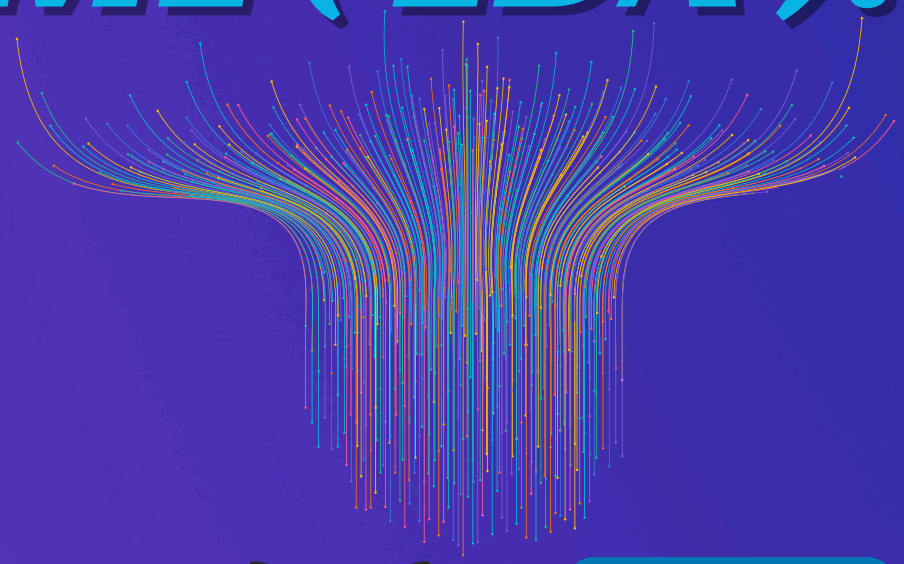
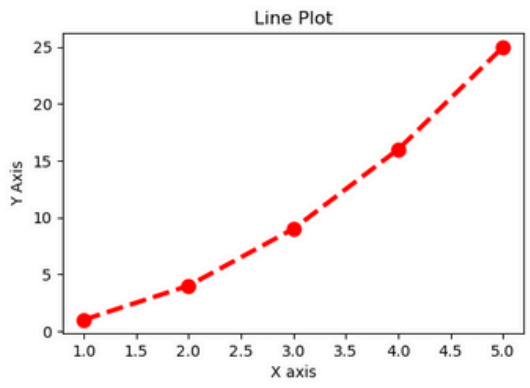
Matplotlib

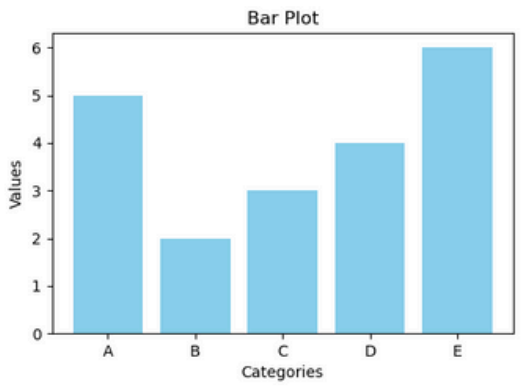
ML (EDA)!

