# VIDHI ROHIRA

# SY BTECH

# COMPUTER ENGINEERING

# PD LAB-4

# 231071052

# BATCH C

# PD LAB 4

**AIM:-** Plot different graphs using matplotlib. Select any dataset from data world website and create corresponding dashboard using streamlit.

## THEORY:-

### Q] WHAT IS MATPLOTLIB?

**Matplotlib** is a Python library for creating a wide range of static, animated, and interactive visualizations. It's known for its flexibility and customization options, allowing users to produce various plots like line charts, bar charts, and histograms. It integrates well with other libraries such as NumPy and Pandas and is widely used in data analysis and scientific research.

### Q] WHAT IS STREAMLIT?

**Streamlit** is an open-source Python library that allows you to create interactive web applications for data science and machine learning projects with minimal effort. It's designed to help you quickly build and

deploy data-driven applications and dashboards directly from Python scripts.

**Key Features of Streamlit**

1. **Ease of Use**: Streamlit allows you to create web apps with simple Python code. It automatically handles the UI, allowing you to focus on writing Python code rather than dealing with HTML, CSS, or JavaScript.
2. **Real-Time Interactivity**: It provides interactive widgets like sliders, buttons, and text inputs that update in real time. This makes it easy to create dynamic and interactive data visualizations.
3. **Data Visualization**: Streamlit integrates seamlessly with libraries like Matplotlib, Plotly, and Altair, making it straightforward to display charts and graphs.
4. **Rapid Development**: You can build and test your app rapidly because Streamlit supports hot-reloading, which means your app updates automatically as you make changes to the code.
5. **Deployment**: Streamlit apps can be deployed easily to the cloud using Streamlit's sharing platform or other cloud services like Heroku, AWS, and Google Cloud.

## Q] WHAT IS A DASHBOARD?

A **dashboard** is a visual tool that displays data through charts, graphs, and tables, allowing users to monitor, analyze, and make decisions based on key metrics and indicators. It provides a consolidated view of important information, often updating in real-time and offering interactive features for detailed exploration.

### Key Features

- **Data Visualization**: Uses charts, graphs, and maps to simplify complex information.
- **Interactivity**: Allows users to filter and explore data.
- **Real-Time Updates**: Displays current or near-current data.
- **Customization**: Tailors views to specific metrics or needs.
- **Integration**: Combines data from various sources.

### Common Uses

- **Business Intelligence**: Track performance and KPIs.
- **Data Analysis**: Identify trends and patterns.
- **Project Management**: Monitor progress and resources.
- **Web Analytics**: Analyze website and marketing metrics.

## Q] WHAT IS A DATASET?

A **dataset** is a collection of related data organized in a structured format, often in tabular form. It typically consists of multiple records (rows) and attributes (columns) that collectively represent information on a particular subject or area.

### Key Features of a Dataset

- **Records (Rows)**: Individual entries or observations in the dataset. Each record represents a single instance or data point.
- **Attributes (Columns)**: Features or variables associated with the records. Each column represents a different characteristic or measurement of the data.
- **Structure**: Datasets can be structured in various formats, such as spreadsheets, databases, or CSV files, where data is arranged in rows and columns.
- **Types of Data**: Can include various types such as numerical, categorical, text, dates, and more.

### Common Uses

- **Data Analysis**: To perform statistical analyses, identify trends, and derive insights.
- **Machine Learning**: To train models and validate predictions.

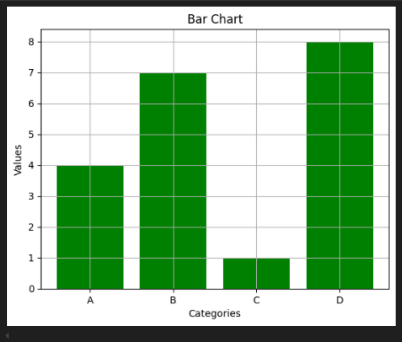
- **Reporting**: To create summaries, visualizations, and reports.
- **Research**: To collect and analyze data for studies and experiments.

