EXNO-6-DS-DATA VISUALIZATION USING SEABORN LIBRARY

Aim:

To Perform Data Visualization using seaborn python library for the given datas.

EXPLANATION:

Data visualization is the graphical representation of information and data. By using visual elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data.

Algorithm:

STEP 1:Include the necessary Library.
STEP 2:Read the given Data.
STEP 3:Apply data visualization techniques to identify the patterns of the data.
STEP 4:Apply the various data visualization tools wherever necessary.
STEP 5:Include Necessary parameters in each functions.

Coding and Output:

from google.colab import drive drive.mount('/content/drive')

```
Mounted at /content/drive
```

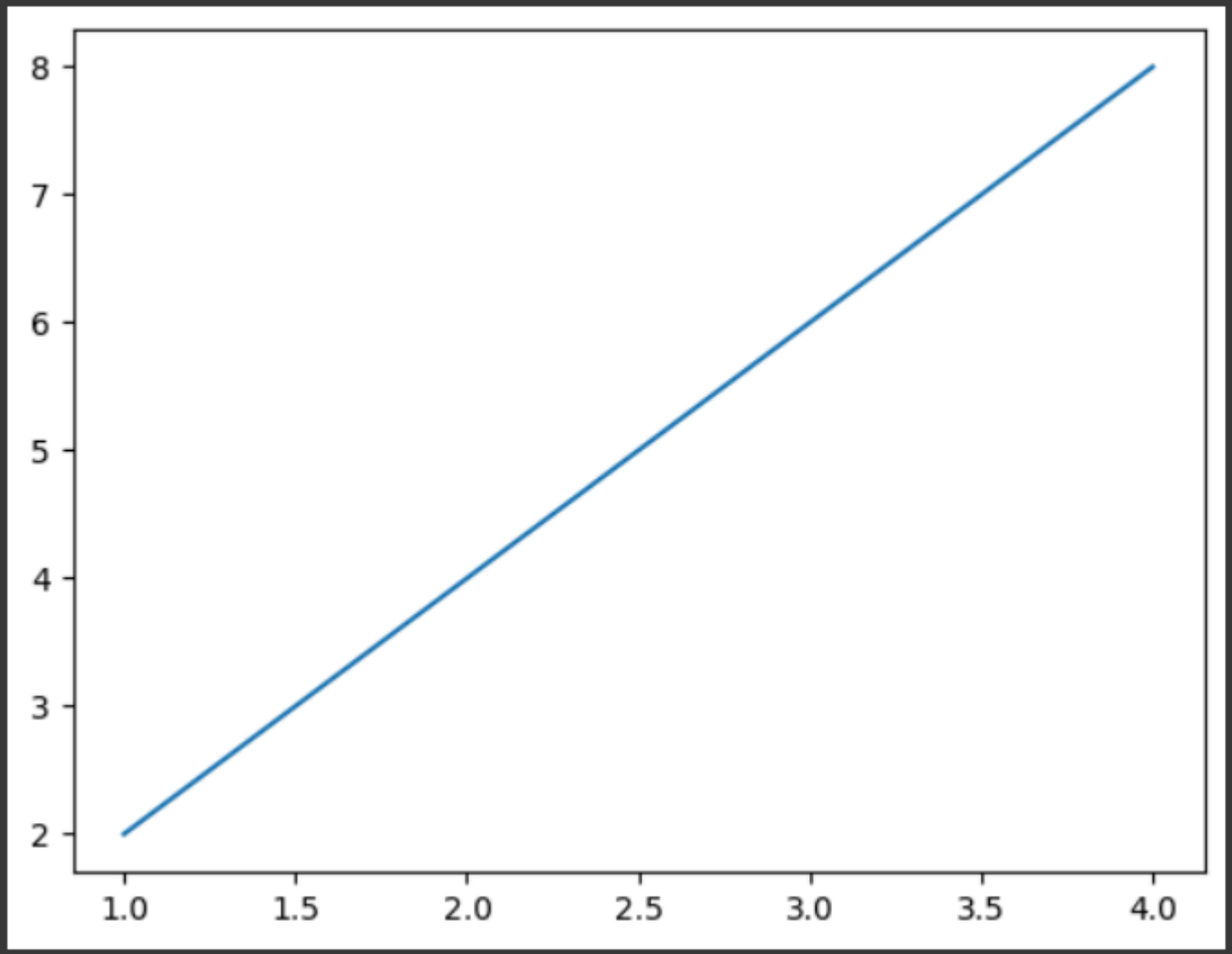
import seaborn as sns !pip install matplotlib import matplotlib.pyplot as plt

```
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-packages (3.8.0)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (1.3.1)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (4.55.3)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (1.4.7)
Requirement already satisfied: numpy<2,>=1.21 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (1.26.4)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (24.2)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (11.0.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (3.2.0)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (2.8.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.7->matplotlib) (1.17.0)
```

Code cell output actions

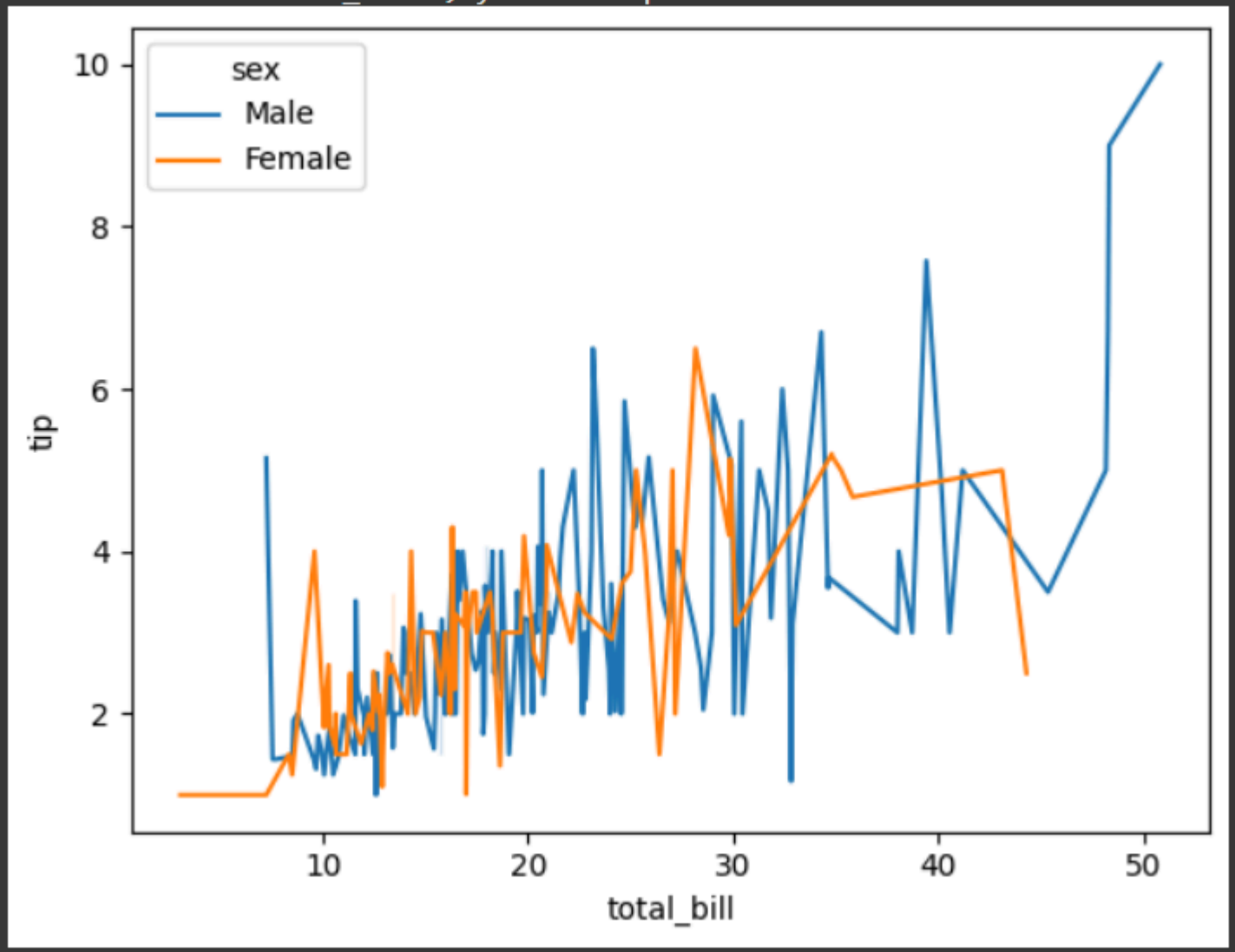
x=[1,2,3,4] y=[2,4,6,8] sns.lineplot(x=x,y=y)