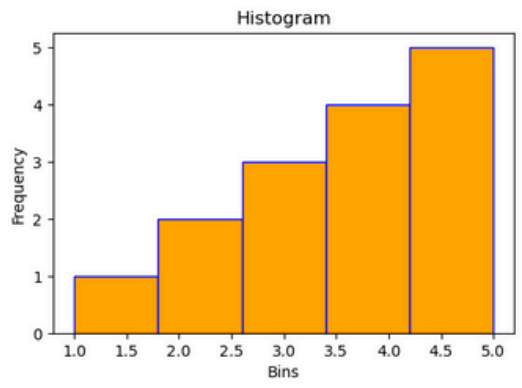


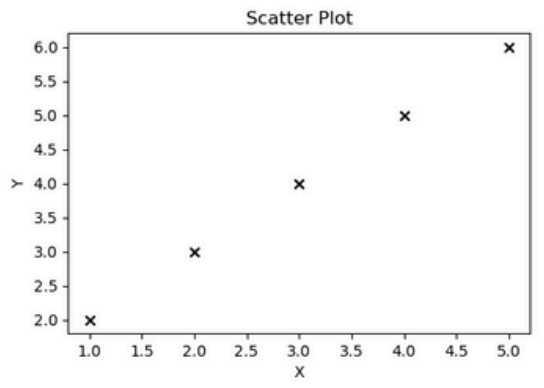
CREATED BY

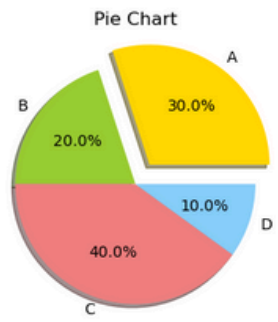
UDAYABHANU NAYAK

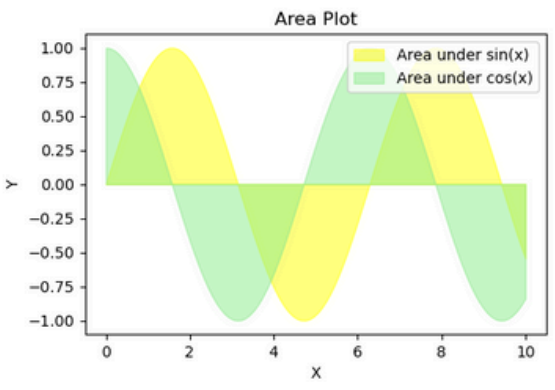

Line Plot
Y Axis
X axis

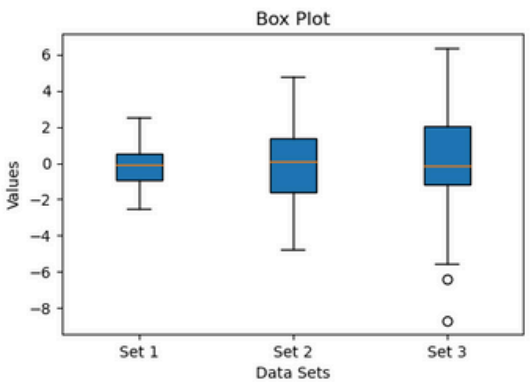

Bar Plot
Values
Categories
A
B
C
D
E

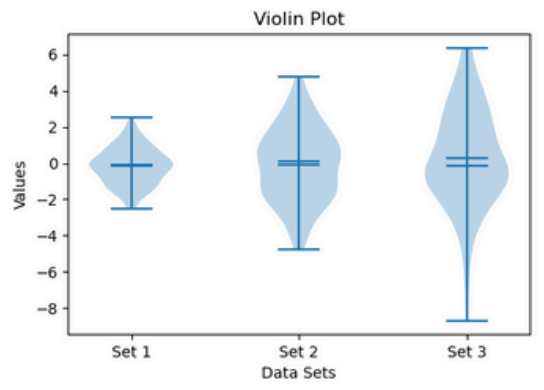

Histogram
Frequency
Bins

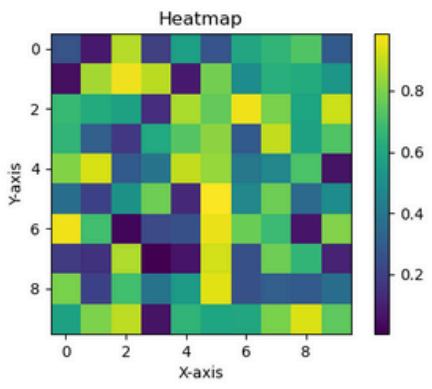

Scatter Plot
Y
X

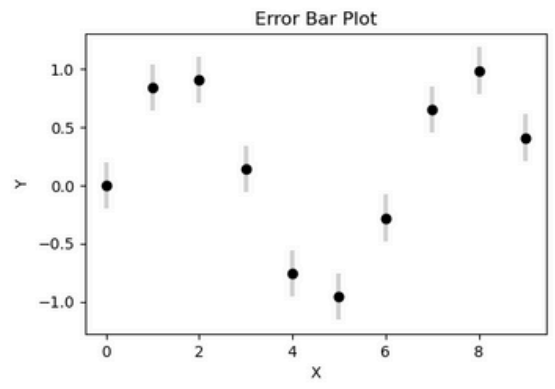

Pie Chart
A
B
C
D
30.0%
20.0%
40.0%
10.0%

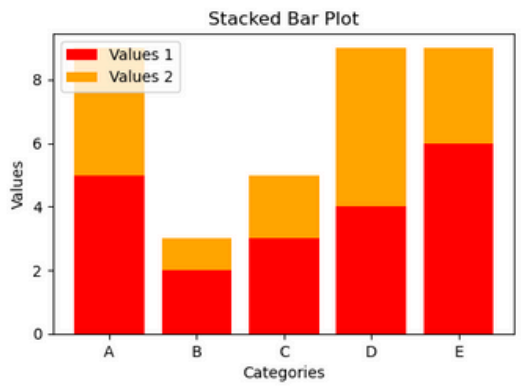

Area Plot
Area under sin(x)
Area under cos(x)
Y
X

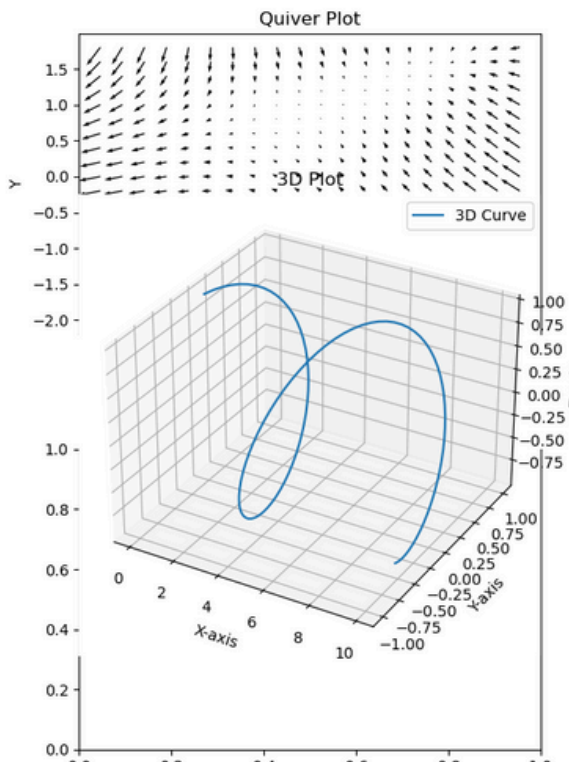

Box Plot
Values
Set 1
Set 2
Set 3
Data Sets

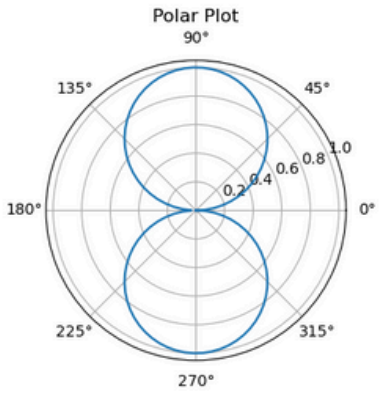

Violin Plot
Values
Set 1
Set 2
Set 3
Data Sets

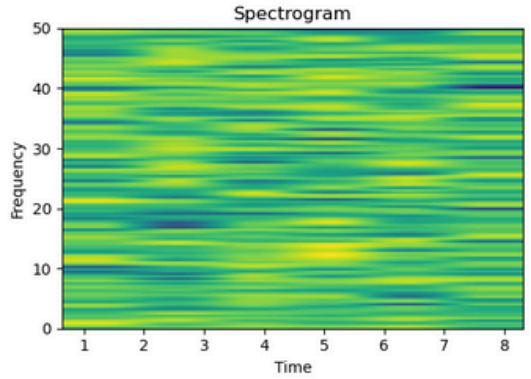

Heatmap
Y-axis
X-axis

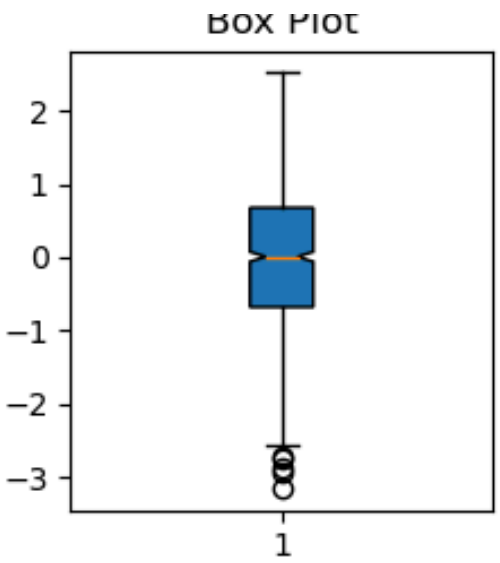

Error Bar Plot
Y
X

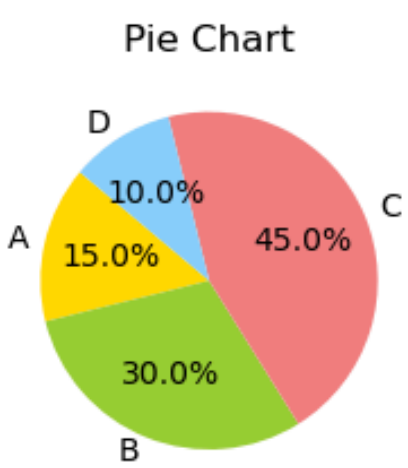

Stacked Bar Plot
Values 1
Values 2
Values
A
B
C
D
E
Categories

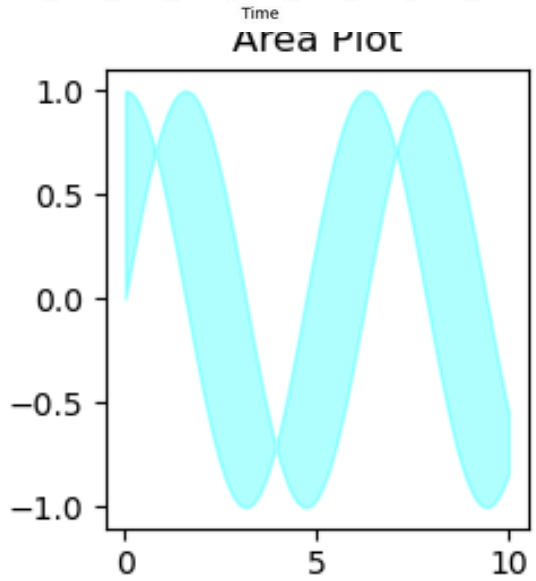

Quiver Plot
3D Plot
3D Curve
X-axis
Y-axis

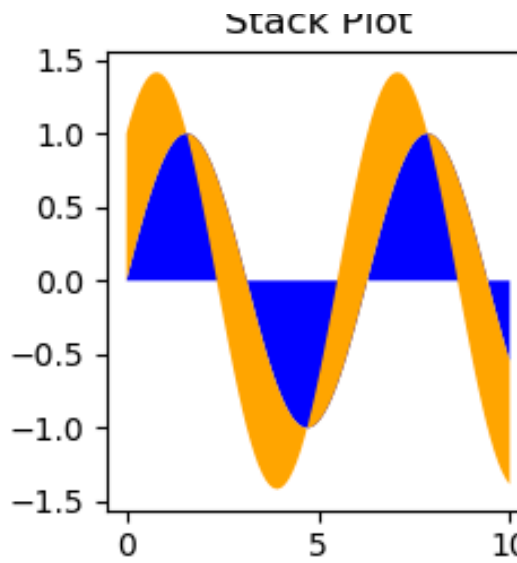

Polar Plot
90°
135°
45°
180°
0°
225°
315°
270°

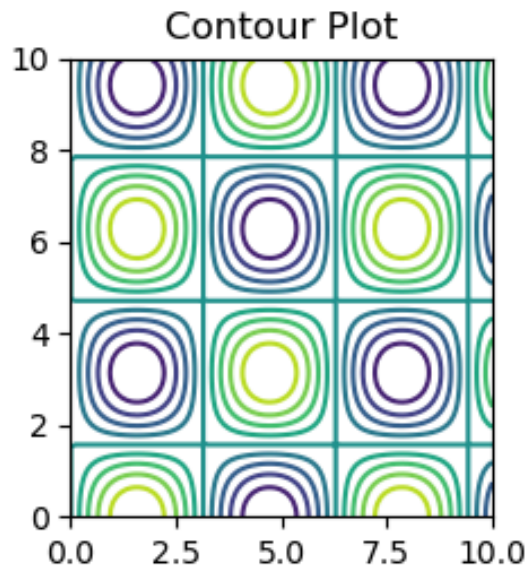

Spectrogram
Frequency
Time

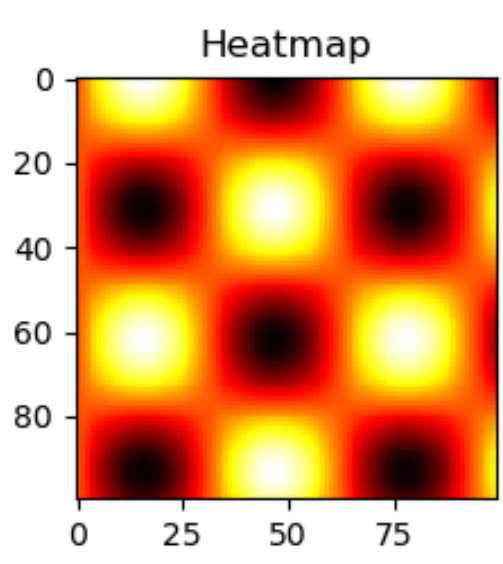

Box Plot

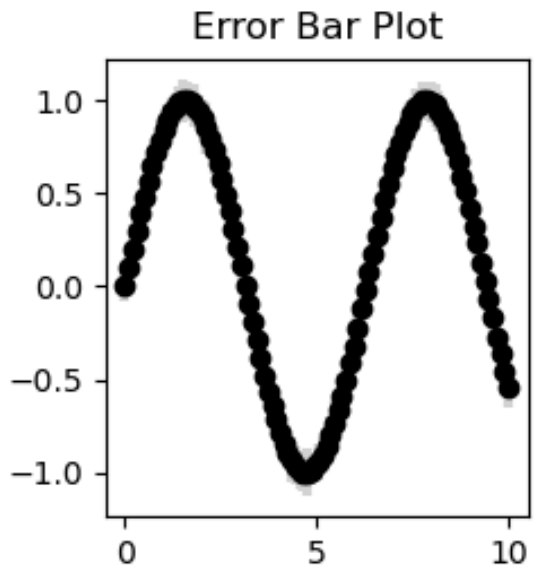

Pie Chart
D
10.0%
C
45.0%
A
15.0%
30.0%
B

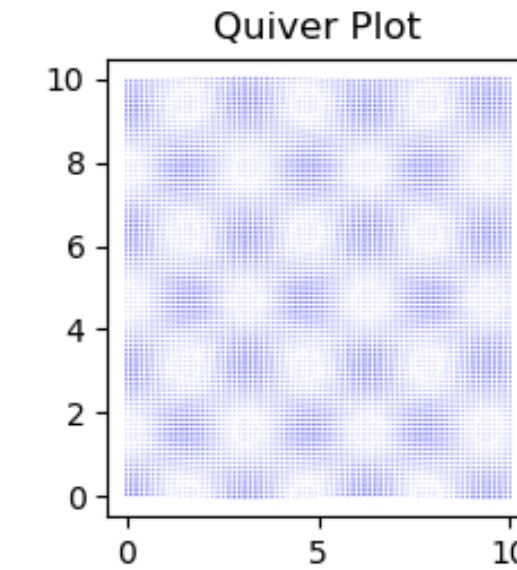

Area Plot


Stack Plot


Contour Plot


Heatmap


Error Bar Plot


Quiver Plot

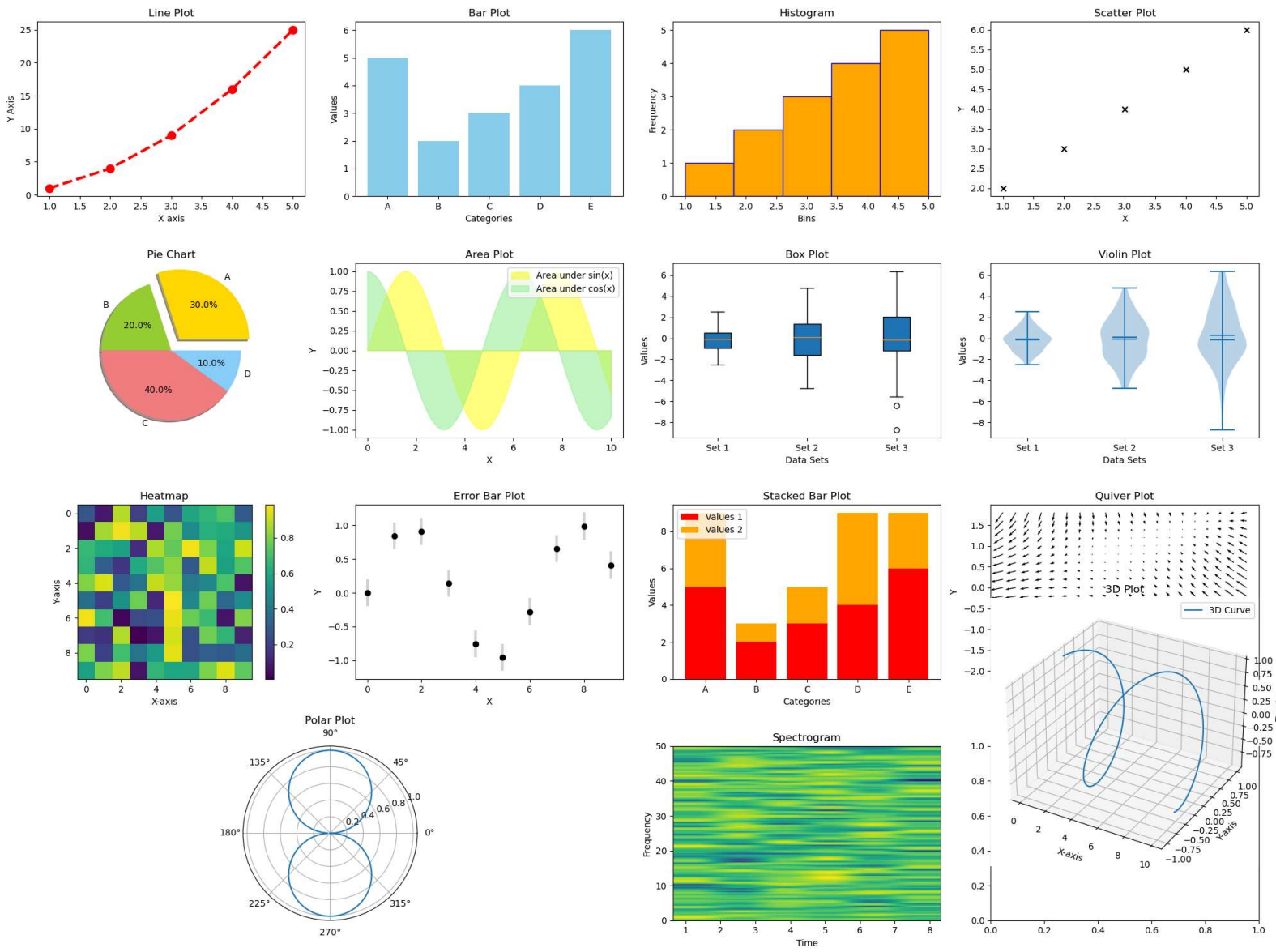
EDA 📊 using Matplotlib

by ✍️ **Udaya**

Tech Stack : Python, matplotlib

Let's create those plots

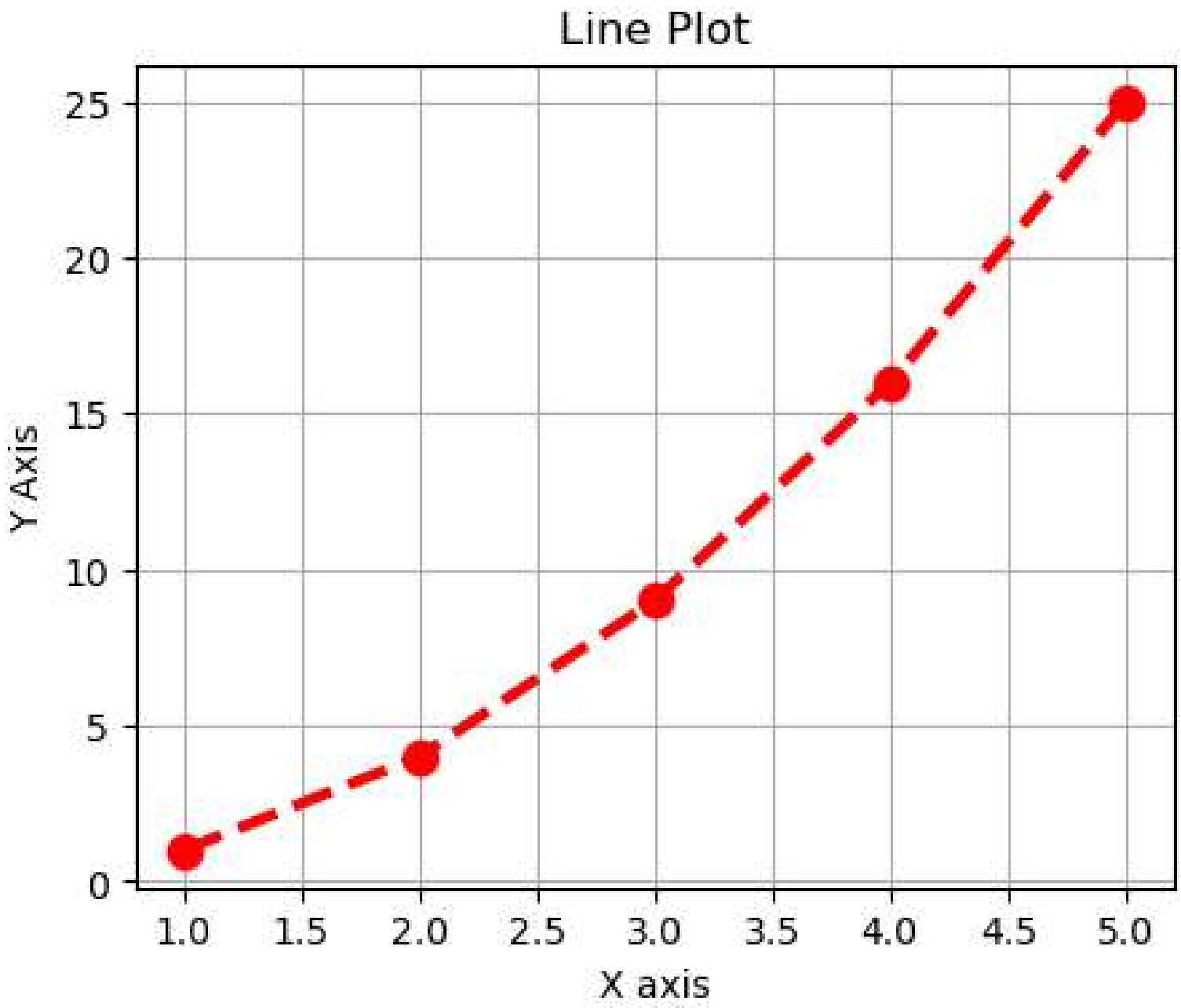
In [1]:
```
from IPython.display import Image, display
image_path = 'C:/Users/udaya/Downloads/all plots.png'

display(Image(filename=image_path))
```



In [2]:
```
import matplotlib.pyplot as plt

import pandas as pd
import numpy as np
```

