

# BASICS OF PYTHON VISUALIZATION WITH SEABORN

## SEABORN

Seaborn is an amazing visualization library for statistical graphics plotting in Python. It provides beautiful default styles and colour palettes to make statistical plots more attractive.

Data Visualization is the presentation of data in pictorial format. It is extremely important for Data Analysis, primarily because of the fantastic ecosystem of data-centric Python packages. Line plots are the most popular plot to draw a relationship between x and y with the possibility of several semantic groupings. Let's visualize the data with a line plot and pandas.



## INSTALLATION STEPS FOR SEABORN

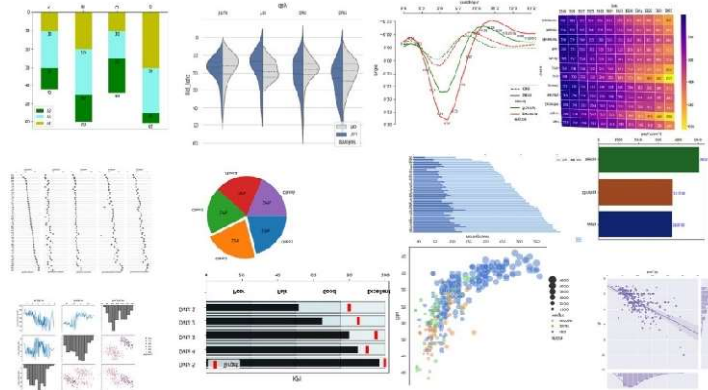
**Step1:** go to command prompt.

**Note:** To install seaborn you need to have internet

**Step2:** Used this command to install sea born "pip install seaborn".

**Step3:** Click enter and wait until it get installed.

**Step4:** Once it is installed you can run seaborn operation program in vs Code.



## WHEN SHOULD WE USE SEABORN IN PYTHON?

Seaborn is a library mostly used for statistical plotting in Python. It is built on top of Matplotlib and provides beautiful default styles and colour palettes to make statistical plots more attractive.

# BASICS OF PYTHON VISUALIZATION WITH MATPLOTLIB.

## MATPLOTLIB

Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack.

Matplotlib comes with a wide variety of plots. Plots help to understand trends, and patterns, and to make correlations. They're typically instruments for reasoning about quantitative information. Some of the sample plots are covered here.



**MATPLOTLIB**

## INSTALLATION STEPS FOR SEABORN

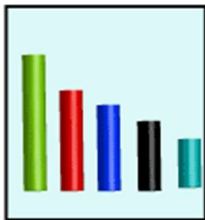
**Step1:** go to command prompt.

**Note:** To install seaborn you need to have internet

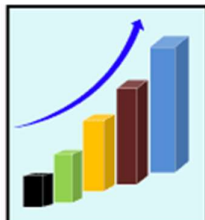
**Step2:** Used this command to install matplotlib "pip install matplotlib".

**Step3:** Click enter and wait until it get installed.

**Step4:** Once it is installed you can run matplotlib operation program in vs Code.



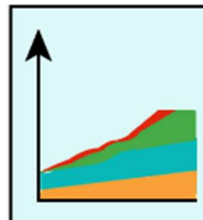
Bar Graph



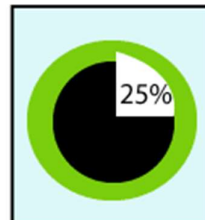
Histogram



Scatter Plot



Area Plot



Pie Plot

## WHEN SHOULD WE USE MATPLOTLIB IN PYTHON?

A Matplotlib module that provides a MATLAB-like interface. Matplotlib is designed to be as usable as MATLAB, with the ability to use Python and the advantage of being free and open source.

# Data Visualization Using Seaborn And Matplotlib

## PROBLEM STATEMENT:

### 1. WRITE A PYTHON PROGRAM USING NUMPY MATPLOTLIB AND SEABORN.

- (a) Create a simple plot sin and cos in single graph using (x=0,10,100).
- (b) Make a graph sin and cos, subplot using row, column, panel number.
- (c) Draw a graph of Visualizing amount method some set using bars, either vertically, horizontally and dot.

YEAR	GROSS AMOUNT
2014	18500
2015	12700
2016	600
2017	14560
2018	8550
2019	11420

- (d) Read table plot grouped and stacked bar.

STATES	YEAR	GROSS AMOUNT
KARNATAKA	2014	18500
DELHI	2015	12700
HYDERABAD	2016	600
TAMIL NADU	2017	14560
MAHARASHTRA	2018	8550
GUJARAT	2019	11420

- (e) Create a dot plot and heat maps to using below tables.

STATES	YEAR	GROSS AMOUNT
KARNATAKA	2014	18500
DELHI	2015	12700
HYDERABAD	2016	600
TAMIL NADU	2017	14560
MAHARASHTRA	2018	8550
GUJARAT	2019	11420

- (f) plot a graph using two histograms on single chart with matplotlib take a random number X1(-2 to 2,1000) and Y2(2 to 2,1000).

(g) Visualization a single distribution method for plotting a graph, using the table below.

AGE	COUNT
0-5	36
6-10	19
11-15	18
16-20	99
21-25	139
26-30	121
31-35	75
36-40	54
41-45	64
46-50	50

(h) Create a pie Chart below the table using proportions visualization

STATES	MEDIAN INCOME
KARNATAKA	18500
DELHI	12700
HYDERABAD	600
TAMIL NADU	14560
MAHARASHTRA	8550
GUJARAT	11420

(i) Create a data set, and using visualizing proportions Visualizing Uncertainly plot graph.

(j) Using Visualization uncertainly of curve fitting method find sine function coefficients as well as graph.

(k) create different types of graph using seaborn and matplotlib.

