1.Demonstrate three different methods for creating identical 2D arrays in NumPy❱ Provide the code for each method and the final output after each methods

```
#1.Using numpy.zeros

import numpy as np

array1 = np.zeros((3, 4))
print(array1)
```

```
[[0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]]
```

```
#2.Using numpy.ones

array2 = np.ones((3, 4))
print(array2)
```

```
[[1. 1. 1. 1.]
 [1. 1. 1. 1.]
 [1. 1. 1. 1.]]
```

```
#3.Using numpy.full

array3 = np.full((3, 4), 2)

print(array3)
```

```
[[2 2 2 2]
 [2 2 2 2]
 [2 2 2 2]]
```

2.Using the numpy function generate an array of 100 evenly spaced numbers between 1 and 10 and reshape that 1d array into 2d array

```
import numpy as np


array1d = np.linspace(1, 10, 100)


array2d = array1d.reshape(-1, 10)
```

```
print('1DArray',array1d)
print('2DArray',array2d)
```

3.Explain the following terms.

1.The difference in np.array, np.asarray and np.asanyarray

```
# 2. The difference between Deep copy and shallow copy
```

4.generate a 3x3 array with random floating point number between 5 and 20 then roundeach number in the array to 2 decimal places.

```
import numpy as np


array = np.random.uniform(5, 20, size=(3, 3))

rarray = np.round(array, decimals=2)

print("Original Array")
print(array)
print("Rounded Array")
print(rarray)
```

```
Original Array
[[17.422127    6.60145432 19.64891644]
 [16.74510276 19.51294778 11.43220879]
 [ 6.16124531  9.98963659 18.59252679]]
Rounded Array
[[17.42  6.6  19.65]
 [16.75 19.51 11.43]
 [ 6.16  9.99 18.59]]
```

5.Certainly! Here's how you can create a NumPy array with random integers between 1 and 10 of shape (5, 6), and then perform the specified operation:

```
array = np.random.randint(1, 11, size=(5, 6))
print(array)


evens = array[array % 2 == 0]
odds = array[array % 2 != 0]
print(evens)
print(odds)
```

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

6.create a 3d numpy array of shape (3,3,3) containing random integers between 1 and 10 and perform the following operation 1.find the indices of maximum valuesalong each depth level(third axis). 2.perform element wise multiplication between both array.

```
array = np.random.randint(1, 11, size=(3, 3, 3))

# 1. Indices of maximum values along each depth level (third axis)
max_indices = np.argmax(array, axis=2)
print(max_indices)


# 2. Element-wise multiplication between both array
multiplied_array = array * max_indices[:, :, np.newaxis]
print(max_indices)
```

```
[[2 2 0]
 [1 0 1]
 [0 0 1]]
[[2 2 0]
 [1 0 1]
 [0 0 1]]
```

8.Perform the following tasks using the peoples dataset.

```
# a) Read the 'data.csv' file using pandas, skipping the first 50
rows.
import pandas as pd

df = pd.read_csv('data.csv',skiprows=50)
df

# b) Only read the columns: 'Last Name', ‘Gender’,’Email’,‘Phone’ and
‘Salary’ from the file.
data = pd.read_csv('data.csv', usecols=['Gender','Email','Phone',
'Salary',])
data.head()
```

```
   Gender                             Email                Phone  Salary
0    Male            pwarner@example.org        857.139.8239   90000
1  Female  fergusonkatherine@example.net                 NaN   80000
2  Female            fhoward@example.org       (599)782-0605   50000
3    Male          zjohnston@example.com                 NaN   65000
4  Female               elin@example.net  (390)417-1635x3010  100000
```

```
# c) Display the first 10 rows of the filtered dataset.
data.head(10)
```

```
   Gender                             Email                Phone  Salary
0    Male            pwarner@example.org        857.139.8239   90000
1  Female  fergusonkatherine@example.net                 NaN   80000
2  Female            fhoward@example.org       (599)782-0605   50000
3    Male          zjohnston@example.com                 NaN   65000
4  Female               elin@example.net  (390)417-1635x3010  100000
5    Male          kaitlin13@example.net          8537800927   50000
6    Male         jeffharvey@example.com   093.655.7480x7895   60000
7    Male           alicia33@example.org          4709522945   65000
8    Male             jake50@example.com        013.820.4758   50000
9    Male      lanechristina@example.net  (560)903-5068x4985   50000
```

```
# d) Extract the ‘Salary’' column as a Series and display its last 5
values.

salary = data['Salary']
salary.tail()
```

```
995     90000
996     50000
997     60000
998    100000
999     90000
Name: Salary, dtype: int64
```

