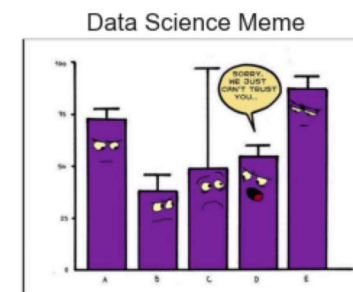
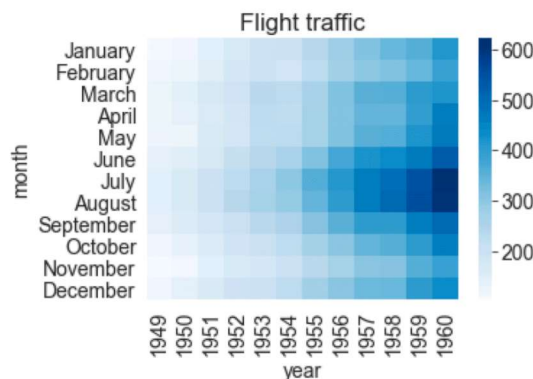
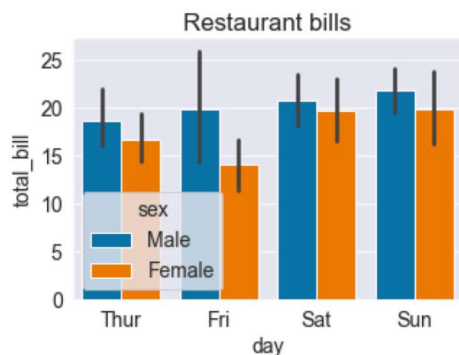
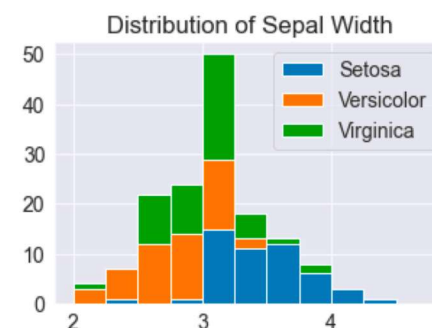
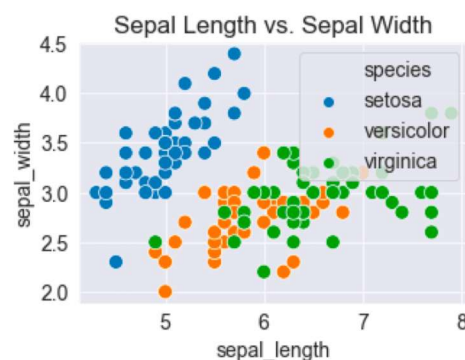
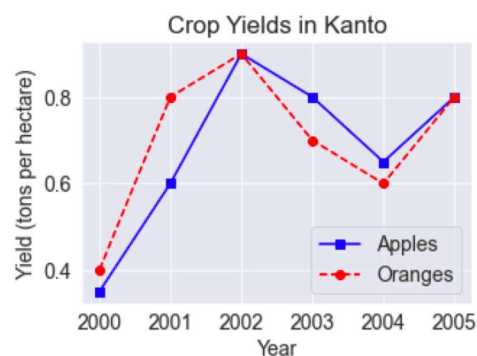


What is Matplotlib?

- Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. It is an open-source and free to use.
- Plotting of data can be extensively made possible in an interactive way by Matplotlib, which is a plotting library that can be demonstrated in Python scripts.
- Plotting of graphs is a part of data visualization, and this property can be achieved by making use of Matplotlib.



In []: ▶

In [1]: `!pip install matplotlib`

```
Requirement already satisfied: matplotlib in c:\users\admin\anaconda3\lib\site-packages (3.7.1)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\admin\anaconda3\lib\site-packages (from matplotlib) (1.0.5)
Requirement already satisfied: cyclor>=0.10 in c:\users\admin\anaconda3\lib\site-packages (from matplotlib) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\admin\anaconda3\lib\site-packages (from matplotlib) (4.25.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\admin\anaconda3\lib\site-packages (from matplotlib) (1.4.4)
Requirement already satisfied: numpy>=1.20 in c:\users\admin\anaconda3\lib\site-packages (from matplotlib) (1.24.3)
Requirement already satisfied: packaging>=20.0 in c:\users\admin\anaconda3\lib\site-packages (from matplotlib) (23.0)
Requirement already satisfied: pillow>=6.2.0 in c:\users\admin\anaconda3\lib\site-packages (from matplotlib) (9.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\admin\anaconda3\lib\site-packages (from matplotlib) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\admin\anaconda3\lib\site-packages (from matplotlib) (2.8.2)
Requirement already satisfied: six>=1.5 in c:\users\admin\anaconda3\lib\site-packages (from python-dateutil>=2.7->matplotlib) (1.16.0)
```

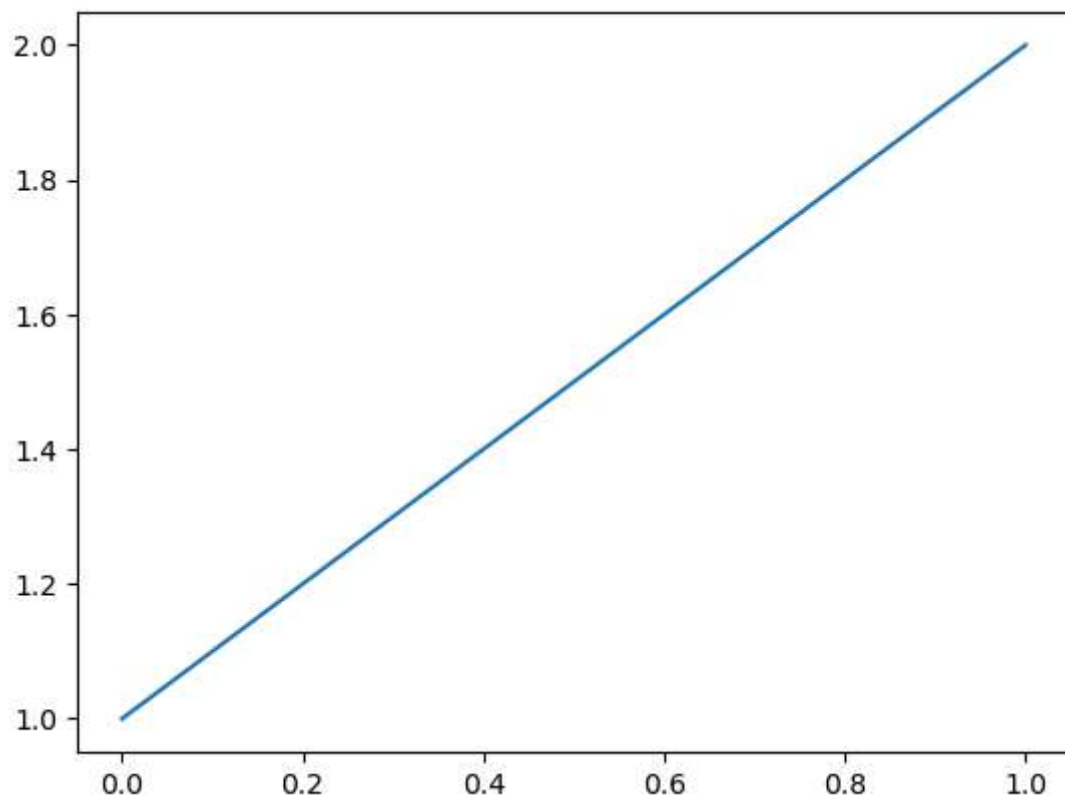
In [3]: `import numpy as np`
`import matplotlib.pyplot as plt`