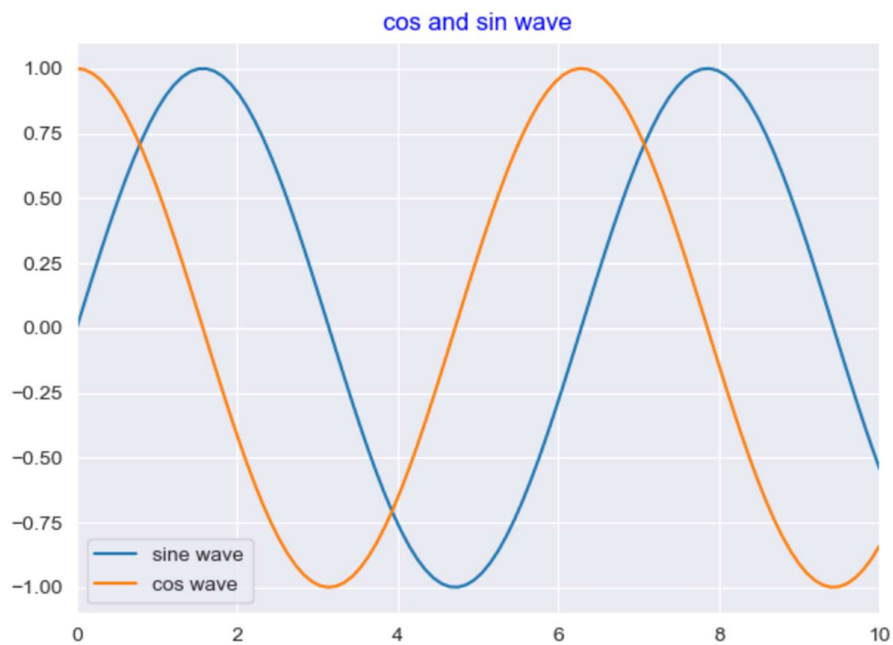
## 1. WRITE A PYTHON PROGRAM USING NUMPY, MATPLOTLIB AND SEABORN.

(a). Create a simple sin and cos wave plot in single graph using(x=0,10,100)

INPUT:

```
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
# Creating a style
sns.set_style("darkgrid")
# Creating subplots
fig, ax = plt.subplots(figsize=(7, 5))
# Range for x
x = np.linspace(0, 10, 100)
# Creating sine wave
y = np.sin(x)
ax.plot(x, y, label="sine wave")
# Creating cosine wave
y1 = np.cos(x)
ax.plot(x, y1, label="cos wave")
# Setting x-axis limits
ax.set_xlim(0, 10)
# Setting title
plt.title('cos and sin wave', color="b")
# Creating a Legend
ax.legend()
# Displaying the plot
plt.show()
```

OUTPUT:

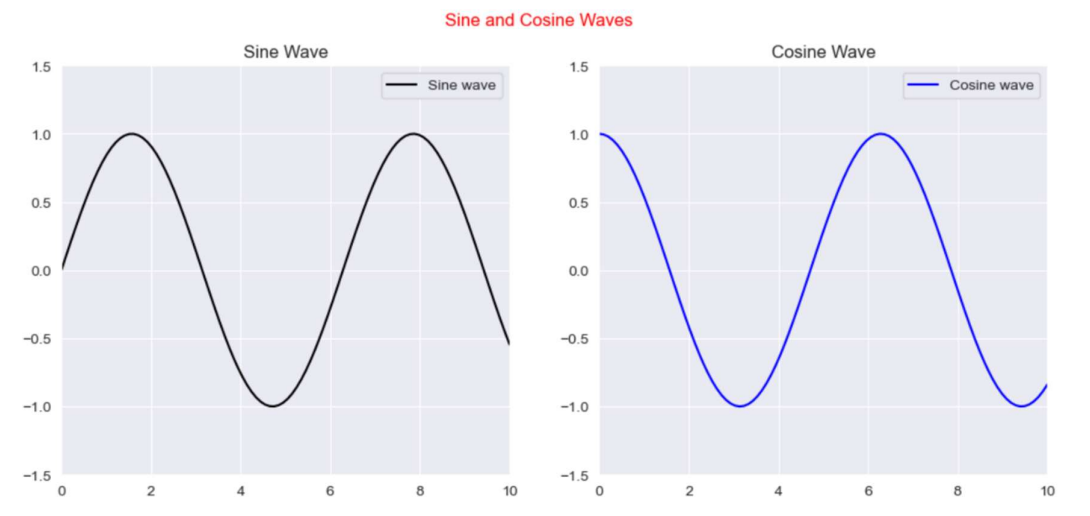


(b). Make a graph of sin and cos subplots, using row, column, panel number.

INPUT:

```
import numpy as np
import seaborn as sns
from matplotlib import pyplot as plt
sns.set_style('darkgrid')
# Creating subplots (2 rows, 1 column)
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 5))
# Range for sine and cosine waves
x = np.linspace(0, 10, 100)
# Creating sine and cosine waves
y_sin = np.sin(x)
y_cos = np.cos(x)
# Plotting sine wave on the first subplot
ax1.plot(x, y_sin, label="Sine wave", color="black")
ax1.set_title('Sine Wave')
ax1.set_xlim(0, 10)
ax1.set_ylim(-1.5, 1.5)
ax1.legend()
# Plotting cosine wave on the second subplot
ax2.plot(x, y_cos, label="Cosine wave", color="blue")
ax2.set_title('Cosine Wave')
ax2.set_xlim(0, 10)
ax2.set_ylim(-1.5, 1.5)
ax2.legend()
# Adding a main title for the entire figure
fig.suptitle('Sine and Cosine Waves', color='r')
plt.show()
```

OUTPUT:



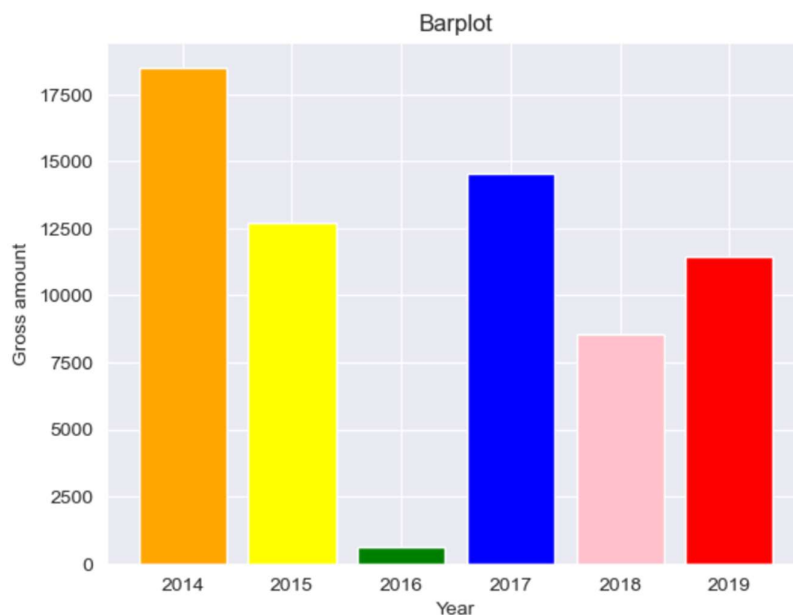
(c) Draw a graph of Visualizing amount method some set using bars, either vertically, horizontally and dot.