<Axes: >



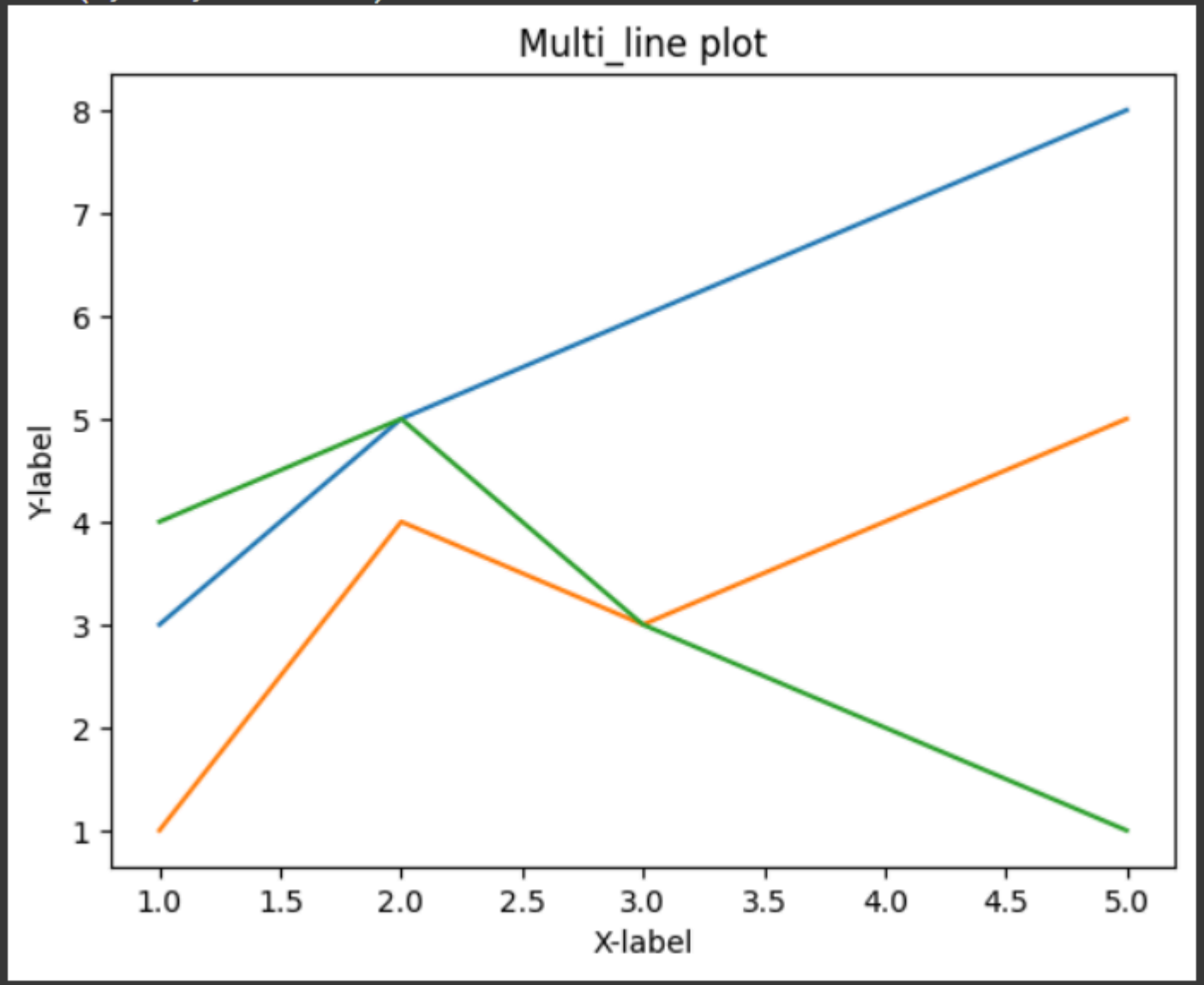

```
df=sns.load_dataset("tips") df
sns.lineplot(x="total_bill",y="tip",data=df,hue="sex",linestyle="solid",legend="auto")
```

<Axes: xlabel='total_bill', ylabel='tip'>



x=[1,2,3,4,5] y1=[3,5,6,7,8] y2=[1,4,3,4,5] y3=[4,5,3,2,1]
sns.lineplot(x=x,y=y1) sns.lineplot(x=x,y=y2) sns.lineplot(x=x,y=y3) plt.title("Multi_line plot") plt.xlabel("X-label") plt.ylabel("Y-label")

```
Text(0, 0.5, 'Y-label')
```



import seaborn as sns import matplotlib.pyplot as plt #Load the tips dataset tips =sns.load_dataset('tips') #Calculate the average total bill and tip for each day of the week avg_total_bill =tips.groupby('day') ['total_bill'].mean() avg_tip =tips.groupby('day') ['tip'].mean()

```
<ipython-input-9-492dfcd0c6aa>:6: FutureWarning: The default of observed=False is deprecated and will be changed to True in a future version of pandas. Pass observed=False to retain cu
  avg_total_bill =tips.groupby('day') ['total_bill'].mean()
<ipython-input-9-492dfcd0c6aa>:7: FutureWarning: The default of observed=False is deprecated and will be changed to True in a future version of pandas. Pass observed=False to retain cu
  avg_tip =tips.groupby('day') ['tip'].mean()
```