In []:

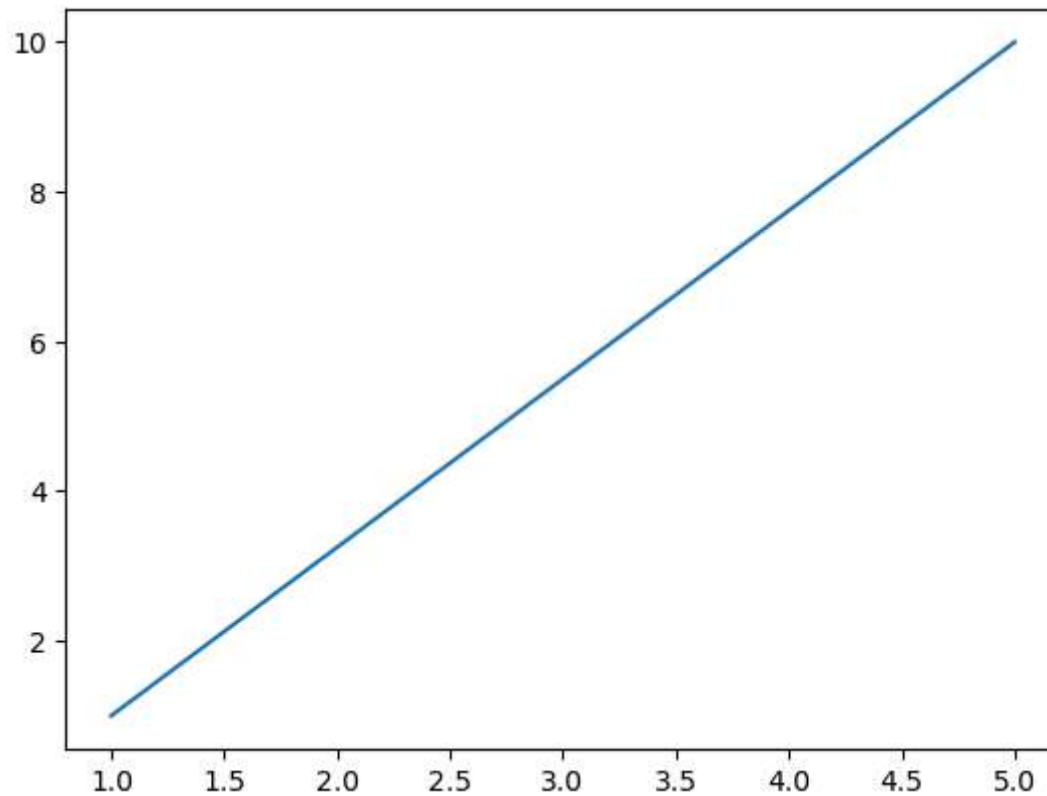
In [4]: `[1,2]`

Out[4]: `[1, 2]`

```
In [9]: ▶ plt.plot([1,2])  
plt.show()
```

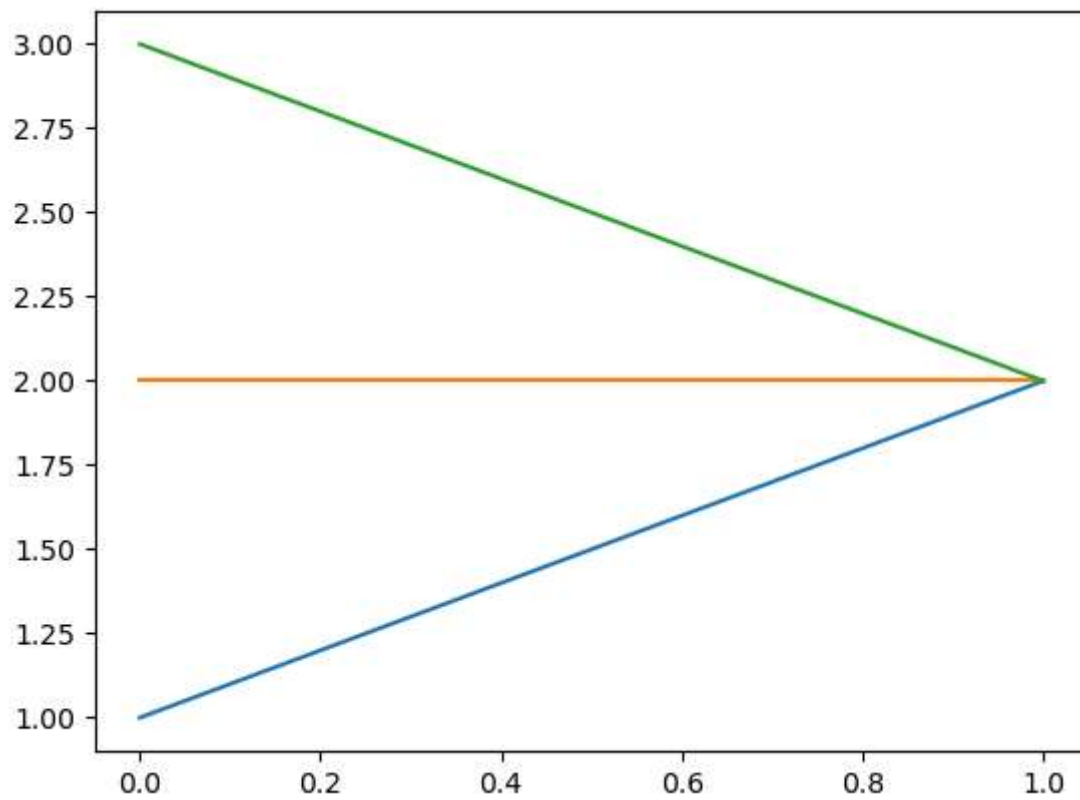


```
In [12]: ▶ plt.plot([1,5], [1,10])  
plt.show()
```



```
In [ ]: ▶
```

```
In [15]: ▶ plt.plot([1,2])  
plt.plot([2,2])  
plt.plot([3,2])  
  
plt.show()
```