YEAR	GROSS AMOUNT
2014	18500
2015	12700
2016	600
2017	14560
2018	8550
2019	11420

INPUT:

```
#Bar plot (Horizontal)
import matplotlib.pyplot as plt
import seaborn as sns
#creating a style
sns.set_style('darkgrid')
#assigning the values for barplot
x=[2014,2015,2016,2017,2018,2019]
y=[18500,12700,600,14560,8550,11420]
colors=['orange','yellow','green','blue','pink','red']
plt.bar(x,y,color=colors)
plt.title('Barplot')
plt.xlabel("Year")
plt.ylabel("Gross amount")
plt.show()
```

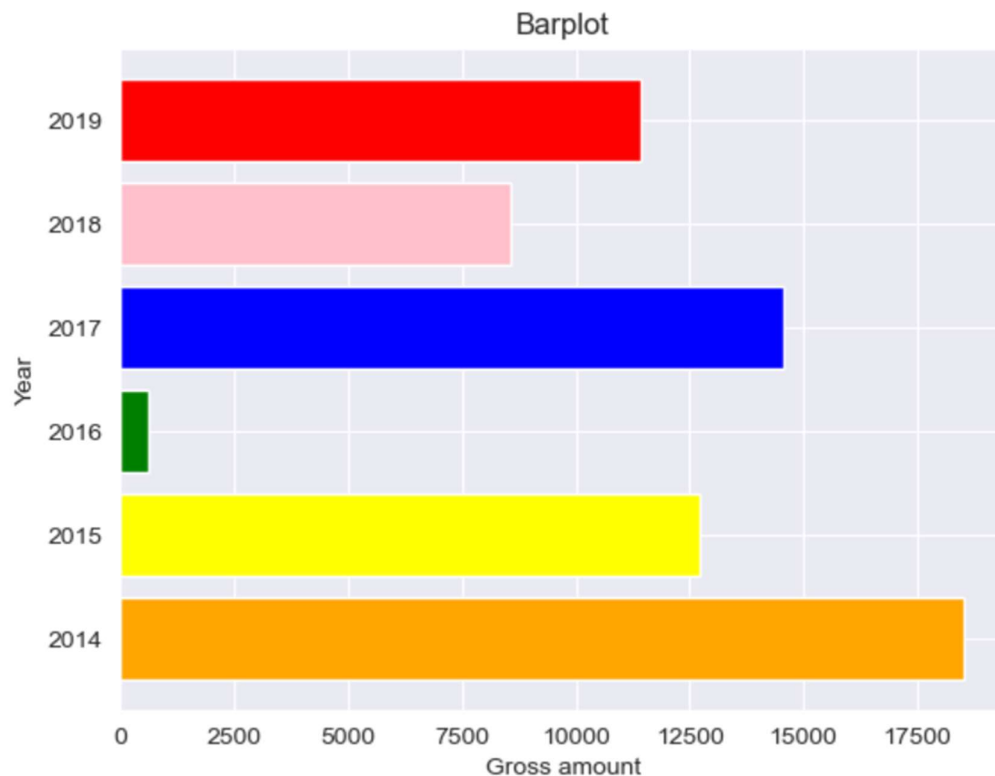
OUTPUT:



## INPUT:

```
#bar plot (Vertical)
import matplotlib.pyplot as plt
import seaborn as sns
# Creating a style
sns.set_style('darkgrid')
# Assigning the values for barplot
x = [2014, 2015, 2016, 2017, 2018, 2019]
y = [18500, 12700, 600, 14560, 8550, 11420]
colors = ['orange', 'yellow', 'green', 'blue', 'pink', 'red']
# Creating a horizontal bar plot
plt.barh(x, y, color=colors)
# Adding title and Labels
plt.title('Barplot')
plt.xlabel("Gross amount")
plt.ylabel("Year")
# Display the plot
plt.show()
```

## OUTPUT:

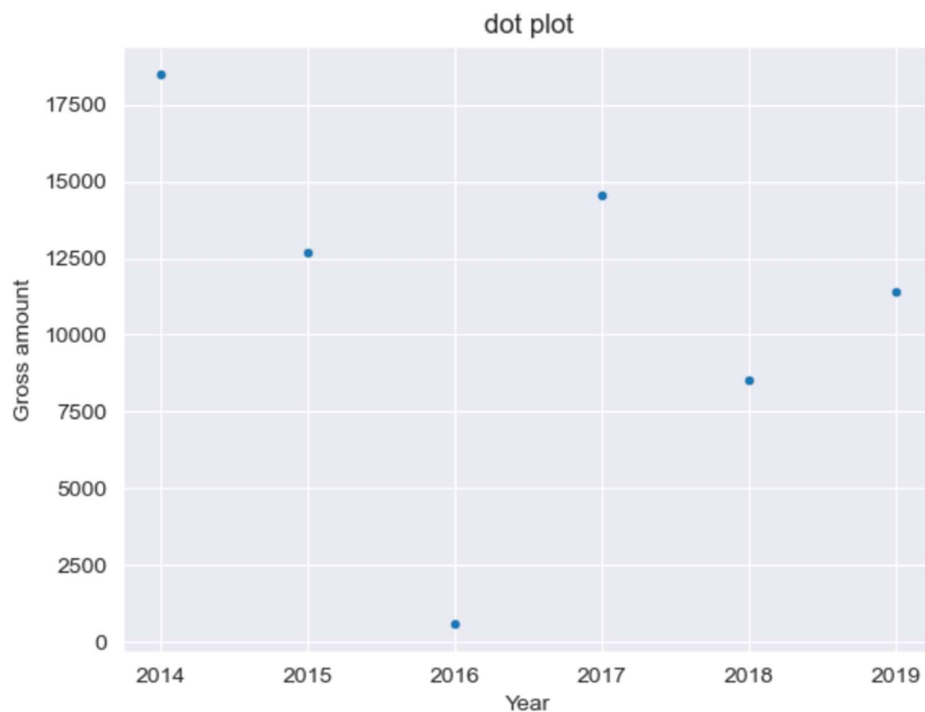




## INPUT:

```
#Dot plot
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
#assigning values
x=[2014, 2015, 2016, 2017, 2018, 2019]
y=[18500, 12700, 600, 14560, 8550, 11420]
#creating a dot plot using a scatter
plt.scatter(x,y, marker=".")
plt.xticks(x)
#creating a style
sns.set_style('darkgrid')
plt.title("dot plot")
plt.xlabel("Year")
plt.ylabel("Gross amount")
plt.show()
```