Line Plot

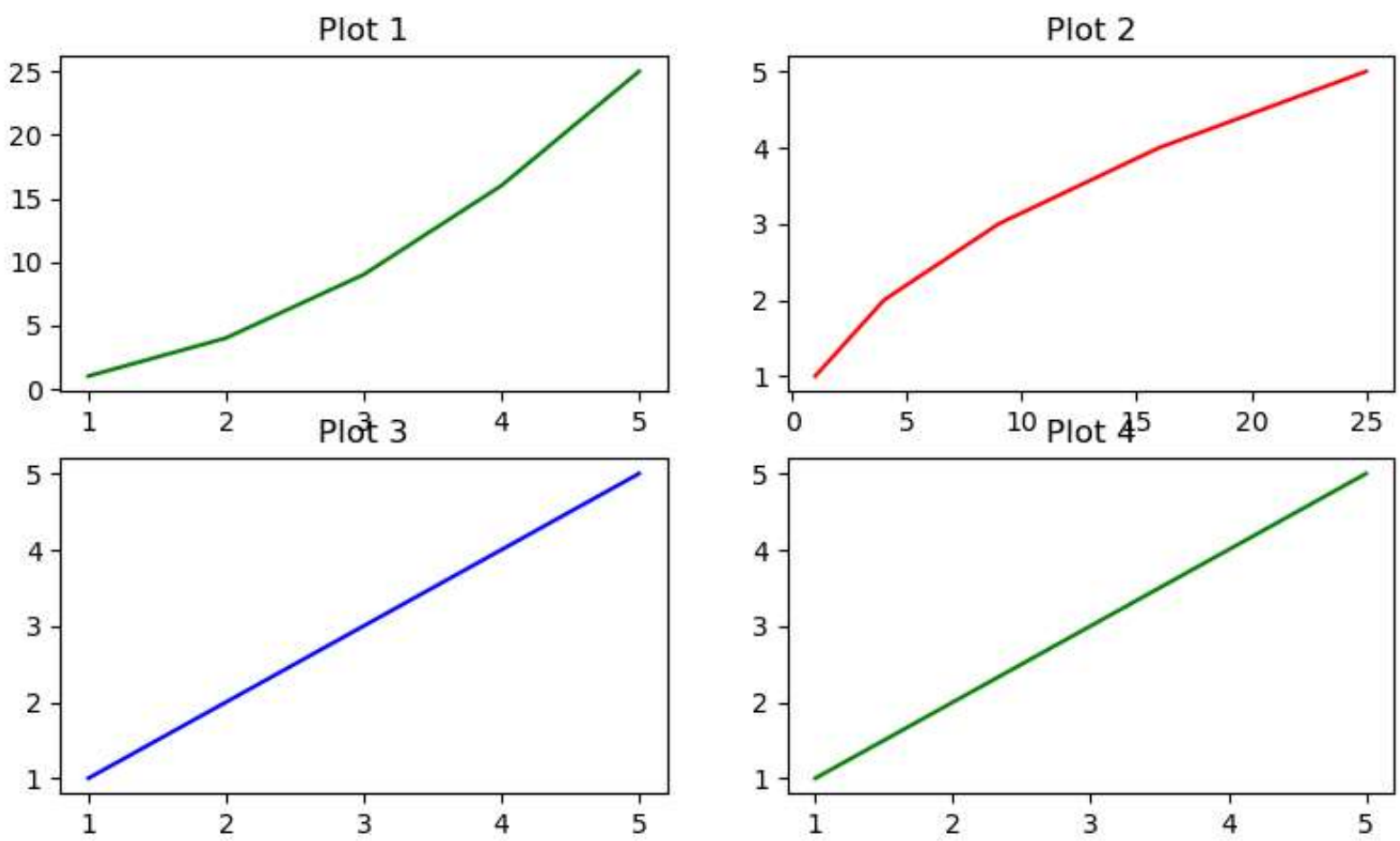
In [3]:
```
x=[1,2,3,4,5]
y=[1,4,9,16,25]

plt.figure(figsize=(5, 4))
plt.plot(x, y, color='red',linestyle='--',marker='o',linewidt
plt.grid()
plt.xlabel('X axis')
plt.ylabel('Y Axis')
plt.title("Line Plot")
plt.show()
```



subplots

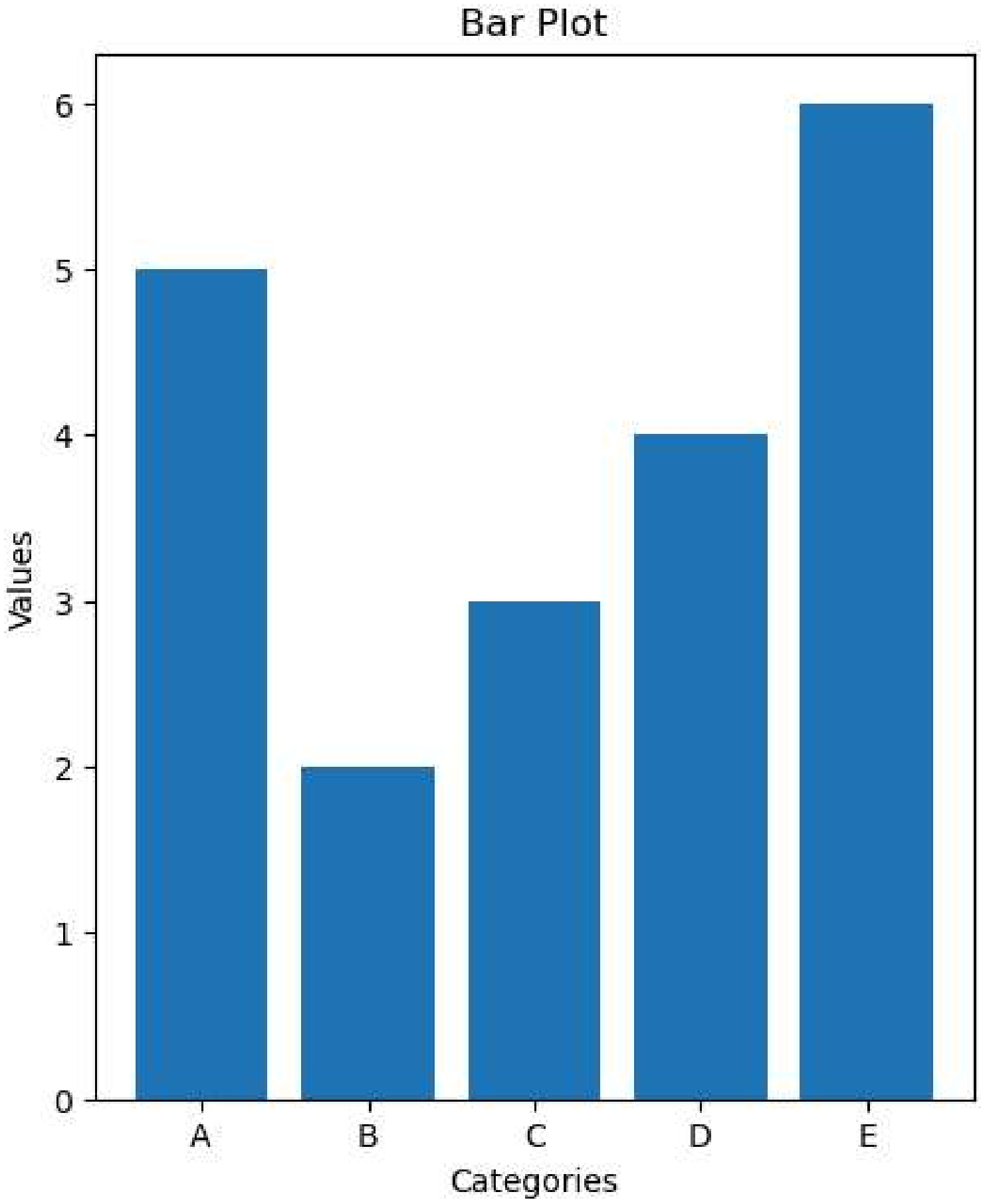
In [4]:
```
x = [1, 2, 3, 4, 5]
y1 = [1, 4, 9, 16, 25]
y2 = [1, 2, 3, 4, 5]

plt.figure(figsize=(9,5))

plt.subplot(2,2,1)
plt.plot(x,y1,color='green')
plt.title("Plot 1")

plt.subplot(2,2,2)
plt.plot(y1,x,color='red')
plt.title("Plot 2")

plt.subplot(2,2,3)
plt.plot(x,y2,color='blue')
plt.title("Plot 3")

plt.subplot(2,2,4)
plt.plot(x,y2,color='green')
plt.title("Plot 4")
```

Out[4]:
```
Text(0.5, 1.0, 'Plot 4')
```



Bar Plot

In [5]:
```
categories=['A','B','C','D','E']
values=[5,2,3,4,6]

plt.figure(figsize=(5,6))
plt.bar(categories,values)
plt.xlabel('Categories')
plt.ylabel('Values')
plt.title('Bar Plot')
plt.show()
```

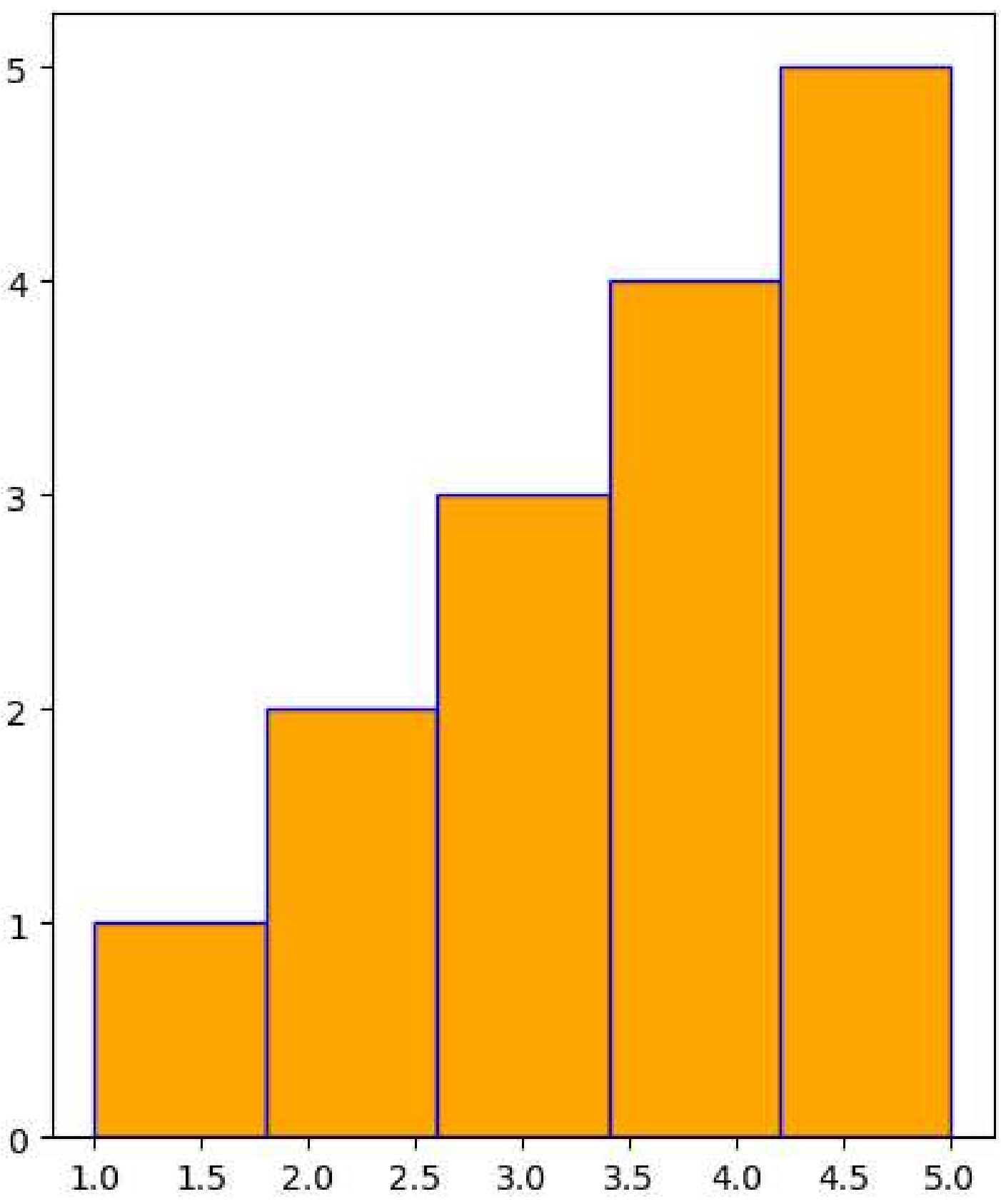


Histogram

Histograms are used to represent the distribution of a dataset. They divide the data into bins and count the number of data points in each bin.

In [6]:
```
data = [1, 2, 2, 3, 3, 3, 4, 4, 4, 4, 5, 5, 5, 5, 5]

plt.figure(figsize=(5,6))
plt.hist(data,bins=5,color='orange',edgecolor='blue')
```

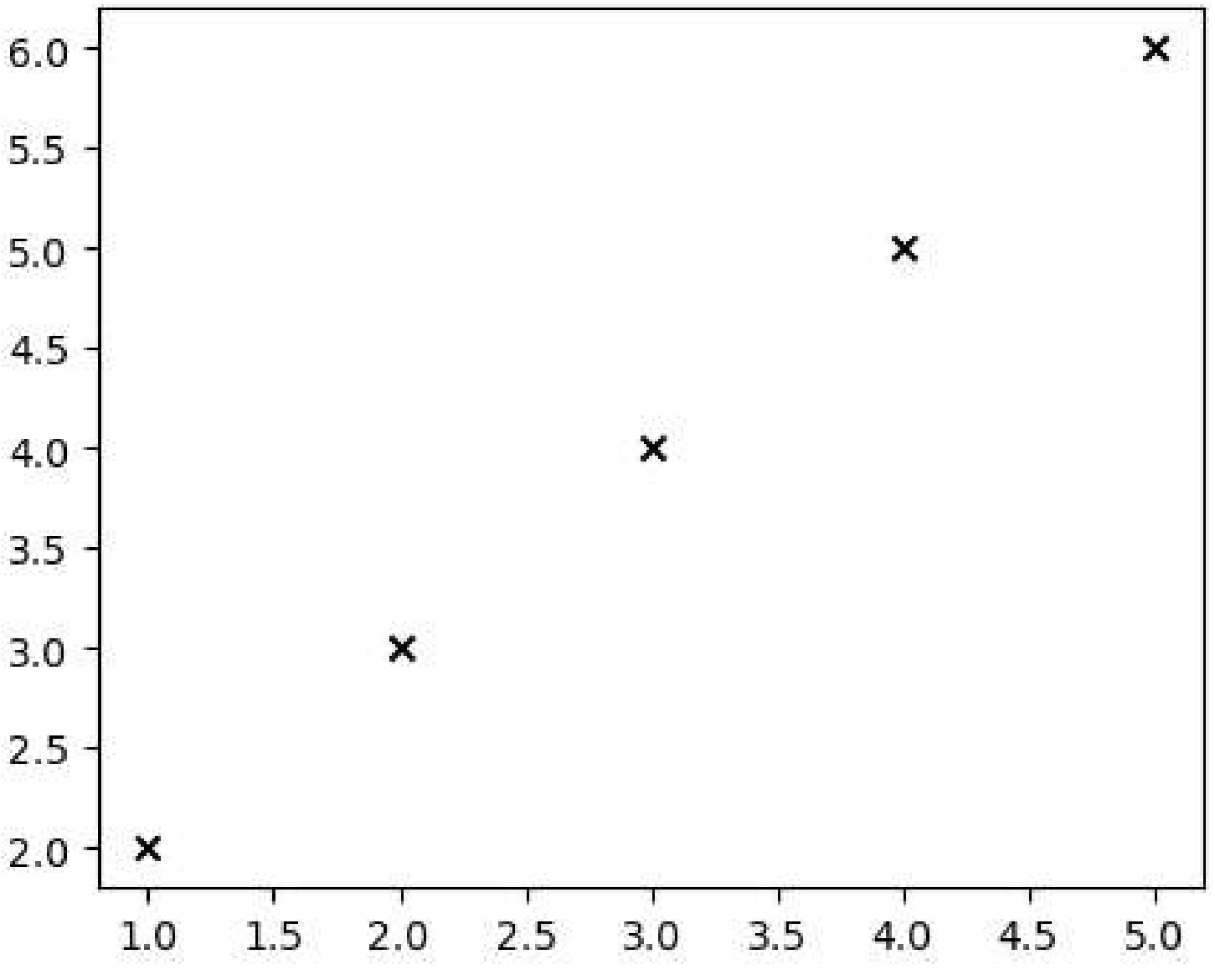
Out[6]:
```
(array([1., 2., 3., 4., 5.]),
 array([1. , 1.8, 2.6, 3.4, 4.2, 5. ]),
 <BarContainer object of 5 artists>)
```



scatter plot

In [7]:
```
x = [1, 2, 3, 4, 5]
y = [2, 3, 4, 5, 6]

plt.figure(figsize=(5,4))
plt.scatter(x, y, color="black", marker='x')
```