9.Filter and select rows from the People_Dataset, where the “Last Name' column contains the name 'Duke', 'Gender' column contains the word Female and ‘salary’ should Pe less than 85000

```
df = pd.read_csv('data.csv')

filtered_data = df[(df['Last Name'].str.contains('Duke')) &
                                (df['Gender'] == 'Female') &
                                (df['Salary'] < 85000)]

filtered_data
```

```
     Index          User Id First Name Last Name  Gender  \
45      46  99A502C175C4EBd     Olivia      Duke  Female
210    211  DF17975CC0a0373    Katrina      Duke  Female
457    458  dcE1B7DE83c1076      Traci      Duke  Female
```

```
729    730  c9b482D7aa3e682     Lonnie      Duke  Female

                         Email                    Phone Date of birth  \
45         diana26@example.net  001-366-475-8607x04350    13-10-1934
210        robin78@example.com            740.434.0212    21-09-1935
457  perryhoffman@example.org     +1-903-596-0995x489    11-02-1997
729   kevinkramer@example.net             982.692.6257    12-05-2015

           Job Title  Salary
45           Dentist   60000
210  Producer, radio   50000
457        Herbalist   50000
729     Nurse, adult   70000
```

10.Create a 7*5 Dataframe in pandas using a series generated from 35 random integers between 1 and 6

```
import numpy as np
import pandas as pd

random_data = np.random.randint(1, 6 + 1, size=(7, 5))
df = pd.DataFrame(random_data)
df
```

```
   0  1  2  3  4
0  1  3  1  6  3
1  1  2  4  4  3
2  2  6  4  5  5
3  4  1  5  1  6
4  1  4  2  3  6
5  5  1  6  2  6
6  3  2  5  6  2
```

11.Create a two diffrent series, each of length 50, with the following criteria.

```
import pandas as pd
import numpy as np

# a) The first Series should contain random numbers ranging from 10 to
50.

series1 = pd.Series(np.random.randint(10, 51, size=50))

# b) The second Series should contain random numbers ranging from 100
to 1000.
series2 = pd.Series(np.random.randint(100, 1001, size=50))
```

```python
# c) Create a DataFrame by 'oining these Series by column, and, change the names of the columns to 'col1', 'col2',
# etc.
df = pd.DataFrame({'col1': series1, 'col2': series2})

print(df)
```

```
    col1  col2
0     38   960
1     47   402
2     42   693
3     37   514
4     10   258
5     33   400
6     17   629
7     37   420
8     36   128
9     14   655
10    16   266
11    26   880
12    50   963
13    30   958
14    40   834
15    44   426
16    24   516
17    30   354
18    18   990
19    15   634
20    29   874
21    44   520
22    45   562
23    38   953
24    22   403
25    12   324
26    46   563
27    28   901
28    17   205
29    46   955
30    26   495
31    42   579
32    22   568
33    16   823
34    10   720
35    37   425
36    45   375
37    24   396
38    34   551
39    26   429
40    24   299
```

```
41    41   167
42    31   897
43    30   561
44    46   579
45    43   630
46    42   469
47    39   864
48    23   505
49    14   124
```

12.Perform the following operations using peoples dataset.

```
# a) Delete the 'Email', 'Phone', and 'Date of birth' columns from the
dataset.

df = pd.read_csv('data.csv')
df2 = df.drop(columns=['Email','Phone','Date of birth'])
df2
```

```
     Index          User Id First Name Last Name  Gender  \
0        1  8717bbf45cCDbEe     Shelia   Mahoney    Male
1        2  3d5AD30A4cD38ed         Jo    Rivers  Female
2        3  810Ce0F276Badec     Sheryl    Lowery  Female
3        4  BF2a889C00f0cE1    Whitney    Hooper    Male
4        5  9afFEafAe1CBBB9    Lindsey      Rice  Female
..     ...              ...        ...       ...     ...
995    996  fedF4c7Fd9e7cFa       Kurt    Bryant  Female
996    997  ECddaFEDdEc4FAB      Donna     Barry  Female
997    998  2adde51d8B8979E      Cathy  Mckinney  Female
998    999  Fb2FE369D1E171A   Jermaine    Phelps    Male
999   1000  8b756f6231DDC6e        Lee      Tran  Female

                           Job Title  Salary
0                  Probation officer   90000
1                            Dancer   80000
2                              Copy   50000
3           Counselling psychologist   65000
4                Biomedical engineer  100000
..                               ...     ...
995                Personnel officer   90000
996          Education administrator   50000
997  Commercial/residential surveyor   60000
998                 Ambulance person  100000
999       Nurse, learning disability   90000

[1000 rows x 7 columns]
```