## OUTPUT:



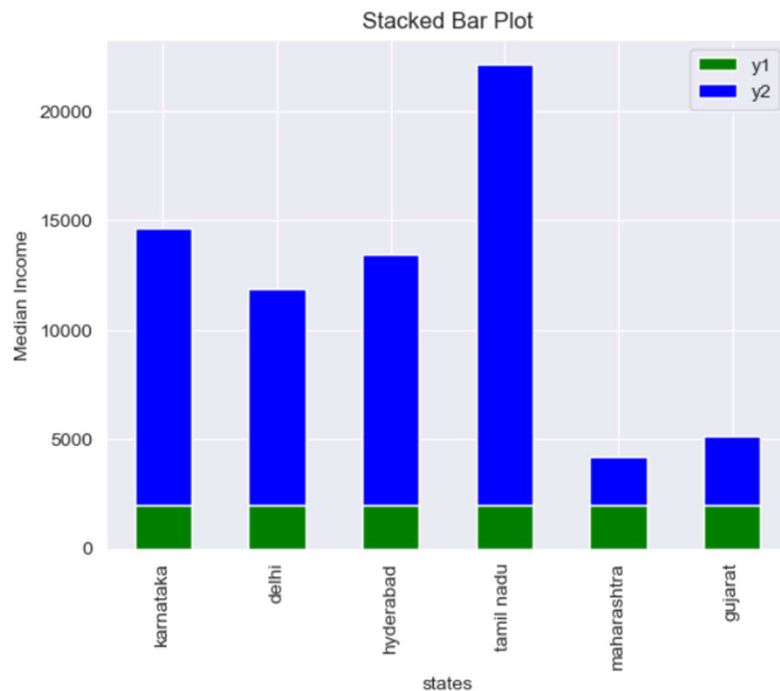
(d) Read table plot grouped and stacked bar.

STATES	YEAR	GROSS AMOUNT
KARNATAKA	2014	18500
DELHI	2015	12700
HYDERABAD	2016	600
TAMIL NADU	2017	14560
MAHARASHTRA	2018	8550
GUJARAT	2019	11420

INPUT:

```
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
# Creating a style
sns.set_style("darkgrid")
# Creating a DataFrame with data
data = pd.DataFrame({
    'y1': [2014, 2015, 2016, 2017, 2018, 2019],
    'y2': [12645, 9877, 11433, 20133, 2206, 3097]
}, index=['karnataka', 'delhi', 'hyderabad', 'tamil nadu', 'maharashtra', 'gujarat'])
# Plotting stacked bar chart
data.plot(kind='bar', stacked=True, color=['green', 'blue'])
# Adding title and labels
plt.title('Stacked Bar Plot')
plt.xlabel("states")
plt.ylabel("Median Income")
# Display the plot
plt.show()
```

OUTPUT:



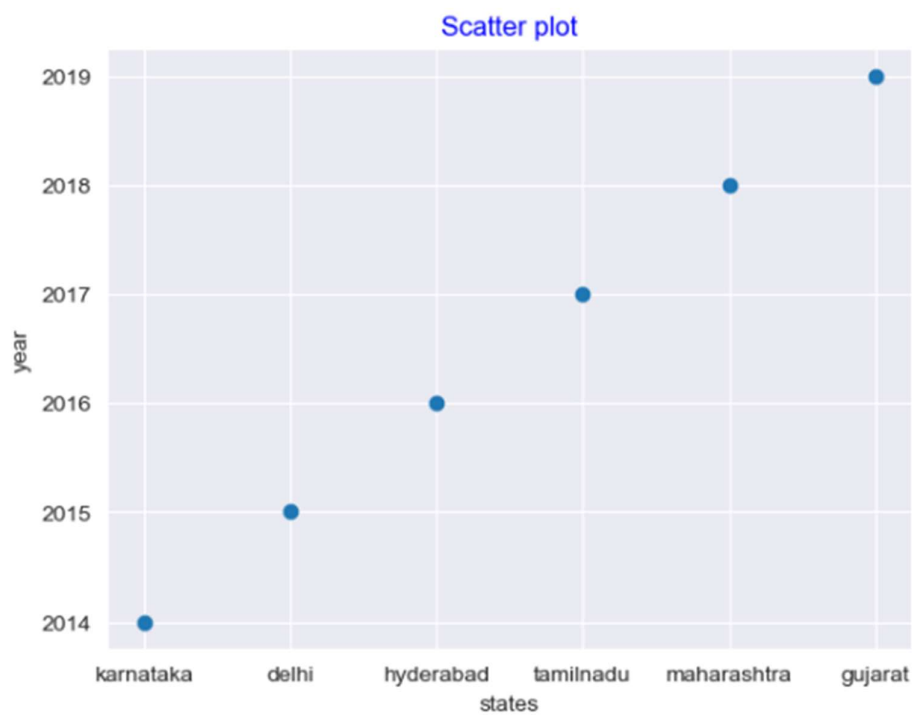
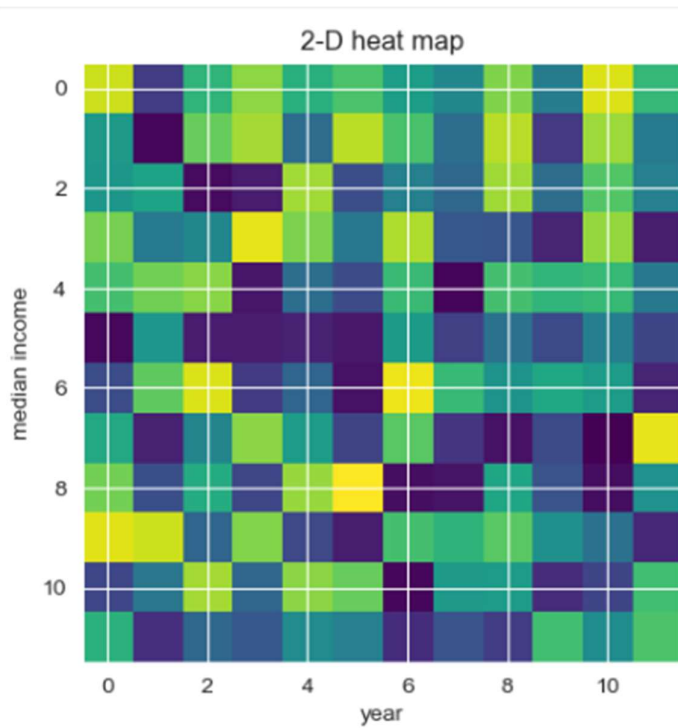
(e) Create a dot plot and heat maps to using below tables.