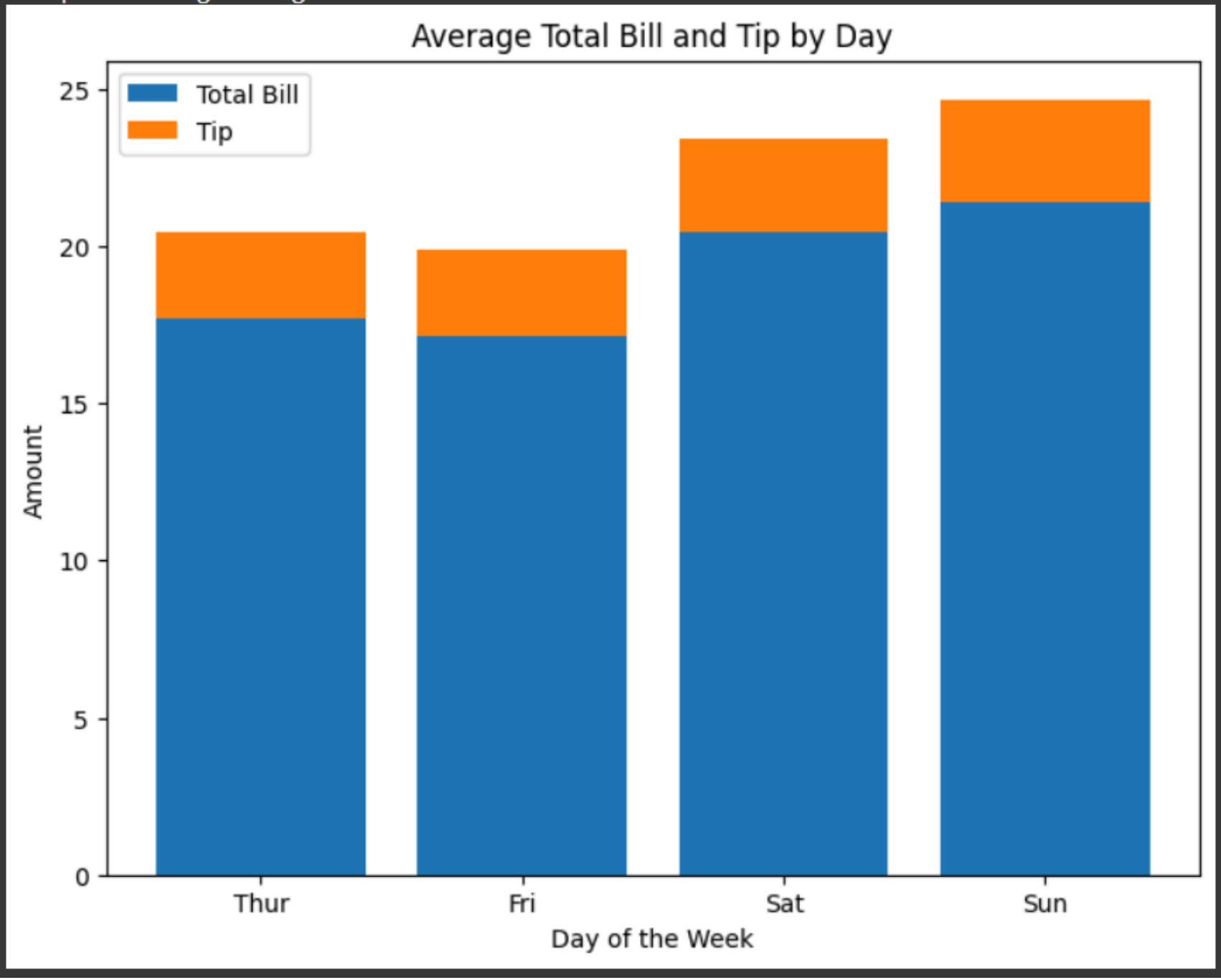
plt.figure(figsize=(8, 6)) p1= plt.bar(avg_total_bill.index, avg_total_bill, label='Total Bill') p2= plt.bar(avg_tip.index, avg_tip, bottom=avg_total_bill, label='Tip')

Set the labels and title

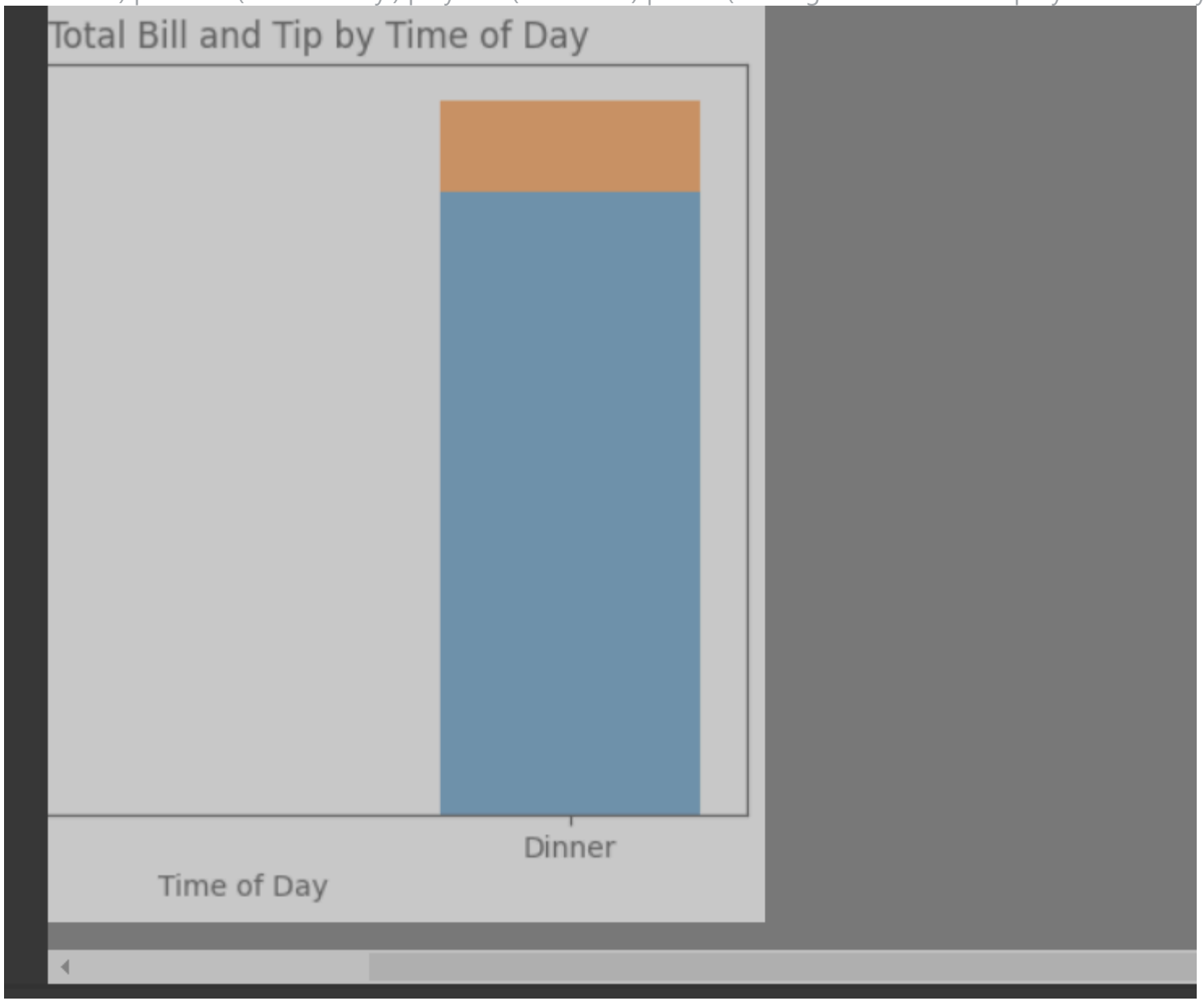
plt.xlabel('Day of the Week') plt.ylabel('Amount') plt.title('Average Total Bill and Tip by Day') plt.legend()