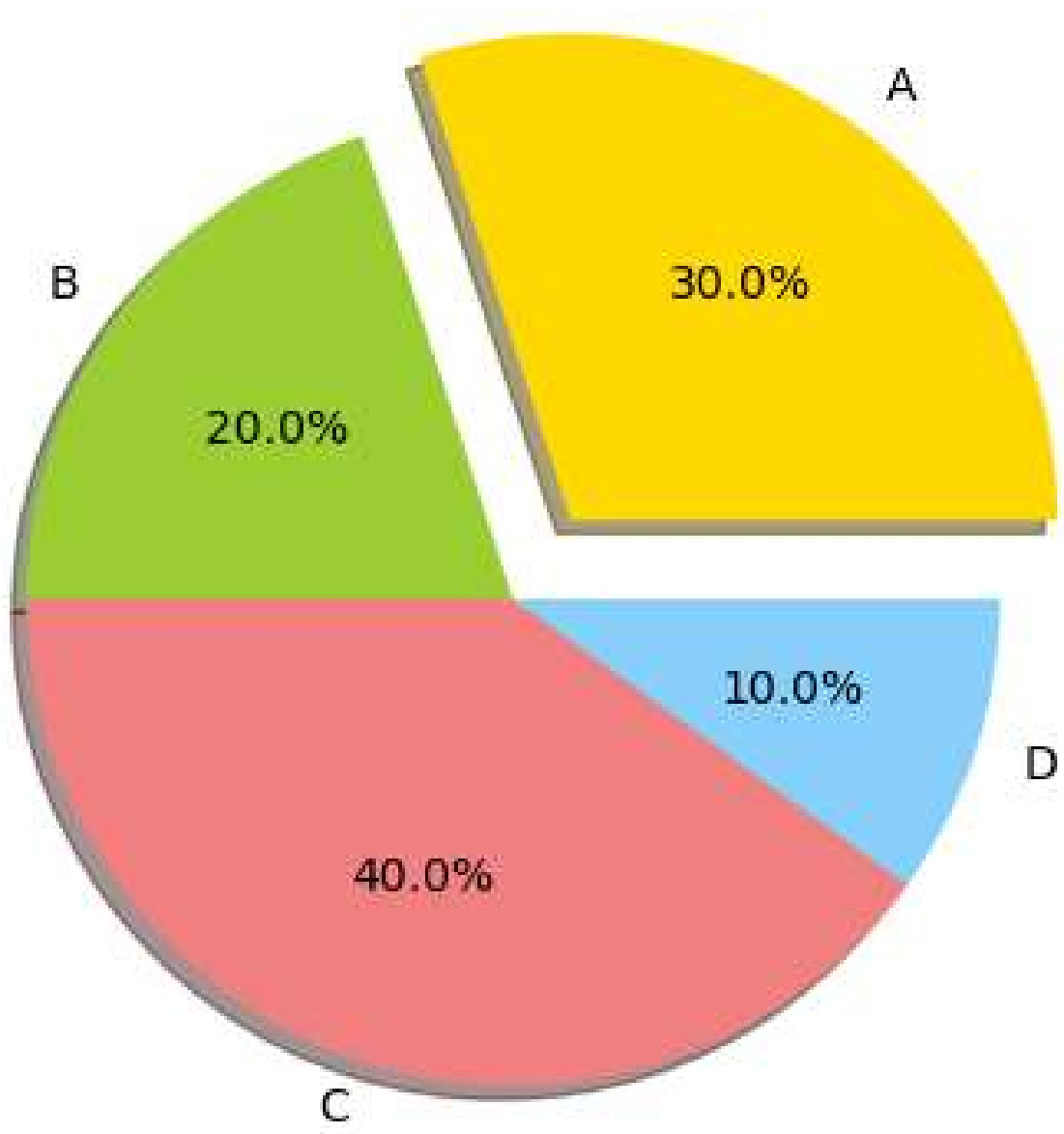
Out[7]: <matplotlib.collections.PathCollection at 0x1e3ec3fee10>



pie chart

In [8]:
```
labels=['A','B','C','D']
sizes=[30,20,40,10]
colors=['gold','yellowgreen','lightcoral','lightskyblue']
explode=(0.2,0,0,0) ## move-out the 1st slice


plt.pie(sizes, explode=explode, labels=labels, colors=colors,
```

Out[8]:
```
([<matplotlib.patches.Wedge at 0x1e3ec0fc250>,
  <matplotlib.patches.Wedge at 0x1e3ec636e10>,
  <matplotlib.patches.Wedge at 0x1e3ec641790>,
  <matplotlib.patches.Wedge at 0x1e3ec643f90>],
 [Text(0.764120788592483, 1.051722121304293, 'A'),
  Text(-0.889918748294541 9, 0.646563702533536 9, 'B'),
  Text(-0.339918576273915 3, -1.046162206115244, 'C'),
  Text(1.046162214071612 7, -0.339918551786720 9, 'D')],
 [Text(0.470228177595374 16, 0.647213613110334 1, '30.0%'),
  Text(-0.485410226342477 3, 0.352671110472838 3, '20.0%'),
  Text(-0.185410132513044 7, -0.570633930608314 9, '40.0%'),
  Text(0.570633934948152 3, -0.185410119156393 22, '10.0%')])
```



In [9]:
```
sales_data_df=pd.read_csv('sales_data.csv')
sales_data_df.head(5)
```

Out[9]:

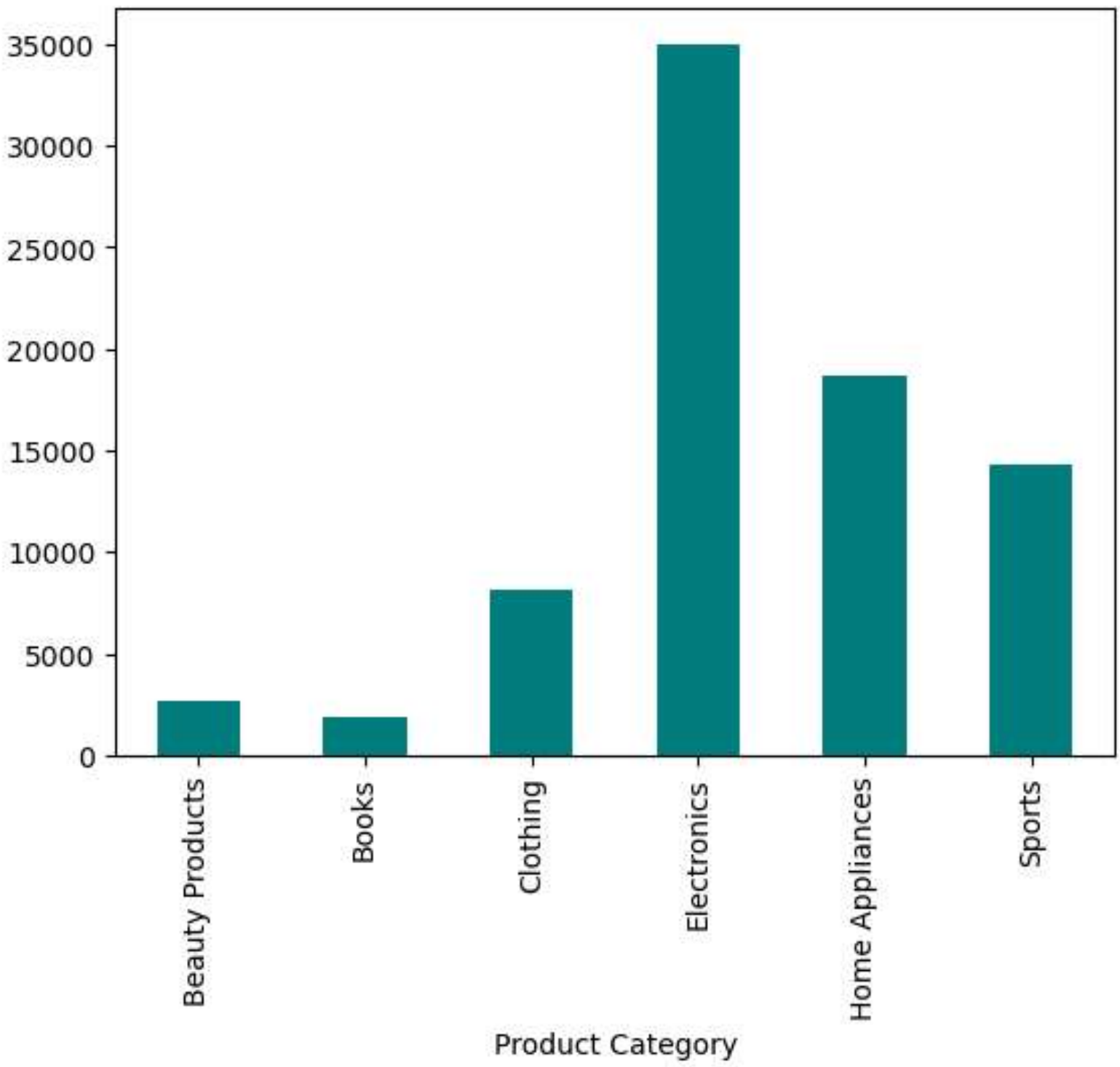
	Transaction ID	Date	Product Category	Product Name	Units Sold	Unit Price	Total Revenue	Reg
0	10001	2024-01-01	Electronics	iPhone 14 Pro	2	999.99	1999.98	N Ame
1	10002	2024-01-02	Home Appliances	Dyson V11 Vacuum	1	499.99	499.99	Eur
2	10003	2024-01-03	Clothing	Levi's 501 Jeans	3	69.99	209.97	A
3	10004	2024-01-04	Books	The Da Vinci Code	4	15.99	63.96	N Ame
4	10005	2024-01-05	Beauty Products	Neutrogena Skincare Set	1	89.99	89.99	Eur

In [10]:
```
total_sales_by_product = sales_data_df.groupby('Product Categ
total_sales_by_product
```

Out[10]:
```
Product Category
Beauty Products     2621.90
Books               1861.93
Clothing            8128.93
Electronics        34982.41
Home Appliances    18646.16
Sports             14326.52
Name: Total Revenue, dtype: float64
```

In [11]:
```
total_sales_by_product.plot(kind='bar',color='teal')
```

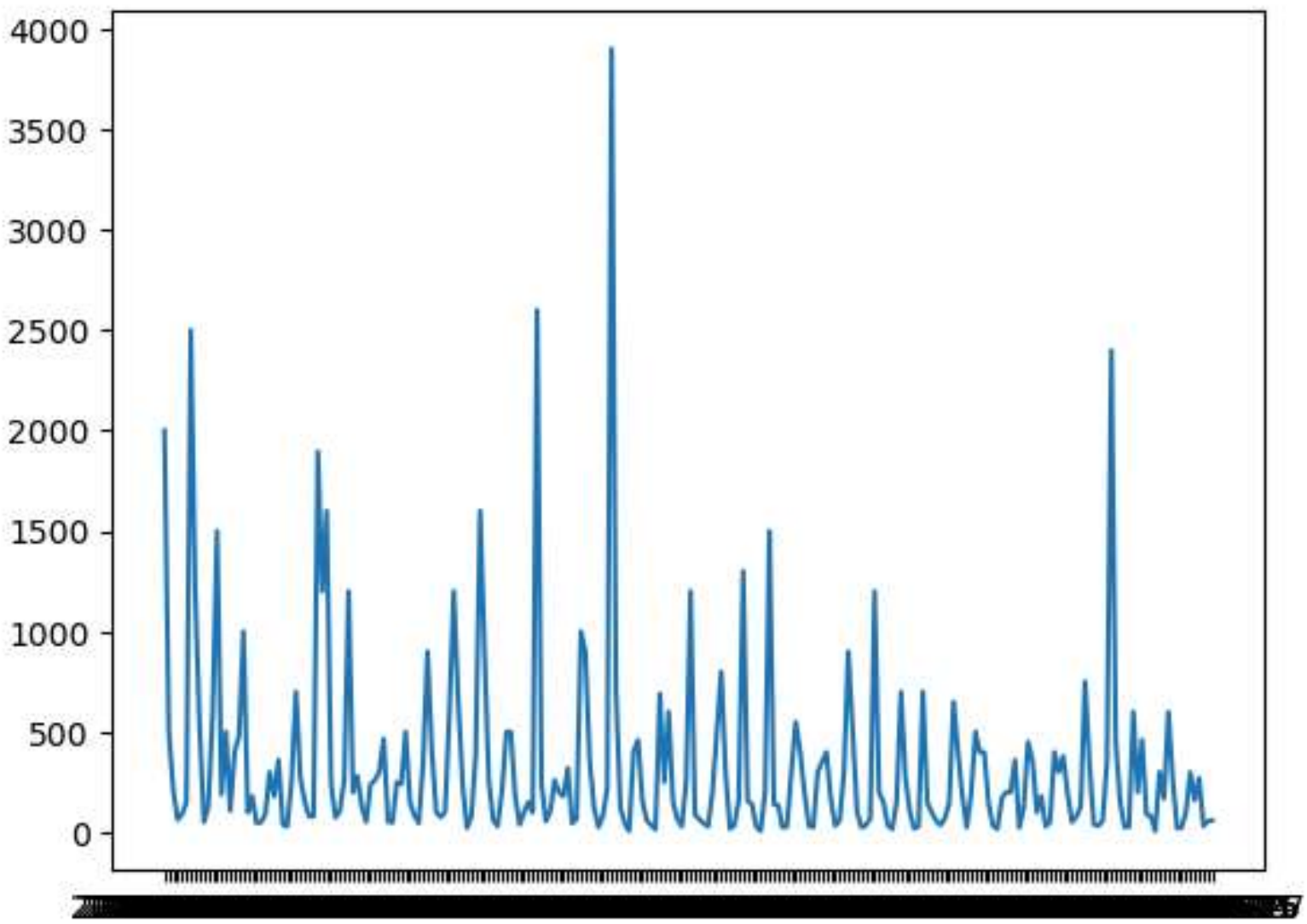
Out[11]: <Axes: xlabel='Product Category'>