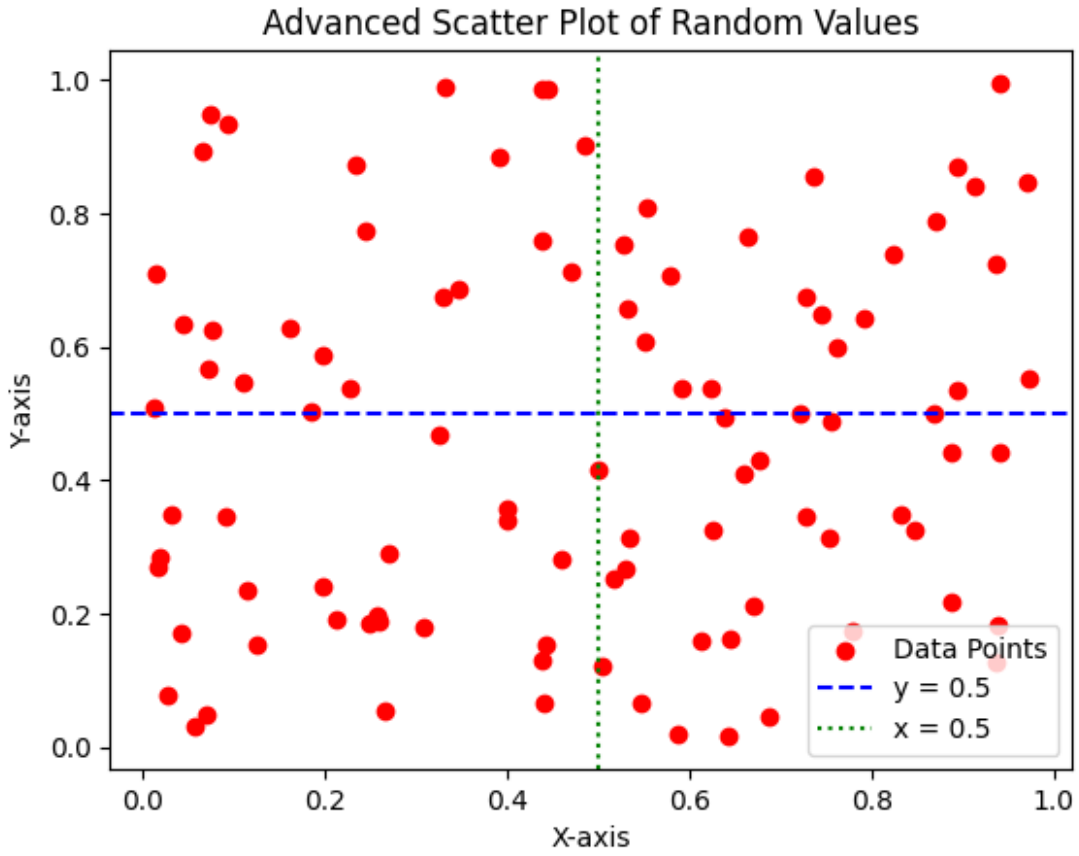
```
# b) Delete the rows containing any missing values.

df3 = df2.dropna()

# c.Print the final output.
# df3.isnull().sum()
df3
```

```
     Index          User Id First Name Last Name  Gender  \
0        1  8717bbf45cCDbEe     Shelia   Mahoney    Male
1        2  3d5AD30A4cD38ed         Jo    Rivers  Female
2        3  810Ce0F276Badec     Sheryl    Lowery  Female
3        4  BF2a889C00f0cE1    Whitney    Hooper    Male
4        5  9afFEafAe1CBBB9    Lindsey      Rice  Female
..     ...              ...        ...       ...     ...
995    996  fedF4c7Fd9e7cFa       Kurt    Bryant  Female
996    997  ECddaFEDdEc4FAB      Donna     Barry  Female
997    998  2adde51d8B8979E      Cathy  Mckinney  Female
998    999  Fb2FE369D1E171A   Jermaine    Phelps    Male
999   1000  8b756f6231DDC6e        Lee      Tran  Female

                           Job Title  Salary
0                  Probation officer   90000
1                            Dancer   80000
2                              Copy   50000
3           Counselling psychologist   65000
4                Biomedical engineer  100000
..                               ...     ...
995                Personnel officer   90000
996          Education administrator   50000
997  Commercial/residential surveyor   60000
998                 Ambulance person  100000
999       Nurse, learning disability   90000

[1000 rows x 7 columns]
```

1. Create two numpy arrays x and y each containing 100 random float values between 0 and 1Perform the following tasks using matplotlib and numpy.