0s

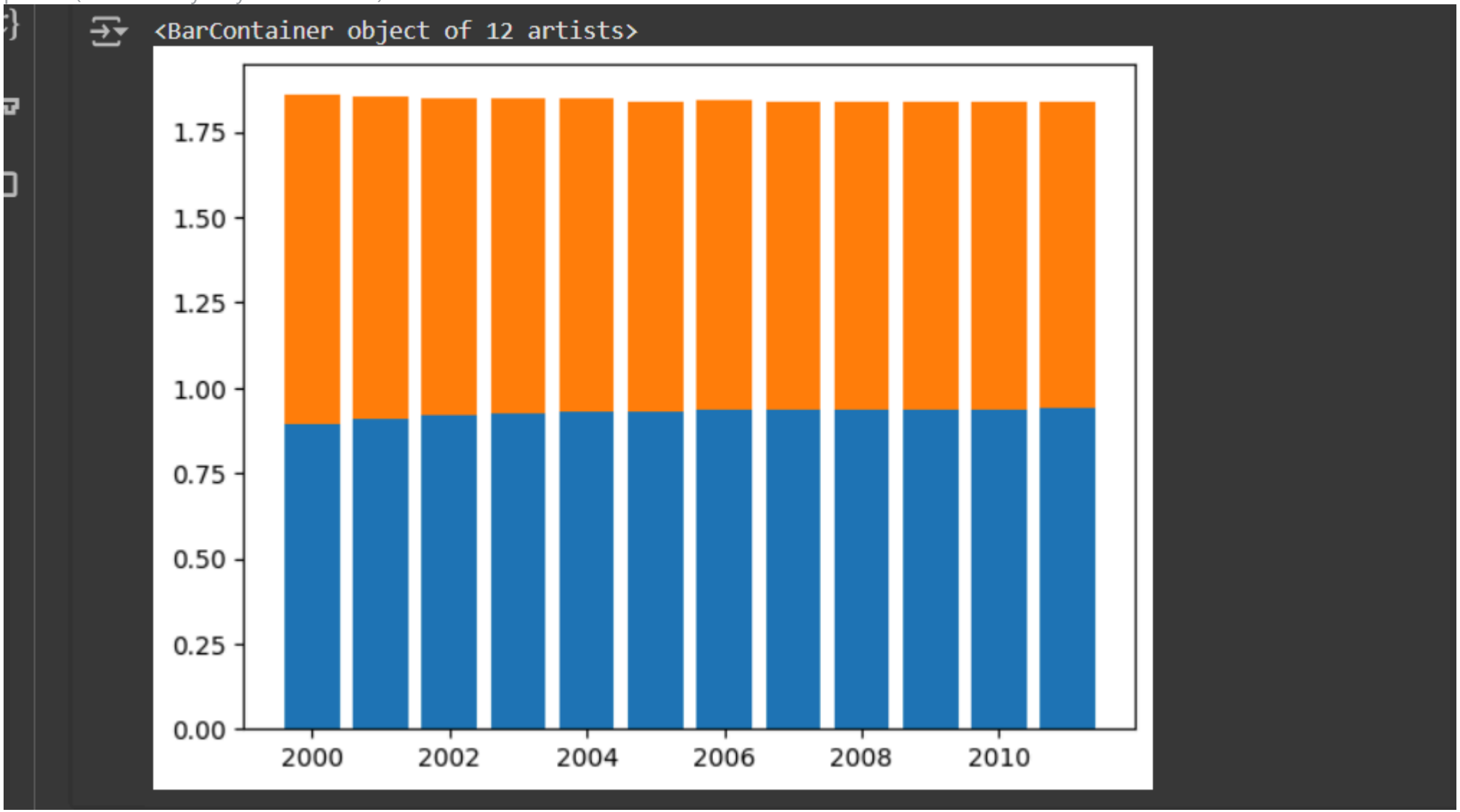
```
<matplotlib.legend.Legend at 0x788aba06c2b0>
```



```
avg_total_bill =tips.groupby('time') ['total_bill'].mean() avg_tip =tips.groupby('time') ['tip'].mean() #Create a grouped bar chart p1 = plt.bar(avg_total_bill.index, avg_total_bill, label='Total Bill', width=0.4) p2 = plt.bar(avg_tip.index, avg_tip, bottom = avg_total_bill, label='Tip', width=0.4) plt.xlabel('Time of Day') plt.ylabel("Amount") plt.title('Average Total Bill and Tip by Time of Day') plt.legend()
```



years =range (2000, 2012) apples= [0.895, 0.91, 0.919, 0.926, 0.929, 0.931, 0.934, 0.936, 0.937, 0.9375, 0.9372, 0.939] oranges = [0.962, 0.941, 0.930, 0.923, 0.918, 0.908, 0.907, 0.904, 0.901, 0.898, 0.9, 0.896,] import seaborn as sns dt= sns.load_dataset('tips') #Bar plot with hue parameter sns.barplot(x='day', y='total_bill', hue='sex', data=dt, palette='Set1') #Set labels and title plt.xlabel('Day of the week') plt.ylabel('Total Bill') plt.title('Total Bill by Day and Gender')