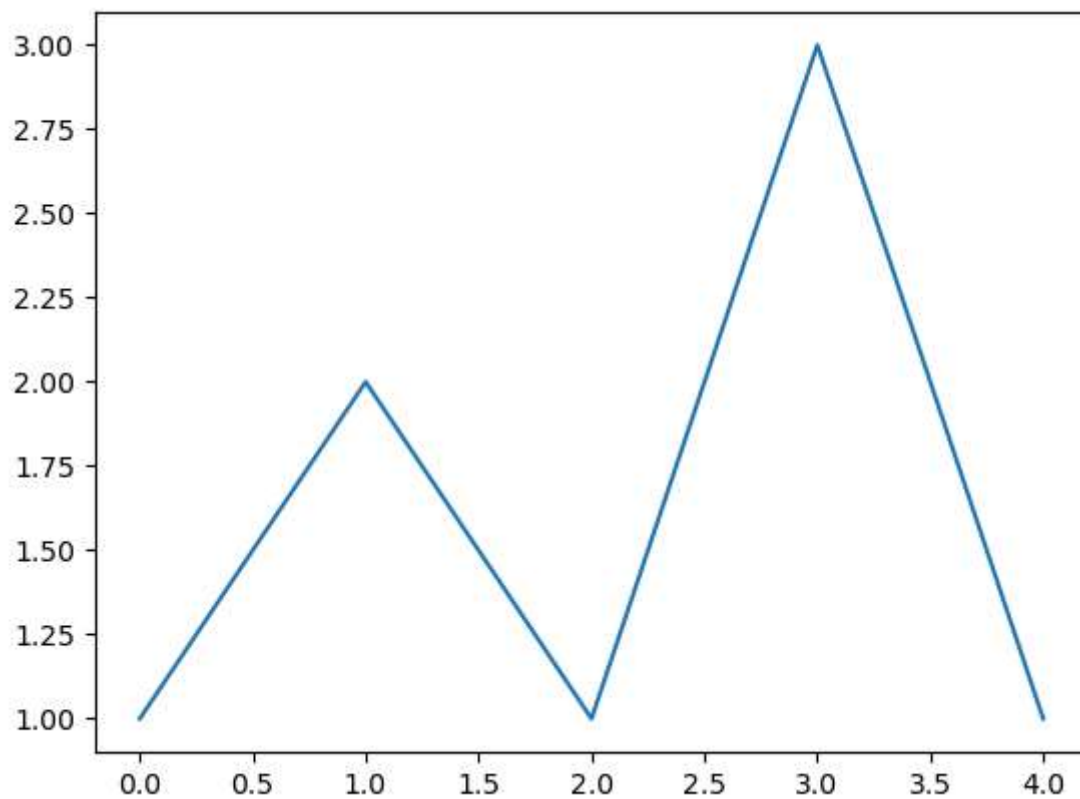


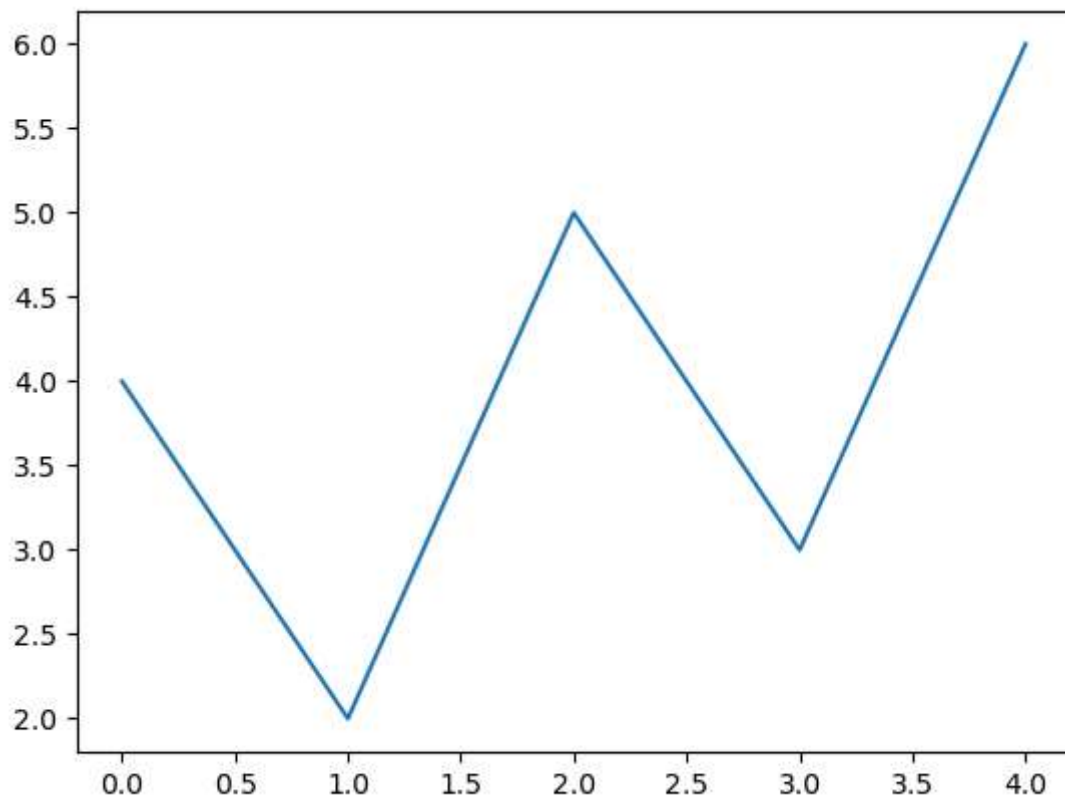
```
In [ ]: ▶
```

```
In [18]: ▶ plt.figure(1)
plt.plot([1,2,1,3,1])

plt.figure(2)
plt.plot([4,2,5,3,6])

plt.show()
```