EXAMPLE OF A DATASET

| Name | Age | City |
|---|---|---|
| Alice | 30 | New York |
| Bob | 25 | Los Angeles |
| Charlie | 35 | Chicago |

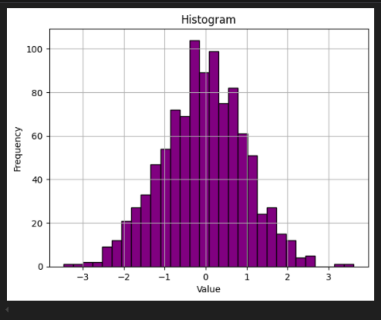
# CODE AND OUTPUT:-

```
import matplotlib.pyplot as plt

# Sample data
categories = ['A', 'B', 'C', 'D']
values = [4, 7, 1, 8]

plt.bar(categories, values, color='green')
plt.title('Bar Chart')
plt.xlabel('Categories')
plt.ylabel('Values')
plt.grid(True)
plt.show()
```



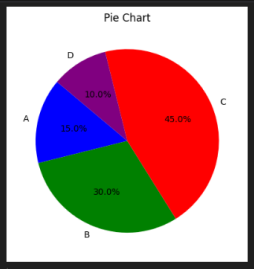

The above shows the code and graph for a bar graph that has been plot for a sample data A,B,C,D with values ranging between 0-8.

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

# Sample data
data = np.random.randn(1000)

plt.hist(data, bins=30, color='purple', edgecolor='black')
plt.title('Histogram')
plt.xlabel('Value')
plt.ylabel('Frequency')
plt.grid(True)
plt.show()
```

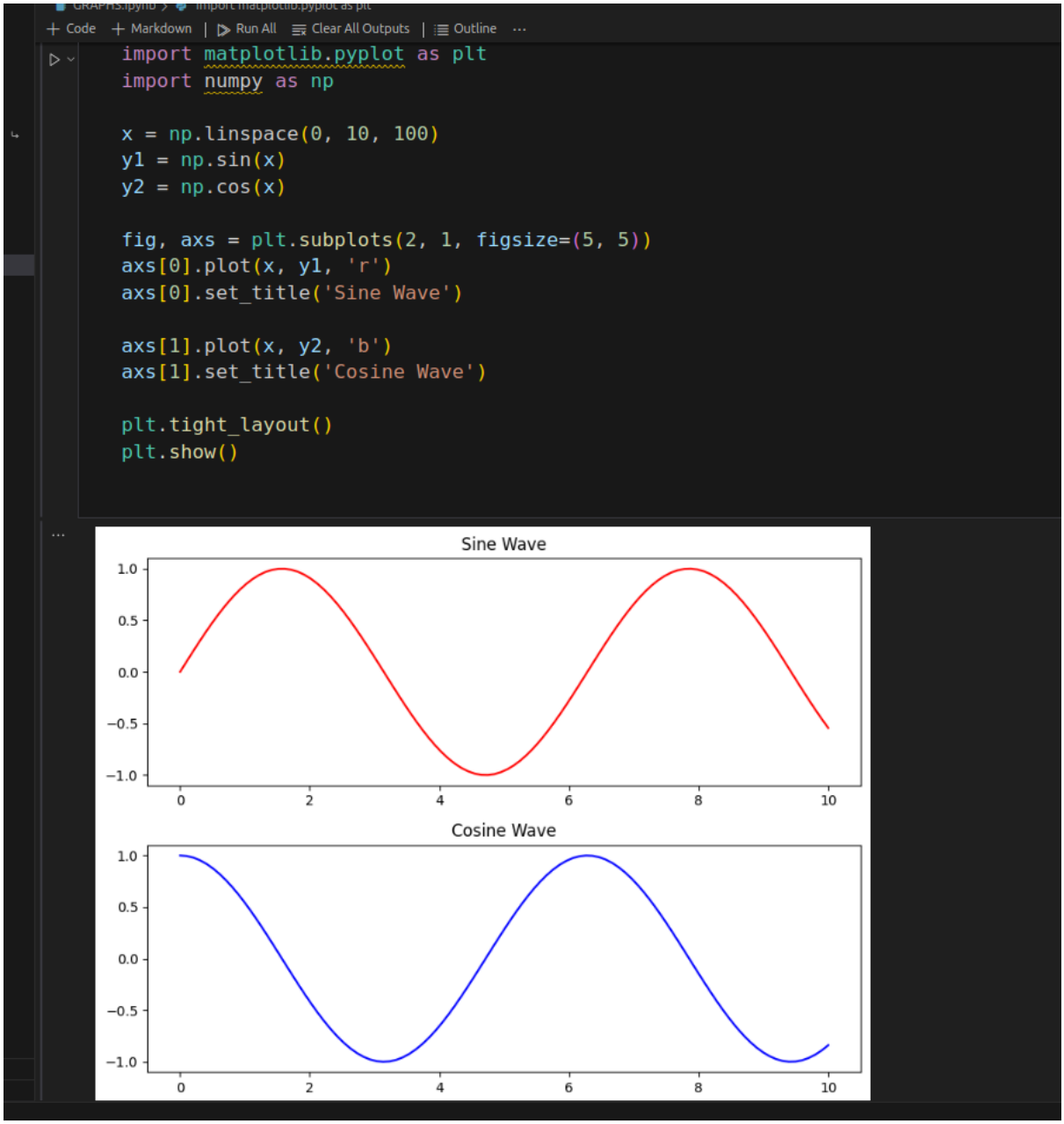



The above shows the code and graph for a histogram for random sample data.