STATES	YEAR	GROSS AMOUNT
KARNATAKA	2014	18500
DELHI	2015	12700
HYDERABAD	2016	600
TAMIL NADU	2017	14560
MAHARASHTRA	2018	8550
GUJARAT	2019	11420

INPUT:

```
#creating a dot plots and heatmaps
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
#heat map
#creating a heat map
data=np.random.random((12,12))
plt.imshow(data,cmap='viridis', interpolation = 'nearest')
plt.title("2-D heat map")
plt.xlabel('year')
plt.ylabel('median income')
plt.show()
#dot graph
#assigning a value to do plot
x=['karnataka', 'delhi', 'hyderabad', 'tamilnadu', 'maharashtra','gujarat']
y=[2014,2015,2016,2017,2018,2019]
#creating a dot plot using a scatter
plt.scatter(x,y)
#creating a style
sns.set_style('darkgrid')
plt.title("Scatter plot", color='blue')
plt.xlabel('states')
plt.ylabel('year')
plt.show()
```

OUTPUT:

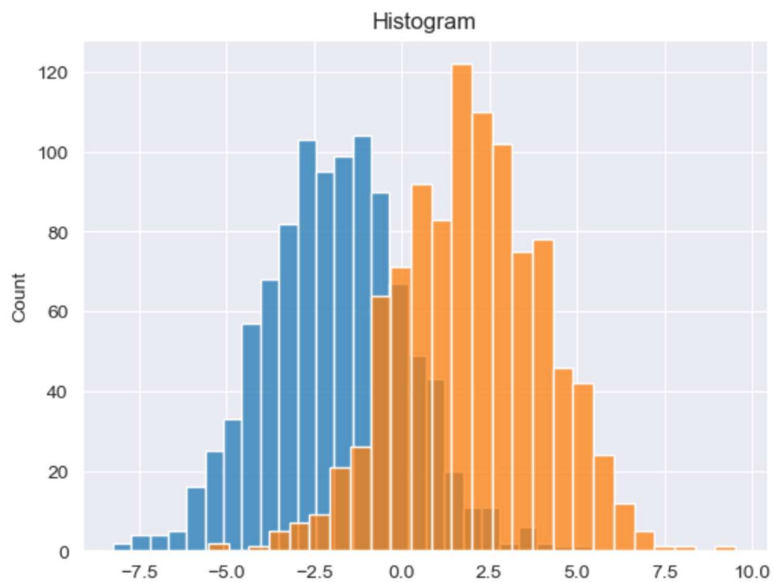


(f) plot a graph using two histograms on single chart with matplotlib take a random number X1(-2 to 2,1000) and Y2(2 to 2,1000).

INPUT:

```
#Histograms
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
#creating a style
sns.set_style("darkgrid")
#declaring values
y1=np.random.normal(-2,2,1000)
y2=np.random.normal(2,2,1000)
#creating histogram
sns.histplot(y1)
sns.histplot(y2)
plt.title("Histogram")
plt.show()
```

OUTPUT:



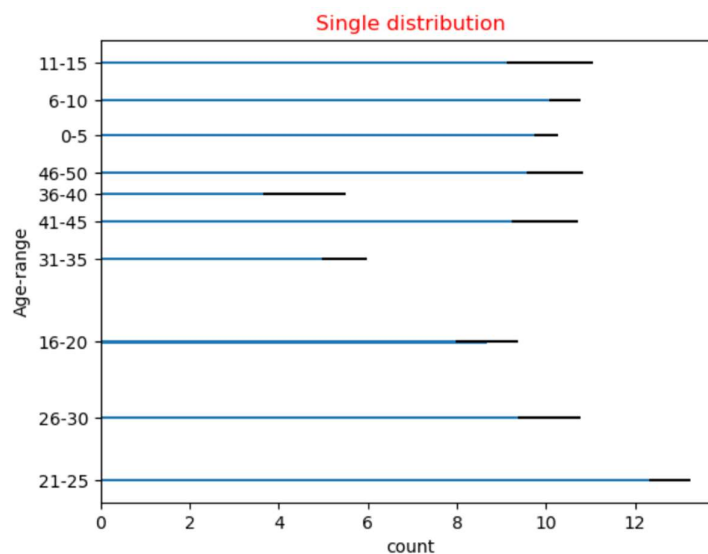
(g) Visualization a single distribution method for plotting a graph, using the table below.

AGE	COUNT
0-5	36
6-10	19
11-15	18
16-20	99
21-25	139
26-30	121
31-35	75
36-40	54
41-45	64
46-50	50

INPUT:

```
#visualizing a single distribution method for plotting a graph
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
#creating a style
sns.set_style('darkgrid')
np.random.seed(19680801)
plt.rcParams()
ax=plt.subplot()
people=['0-5', '6-10', '11-15', '16-20', '21-25', '26-30', '31-35', '36-40', '41-45', '46-50']
y_pos=np.array([39,29,18,99,139,121,75,56,64,50])
performance=3+10*np.random.rand(len(people))
error=np.random.rand(len(people))
ax.barh(y_pos, performance, xerr=error, align='center')
ax.set_yticks(y_pos, labels=people)
ax.invert_yaxis()
#setting a x Label
ax.set_xlabel('count')
#setting a y Label
ax.set_ylabel('Age-range')
ax.set_title('Single distribution', color='red')
plt.show()
```

OUTPUT:



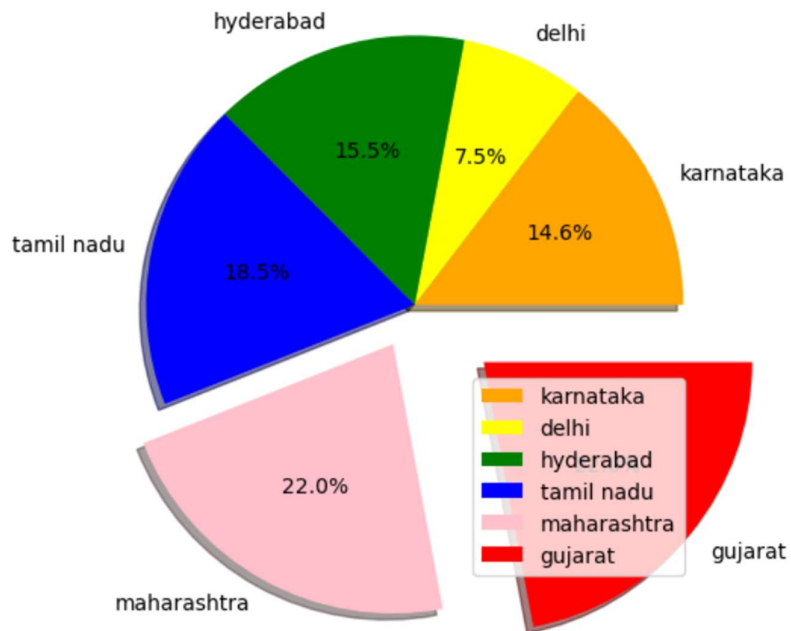
(h) Create a pie Chart below the table using proportions visualization

