

EXPERIMENT- 3

AIM

Create various types of plots/charts like histograms, plot based on sine/cosine function based on data from a matrix. Further, label different axes in a plot and data in a plot.

SOFTWARE USED

Jupyter Platform - Python Programming Language

PROGRAM CODE

```
# Exp 3: Create various types of plots/charts like histograms, plot based on sine/cosine function based on data from a matrix. Further, label different axes in a plot and data in a plot
```

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

```
x = np.arange(-np.pi,np.pi,np.pi/40)
print(x)
```

```
mat = np.array([x, np.sin(x), np.cos(x)])
print(mat)
```

```
plt.plot(mat[0],mat[1], label="sin(x)")
plt.plot(mat[0],mat[2], label="cos(x)")
```

```
#Line Plot
```

```
plt.title("Line plot on Trigonometric Functions",fontsize='20')
plt.xlabel("Values of x"+r"$\rightarrow$")
plt.ylabel("Function values"+r"$\rightarrow$")
plt.axhline(y=0, color = "black")
plt.axvline(x=0, color = "black")
plt.legend()
plt.grid()
```

```
#Scatter Plot
```

```
plt.scatter(mat[0],mat[1], label="sin(x)")
plt.scatter(mat[0],mat[2], label="cos(x)")
```

```
plt.title("Scatter Plot on Trigonometric Functions",fontsize='20')
plt.xlabel("Values of x"+r"$\rightarrow$")
plt.ylabel("Function values"+r"$\rightarrow$")
plt.axhline(y=0, color = "black")
plt.axvline(x=0, color = "black")
plt.legend()
plt.grid()
```

```
#Box Plot
```

```
plt.boxplot([mat[0],mat[1],mat[2]], labels = ['x','sin(x)','cos(x)'])
plt.title("Box Plot",fontsize='20')
plt.xlabel("X"+r"$\rightarrow$")
```

```
plt.ylabel("Y"+r"$\rightarrow$")
plt.grid()

a = np.array(["A", "B", "C", "D"])
b = np.array([3, 8, 1, 10])
print(a)
print(b)

#Bar Graph
plt.bar(a,b, color='red')
plt.title('Bar Graph',fontsize='20')
plt.xlabel("Product Name"+r"$\rightarrow$")
plt.ylabel("Quantity"+r"$\rightarrow$")
plt.grid()

#Pie Chart
plt.figure(figsize=(6,6))
plt.pie(b, labels=a, autopct = "%1.1f%%",wedgeprops={'linewidth': 3.0, 'edgecolor': 'black'})
plt.title("Pie Chart",fontsize='20')

#Histogram
plt.hist(np.random.normal(23,19,22), color='green')
plt.title("Histogram",fontsize='20')
plt.xlabel("X"+r"$\rightarrow$")
plt.ylabel("Y"+r"$\rightarrow$")
plt.grid()
```

OUTPUT

Jupyter Experiment 3 Last Checkpoint: 6 minutes ago (autosaved) Python 3 (ipykernel)

File Edit View Insert Cell Kernel Help Trusted

Run

Exp 3: Create various types of plots/charts like histograms, plot based on sine/cosine function based on data from a matrix. Further, lable different axes in a plot and data in a plot

In [65]: `import numpy as np
import matplotlib.pyplot as plt`

In [4]: `x = np.arange(-np.pi,np.pi,np.pi/40)
print(x)`

```
[ -3.14159265 -3.06305284 -2.98451302 -2.9059732  -2.82743339 -2.74889357
 -2.67035376 -2.59181394 -2.51327412 -2.43473431 -2.35619449 -2.27765467
 -2.19911486 -2.12057504 -2.04203522 -1.96349541 -1.88495559 -1.80641578
 -1.72787596 -1.64933614 -1.57079633 -1.49225651 -1.41371669 -1.33517688
 -1.25663706 -1.17809725 -1.09955743 -1.02101761 -0.9424778  -0.86393798
 -0.78539816 -0.70685835 -0.62831853 -0.54977871 -0.4712389   -0.39269908
 -0.31415927 -0.23561945 -0.15707963 -0.07853982  0.          0.07853982
 0.15707963 0.23561945 0.31415927 0.39269908 0.4712389   0.54977871
 0.62831853 0.70685835 0.78539816 0.86393798 0.9424778  1.02101761
 1.09955743 1.17809725 1.25663706 1.33517688 1.41371669 1.49225651
 1.57079633 1.64933614 1.72787596 1.80641578 1.88495559 1.96349541
 2.04203522 2.12057504 2.19911486 2.27765467 2.35619449 2.43473431
 2.51327412 2.59181394 2.67035376 2.74889357 2.82743339 2.9059732
 2.98451302 3.06305284]
```