```python
import matplotlib.pyplot as plt

# Sample data
labels = ['A', 'B', 'C', 'D']
sizes = [15, 30, 45, 10]

plt.pie(sizes, labels=labels, autopct='%1.1f%%', startangle=140, colors=['blue', 'green',
plt.title('Pie Chart')
plt.show()
```

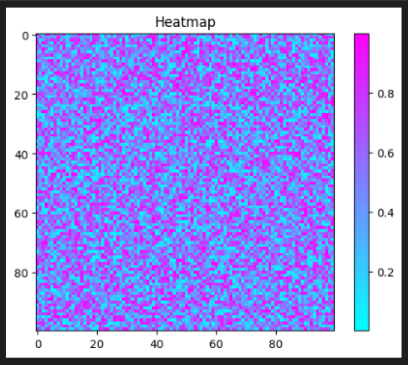



The above shows the code and graph for a pie chart on a random dataset. Here I have chosen the colours red blue green and purple but we have the flexibility of choosing multiple colours of our choice by seeing their code from the official website.



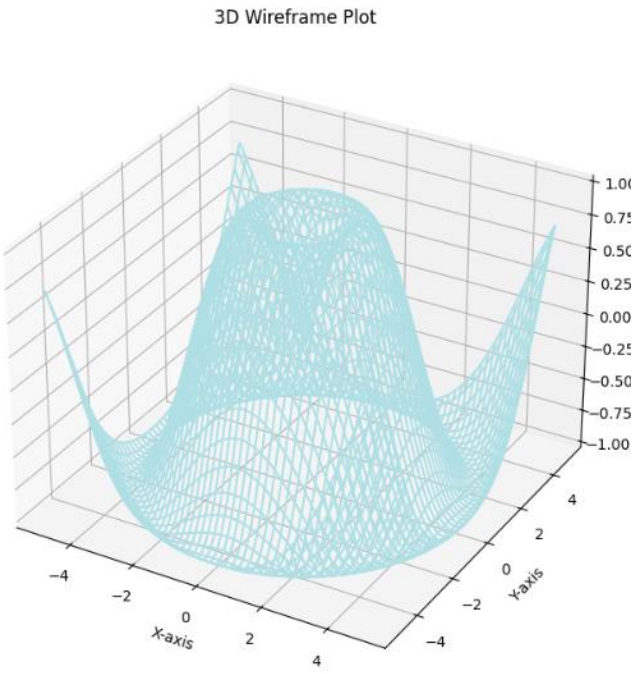

The above shows the graph and code for sin(x) and cos(x) functions.

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

# Sample data
data = np.random.rand(100, 100)

plt.imshow(data, cmap='cool', interpolation='nearest')
plt.title('Heatmap')
plt.colorbar()
plt.show()
```