STATES	MEDIAN INCOME
KARNATAKA	18500
DELHI	12700
HYDERABAD	600
TAMIL NADU	14560
MAHARASHTRA	8550
GUJARAT	11420

INPUT:

```
#Pie chart
import matplotlib.pyplot as plt
colors=['r', 'y', 'g', 'b']
city=["karnataka", "delhi", "hyderabad", "tamil nadu", "maharashtra","gujarat"]
colors=['orange', 'yellow', 'green', 'blue', 'pink','red']
Median_Income=[200211,102344,212200,254103,301552,302567]
plt.pie(Median_Income,labels=city,colors=colors,shadow = True,
        explode=(0,0,0,0,0.2,0.4),radius=1.2,autopct='%1.1f%%')
plt.legend()
plt.show()
```

OUTPUT:

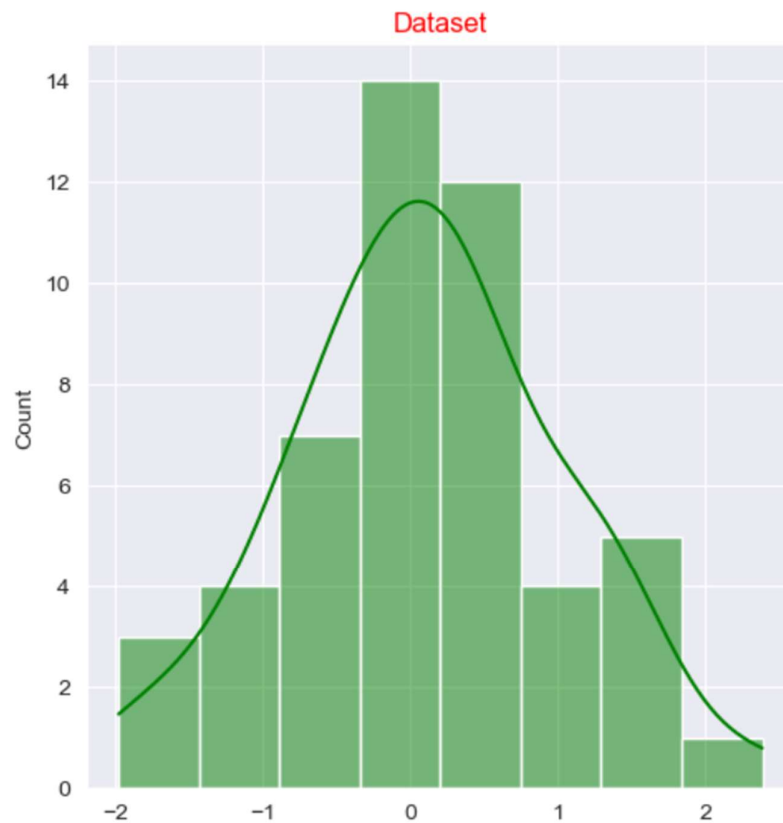


(i) Create a data set, and using visualizing proportions Visualizing Uncertainty plot graph.

**OUTPUT:**

```
#Create dataset, and using visualizing uncertainty plot graph
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
#creating a style
sns.set_style("darkgrid")
#Assigning a range
rs=np.random. RandomState (10)
d=rs.normal(size=50)
sns.displot(d,kde=True,color="green")
plt.title("Dataset", color='red')
plt.show()
```

**INPUT:**





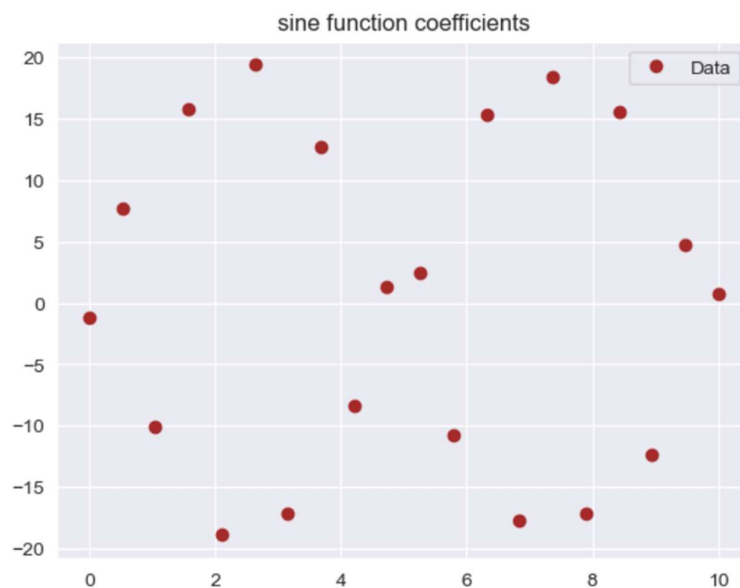
(j) Using Visualization uncertainty of curve fitting method find sine function coefficients as well as graph.

INPUT:

```
import numpy as np
import seaborn as sns
from scipy.optimize import curve_fit
from matplotlib import pyplot as plt
x=np.linspace(0, 10, num=20)
#the coefficients are much bigger
#creating a style
sns.set_style("darkgrid")
y=18.45*np.sin(5.334*x)+np.random.normal(size=20)
def test(x,a,b):
    return a*np.sin(b*x)
param,param_cuv=curve_fit(test,x,y)
print("sine function coefficients")
print(param)
print("sine function coefficients")
print(param_cuv)
ns=(param[0]*(np.sin(param[1]*x)))
#plotting a graph using plot
plt.plot(x,y, 'o', color='brown', label="Data")
#Creating a Line plot
plt.plot(label="Optimize data")
plt.title("sine function coefficients")
plt.legend()
plt.show()
```

OUTPUT:

```
sine function coefficients
[0.44058945 0.88325169]
sine function coefficients
[[18.79809547 0.62747019]
 [ 0.62747019 3.16883449]]
```



(k) create different types of graph using seaborn and matplotlib.