sales trend analysis

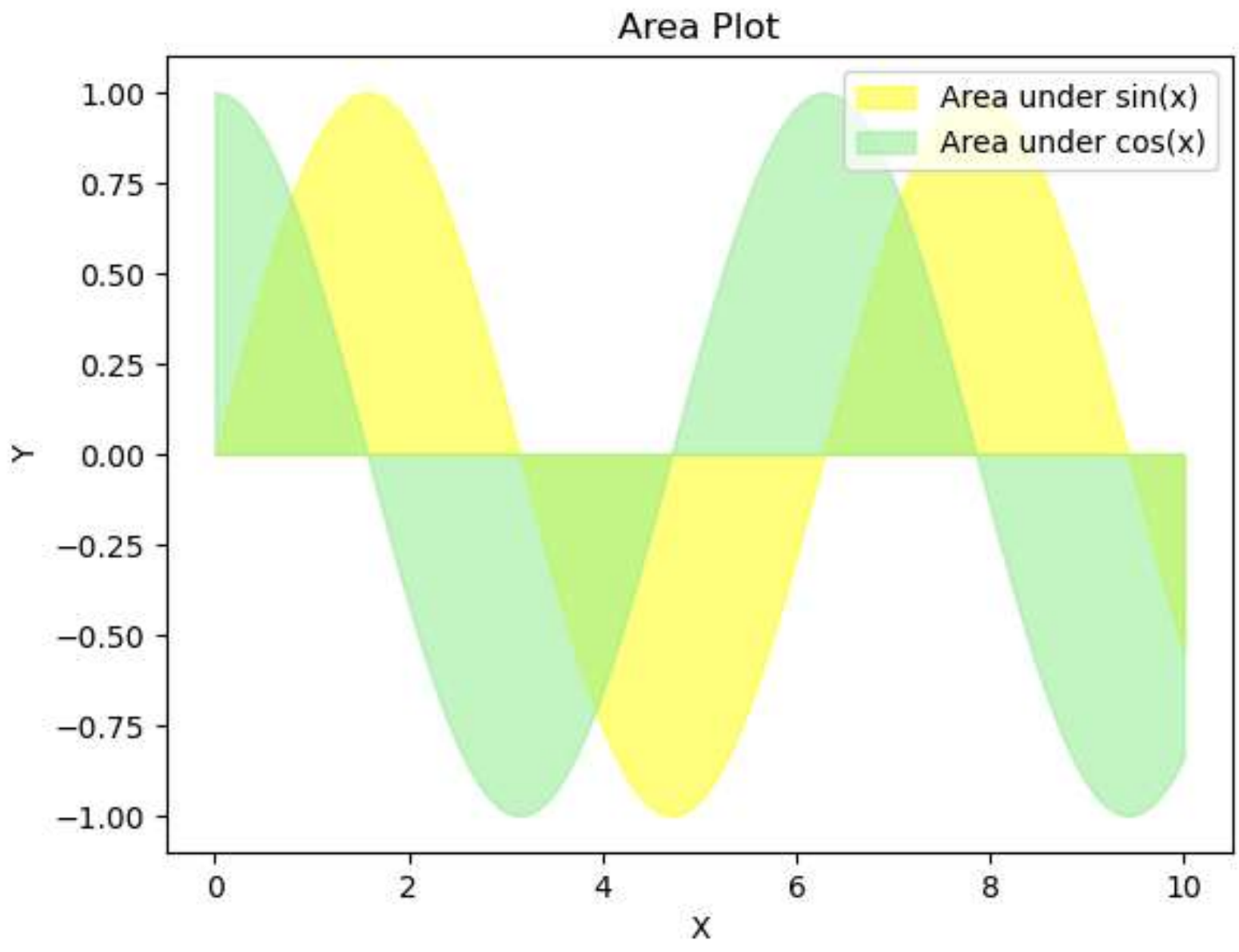
In [12]:
```
sales_trend=sales_data_df.groupby('Date')['Total Revenue'].su
plt.plot(sales_trend['Date'],sales_trend['Total Revenue'])
```

Out[12]:
```
[<matplotlib.lines.Line2D at 0x1e3ec158cd0>]
```



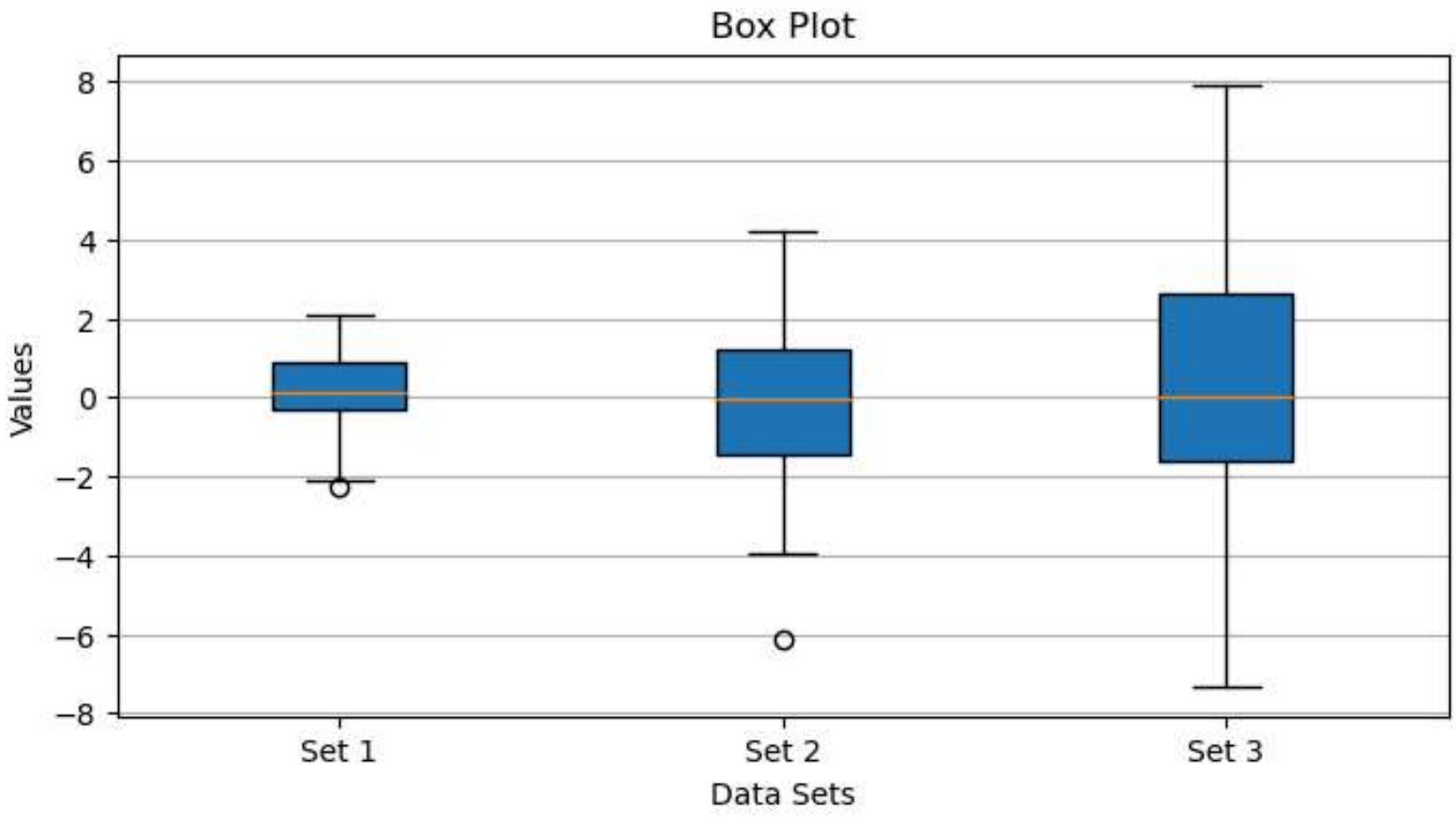
Area Plot

In [13]:
```
x = np.linspace(0, 10, 100)
y1 = np.sin(x)
y2 = np.cos(x)

# Create the area plot
plt.fill_between(x, y1, alpha=0.5, color='yellow', label='Are
plt.fill_between(x, y2, alpha=0.5, color='lightgreen', label=

# Add labels and title
plt.xlabel('X')
plt.ylabel('Y')
plt.title('Area Plot')
plt.legend()

# Show plot
plt.show()
```



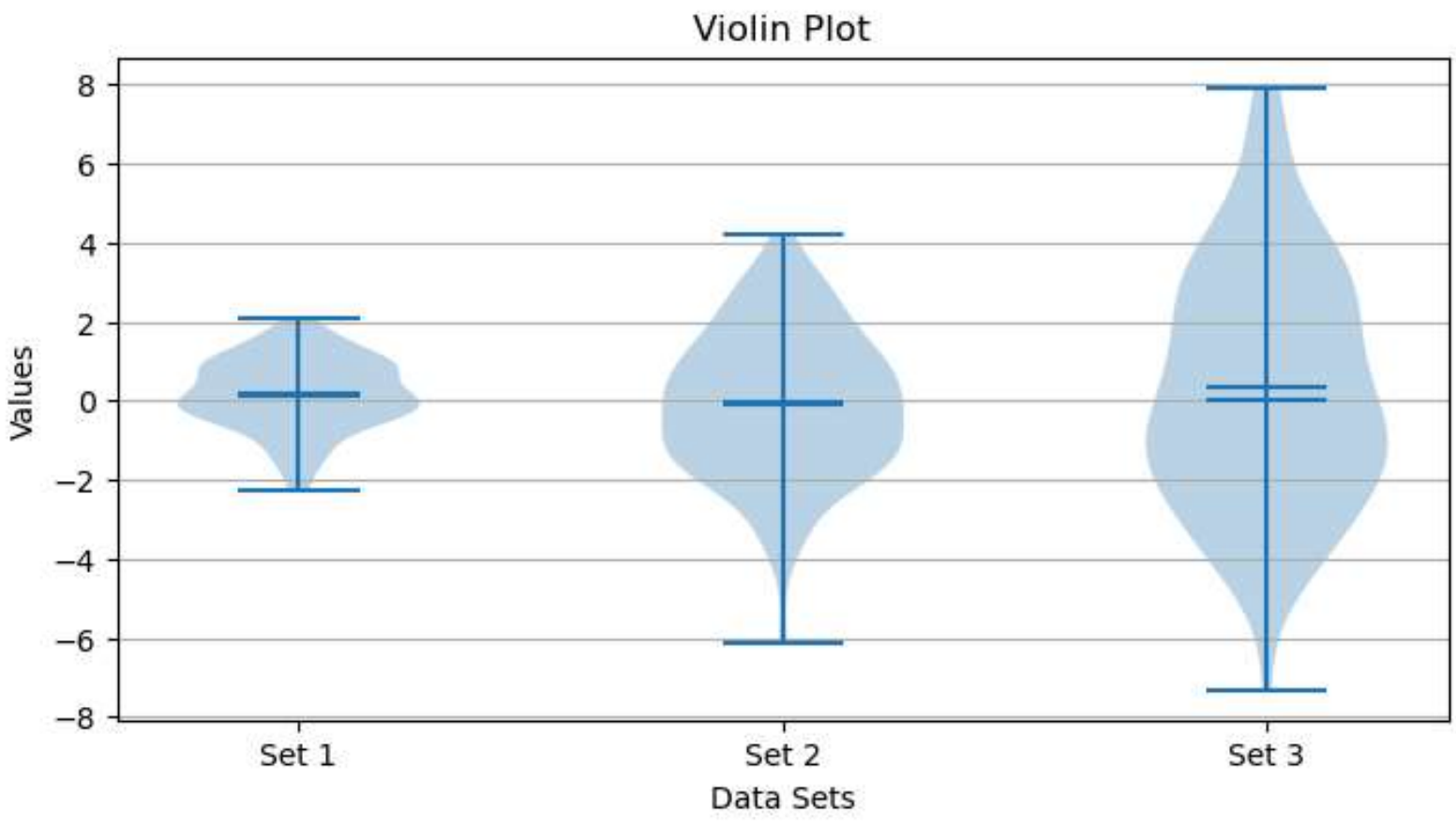
Box Plot

In [14]:
```
data = [np.random.normal(0, std, 100) for std in range(1, 4)]

plt.figure(figsize=(8, 4))
plt.boxplot(data, vert=True, patch_artist=True)
plt.title('Box Plot')
plt.xlabel('Data Sets')
plt.ylabel('Values')
plt.xticks([1, 2, 3], ['Set 1', 'Set 2', 'Set 3'])
plt.grid(axis='y')
plt.show()
```