The above shows the code and graph of a heatmap we can also adjust this graph by changing the input and making it sparse. The above image is dense and shows the cool version of it.

```python
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

# Generate sample data
x = np.linspace(-5, 5, 100)
y = np.linspace(-5, 5, 100)
X, Y = np.meshgrid(x, y)
Z = np.sin(np.sqrt(X**2 + Y**2))

# Create a 3D wireframe plot
fig = plt.figure(figsize=(4, 4))
ax = fig.add_subplot(111, projection='3d')
ax.plot_wireframe(X, Y, Z, color='powderblue')

ax.set_title('3D Wireframe Plot')
ax.set_xlabel('X-axis')
ax.set_ylabel('Y-axis')
ax.set_zlabel('Z-axis')
plt.show()
```




The above shows the code and graph of a 3D wireframe. This tells us that we can plot both 2d and 3d graphs using matplotlib.

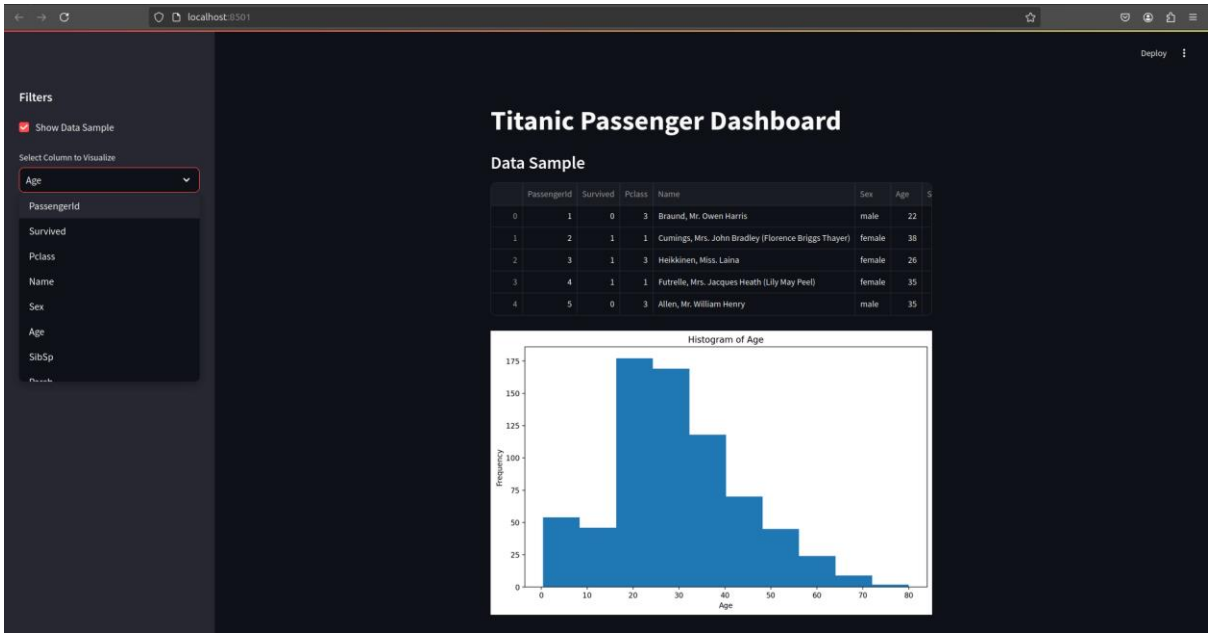
We can also change the input sizes and colours which tells us about the flexibility matplotlib provides us with

