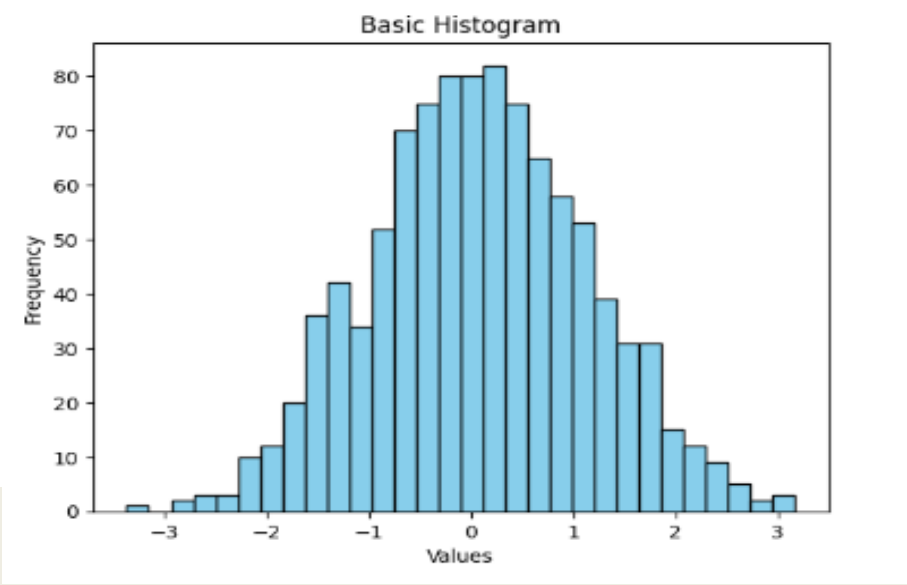
Pie Chart

```
# Import libraries
from matplotlib import pyplot as plt
import numpy as np


# Creating dataset
cars = ['AUDI', 'BMW', 'FORD',
        'TESLA', 'JAGUAR', 'MERCEDES']

data = [23, 17, 35, 29, 12, 41]

# Creating plot
fig = plt.figure(figsize=(10, 7))
plt.pie(data, labels=cars)

# show plot
plt.show()
```



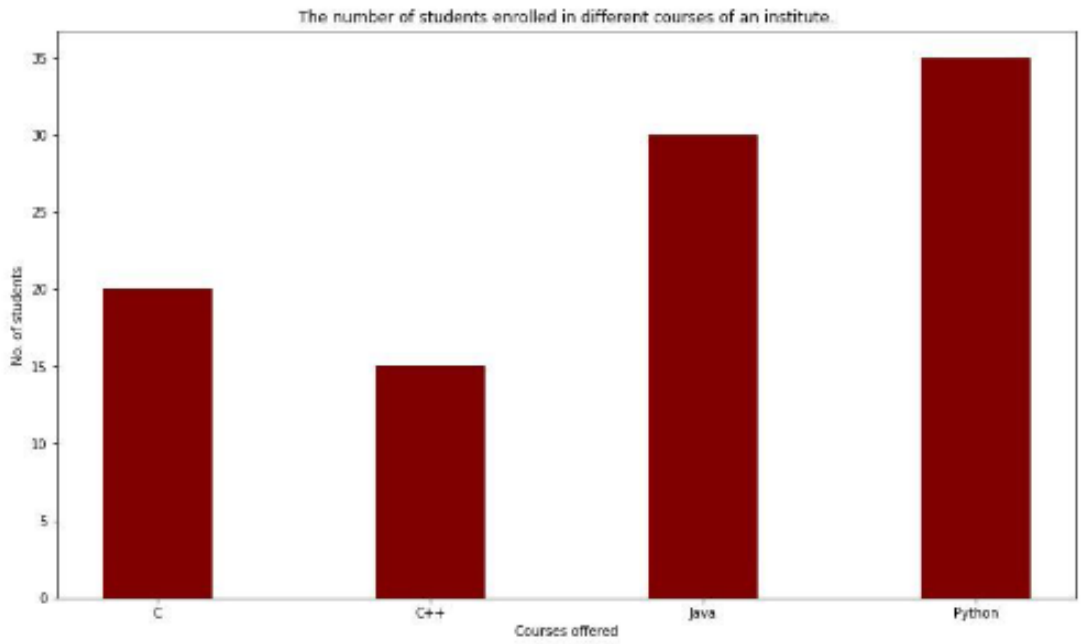
Create a Basic Histogram in Matplotlib

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

# Generate random data for the histogram
data = np.random.randn(1000)

# Plotting a basic histogram
plt.hist(data, bins=30, color='skyblue', edgecolor='black')

# Adding labels and title
plt.xlabel('Values')
plt.ylabel('Frequency')
plt.title('Basic Histogram')

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



Bar Plot using Matplotlib

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


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

fig = plt.figure(figsize = (10, 5))

# creating the bar plot
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()
```



3D Plots using Plotly Libraries

```
import numpy as np
import plotly.graph_objs as go
from plotly.offline import iplot

x = np.outer(np.linspace(-2, 2, 30), np.ones(30))
y = x.copy().T
z = np.sin(x ** 2 + y ** 2)

trace = go.Surface(x = x, y = y, z =z )
data = [trace]

layout = go.Layout(title = '3D Surface plot')

fig = go.Figure(data = data)

iplot(fig)
```