In []: ▶

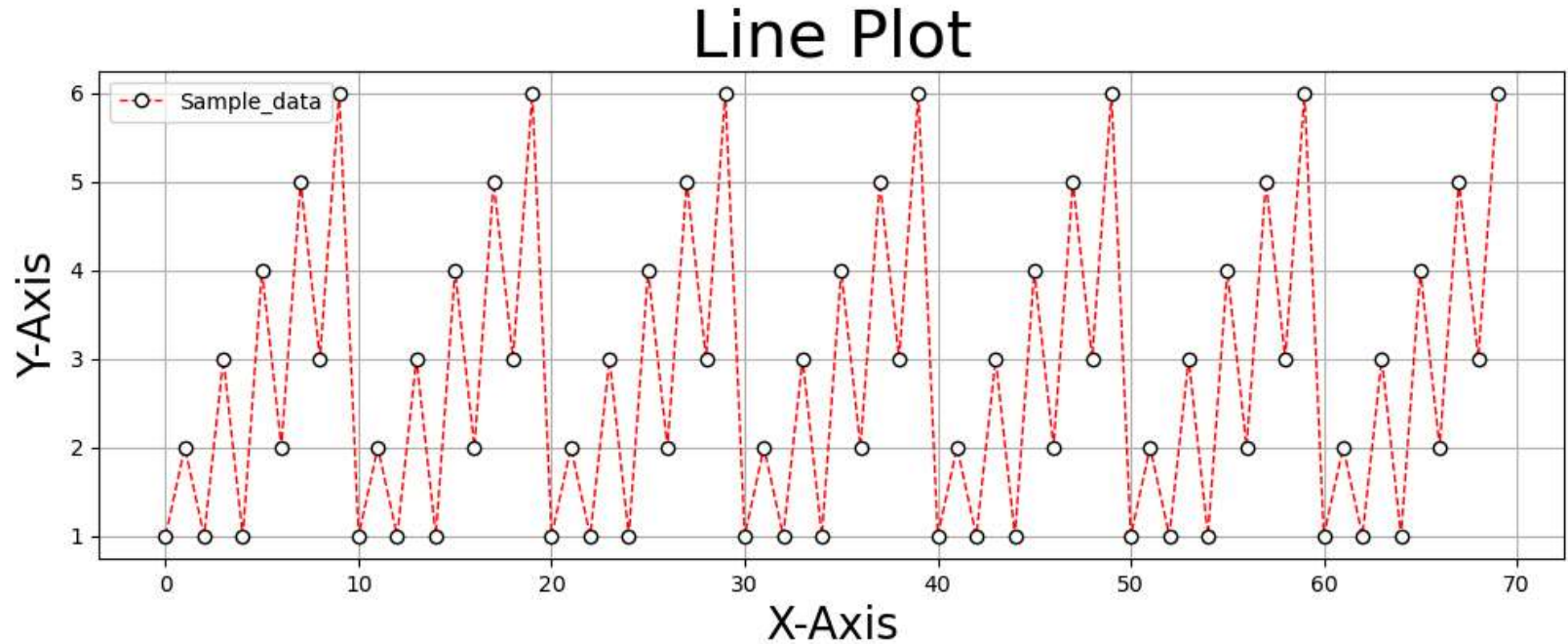
```
In [78]: ► y = [1,2,1,3,1,4,2,5,3,6,1,2,1,3,1,4,2,5,3,6,1,2,1,3,1,4,2,5,3,6,1,2,1,3,
              1,4,2,5,3,6,1,2,1,3,1,4,2,5,3,6,1,2,1,3,1,4,2,5,3,6,1,2,1,3,1,4,2,5,3,6]

plt.figure(figsize=(12, 4))

plt.plot(y, color='r', marker='o', mfc = 'w', mec='k', markersize=6, lw = 1, ls='dashed', label='Sample_da

plt.title('Line Plot', color='k', fontsize=30)

plt.xlabel('X-Axis', fontsize=20)
plt.ylabel('Y-Axis', fontsize=20)
plt.grid()
plt.legend()
plt.show()
```



In []: ►


```
In [88]: ▶ # Line Plot  
          # Bar Plot  
          # Scatter Plot  
          # Pie Chart  
          # 3D Plots
```