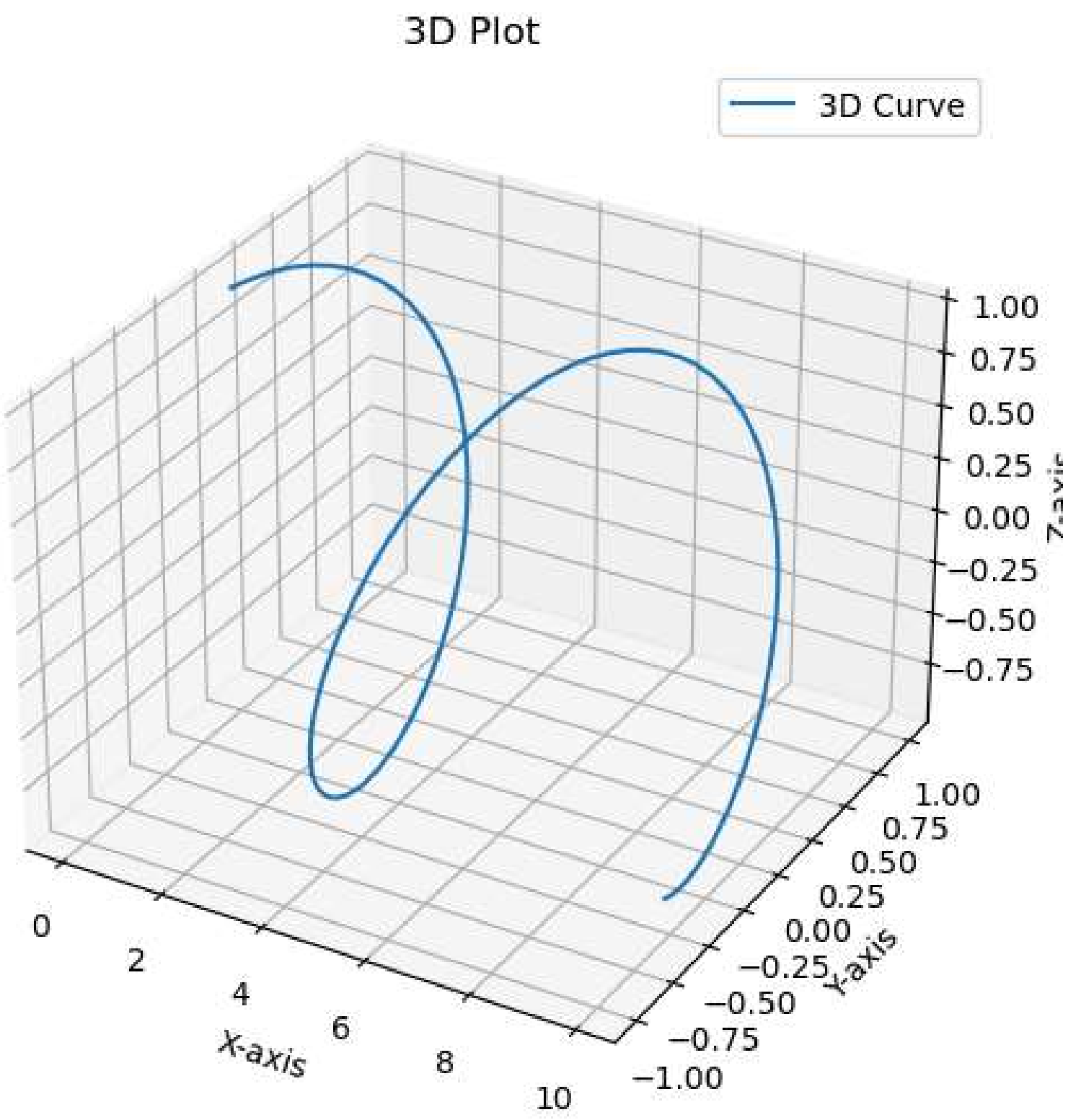
Violin Plot

In [15]:
```
plt.figure(figsize=(8, 4))
plt.violinplot(data, showmeans=True, showmedians=True)
plt.title('Violin Plot')
plt.xlabel('Data Sets')
plt.ylabel('Values')
plt.xticks([1, 2, 3], ['Set 1', 'Set 2', 'Set 3'])
plt.grid(axis='y')
plt.show()
```



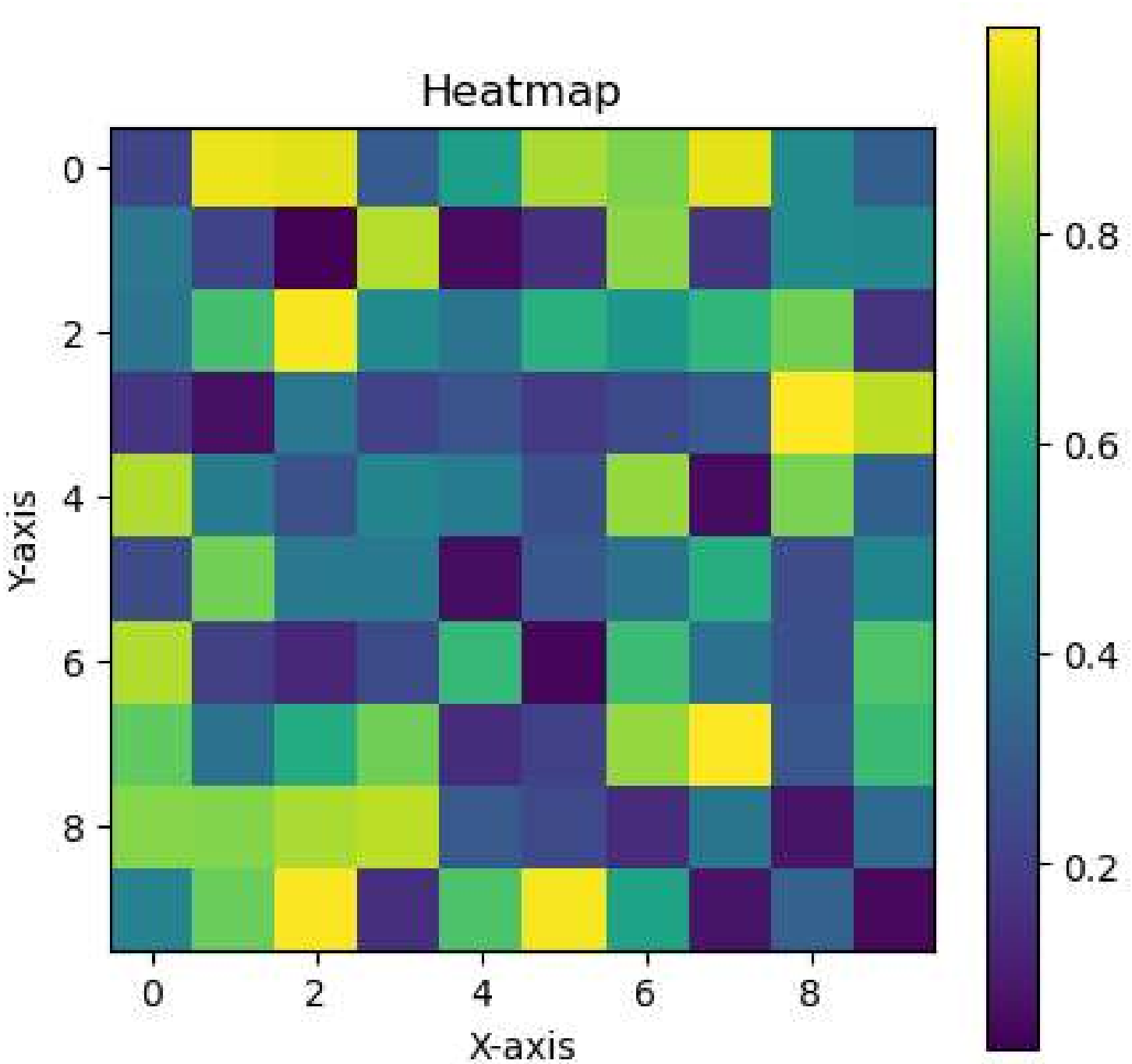
3D Plot

In [16]:
```
from mpl_toolkits.mplot3d import Axes3D

fig = plt.figure(figsize=(8, 6))
ax = fig.add_subplot(111, projection='3d')
x3d = np.linspace(0, 10, 100)
y3d = np.sin(x3d)
z3d = np.cos(x3d)
ax.plot(x3d, y3d, z3d, label='3D Curve')
ax.set_title('3D Plot')
ax.set_xlabel('X-axis')
ax.set_ylabel('Y-axis')
ax.set_zlabel('Z-axis')
ax.legend()
plt.show()
```



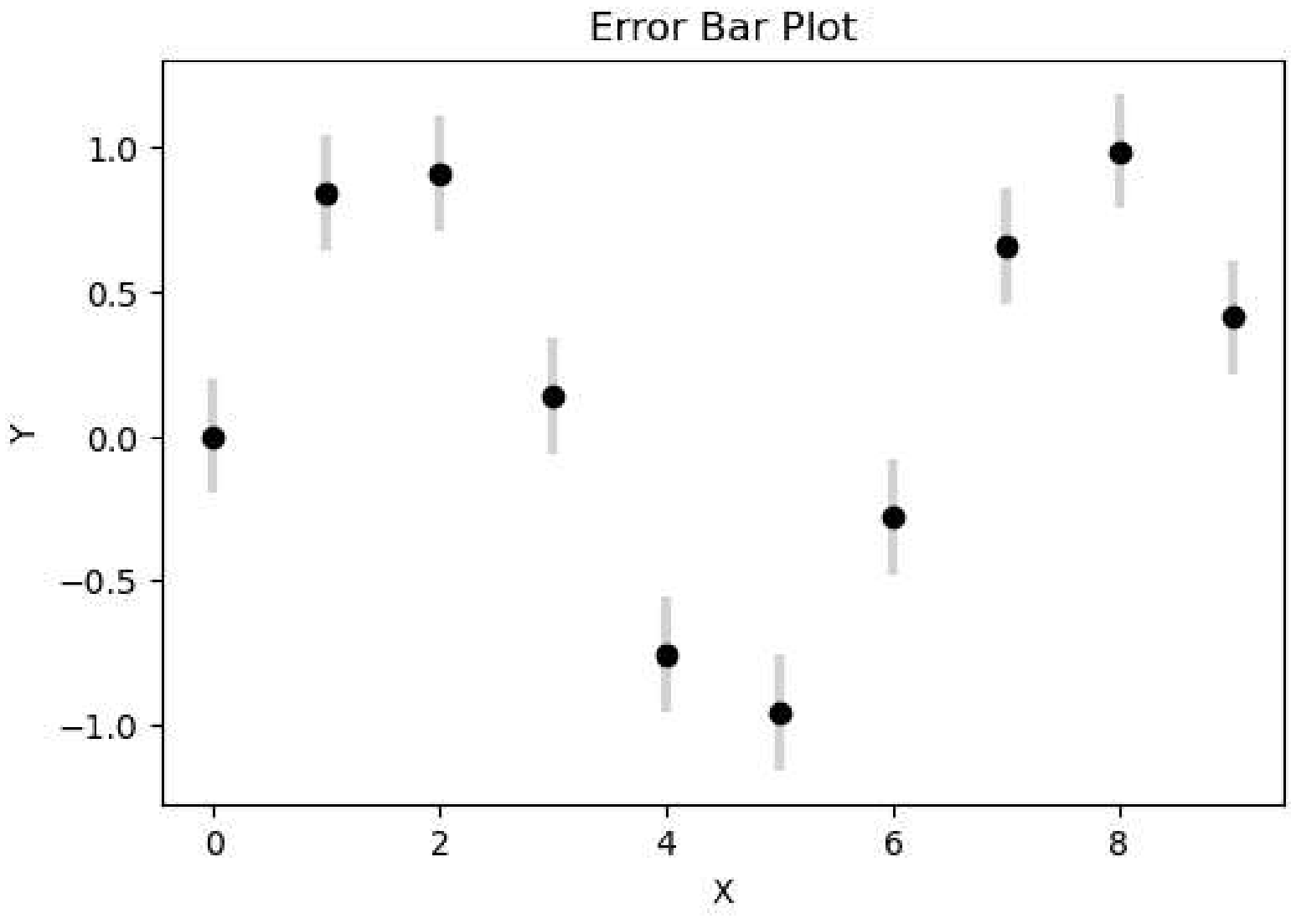
Heatmap

In [17]:
```
data_matrix = np.random.rand(10, 10)

plt.figure(figsize=(5, 5))
plt.imshow(data_matrix, cmap='viridis', interpolation='neares
plt.colorbar()
plt.title('Heatmap')
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.show()
```



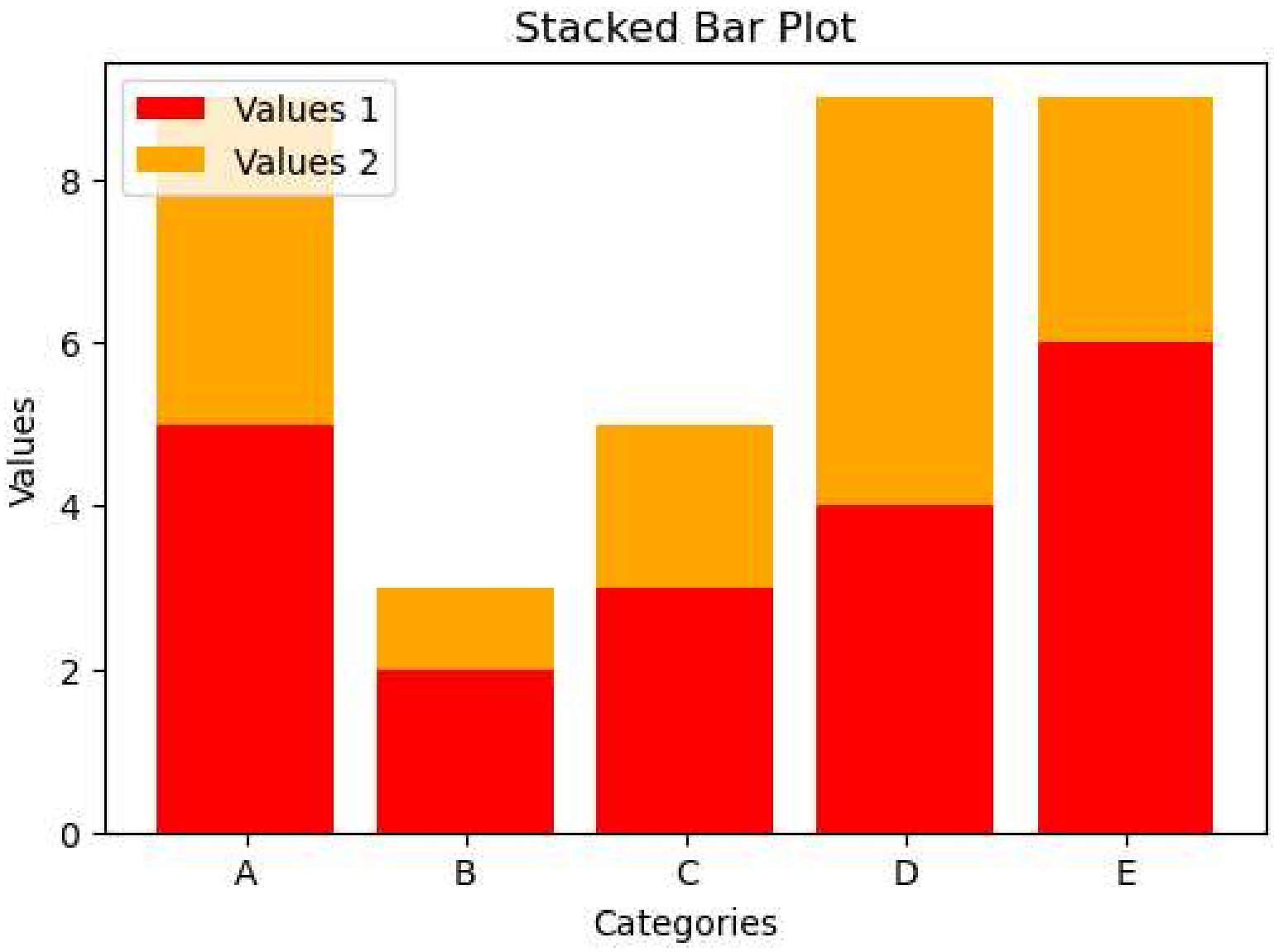
Error Bar Plot

In [18]:
```
x = np.arange(0, 10, 1)
y = np.sin(x)
yerr = 0.2

plt.figure(figsize=(6, 4))
plt.errorbar(x, y, yerr=yerr, fmt='o', color='black', ecolor=
plt.xlabel('X')
plt.ylabel('Y')
plt.title('Error Bar Plot')
plt.show()
```