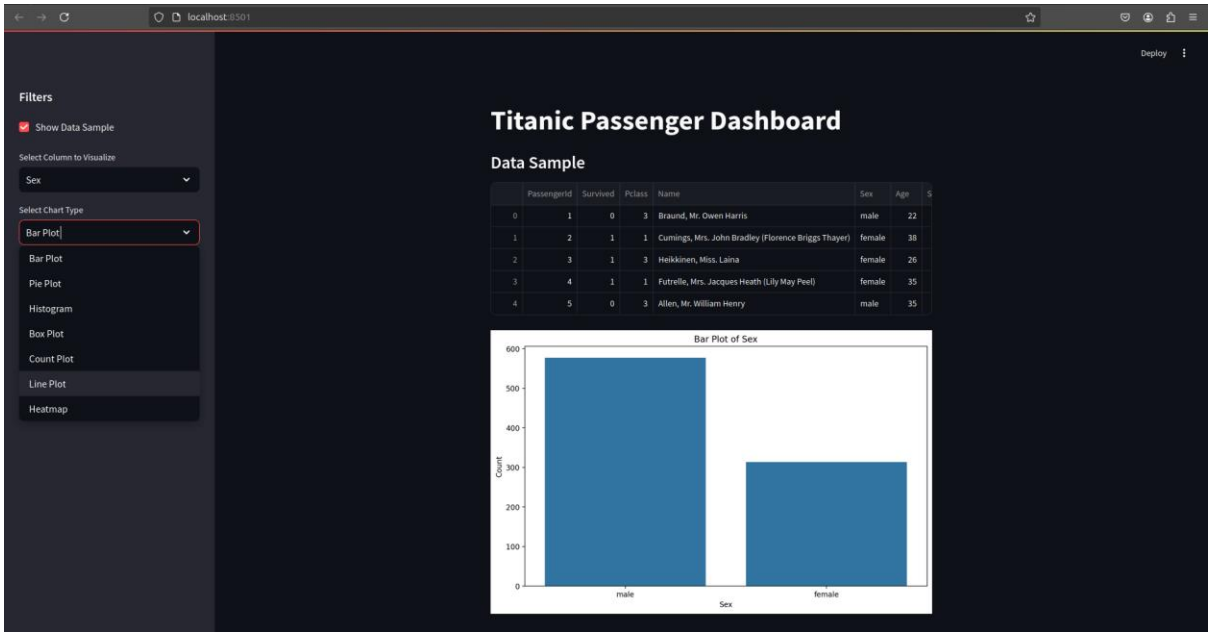
# DASHBOARD:-

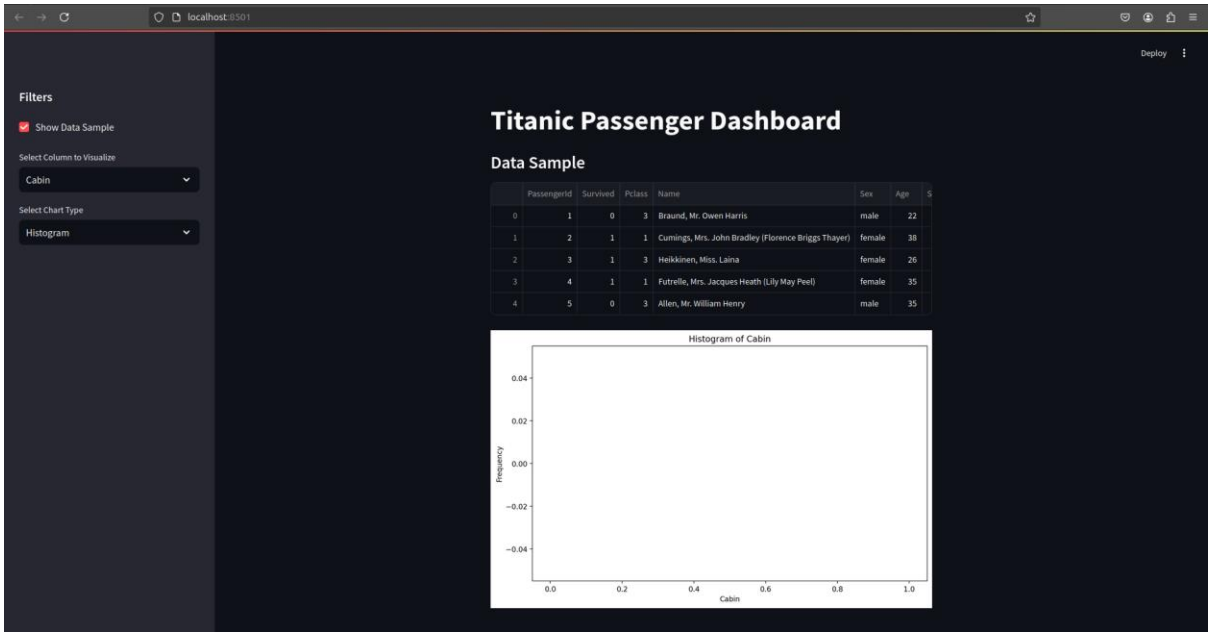
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import streamlit as st