import seaborn as sns

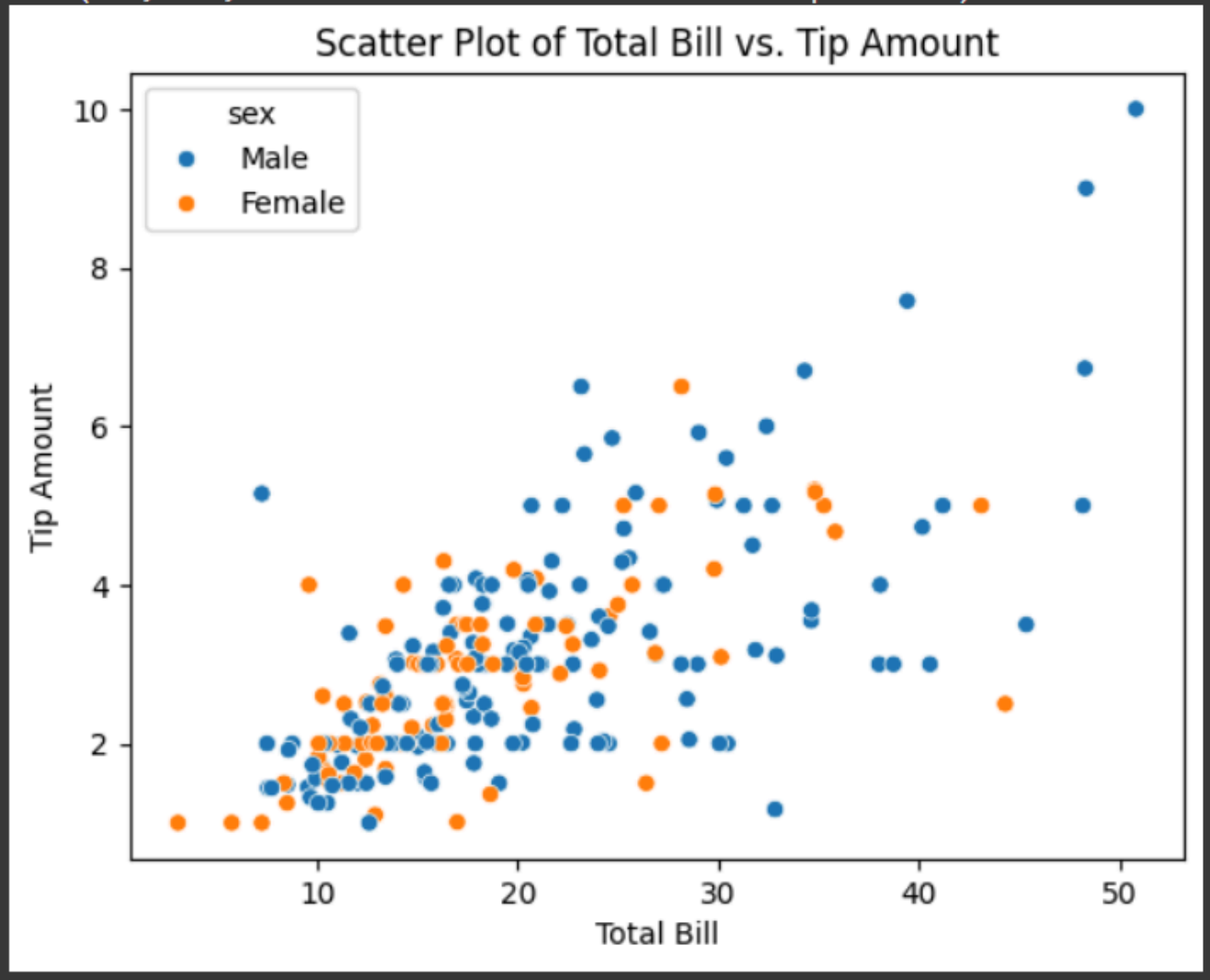
Load the tips dataset

tips =sns.load_dataset('tips')

Scatter plot of total bill vs. tip amount

sns.scatterplot(x='total_bill', y='tip', hue='sex',data=tips) #Set labels and title plt.xlabel('Total Bill') plt.ylabel('Tip Amount') plt.title('Scatter Plot of Total Bill vs. Tip Amount')

```
Text(0.5, 1.0, 'Scatter Plot of Total Bill vs. Tip Amount')
```



import seaborn as sns import numpy as np import pandas as pd

```
np.random.seed (1) num_var= np.random.randn(1000) num_var = pd.Series (num_var, name = "Numerical variable") num_var
```

	Numerical variable
0	1.624345
1	-0.611756
2	-0.528172
3	-1.072969
4	0.865408
...	...
995	-0.116444
996	-2.277298
997	-0.069625
998	0.353870
999	-0.186955

1000 rows × 1 columns

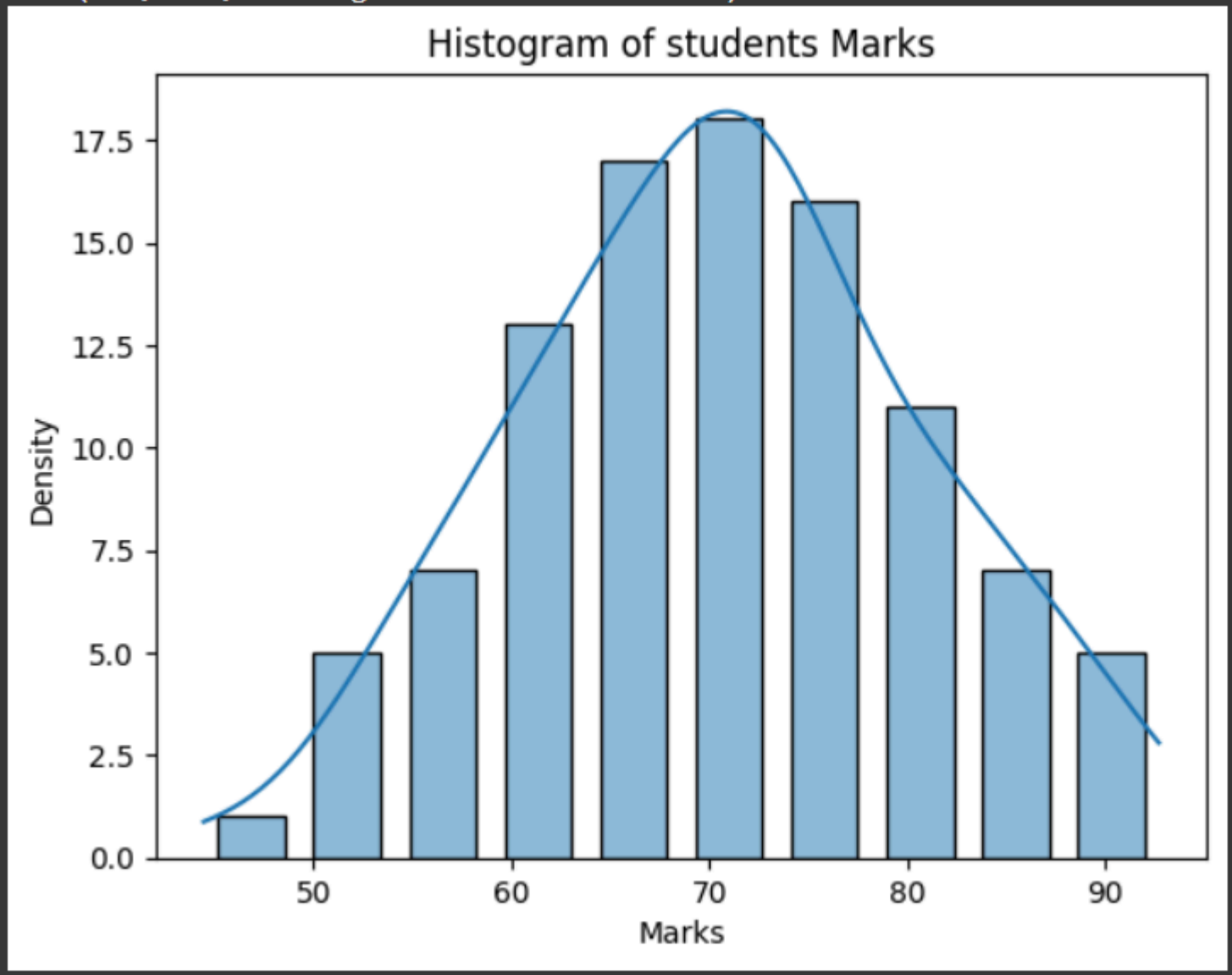
dtype: float64

```
import seaborn as sns import numpy as np import pandas as pd import matplotlib.pyplot as plt np.random.seed(0) marks = np.random.normal(loc=70, scale=10, size=100) marks
```