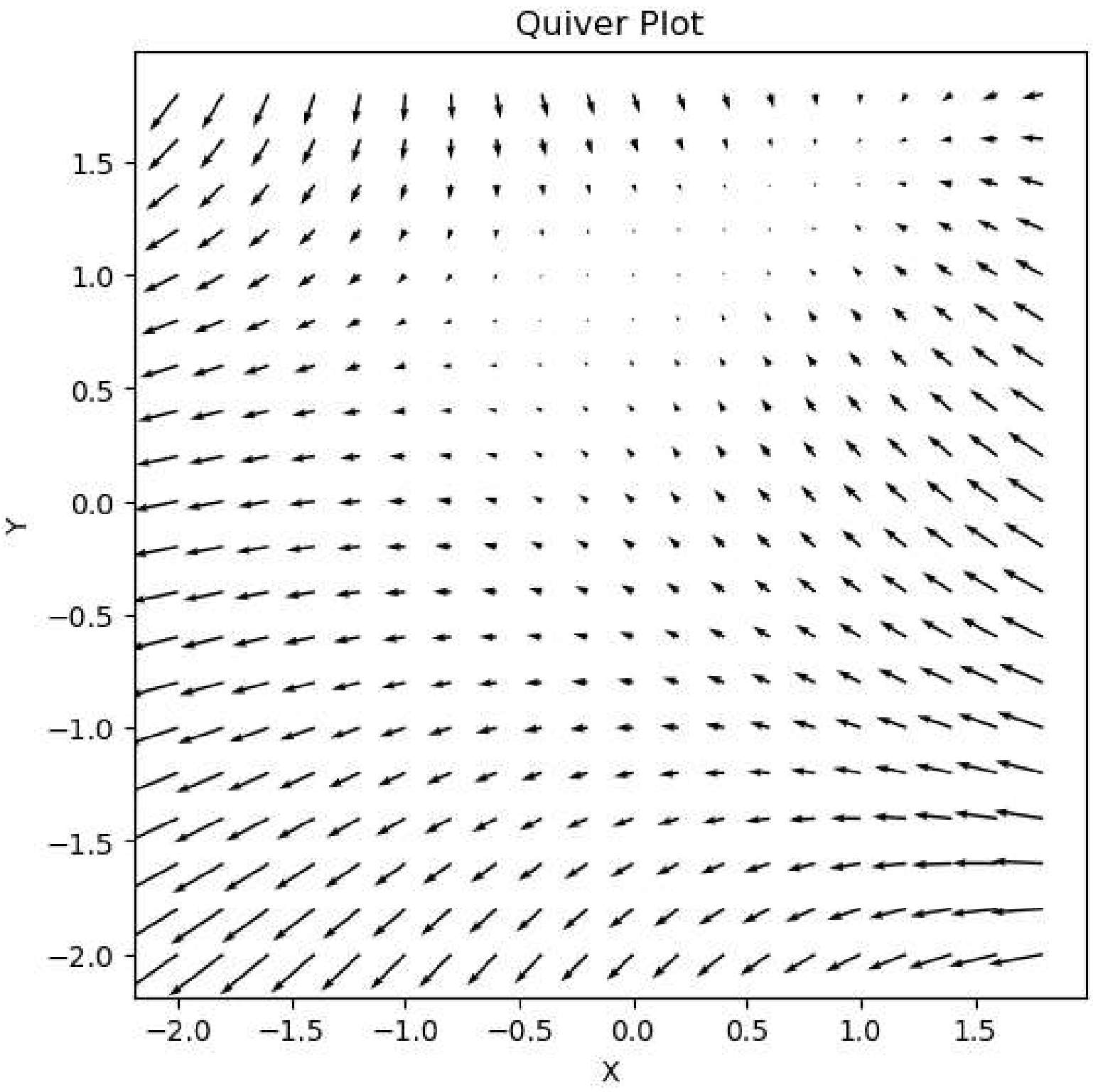
Stacked Bar Plot

In [19]:
```
categories = ['A', 'B', 'C', 'D', 'E']
values1 = [5, 2, 3, 4, 6]
values2 = [4, 1, 2, 5, 3]

plt.figure(figsize=(6, 4))
plt.bar(categories, values1, color='red', label='Values 1')
plt.bar(categories, values2, bottom=values1, color='orange',
plt.xlabel('Categories')
plt.ylabel('Values')
plt.title('Stacked Bar Plot')
plt.legend()
plt.show()
```



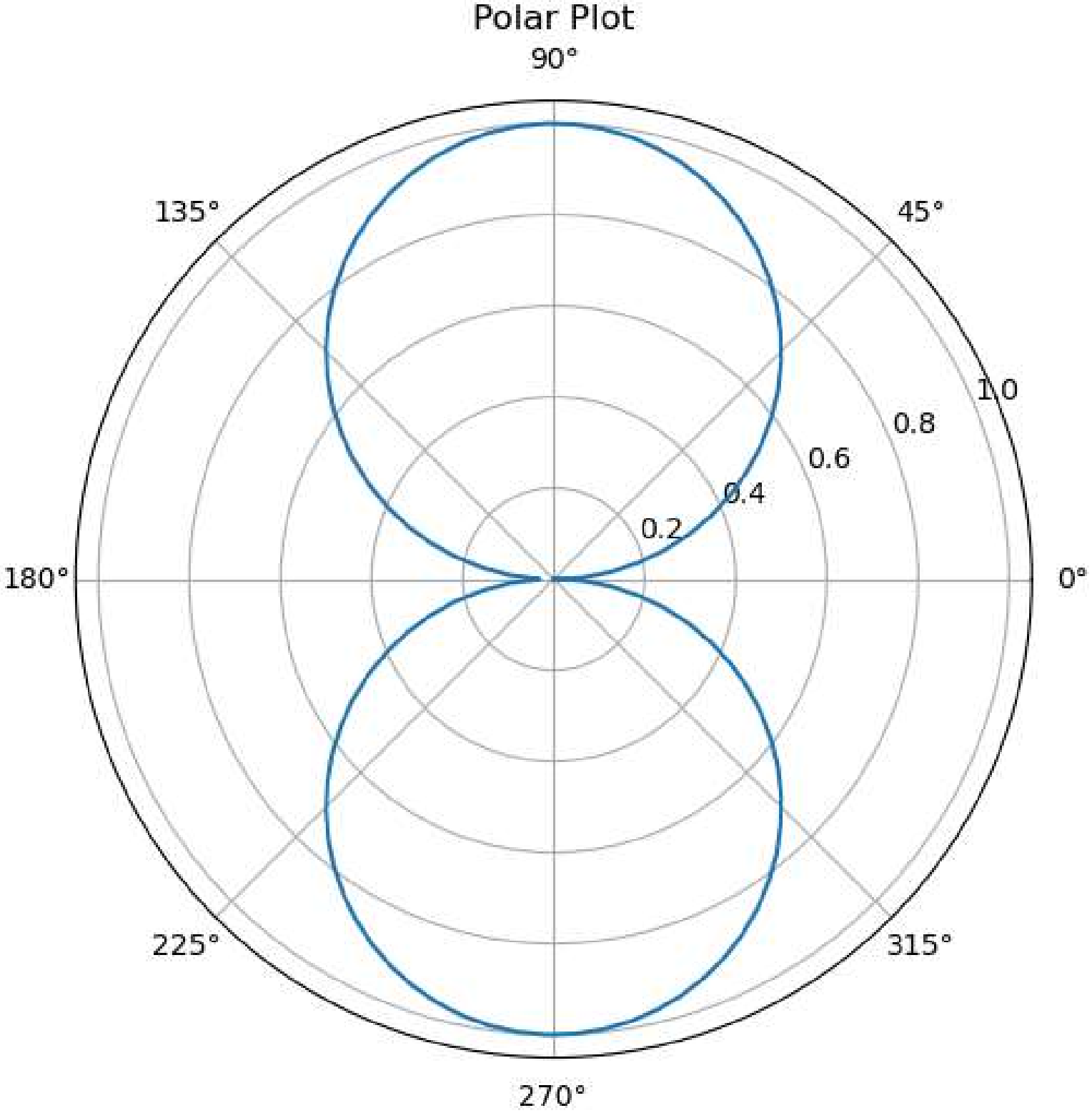
Quiver Plot

In [20]:
```
X, Y = np.meshgrid(np.arange(-2, 2, 0.2), np.arange(-2, 2, 0.
U = -1 - X**2 + Y
V = 1 + X - Y**2

plt.figure(figsize=(6, 6))
plt.quiver(X, Y, U, V)
plt.xlabel('X')
plt.ylabel('Y')
plt.title('Quiver Plot')
plt.show()
```



Polar Plot

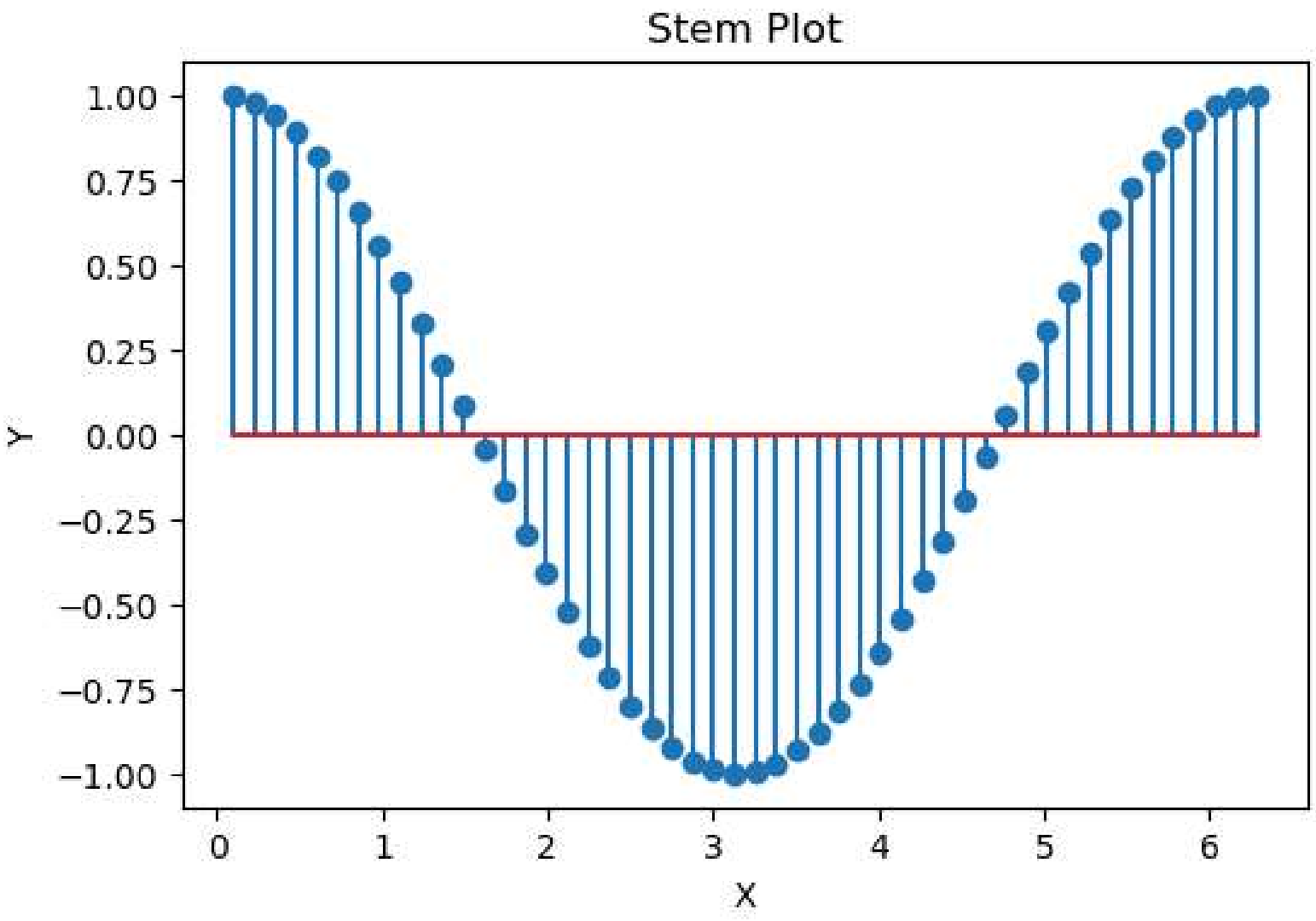
In [21]:
```
theta = np.linspace(0, 2*np.pi, 100)
r = np.abs(np.sin(theta))

plt.figure(figsize=(6, 6))
ax = plt.subplot(111, polar=True)
ax.plot(theta, r)
ax.set_title('Polar Plot')
plt.show()
```