# Load the dataset from the internet
DATA_URL =
'https://raw.githubusercontent.com/datasciencedojo/datasets/master/titanic.csv
'
df = pd.read_csv(DATA_URL)

# Streamlit App
st.title("Titanic Passenger Dashboard")

# Display a sample of the dataset
st.sidebar.header("Filters")
if st.sidebar.checkbox("Show Data Sample"):
    st.write("### Data Sample")
    st.write(df.head())

# Dropdown for selecting the column to visualize
column = st.sidebar.selectbox(
    'Select Column to Visualize',
    df.select_dtypes(include=['object', 'int64', 'float64']).columns
)

# Dropdown for selecting the chart type
chart_type = st.sidebar.selectbox(
    'Select Chart Type',
    ['Bar Plot', 'Pie Plot', 'Histogram', 'Box Plot', 'Count Plot', 'Line
Plot', 'Heatmap']
)

# Create the plot based on user selection
fig, ax = plt.subplots(figsize=(10, 6))

if chart_type == 'Bar Plot':
    sns.countplot(data=df, x=column, ax=ax)
    ax.set_title(f'Bar Plot of {column}')
    ax.set_xlabel(column)
    ax.set_ylabel('Count')

elif chart_type == 'Pie Plot':
    df[column].value_counts().plot(kind='pie', autopct='%1.1f%%', ax=ax,
wedgeprops={'edgecolor': 'black'})
```

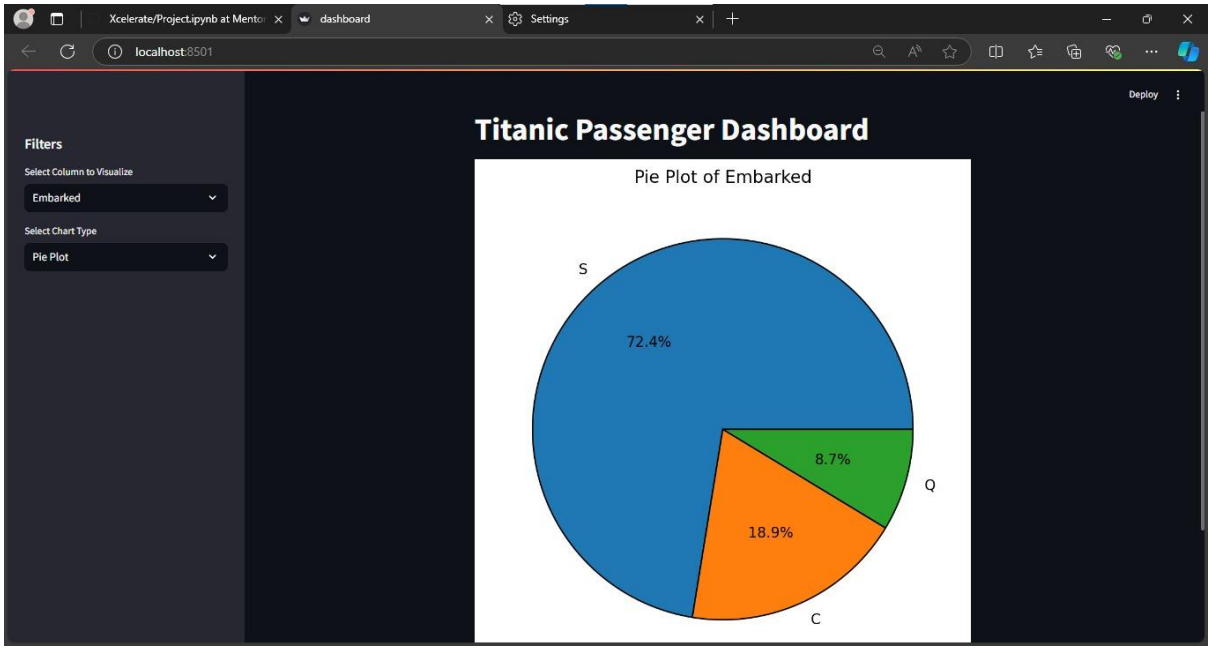
```
    ax.set_title(f'Pie Plot of {column}')
    ax.set_ylabel('')