In [6]: `mat = np.array([x, np.sin(x), np.cos(x)])
print(mat)`

```
In [6]: mat = np.array([x, np.sin(x), np.cos(x)])
print(mat)
```

```

-9.51056516e-01 -9.23879533e-01 -8.91006524e-01 -8.52640164e-01
-8.09016994e-01 -7.60405966e-01 -7.07106781e-01 -6.49448048e-01
-5.87785252e-01 -5.22498565e-01 -4.53990500e-01 -3.82683432e-01
-3.09016994e-01 -2.33445364e-01 -1.56434465e-01 -7.84590957e-02
 6.12323400e-17  7.84590957e-02  1.56434465e-01  2.33445364e-01
 3.09016994e-01  3.82683432e-01  4.53990500e-01  5.22498565e-01
 5.87785252e-01  6.49448048e-01  7.07106781e-01  7.60405966e-01
 8.09016994e-01  8.52640164e-01  8.91006524e-01  9.23879533e-01
 9.51056516e-01  9.72369920e-01  9.87688341e-01  9.96917334e-01
 1.00000000e+00  9.96917334e-01  9.87688341e-01  9.72369920e-01
 9.51056516e-01  9.23879533e-01  8.91006524e-01  8.52640164e-01
 8.09016994e-01  7.60405966e-01  7.07106781e-01  6.49448048e-01
 5.87785252e-01  5.22498565e-01  4.53990500e-01  3.82683432e-01
 3.09016994e-01  2.33445364e-01  1.56434465e-01  7.84590957e-02
 6.12323400e-17 -7.84590957e-02 -1.56434465e-01 -2.33445364e-01
-3.09016994e-01 -3.82683432e-01 -4.53990500e-01 -5.22498565e-01
-5.87785252e-01 -6.49448048e-01 -7.07106781e-01 -7.60405966e-01
-8.09016994e-01 -8.52640164e-01 -8.91006524e-01 -9.23879533e-01
-9.51056516e-01 -9.72369920e-01 -9.87688341e-01 -9.96917334e-01]]

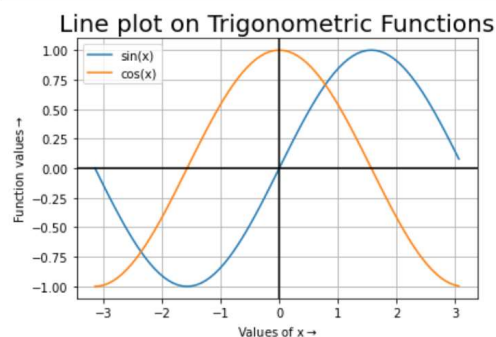
```

```
In [68]: plt.plot(mat[0],mat[1], label="sin(x)")
plt.plot(mat[0],mat[2], label="cos(x)")

plt.title("Line plot on Trigonometric Functions",fontsize='20')
plt.xlabel("Values of x"+r"$\rightarrow$")
```

```
In [68]: plt.plot(mat[0],mat[1], label="sin(x)")
plt.plot(mat[0],mat[2], label="cos(x)")

plt.title("Line plot on Trigonometric Functions",fontsize='20')
plt.xlabel("Values of x"+r"$\rightarrow$")
plt.ylabel("Function values"+r"$\rightarrow$")
plt.axhline(y=0, color = "black")
plt.axvline(x=0, color = "black")
plt.legend()
plt.grid()
```



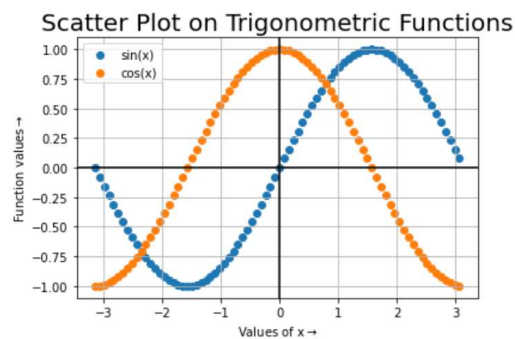
File Edit View Insert Cell Kernel Help

Trusted Python 3 (ipykernel)

Run Stop Restart Clear All

```
In [86]: plt.scatter(mat[0],mat[1], label="sin(x)")
plt.scatter(mat[0],mat[2], label="cos(x)")

plt.title("Scatter Plot on Trigonometric Functions",fontsize='20')
plt.xlabel("Values of x"+r"$\rightarrow$")
plt.ylabel("Function values"+r"$\rightarrow$")
plt.axhline(y=0, color = "black")
plt.axvline(x=0, color = "black")
plt.legend()
plt.grid()
```