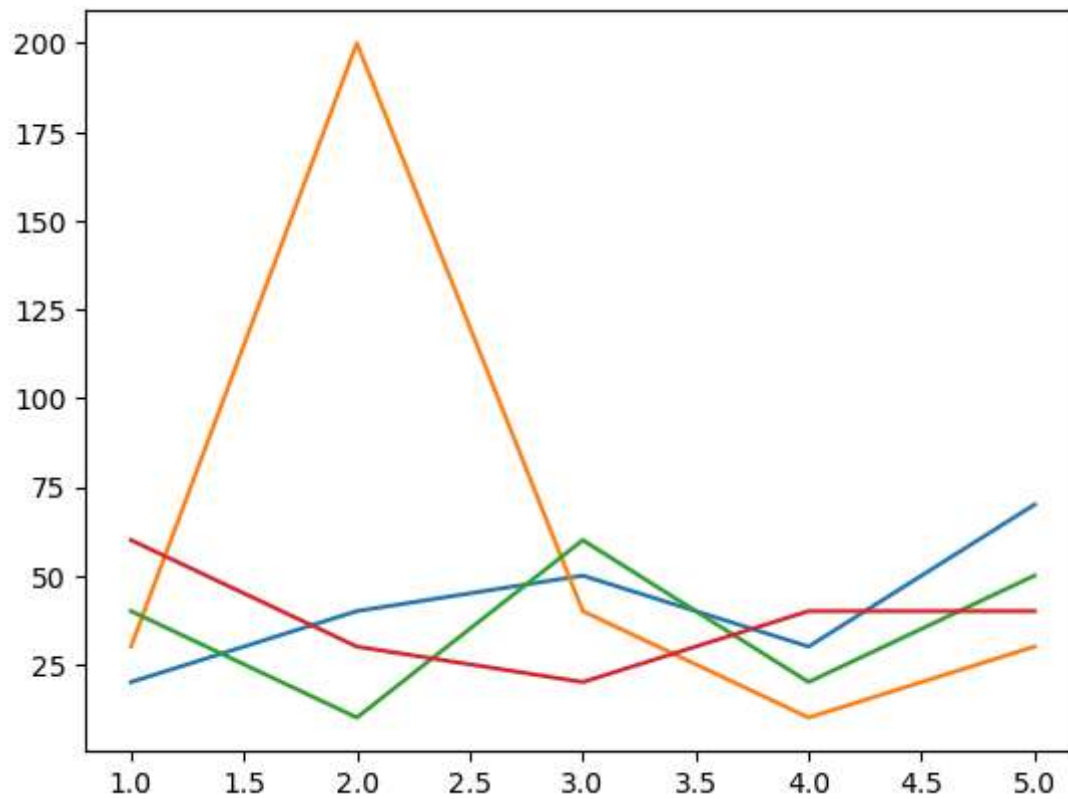
```
In [ ]: ▶
```

Line Plot

```
In [92]: x = [1, 2, 3, 4, 5]

y1 = [20, 40, 50, 30, 70]
y2 = [30, 200, 40, 10, 30]
y3 = [40, 10, 60, 20, 50]
y4 = [60, 30, 20, 40, 40]

plt.plot(x, y1)
plt.plot(x, y2)
plt.plot(x, y3)
plt.plot(x, y4)
plt.show()
```



```
In [112]: x = [1, 2, 3, 4, 5]

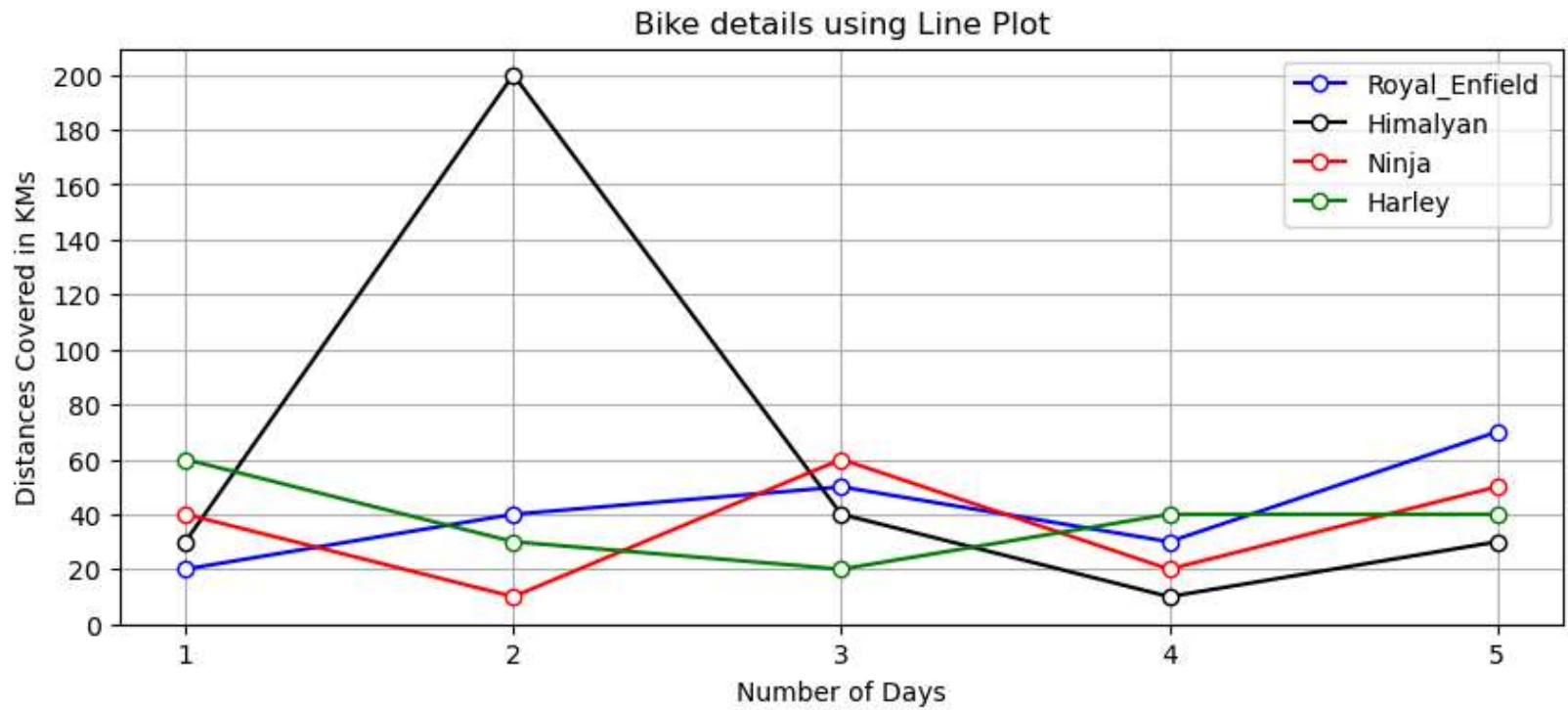
y1 = [20, 40, 50, 30, 70]
y2 = [30, 200, 40, 10, 30]
y3 = [40, 10, 60, 20, 50]
y4 = [60, 30, 20, 40, 40]

plt.figure(figsize=(10, 4))

plt.plot(x, y1, color='b', marker='o', label='Royal_Enfield', mfc='w')
plt.plot(x, y2, color='k', marker='o', label='Himalyan', mfc='w')
plt.plot(x, y3, color='r', marker='o', label='Ninja', mfc='w')
plt.plot(x, y4, color='g', marker='o', label='Harley', mfc='w')

plt.title('Bike details using Line Plot')
plt.xlabel('Number of Days')
plt.ylabel('Distances Covered in KMs')
plt.xticks(np.arange(1, 6, 1))
plt.yticks(np.arange(0, 201, 20))

plt.legend()
plt.grid()
plt.show()
```