### INPUT:

```
#Plot graph
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
sns.set_style("darkgrid")
x=["karnataka", "delhi", "hyderabad", "tamil nadu", "maharashtra"]
y=[200211, 102344, 1220, 254103, 301552]
plt.plot(x,y)
sns.set_style("darkgrid")
plt.title("Line Chart")
plt.xlabel("State")
plt.ylabel("Median Income")
plt.show()

#Bar graph
sns.set_style("darkgrid")
x1=["karnataka", "delhi", "hyderabad", "tamil nadu", "maharashtra"]
y1=[200211, 102344, 1220, 254103, 301552]
plt.bar(x1,y1)
plt.title("Bar Chart")
plt.xlabel("State")
plt.ylabel("Median Income")
plt.show()

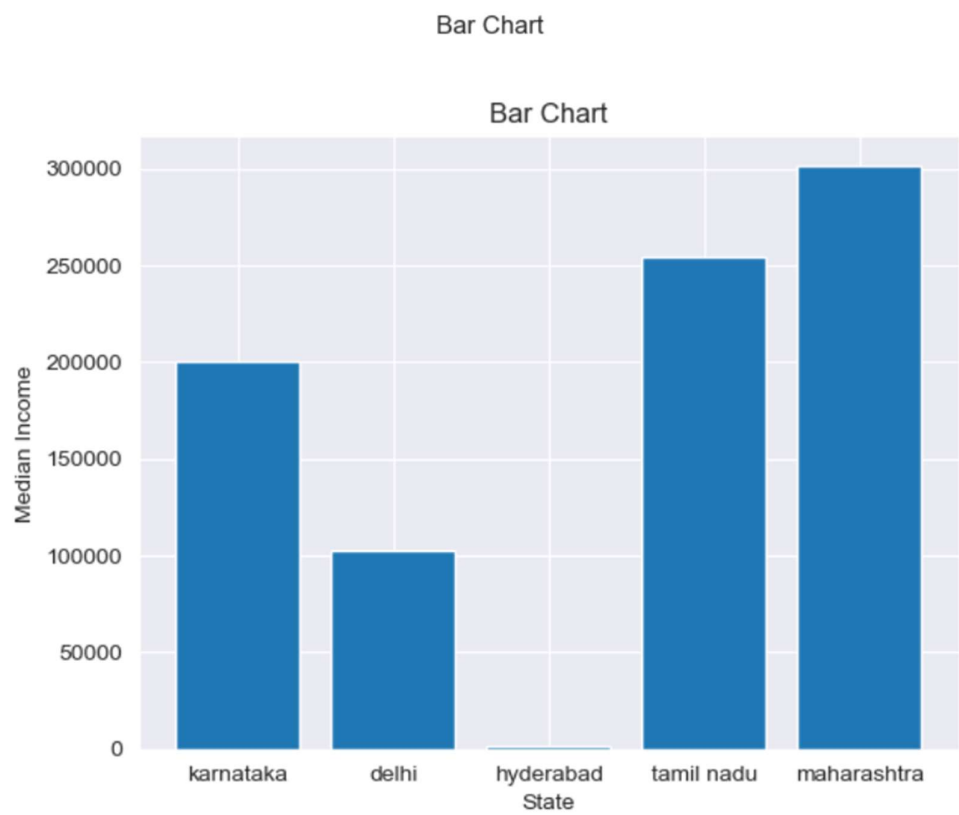
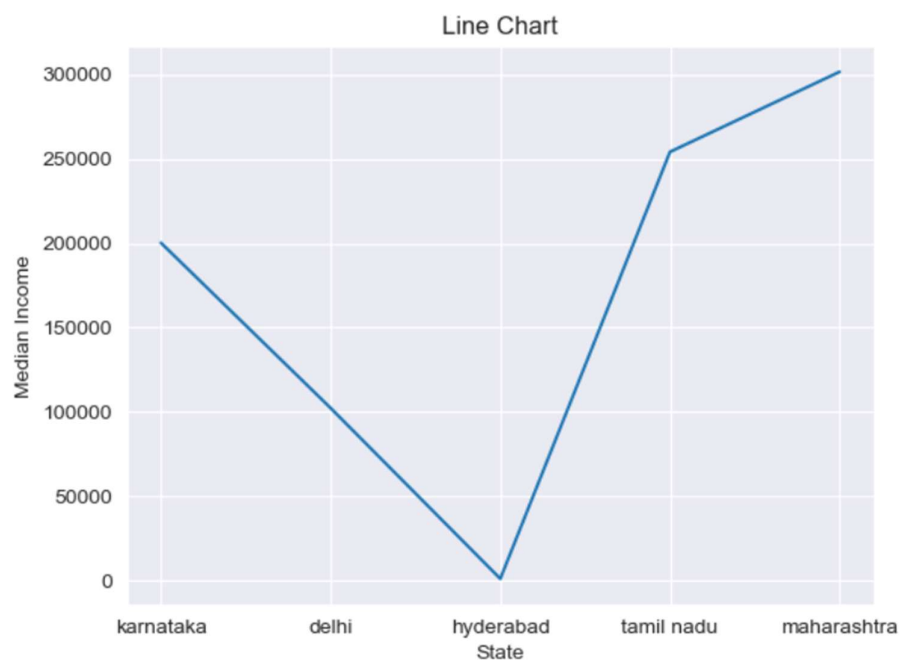
#Scatter graph
sns.set_style("darkgrid")
x2=["karnataka", "delhi", "hyderabad", "tamil nadu", "maharashtra"]
y2=[200211, 102344, 1220, 254103, 301552]