Stem Plot

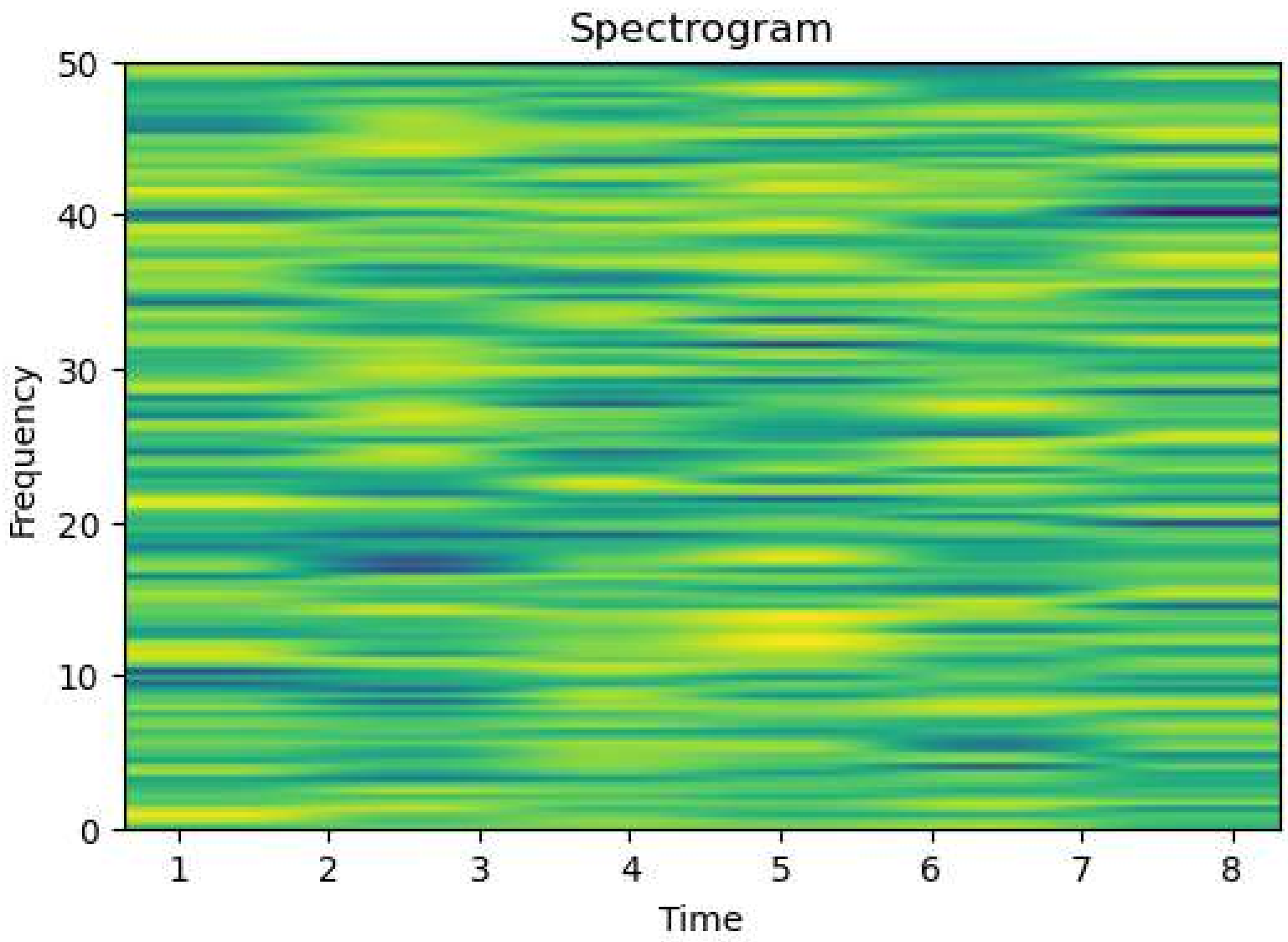
In [22]:
```
x = np.linspace(0.1, 2 * np.pi, 50)
y = np.cos(x)

plt.figure(figsize=(6, 4))
plt.stem(x, y, use_line_collection=True)
plt.xlabel('X')
plt.ylabel('Y')
plt.title('Stem Plot')
plt.show()
```

```
C:\Users\udaya\AppData\Local\Temp\ipykernel_12284\112611512
2.py:5: MatplotlibDeprecationWarning: The 'use_line_collecti
on' parameter of stem() was deprecated in Matplotlib 3.6 and
will be removed two minor releases later. If any parameter f
ollows 'use_line_collection', they should be passed as keywo
rd, not positionally.
  plt.stem(x, y, use_line_collection=True)
```



Spectrogram

In [23]:
```
np.random.seed(0)
dt = 0.01
t = np.arange(0.0, 10.0, dt)
s1 = np.sin(2 * np.pi * 100 * t)
s2 = 2 * np.sin(2 * np.pi * 400 * t)
s = s1 + s2
s = s + np.random.randn(len(t))

plt.figure(figsize=(6, 4))
plt.specgram(s, NFFT=256, Fs=1/dt, noverlap=128)
plt.xlabel('Time')
plt.ylabel('Frequency')
plt.title('Spectrogram')
plt.show()
```



Thank You 🙏 🤱

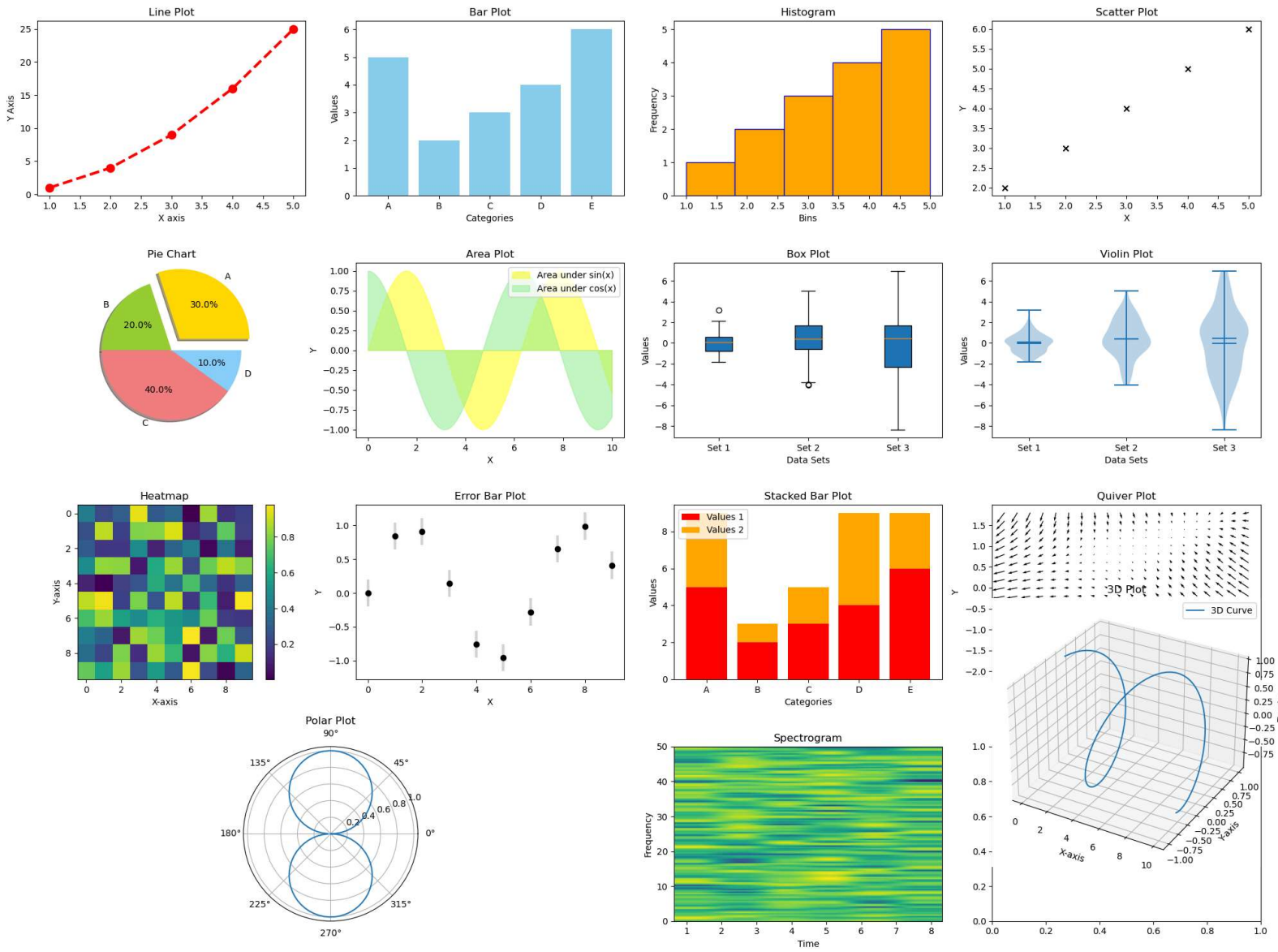
All In a single chart

In [24]:
```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
from mpl_toolkits.mplot3d import Axes3D

# Creating a figure with multiple subplots
fig, axs = plt.subplots(4, 4, figsize=(20, 15))

# Adjust the layout
plt.subplots_adjust(hspace=0.4, wspace=0.4)

# Line Plot
x = [1, 2, 3, 4, 5]
y = [1, 4, 9, 16, 25]
axs[0, 0].plot(x, y, color='red', linestyle='--', marker='o',
axs[0, 0].set_title("Line Plot")
axs[0, 0].set_xlabel('X axis')
axs[0, 0].set_ylabel('Y Axis')

# Bar Plot
categories = ['A', 'B', 'C', 'D', 'E']
values = [5, 2, 3, 4, 6]
axs[0, 1].bar(categories, values, color='skyblue')
axs[0, 1].set_title('Bar Plot')
axs[0, 1].set_xlabel('Categories')
axs[0, 1].set_ylabel('Values')

# Histogram
data = [1, 2, 2, 3, 3, 3, 4, 4, 4, 4, 5, 5, 5, 5, 5]
axs[0, 2].hist(data, bins=5, color='orange', edgecolor='blue'
axs[0, 2].set_title('Histogram')
axs[0, 2].set_xlabel('Bins')
axs[0, 2].set_ylabel('Frequency')

# Scatter Plot
x = [1, 2, 3, 4, 5]
y = [2, 3, 4, 5, 6]
axs[0, 3].scatter(x, y, color="black", marker='x')
axs[0, 3].set_title('Scatter Plot')
axs[0, 3].set_xlabel('X')
axs[0, 3].set_ylabel('Y')

# Pie Chart
labels = ['A', 'B', 'C', 'D']
sizes = [30, 20, 40, 10]
colors = ['gold', 'yellowgreen', 'lightcoral', 'lightskyblue'
explode = (0.2, 0, 0, 0)
axs[1, 0].pie(sizes, explode=explode, labels=labels, colors=c
axs[1, 0].set_title('Pie Chart')

# Area Plot
```

```
x = np.linspace(0, 10, 100)
y1 = np.sin(x)
y2 = np.cos(x)
axs[1, 1].fill_between(x, y1, alpha=0.5, color='yellow', labe
axs[1, 1].fill_between(x, y2, alpha=0.5, color='lightgreen',
axs[1, 1].set_title('Area Plot')
axs[1, 1].set_xlabel('X')
axs[1, 1].set_ylabel('Y')
axs[1, 1].legend()

# Box Plot
data = [np.random.normal(0, std, 100) for std in range(1, 4)]
axs[1, 2].boxplot(data, vert=True, patch_artist=True)
axs[1, 2].set_title('Box Plot')
axs[1, 2].set_xlabel('Data Sets')
axs[1, 2].set_ylabel('Values')
axs[1, 2].set_xticks([1, 2, 3])
axs[1, 2].set_xticklabels(['Set 1', 'Set 2', 'Set 3'])

# Violin Plot
axs[1, 3].violinplot(data, showmeans=True, showmedians=True)
axs[1, 3].set_title('Violin Plot')
axs[1, 3].set_xlabel('Data Sets')
axs[1, 3].set_ylabel('Values')
axs[1, 3].set_xticks([1, 2, 3])
axs[1, 3].set_xticklabels(['Set 1', 'Set 2', 'Set 3'])

# 3D Plot
x3d = np.linspace(0, 10, 100)
y3d = np.sin(x3d)
z3d = np.cos(x3d)
ax3d = fig.add_subplot(2, 4, (8, 8), projection='3d')
ax3d.plot(x3d, y3d, z3d, label='3D Curve')
ax3d.set_title('3D Plot')
ax3d.set_xlabel('X-axis')
ax3d.set_ylabel('Y-axis')
ax3d.set_zlabel('Z-axis')
ax3d.legend()

# Heatmap
data_matrix = np.random.rand(10, 10)
heatmap = axs[2, 0].imshow(data_matrix, cmap='viridis', inter
axs[2, 0].set_title('Heatmap')
axs[2, 0].set_xlabel('X-axis')
axs[2, 0].set_ylabel('Y-axis')
fig.colorbar(heatmap, ax=axs[2, 0])

# Error Bar Plot
x = np.arange(0, 10, 1)
y = np.sin(x)
yerr = 0.2
axs[2, 1].errorbar(x, y, yerr=yerr, fmt='o', color='black', e
```

```
axs[2, 1].set_title('Error Bar Plot')
axs[2, 1].set_xlabel('X')
axs[2, 1].set_ylabel('Y')

# Stacked Bar Plot
categories = ['A', 'B', 'C', 'D', 'E']
values1 = [5, 2, 3, 4, 6]
values2 = [4, 1, 2, 5, 3]
axs[2, 2].bar(categories, values1, color='red', label='Values
axs[2, 2].bar(categories, values2, bottom=values1, color='ora
axs[2, 2].set_title('Stacked Bar Plot')
axs[2, 2].set_xlabel('Categories')
axs[2, 2].set_ylabel('Values')
axs[2, 2].legend()

# Quiver Plot
X, Y = np.meshgrid(np.arange(-2, 2, 0.2), np.arange(-2, 2, 0.
U = -1 - X**2 + Y
V = 1 + X - Y**2
axs[2, 3].quiver(X, Y, U, V)
axs[2, 3].set_title('Quiver Plot')
axs[2, 3].set_xlabel('X')
axs[2, 3].set_ylabel('Y')

# Polar Plot
theta = np.linspace(0, 2*np.pi, 100)
r = np.abs(np.sin(theta))
axs[3, 0] = plt.subplot2grid((4, 4), (3, 0), colspan=2, polar
axs[3, 0].plot(theta, r)
axs[3, 0].set_title('Polar Plot')

# Spectrogram
np.random.seed(0)
dt = 0.01
t = np.arange(0.0, 10.0, dt)
s1 = np.sin(2 * np.pi * 100 * t)
s2 = 2 * np.sin(2 * np.pi * 400 * t)
s = s1 + s2
s = s + np.random.randn(len(t))
axs[3, 2].specgram(s, NFFT=256, Fs=1/dt, noverlap=128)
axs[3, 2].set_title('Spectrogram')
axs[3, 2].set_xlabel('Time')
axs[3, 2].set_ylabel('Frequency')

# Final adjustments and showing the plot
fig.tight_layout()
plt.show()

# Thank You Message (this will appear in Jupyter Notebook or
from IPython.display import display, HTML
display(HTML("<div align='center'><h1>Thank You 🙏 🧣</h1></d
```

```
C:\Users\udaya\AppData\Local\Temp\ipykernel_12284\3407960307.py:131: MatplotlibDeprecationWarning: Auto-removal of overlapping axes is deprecated since 3.6 and will be removed two minor releases later; explicitly call ax.remove() as needed.
  axs[3, 0] = plt.subplot2grid((4, 4), (3, 0), colspan=2, polar=True)
```



Thank You 🙏 🫂