In []: ▶

Bar Plot

```
In [115]: ▶ x = [1, 2, 3, 4, 5]

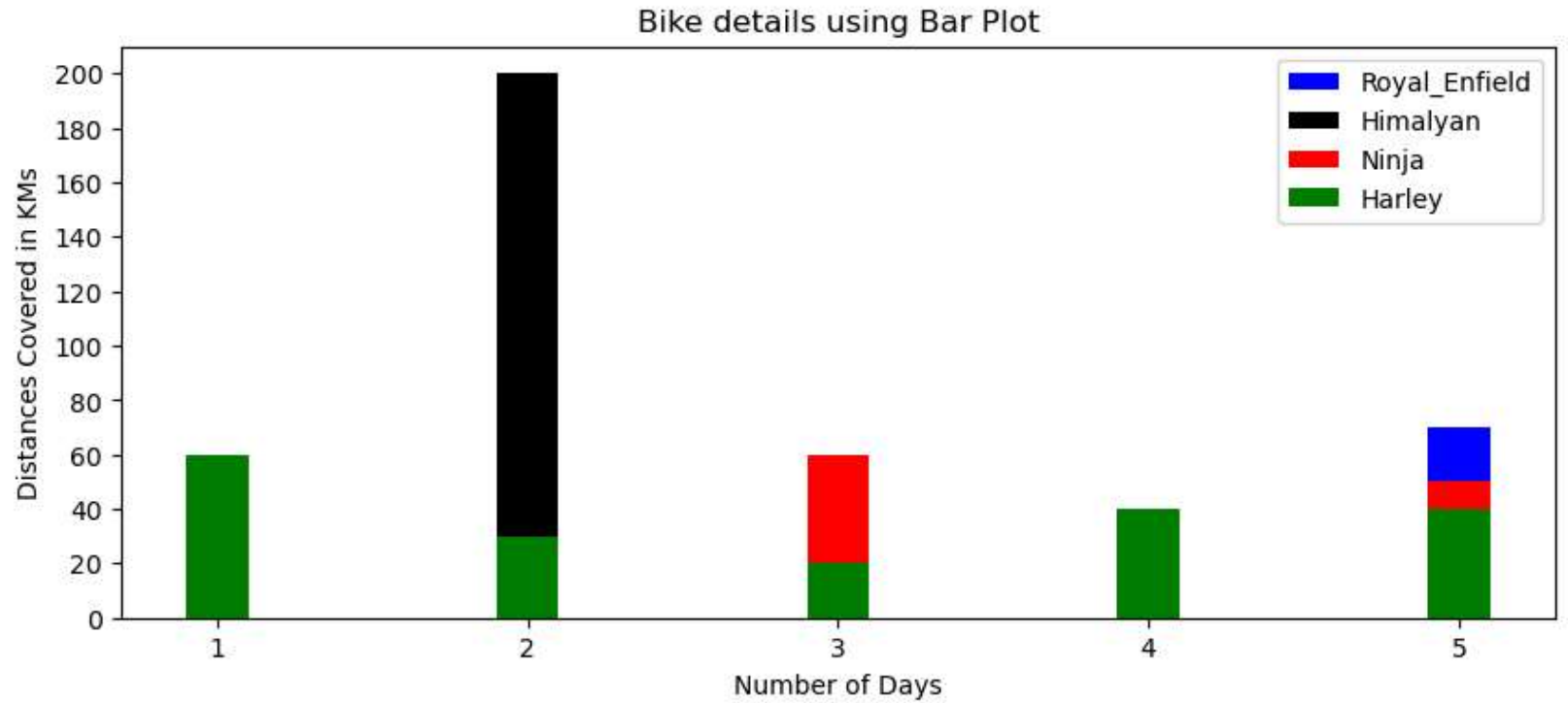
y1 = [20, 40, 50, 30, 70]
y2 = [30, 200, 40, 10, 30]
y3 = [40, 10, 60, 20, 50]
y4 = [60, 30, 20, 40, 40]

plt.figure(figsize=(10, 4))

plt.bar(x, y1, color='b', label='Royal_Enfield', width=0.2)
plt.bar(x, y2, color='k', label='Himalyan', width=0.2)
plt.bar(x, y3, color='r', label='Ninja', width=0.2)
plt.bar(x, y4, color='g', label='Harley', width=0.2)

plt.title('Bike details using Bar Plot')
plt.xlabel('Number of Days')
plt.ylabel('Distances Covered in KMs')
plt.xticks(np.arange(1, 6, 1))
plt.yticks(np.arange(0, 201, 20))

plt.legend()
# plt.grid()
plt.show()
```



