Experiment – 1

Aim:

Write a program to implement normal distribution and histograms for data visualization

Software Required:

Any language compiler Python, Jupyter Notebook, Google Colab, Dataset for visualization.

Description:

The experiment involves writing a program to implement the normal distribution and histograms for data visualization. The aim of this experiment is to gain practical experience in generating and visualizing data that follows a normal distribution, as well as understanding the concept of histograms as a graphical representation of data.

Algorithm / Steps:

Step 1: Setting Up the Environment

Ensure that Python and necessary libraries (such as NumPy, Pandas, and Matplotlib) are installed on your computer.

Launch Jupyter Notebook or your preferred Python IDE.

Step 2: Importing Required Libraries

Begin by importing the necessary libraries like pandas for data manipulation and visualization:

Step 3: Loading the Dataset

If you have a dataset to visualize, load it using the appropriate function (pd.read_csv(), pd.read_excel(), etc.). Make sure the dataset is in a compatible format (CSV, Excel, etc.).

If you don't have a dataset, generate a random dataset using NumPy. For example, to generate 1000 data points from a normal distribution with mean 0 and standard deviation1:

Step 4: Creating a Histogram

1. To create a histogram, use the plt.hist() function, specifying the dataset and the number of bins:
2. Customize the plot by modifying the labels, title, and appearance of the histogram asneeded.

Step 5: Implementing Normal Distribution Plot

.To plot the normal distribution curve, use the plt.plot() function with the appropriate mean and standard deviation values**:**
Customize the plot by modifying labels, title, and appearance.

Step 6: Analyzing Data Distribution

1. Utilize the histogram and normal distribution plot to analyze the distribution of thedataset.
2. Observe the shape of the histogram to identify any skewness or asymmetry in the data.
3. Compare the histogram with the normal distribution plot to assess how closely the datafollows a normal distribution.
4. Calculate key statistical properties such as mean, median, mode, standard deviation,skewness, and kurtosis using appropriate functions(np.mean(), np.median(), np.std(), scipy.stats.skew(), scipy.stats.kurtosis(), etc.).

Step 7: Comparing Data Sets

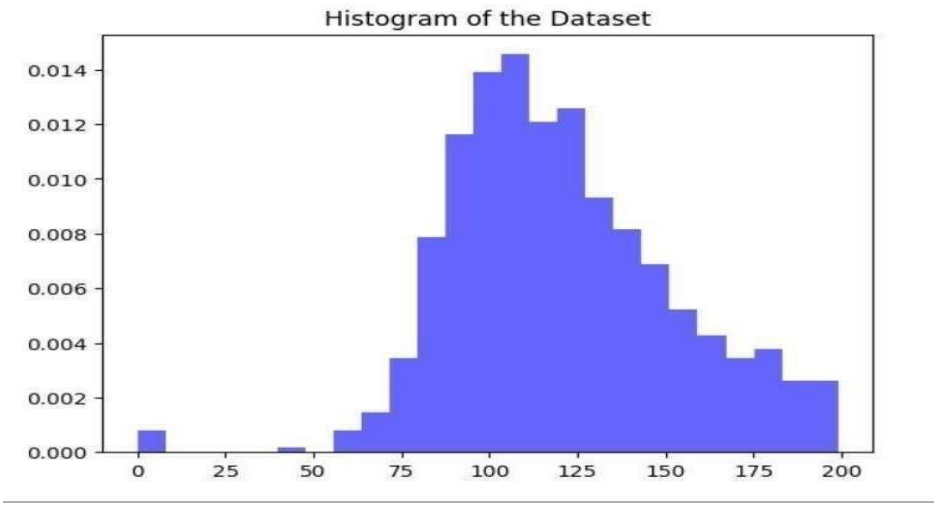
If you have multiple datasets, repeat Steps 3 to 6 for each dataset.
Compare and contrast the histograms and normal distribution plots of different datasetsto identify variations in their distributions

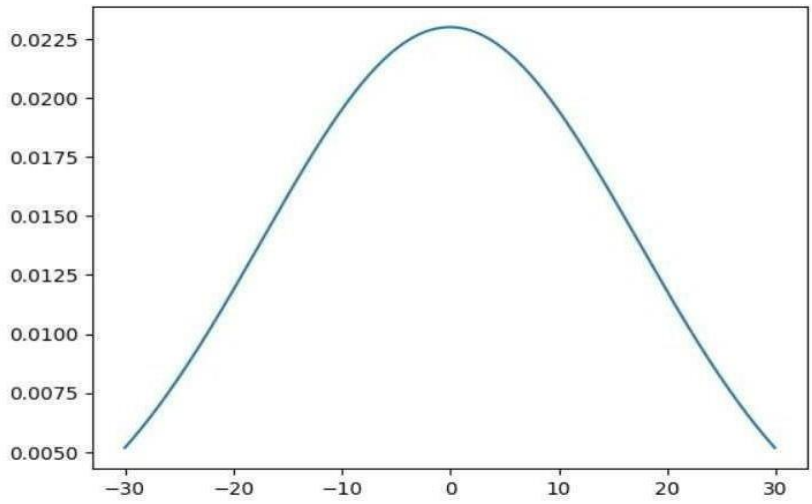
Step of implementation:

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

df = pd.read_csv('diabetes.csv')
data = (df.Glucose)

plt.hist(data, bins=25, density=True, alpha=0.6, color="b")
plt.title('Histogram of the Dataset')
plt.show()
```

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import norm
import statistics

x_axis = np.arange(-30, 30, 0.1)

mean = statistics.mean(x_axis)  # Mean
sd = statistics.stdev(x_axis)  # Standard Devistion

plt.plot(x_axis, norm.pdf(x_axis, mean, sd))

plt.show()
```

OUTPUT:


Histogram of the Dataset
0.014
0.012
0.010
0.008
0.006
0.004
0.002
0.000
0
25
50
75
100
125
150
175
200


0.0225
0.0200
0.0175
0.0150
0.0125
0.0100
0.0075
0.0050
−30
−20
−10
0
10
20
30