plt.scatter(x2,y2)
sns.set_style("darkgrid")
plt.title("Scatter Chart")
plt.xlabel("State")
plt.ylabel("Median Income")
plt.show()

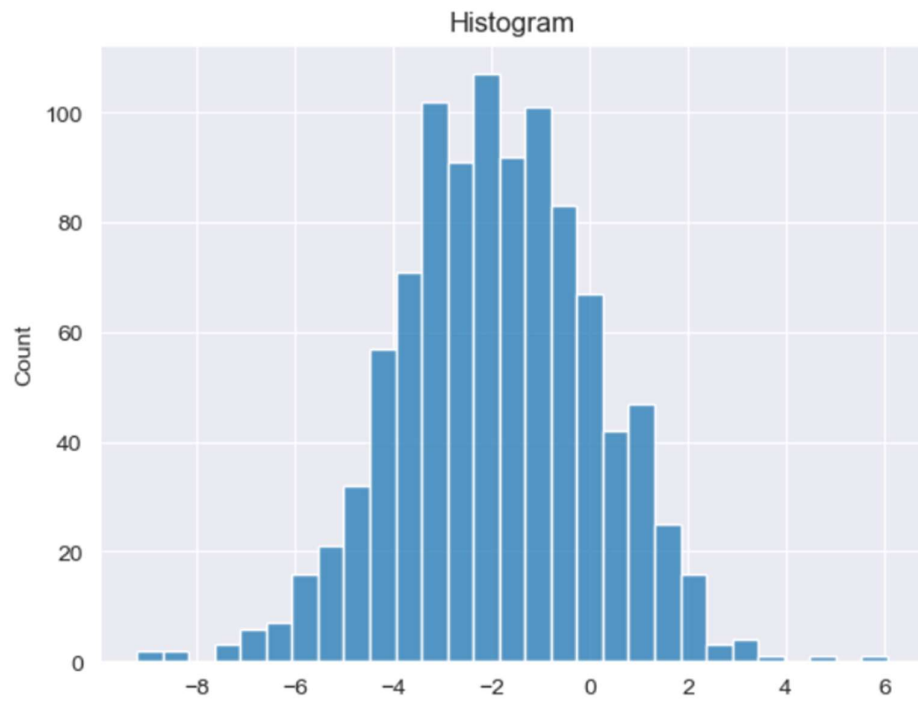
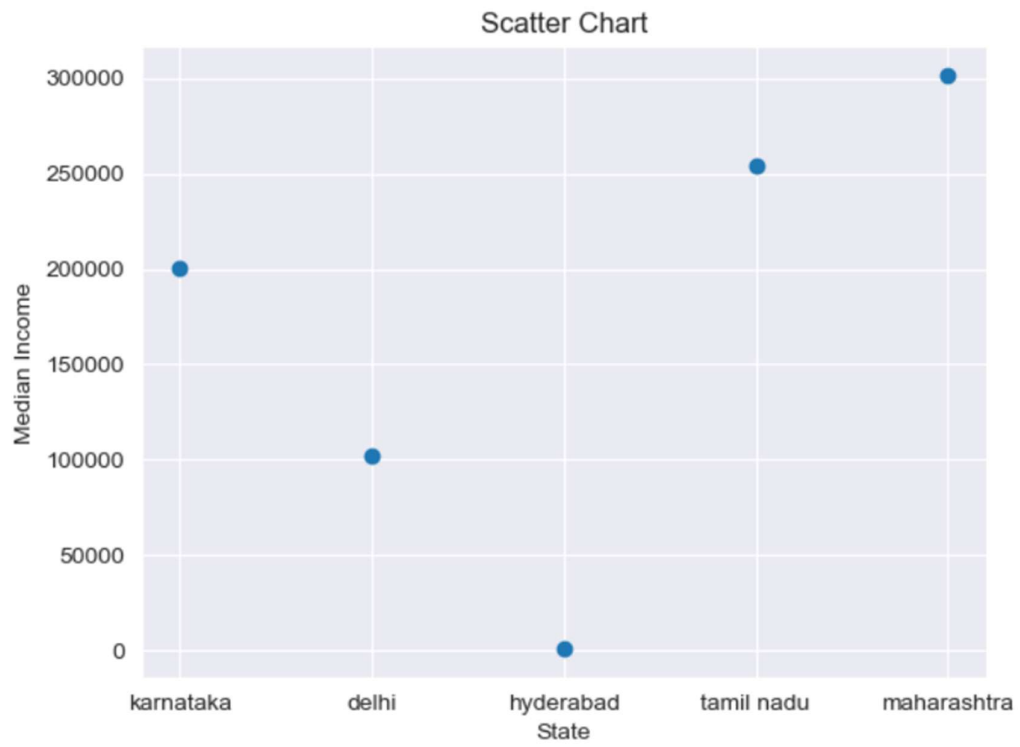
plt.show()

#Histograms
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
#creating a style
sns.set_style("darkgrid")
#declaring values
y1=np.random.normal(-2,2,1000)
#creating histogram
sns.histplot(y1)
plt.title("Histogram")
plt.show()

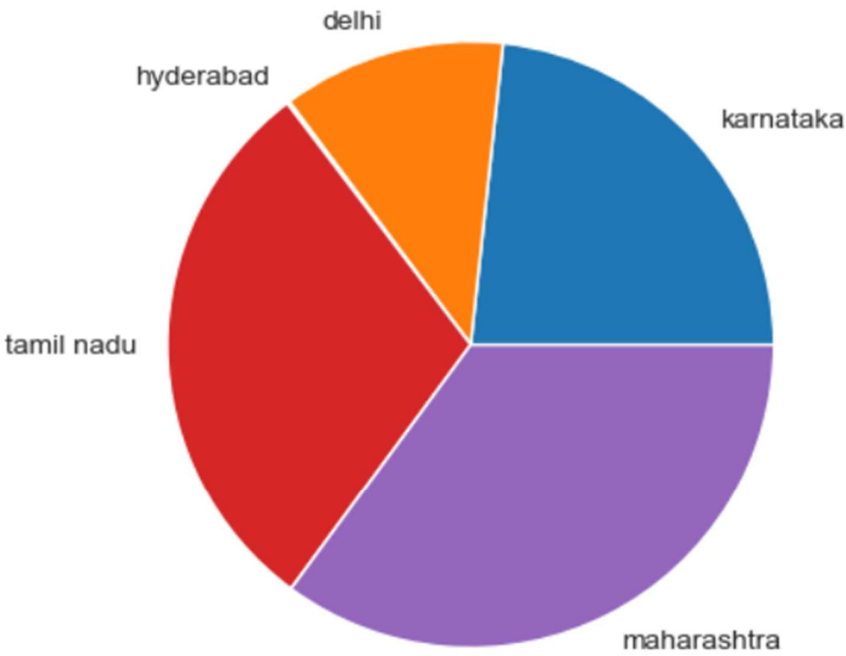
#Piechart
sns.set_style("darkgrid")
#assigning a values to pie chart
x4=["karnataka", "delhi", "hyderabad", "tamil nadu", "maharashtra"]
y4=[200211, 102344, 1220, 254103, 301552]
#creating a pie chart using pie
plt.pie(y4, labels=x4)
plt.title("Pie Chart")
plt.xlabel("State")
plt.show()
```

OUTPUT:





Pie Chart



State