In []: ▶

```
In [120]: ▶ x1 = [0.25, 1.25, 2.25, 3.25, 4.25]
x2 = [0.5, 1.5, 2.5, 3.5, 4.5]
x3 = [0.75, 1.75, 2.75, 3.75, 4.75]
x4 = [1, 2, 3, 4, 5]

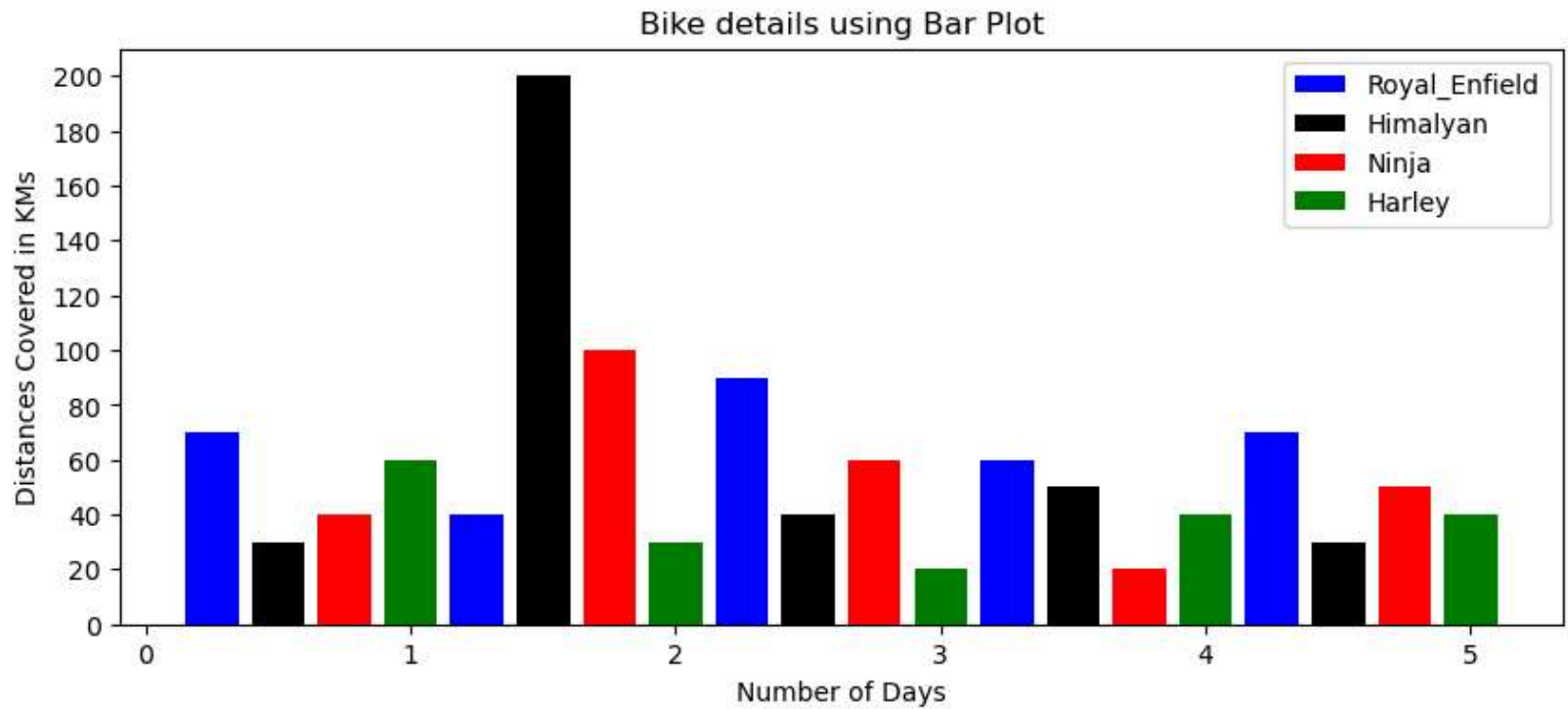
y1 = [70, 40, 90, 60, 70]
y2 = [30, 200, 40, 50, 30]
y3 = [40, 100, 60, 20, 50]
y4 = [60, 30, 20, 40, 40]

plt.figure(figsize=(10, 4))

plt.bar(x1, y1, color='b', label='Royal_Enfield', width=0.2)
plt.bar(x2, y2, color='k', label='Himalyan', width=0.2)
plt.bar(x3, y3, color='r', label='Ninja', width=0.2)
plt.bar(x4, y4, color='g', label='Harley', width=0.2)

plt.title('Bike details using Bar Plot')
plt.xlabel('Number of Days')
plt.ylabel('Distances Covered in KMs')
plt.xticks(np.arange(0, 6, 1))
plt.yticks(np.arange(0, 201, 20))

plt.legend()
# plt.grid()
plt.show()
```

In []: ▶

In []: ▶

Scatter Plot

```
In [123]: ▶ x = [1, 2, 3, 4, 5]

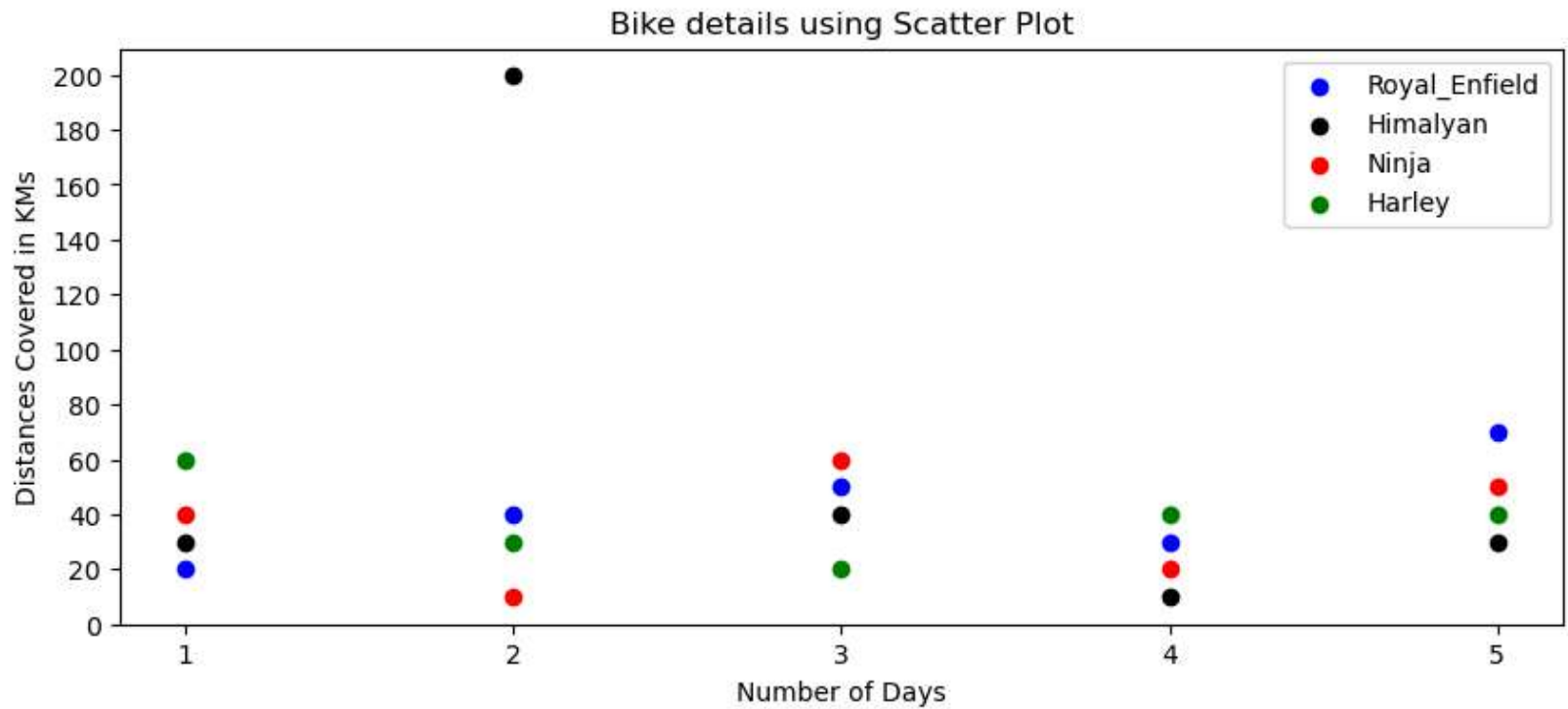
y1 = [20, 40, 50, 30, 70]
y2 = [30, 200, 40, 10, 30]
y3 = [40, 10, 60, 20, 50]
y4 = [60, 30, 20, 40, 40]

plt.figure(figsize=(10, 4))

plt.scatter(x, y1, color='b', marker='o', label='Royal_Enfield')
plt.scatter(x, y2, color='k', marker='o', label='Himalyan')
plt.scatter(x, y3, color='r', marker='o', label='Ninja')
plt.scatter(x, y4, color='g', marker='o', label='Harley')

plt.title('Bike details using Scatter Plot')
plt.xlabel('Number of Days')
plt.ylabel('Distances Covered in KMs')
plt.xticks(np.arange(1, 6, 1))
plt.yticks(np.arange(0, 201, 20))

plt.legend()
# plt.grid()
plt.show()
```