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

# Generate two numpy arrays each containing 100 random float values
between 0 and 1
x = np.random.rand(100)
y = np.random.rand(100)

# 1.Create a scatter plot of x and y with red points and 'o' marker
style
plt.scatter(x, y, color='red', marker='o', label='Data Points')
```

```
# 2.Add a horizontal line at y = 0.5 with dashed line style and label
plt.axhline(y=0.5, color='blue', linestyle='--', label='y = 0.5')

# 3.Add a vertical line at x = 0.5 with dotted line style and label
plt.axvline(x=0.5, color='green', linestyle=':', label='x = 0.5')

# 4.Label the x-axis as 'X-axis' and the y-axis as 'Y-axis'
plt.xlabel('X-axis')
plt.ylabel('Y-axis')

# 5.Set the title of the plot as 'Advanced Scatter Plot of Random
Values'
plt.title('Advanced Scatter Plot of Random Values')

# 6.Display a legend for the scatter plot, the horizontal line, and
the vertical line
plt.legend()

# 7.Show the plot
plt.show()
```



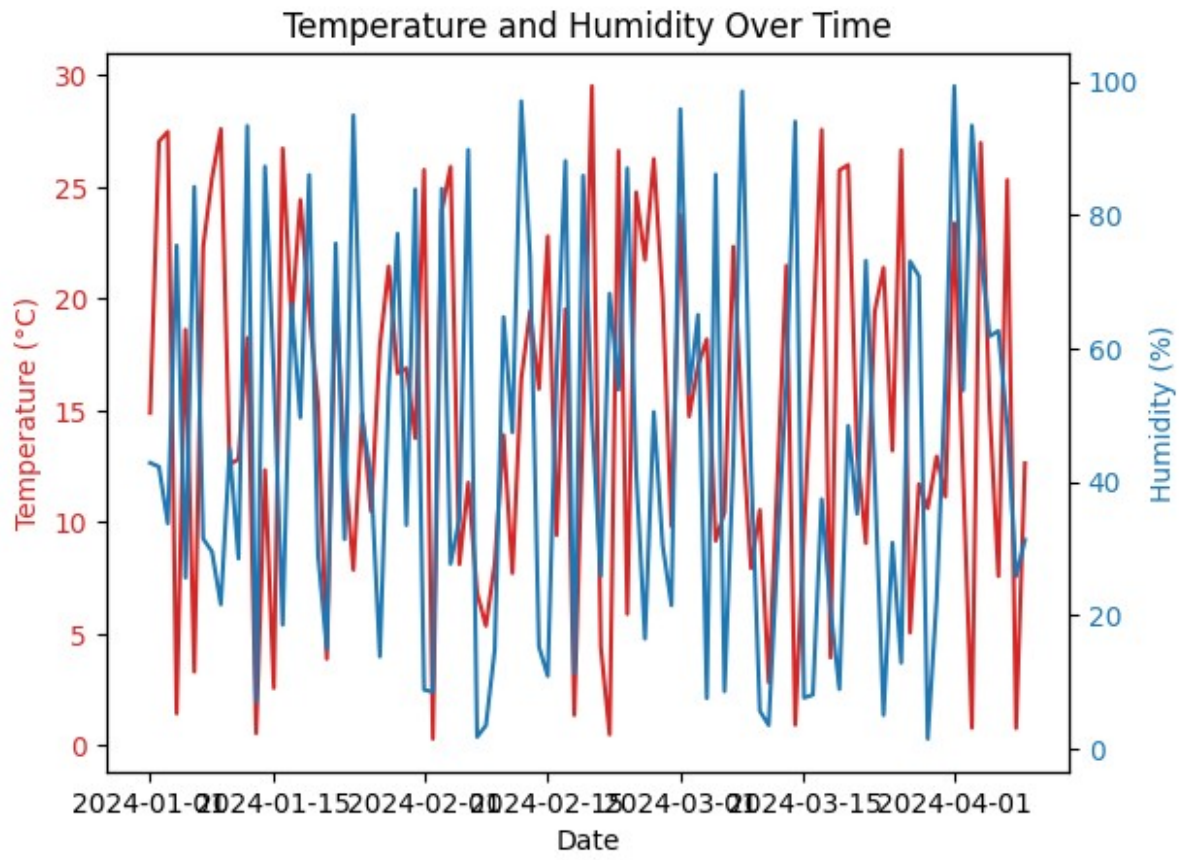
14.Create a time series dataset in a pandas Dataframe with columns : 'Date', 'Temperature', 'Humidity' and Perform the following tasks using Matplotlib

a) Plot the 'Temprature' and 'Humidity' on the same plot with different y-axes (left y-axis for 'Temprature' and right y-axis for 'Humidity').

b) Label the x-axis as 'Date'.

c) Set the title of the pot as 'Temprature and Humidity Over Time

```
import pandas as pd
import matplotlib.pyplot as plt

# Create a time series dataset in a pandas DataFrame
data = {
    'Date': pd.date_range(start='2024-01-01', periods=100),
    'Temperature': np.random.rand(100) * 30,
    'Humidity': np.random.rand(100) * 100,   }
df = pd.DataFrame(data)