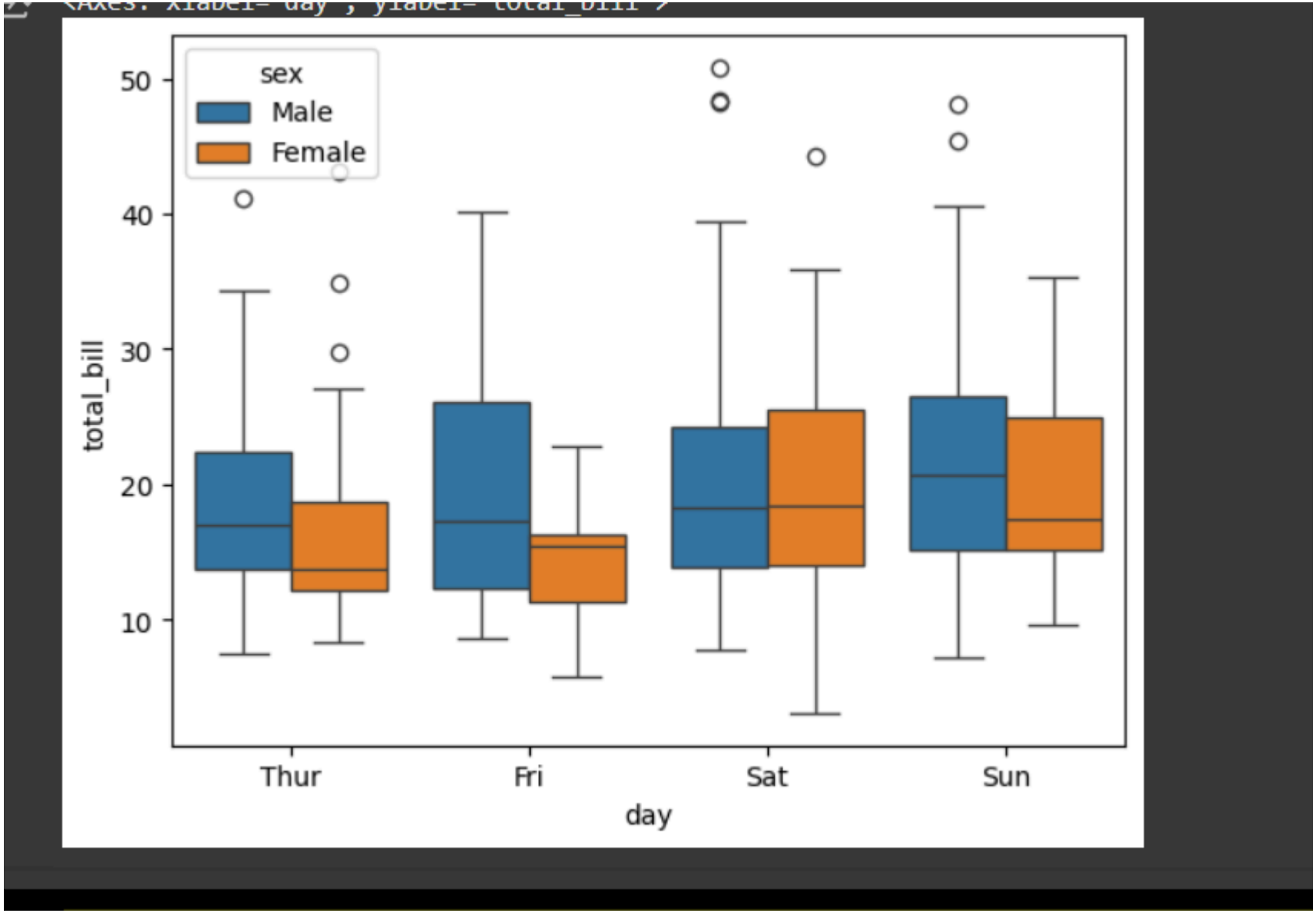
```
array([87.64052346, 74.00157208, 79.78737984, 92.40893199, 88.6755799 ,
       60.2272212 , 79.50088418, 68.48642792, 68.96781148, 74.10598502,
       71.44043571, 84.54273507, 77.61037725, 71.21675016, 74.43863233,
       73.33674327, 84.94079073, 67.94841736, 73.13067702, 61.45904261,
       44.47010184, 76.53618595, 78.64436199, 62.5783498 , 92.69754624,
       55.45634325, 70.45758517, 68.1281615 , 85.32779214, 84.6935877 ,
       71.54947426, 73.7816252 , 61.12214252, 50.19203532, 66.52087851,
       71.56348969, 82.30290681, 82.02379849, 66.12673183, 66.97697249,
       59.51447035, 55.79982063, 52.93729809, 89.50775395, 64.90347818,
       65.61925698, 57.4720464 , 77.77490356, 53.86102152, 67.8725972 ,
       61.04533439, 73.86902498, 64.89194862, 58.19367816, 69.71817772,
       74.28331871, 70.66517222, 73.02471898, 63.65677906, 66.37258834,
       63.27539552, 66.40446838, 61.86853718, 52.73717398, 71.77426142,
       65.98219064, 53.69801653, 74.62782256, 60.92701636, 70.51945396,
       77.29090562, 71.28982911, 81.39400685, 57.6517418 , 74.02341641,
       63.15189909, 61.29202851, 64.21150335, 66.88447468, 70.56165342,
       58.34850159, 79.00826487, 74.6566244 , 54.63756314, 84.88252194,
       88.95889176, 81.78779571, 68.20075164, 59.29247378, 80.54451727,
       65.96823053, 82.2244507 , 72.08274978, 79.76639036, 73.56366397,
       77.06573168, 70.10500021, 87.85870494, 71.26912093, 74.01989363])
```

```
sns.histplot(data=marks, bins=10, kde= True, stat='count', cumulative=False, multiple='stack', element='bars', palette='Set1', shrink=0.7)
plt.xlabel('Marks') plt.ylabel('Density') plt.title("Histogram of students Marks")
```

```
  sns.histplot(data=marks, bins=10, kde= True, stat='count', cumulative=False, multiple='stack',
Text(0.5, 1.0, 'Histogram of students Marks')
```