elif chart_type == 'Histogram':
    df[column].apply(pd.to_numeric,
errors='coerce').dropna().plot(kind='hist', ax=ax)
    ax.set_title(f'Histogram of {column}')
    ax.set_xlabel(column)

elif chart_type == 'Box Plot':
    if pd.api.types.is_numeric_dtype(df[column]):
        sns.boxplot(data=df, x=column, ax=ax)
        ax.set_title(f'Box Plot of {column}')
        ax.set_xlabel(column)
    else:
        st.write("Box Plot is not available for non-numeric columns.")

elif chart_type == 'Count Plot':
    sns.countplot(data=df, x=column, ax=ax)
    ax.set_title(f'Count Plot of {column}')
    ax.set_xlabel(column)
    ax.set_ylabel('Count')

elif chart_type == 'Line Plot':
    if pd.api.types.is_numeric_dtype(df[column]):
        df[column].plot(kind='line', ax=ax)
        ax.set_title(f'Line Plot of {column}')
        ax.set_xlabel('Index')
        ax.set_ylabel(column)
    else:
        st.write("Line Plot is not available for non-numeric columns.")

elif chart_type == 'Heatmap':
    # Create a heatmap of correlations if numeric columns are available
    numeric_df = df.select_dtypes(include=['float64', 'int64'])
    if not numeric_df.empty:
        corr = numeric_df.corr()
        sns.heatmap(corr, annot=True, cmap='coolwarm', ax=ax)
        ax.set_title('Heatmap of Correlations')
    else:
        st.write("Heatmap is not available for non-numeric columns.")

# Display the plot
st.pyplot(fig)
```

# OUTPUTS:-


localhost:8501
Deploy
Filters
Show Data Sample
Select Column to Visualize
Age
PassengerId
Survived
Pclass
Name
Sex
Age
SibSp
Titanic Passenger Dashboard
Data Sample
PassengerId Survived Pclass Name Sex Age
0 1 0 3 Braund, Mr. Owen Harris male 22
1 2 1 1 Cumings, Mrs. John Bradley (Florence Briggs Thayer) female 38
2 3 1 3 Heikkinen, Miss. Laina female 26
3 4 1 1 Futrelle, Mrs. Jacques Heath (Lily May Peel) female 35
4 5 0 3 Allen, Mr. William Henry male 35
Histogram of Age
Frequency
Age


localhost:8501
Deploy
Filters
Show Data Sample
Select Column to Visualize
Sex
Select Chart Type
Bar Plot
Bar Plot
Pie Plot
Histogram
Box Plot
Count Plot
Line Plot
Heatmap
Titanic Passenger Dashboard
Data Sample
PassengerId Survived Pclass Name Sex Age
0 1 0 3 Braund, Mr. Owen Harris male 22
1 2 1 1 Cumings, Mrs. John Bradley (Florence Briggs Thayer) female 38
2 3 1 3 Heikkinen, Miss. Laina female 26
3 4 1 1 Futrelle, Mrs. Jacques Heath (Lily May Peel) female 35
4 5 0 3 Allen, Mr. William Henry male 35
Bar Plot of Sex
Count
male
female
Sex




The above is my dashboard which shows a table and a graph.

We have options in the left column side bar that enable us to change the parameters of the graph and we can change the graph type as well

## CONCLUSION:-

Hence we have studied about the two python libraries matplotlib and streamlit. We have looked at how we need to set-up a virtual environment to run matplotlib. It can plot various graphs in various shades and we have also looked at their basic codes briefly. We have used this knowledge to make a dashboard that has been displayed and explained with it features in the end.