# Plot the 'Temperature' and 'Humidity' on the same plot with
different y-axes
fig, ax1 = plt.subplots()

color = 'tab:red'
ax1.set_xlabel('Date')
ax1.set_ylabel('Temperature (°C)', color=color)
ax1.plot(df['Date'], df['Temperature'], color=color)
ax1.tick_params(axis='y', labelcolor=color)

ax2 = ax1.twinx()
color = 'tab:blue'
ax2.set_ylabel('Humidity (%)', color=color)
ax2.plot(df['Date'], df['Humidity'], color=color)
ax2.tick_params(axis='y', labelcolor=color)

plt.title('Temperature and Humidity Over Time')
plt.show()
```



1. Create a NumPy array data containing 1000 samples from a normal distribution. Perform the following tasks using Matplotlib:

a) Plot a histogram of the data with 30 bins.

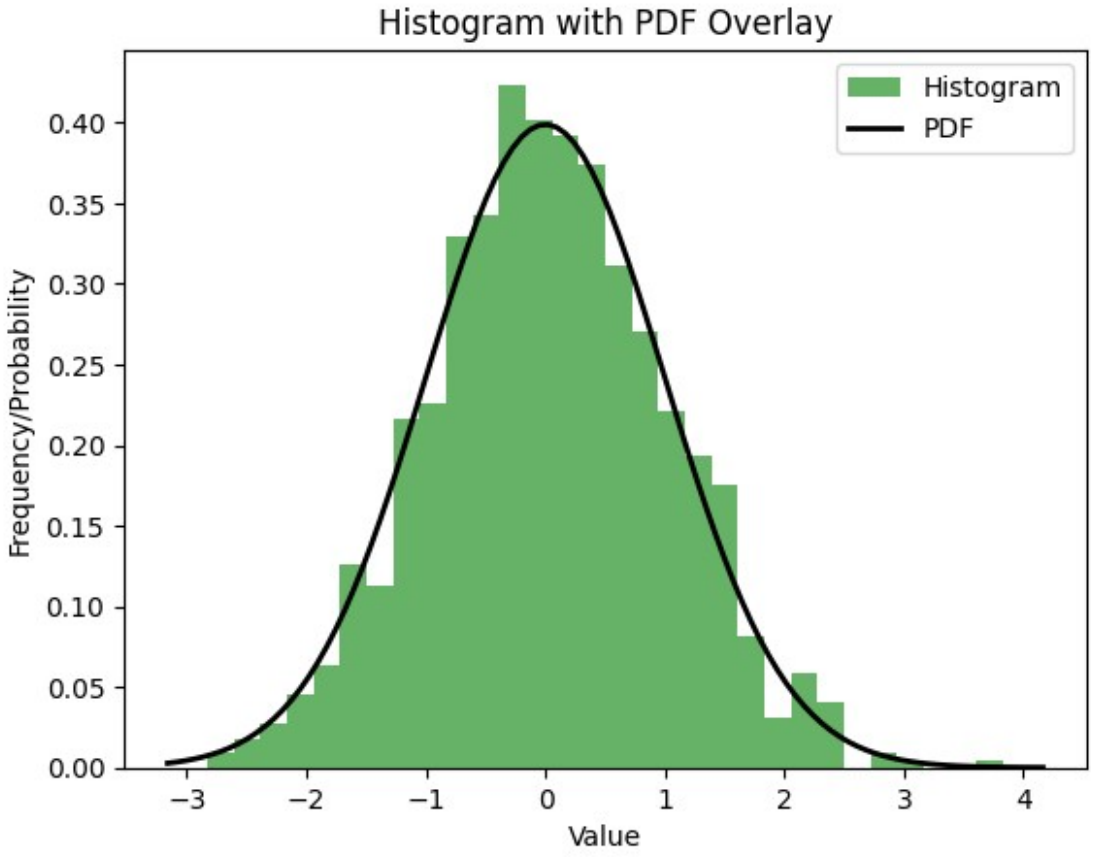
b) Overlay a line plot representing the normal distribution's probability density function (PDF).

c) Label the x-axis as 'Value' and the y-axis as 'Frequency/Probability'.

d) Set the title of the plot as 'Histogram with PDF Overlay'.

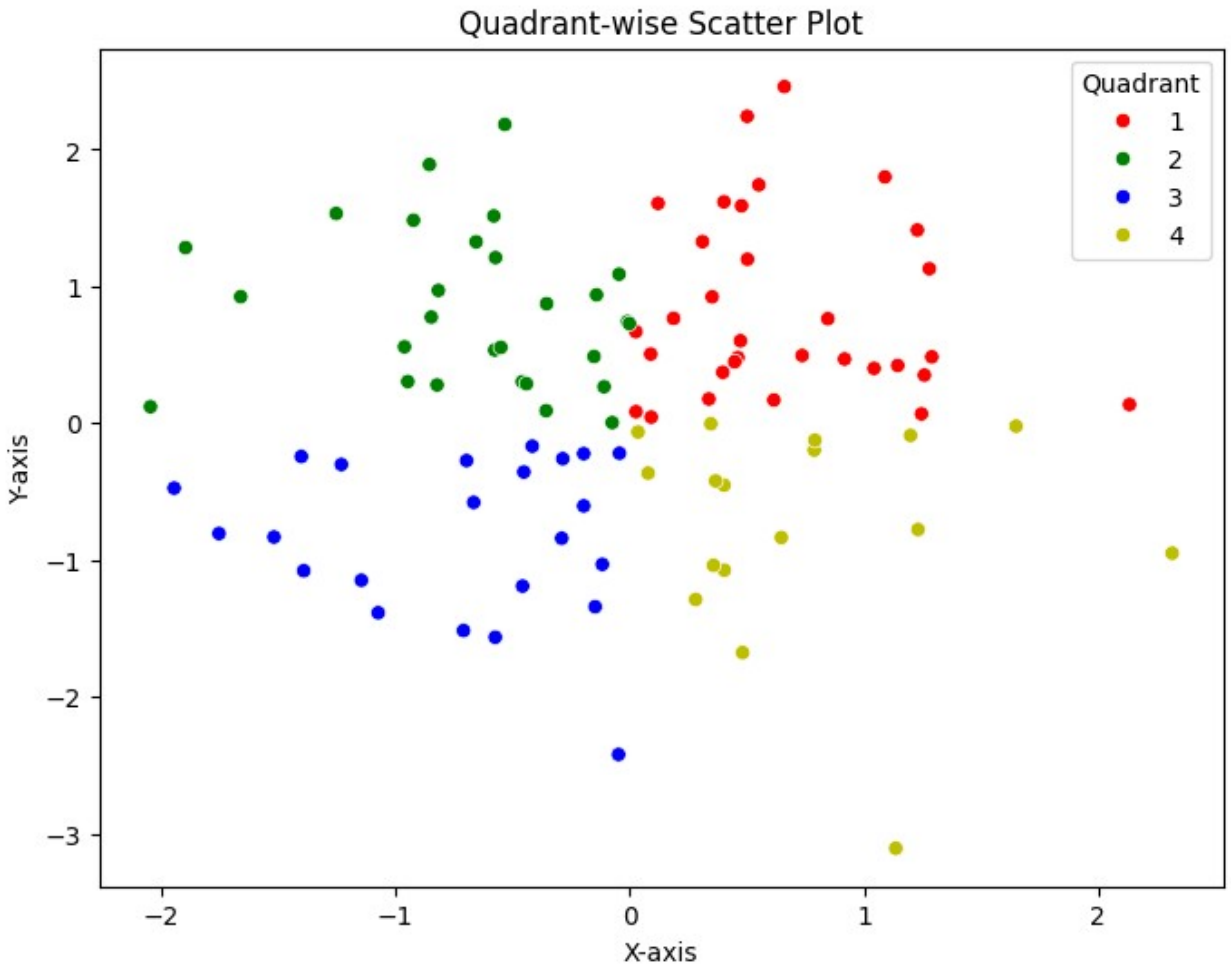
```
import numpy as np
import matplotlib.pyplot as plt