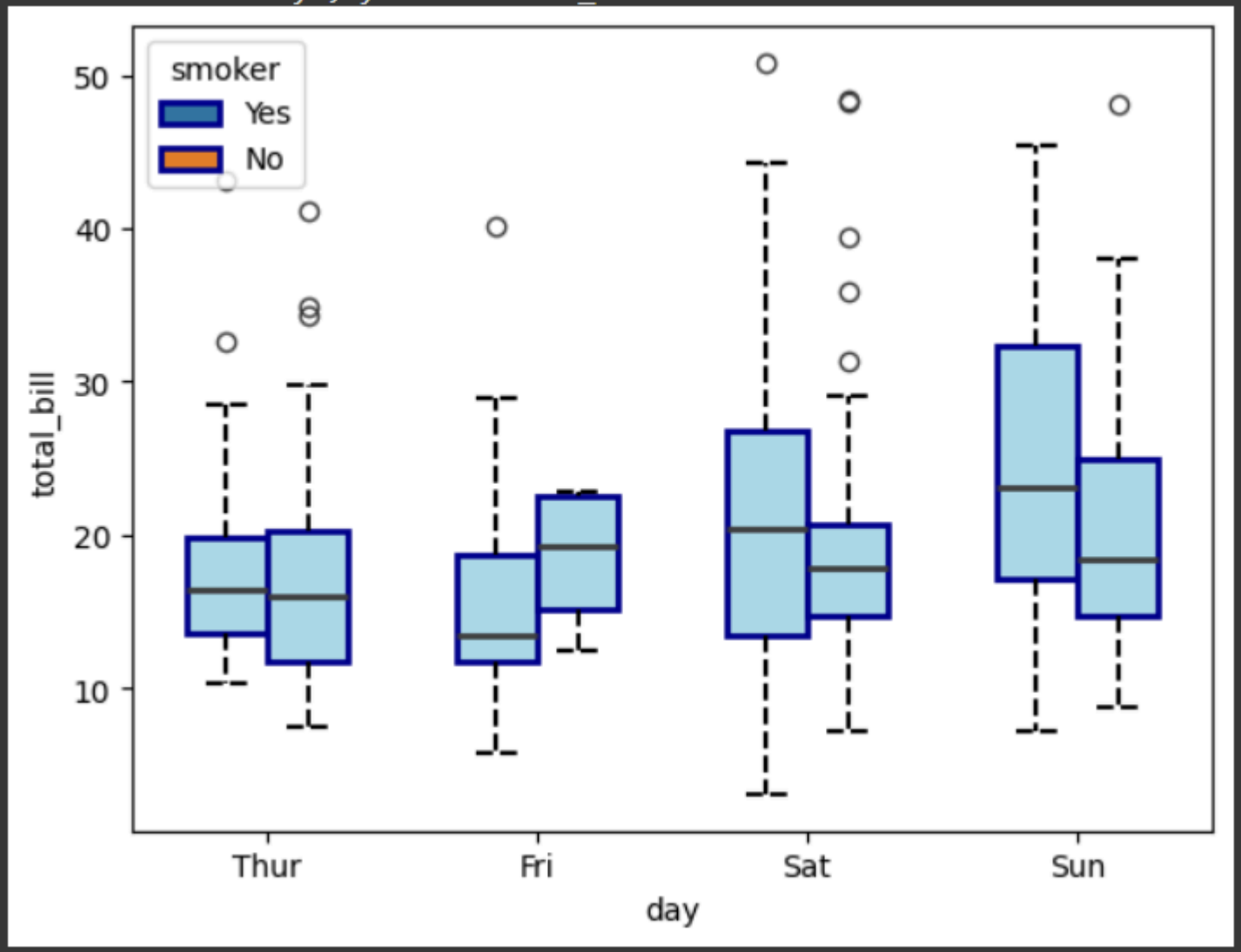


import seaborn as sns import pandas as pd tips =sns.load_dataset('tips') sns.boxplot(x=tips['day'], y=tips['total_bill'], hue=tips['sex'])