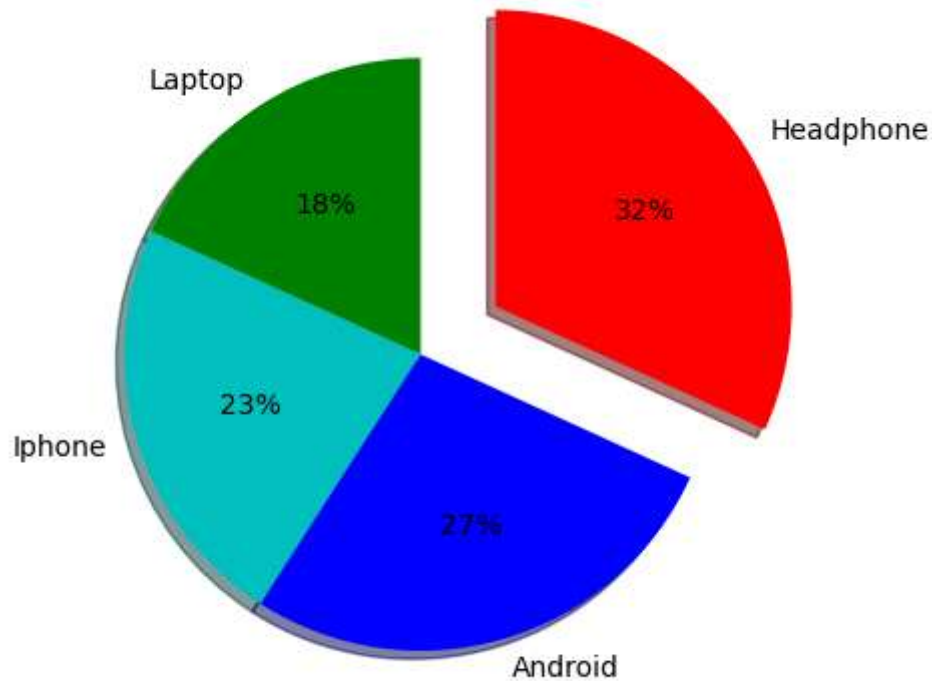


In []: ▶

Pie Chart

```
In [152]: ▶ slices = [4,5,6,7]
labels = ['Laptop', 'Iphone', 'Android', 'Headphone']
cols = ['g', 'c', 'b', 'r']

plt.pie(slices, labels=labels, colors=cols, autopct='%1.0f%%', startangle=90, shadow=True, explode=(0, 0,
# plt.legend()
plt.show()
```



```
In [ ]: ▶
```

3D Plots

```
In [158]: ▶ fig = plt.figure(figsize=(10, 4))
ax = fig.add_subplot(projection='3d')

x = [1, 2, 3, 4, 5]

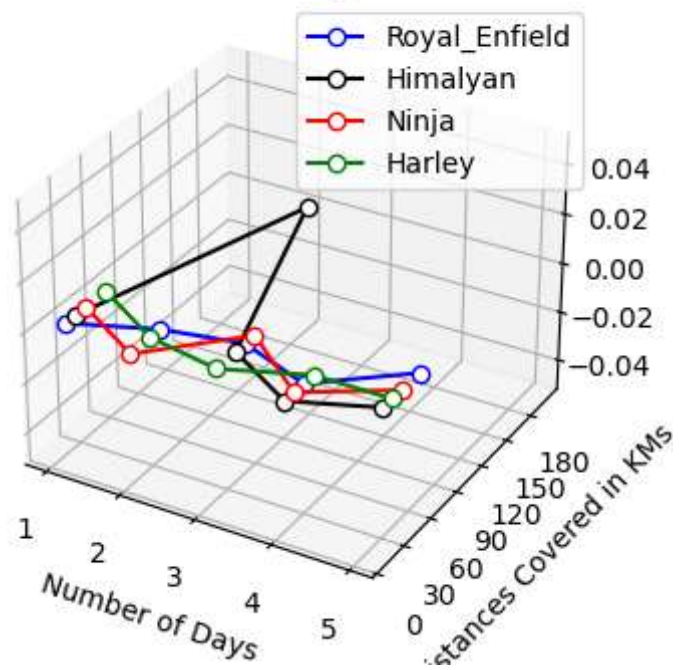
y1 = [20, 40, 50, 30, 70]
y2 = [30, 200, 40, 10, 30]
y3 = [40, 10, 60, 20, 50]
y4 = [60, 30, 20, 40, 40]

plt.plot(x, y1, color='b', marker='o', label='Royal_Enfield', mfc='w')
plt.plot(x, y2, color='k', marker='o', label='Himalyan', mfc='w')
plt.plot(x, y3, color='r', marker='o', label='Ninja', mfc='w')
plt.plot(x, y4, color='g', marker='o', label='Harley', mfc='w')

plt.title('Bike details using Line Plot')
plt.xlabel('Number of Days')
plt.ylabel('Distances Covered in KMs')
plt.xticks(np.arange(1, 6, 1))
plt.yticks(np.arange(0, 201, 30))

plt.legend()
plt.grid()
plt.show()
```

Bike details using Line Plot



In []: ▶

In []: ▶

In []: ▶

In []: ▶

In []: ▶