# Create a NumPy array containing 1000 samples from a normal
distribution
data = np.random.normal(size=1000)

# Plot a histogram of the data with 30 bins
plt.hist(data, bins=30, density=True, alpha=0.6, color='g',
label='Histogram')

# Overlay a line plot representing the normal distribution's
probability density function (PDF)
```

```
xmin, xmax = plt.xlim()
x = np.linspace(xmin, xmax, 100)
p = np.exp(-0.5 * x**2) / np.sqrt(2 * np.pi)
plt.plot(x, p, 'k', linewidth=2, label='PDF')

# Label the x-axis as 'Value' and the y-axis as
'Frequency/Probability'
plt.xlabel('Value')
plt.ylabel('Frequency/Probability')

# Set the title of the plot as 'Histogram with PDF Overlay'
plt.title('Histogram with PDF Overlay')

plt.legend()

plt.show()
```



16.Create a Seaborn scatter plot of two random arrays, color points based on their position relative to the origin (quadrants), add a legend, label the axes, and set the title as 'Quadrant-wise Scatter Plot'.

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

x = np.random.randn(100)
y = np.random.randn(100)

# Determine the quadrant for each point
quadrants = np.zeros_like(x, dtype=int)
quadrants[(x > 0) & (y > 0)] = 1
quadrants[(x < 0) & (y > 0)] = 2

quadrants[(x < 0) & (y < 0)] = 3

quadrants[(x > 0) & (y < 0)] = 4

palette = {1: 'r', 2: 'g', 3: 'b', 4: 'y'}

colors = [palette[q] for q in quadrants]

plt.figure(figsize=(8, 6))
sns.scatterplot(x=x, y=y, hue=quadrants, palette=palette,
legend='full')



plt.xlabel('X-axis')
plt.ylabel('Y-axis')

plt.title('Quadrant-wise Scatter Plot')

plt.legend(title='Quadrant')
plt.show()
```