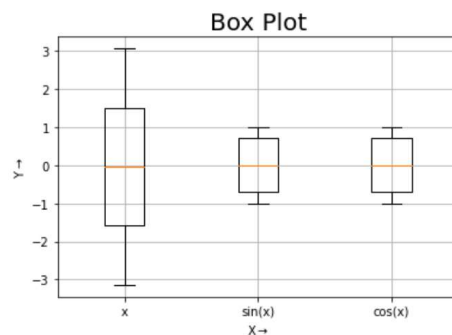


File Edit View Insert Cell Kernel Help

Trusted Python 3 (ipykernel)

Run Stop Restart Clear All

```
In [87]: plt.boxplot([mat[0],mat[1],mat[2]], labels = ['x','sin(x)','cos(x)'])
plt.title("Box Plot",fontsize='20')
plt.xlabel("X"+r"$\rightarrow$")
plt.ylabel("Y"+r"$\rightarrow$")
plt.grid()
```



```
In [11]: a = np.array(["A", "B", "C", "D"])
b = np.array([3, 8, 1, 10])
print(a)
print(b)
```

File Edit View Insert Cell Kernel Help

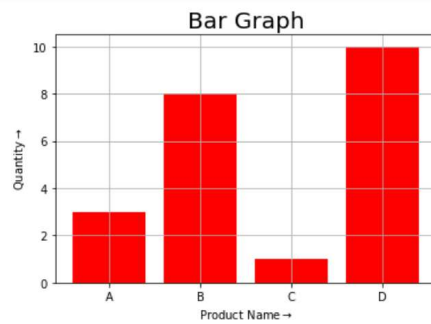
Trusted Python 3 (ipykernel)

Run Stop Restart Clear Output

```
In [11]: a = np.array(["A", "B", "C", "D"])
b = np.array([3, 8, 1, 10])
print(a)
print(b)
```

```
['A' 'B' 'C' 'D']
[ 3  8  1 10]
```

```
In [67]: plt.bar(a,b, color='red')
plt.title('Bar Graph',fontsize='20')
plt.xlabel("Product Name"+r"$\rightarrow$")
plt.ylabel("Quantity"+r"$\rightarrow$")
plt.grid()
```



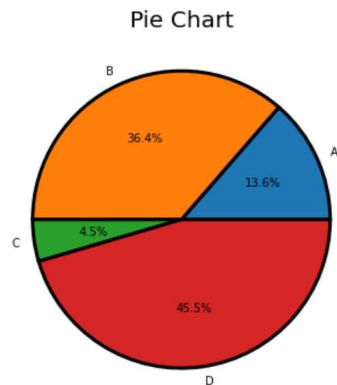
File Edit View Insert Cell Kernel Help

Trusted Python 3 (ipykernel)

Run Stop Restart Clear Output

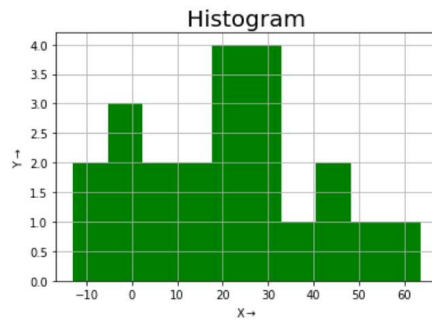
```
In [64]: plt.figure(figsize=(6,6))
plt.pie(b, labels=a, autopct = "%1.1f%%", wedgeprops={'linewidth': 3.0, 'edgecolor': 'black'})
plt.title("Pie Chart",fontsize='20')
```

```
Out[64]: Text(0.5, 1.0, 'Pie Chart')
```



```
In [70]: plt.hist(np.random.normal(23,19,22), color='green')
plt.title("Histogram",fontsize='20')
plt.xlabel("X"+r"$\rightarrow$")
```

```
In [70]: plt.hist(np.random.normal(23,19,22), color='green')
plt.title("Histogram",fontsize='20')
plt.xlabel("X"+r"$\rightarrow$")
plt.ylabel("Y"+r"$\rightarrow$")
plt.grid()
```



DISCUSSION and CONCLUSION

The various types of plots/charts like histograms, plot based on sine/cosine function based on data from a matrix, have been studied and created in the Jupyter platform. Further, labelling of different axes in a plot and data in a plot, have been performed.

CRITERIA	TOTAL MARKS	MARKS OBTAINED	COMMENTS
Concept (A)	2		
Implementation (B)	2		
Performance (C)	2		
Total	6		