1. With Bokhe, plot a line chart of a sine wave function, add grid lines, label the axes, and set the title as 'Sine Wave Function.

```
from bokeh.plotting import figure, show
import numpy as np

x = np.linspace(0, 2*np.pi, 100)
y = np.sin(x)


p = figure(title="Sine Wave Function", x_axis_label='x',
y_axislabel='sin(x)')


p.line(x, y, legend_label="sin(x)", line_width=2)


p.grid.grid_line_alpha = 0.3
```

```
show(p)
```

1. Using Plotly, create a basic line plot of a randomly generated dataset, label the axes, and set the title as 'Simple Line Plot'8

```
import plotly.graph_objects as go
import numpy as np

x = np.linspace(0, 10, 100)
y = np.random.rand(100)


fig = go.Figure(data=go.Scatter(x=x, y=y, mode='lines'))


fig.update_layout(title='Simple Line Plot', xaxis_title='X-axis',
yaxis_title='Y-axis')


fig.show()
```

```
---------------------------------------------------------------------------
-----
ValueError                                Traceback (most recent call
last)
Cell In[4], line 14
     11 fig.update_layout(title='Simple Line Plot', xaxis_title='X-
axis', yaxis_title='Y-axis')
     13 # Show the plot
---> 14 fig.show()

File
~/.local/lib/python3.8/site-packages/plotly/basedatatypes.py:3410, in
BaseFigure.show(self, *args, **kwargs)
   3377 """
   3378 Show a figure using either the default renderer(s) or the
renderer(s)
   3379 specified by the renderer argument
   (...)
   3406 None
   3407 """
   3408 import plotly.io as pio
-> 3410 return pio.show(self, *args, **kwargs)

File ~/.local/lib/python3.8/site-packages/plotly/io/_renderers.py:394,
in show(fig, renderer, validate, **kwargs)
    389         raise ValueError(
    390             "Mime type rendering requires ipython but it is
```

```
not installed"
    391         )
    393     if not nbformat or Version(nbformat.__version__) <
Version("4.2.0"):
--> 394         raise ValueError(
    395             "Mime type rendering requires nbformat>=4.2.0 but
it is not installed"
    396         )
    398     ipython_display.display(bundle, raw=True)
    400 # external renderers

ValueError: Mime type rendering requires nbformat>=4.2.0 but it is not
installed
```

1. Using Bo+eh, generate a bar chart of randomly generated categorical data, color bars based on their values, add hover tooltips to display exact values, label the axes, and set the title as 'Random Categorical Bar Chart.

```
```

1. Using Plotly, create an interactive pie chart of randomly generated data, add labels and percentages, set the title as 'Interactive Pie Chart'.

```
import plotly.express as px
import numpy as np
import pandas as pd

categories = ['Category A', 'Category B', 'Category C', 'Category D']
values = np.random.randint(1, 100, size=len(categories))

data = {'Categories': categories, 'Values': values}
df = pd.DataFrame(data)

fig = px.pie(df, values='Values', names='Categories',
title='Interactive Pie Chart',
             labels={'Categories': 'Category', 'Values': 'Value'})

fig.update_traces(textposition='inside', textinfo='percent+label')

fig.show()
```


```
{"config":{"plotlyServerURL":"https://plot.ly"},"data":[{"domain":
{"x":[0,1],"y":[0,1]},"hovertemplate":"Category=%{label}<br>Value=%
{value}<extra></extra>","labels":["Category A","Category B","Category
C","Category
D"],"legendgroup":"","name":"","showlegend":true,"textinfo":"percent+l
abel","textposition":"inside","type":"pie","values":
[59,32,55,48]}],"layout":{"legend":{"tracegroupgap":0},"template":
{"data":{"bar":[{"error_x":{"color":"#2a3f5f"},"error_y":
{"color":"#2a3f5f"},"marker":{"line":
{"color":"#E5ECF6","width":0.5},"pattern":
```

{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"bar"}],"barpolar":[{"marker":{"line":{"color":"#E5ECF6","width":0.5},"pattern":{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"barpolar"}],"carpet":[{"aaxis":{"endlinecolor":"#2a3f5f","gridcolor":"white","linecolor":"white","minorgridcolor":"white","startlinecolor":"#2a3f5f"},"baxis":{"endlinecolor":"#2a3f5f","gridcolor":"white","linecolor":"white","minorgridcolor":"white","startlinecolor":"#2a3f5f"},"type":"carpet"}],"choropleth":[{"colorbar":{"outlinewidth":0,"ticks":""},"type":"choropleth"}],"contour":[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":[[0,"#0d0887"],[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],[0.8888888888888888,"#fdca26"],[1,"#f0f921"]],"type":"contour"}],"contourcarpet":[{"colorbar":{"outlinewidth":0,"ticks":""},"type":"contourcarpet"}],"heatmap":[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":[[0,"#0d0887"],[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],[0.8888888888888888,"#fdca26"],[1,"#f0f921"]],"type":"heatmap"}],"heatmapgl":[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":[[0,"#0d0887"],[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],[0.8888888888888888,"#fdca26"],[1,"#f0f921"]],"type":"heatmapgl"}],"histogram":[{"marker":{"pattern":{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"histogram"}],"histogram2d":[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":[[0,"#0d0887"],[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],[0.8888888888888888,"#fdca26"],[1,"#f0f921"]],"type":"histogram2d"}],"histogram2dcontour":[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":[[0,"#0d0887"],[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],[0.8888888888888888,"#fdca26"],[1,"#f0f921"]],"type":"histogram2dcontour"}],"mesh3d":[{"colorbar":{"outlinewidth":0,"ticks":""},"type":"mesh3d"}],"parcoords":[{"line":{"colorbar":{"outlinewidth":0,"ticks":""}},"type":"parcoords"}],"pie":[{"automargin":true,"type":"pie"}],"scatter":[{"fillpattern":

{"fillmode":"overlay","size":10,"solidity":0.2},"type":"scatter"}],"sc
atter3d":[{"line":{"colorbar":{"outlinewidth":0,"ticks":""}},"marker":
{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatter3d"}],"scattercarpet":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattercarpet"}],"scattergeo":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattergeo"}],"scattergl":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattergl"}],"scattermapbox":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattermapbox"}],"scatterpolar"
:[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatterpolar"}],"scatterpolargl
":[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatterpolargl"}],"scatterterna
ry":[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatterternary"}],"surface":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"surface"}],"table":[{"cells":{"fill":
{"color":"#EBF0F8"},"line":{"color":"white"}},"header":{"fill":
{"color":"#C8D4E3"},"line":
{"color":"white"}},"type":"table"}]},"layout":{"annotationdefaults":
{"arrowcolor":"#2a3f5f","arrowhead":0,"arrowwidth":1},"autotypenumbers
":"strict","coloraxis":{"colorbar":
{"outlinewidth":0,"ticks":""}},"colorscale":{"diverging":
[[0,"#8e0152"],[0.1,"#c51b7d"],[0.2,"#de77ae"],[0.3,"#f1b6da"],
[0.4,"#fde0ef"],[0.5,"#f7f7f7"],[0.6,"#e6f5d0"],[0.7,"#b8e186"],
[0.8,"#7fbc41"],[0.9,"#4d9221"],[1,"#276419"]],"sequential":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],[1,"#f0f921"]],"sequentialminus":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],[1,"#f0f921"]]},"colorway":
["#636efa","#EF553B","#00cc96","#ab63fa","#FFA15A","#19d3f3","#FF6692"
,"#B6E880","#FF97FF","#FECB52"],"font":{"color":"#2a3f5f"},"geo":
{"bgcolor":"white","lakecolor":"white","landcolor":"#E5ECF6","showlake
s":true,"showland":true,"subunitcolor":"white"},"hoverlabel":
{"align":"left"},"hovermode":"closest","mapbox":

```
{"style":"light"},"paper_bgcolor":"white","plot_bgcolor":"#E5ECF6","po
lar":{"angularaxis":
{"gridcolor":"white","linecolor":"white","ticks":""},"bgcolor":"#E5ECF
6","radialaxis":
{"gridcolor":"white","linecolor":"white","ticks":""}},"scene":
{"xaxis":
{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"lineco
lor":"white","showbackground":true,"ticks":"","zerolinecolor":"white"}
,"yaxis":
{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"lineco
lor":"white","showbackground":true,"ticks":"","zerolinecolor":"white"}
,"zaxis":
{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"lineco
lor":"white","showbackground":true,"ticks":"","zerolinecolor":"white"}
},"shapedefaults":{"line":{"color":"#2a3f5f"}},"ternary":{"aaxis":
{"gridcolor":"white","linecolor":"white","ticks":""},"baxis":
{"gridcolor":"white","linecolor":"white","ticks":""},"bgcolor":"#E5ECF
6","caxis":
{"gridcolor":"white","linecolor":"white","ticks":""}},"title":
{"x":5.0e-2},"xaxis":
{"automargin":true,"gridcolor":"white","linecolor":"white","ticks":"",
"title":
{"standoff":15},"zerolinecolor":"white","zerolinewidth":2},"yaxis":
{"automargin":true,"gridcolor":"white","linecolor":"white","ticks":"",
"title":
{"standoff":15},"zerolinecolor":"white","zerolinewidth":2}}},"title":
{"text":"Interactive Pie Chart"}}}
```