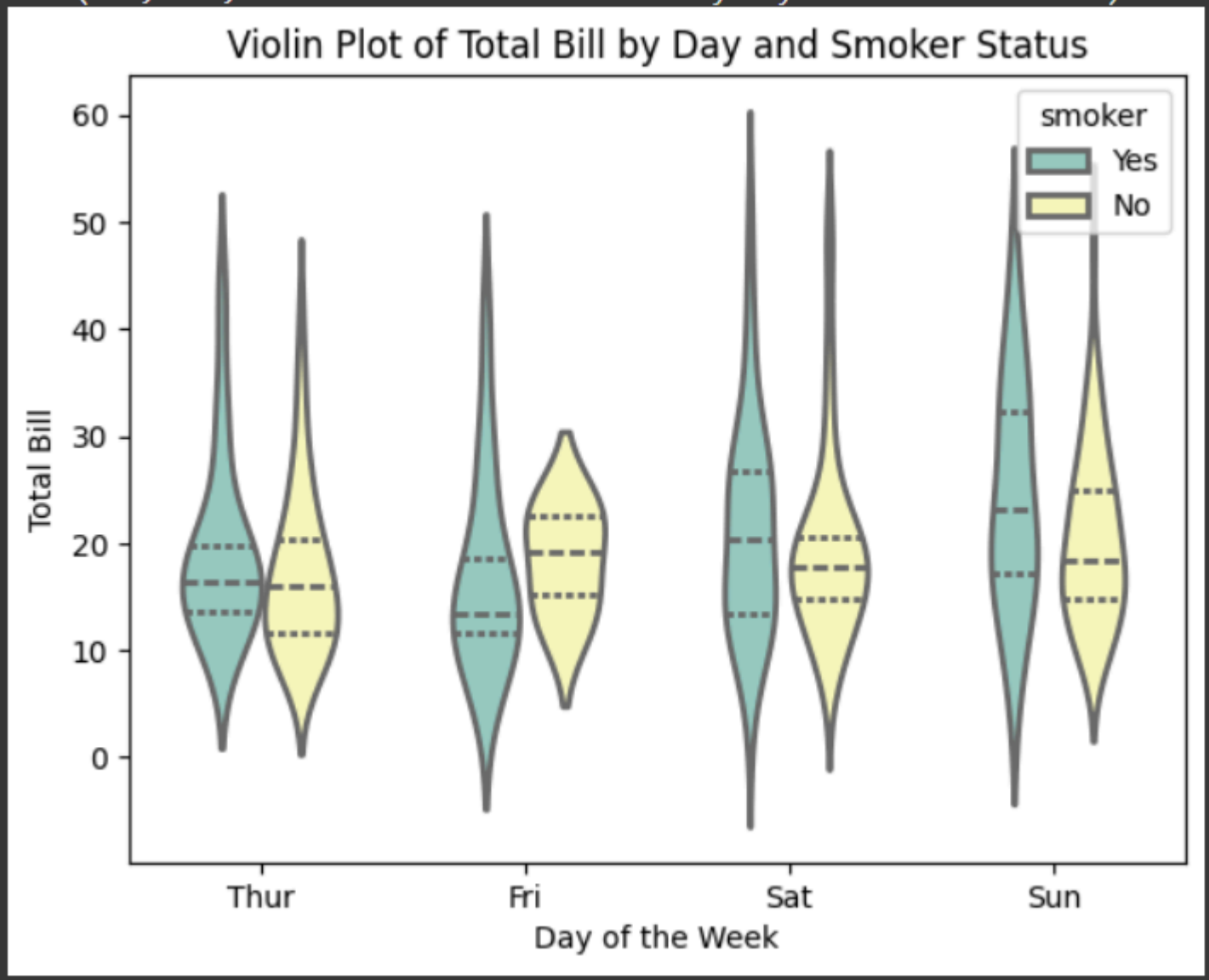

```
sns.boxplot(x="day", y="total_bill", hue="smoker", data =tips, linewidth=2, width=0.6, boxprops={"facecolor": "lightblue", "edgecolor": "darkblue"}, whiskerprops={"color": "black", "linestyle": "--", "linewidth": 1.5}, capprops={"color": "black", "linestyle": "--", "linewidth": 1.5})
```

```
<Axes: xlabel='day', ylabel='total_bill'>
```

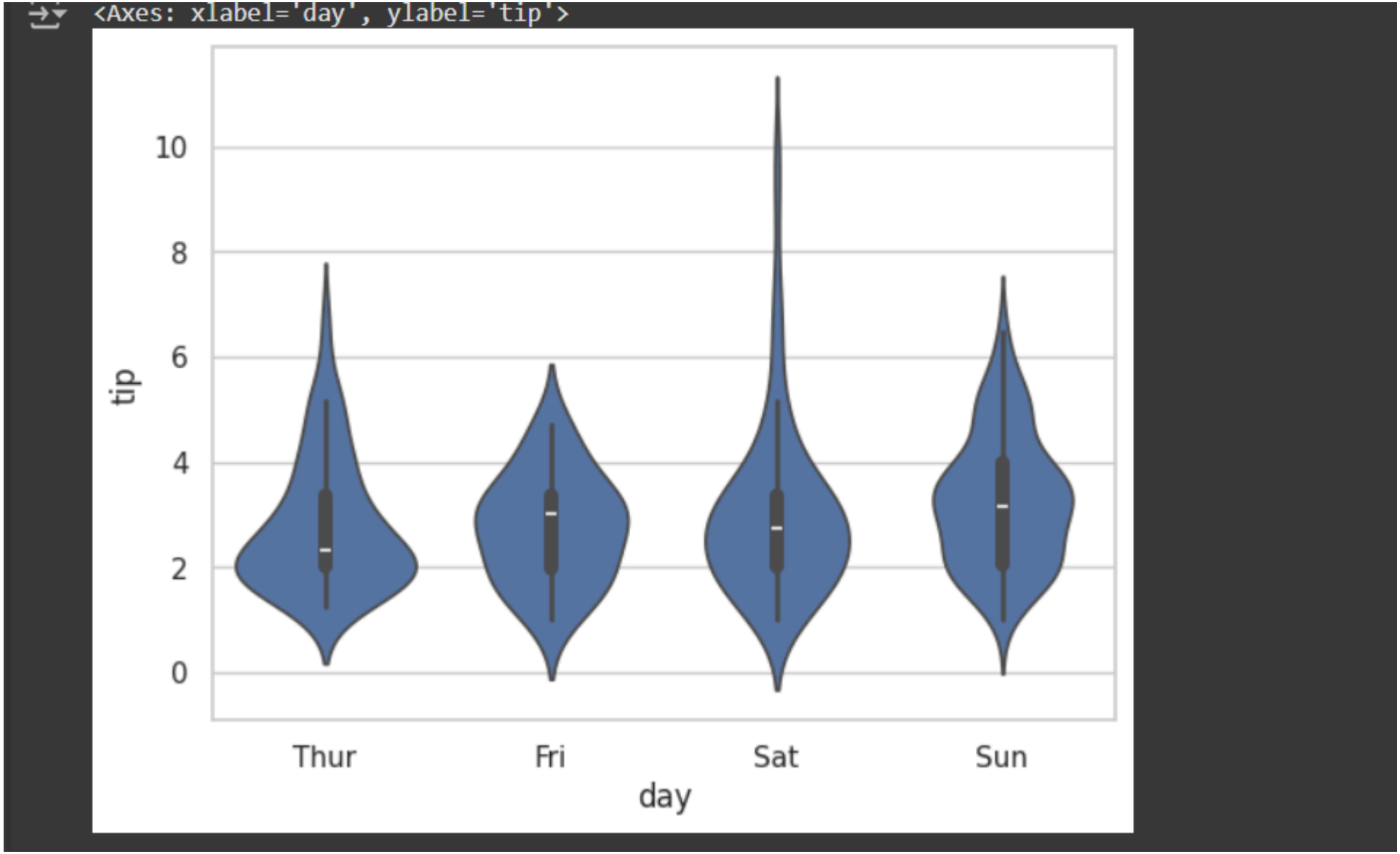



```
sns.violinplot(x="day", y="total_bill", hue="smoker", data =tips, linewidth=2, width=0.6, palette="Set3", inner ="quartile") #Add labels and title
plt.xlabel("Day of the Week") plt.ylabel("Total Bill") plt.title("Violin Plot of Total Bill by Day and Smoker Status")
```

```
Text(0.5, 1.0, 'Violin Plot of Total Bill by Day and Smoker Status')
```




```
import seaborn as sns sns.set(style = 'whitegrid') tip = sns.load_dataset('tips') sns.violinplot(x = 'day', y ='tip', data = tip)
```



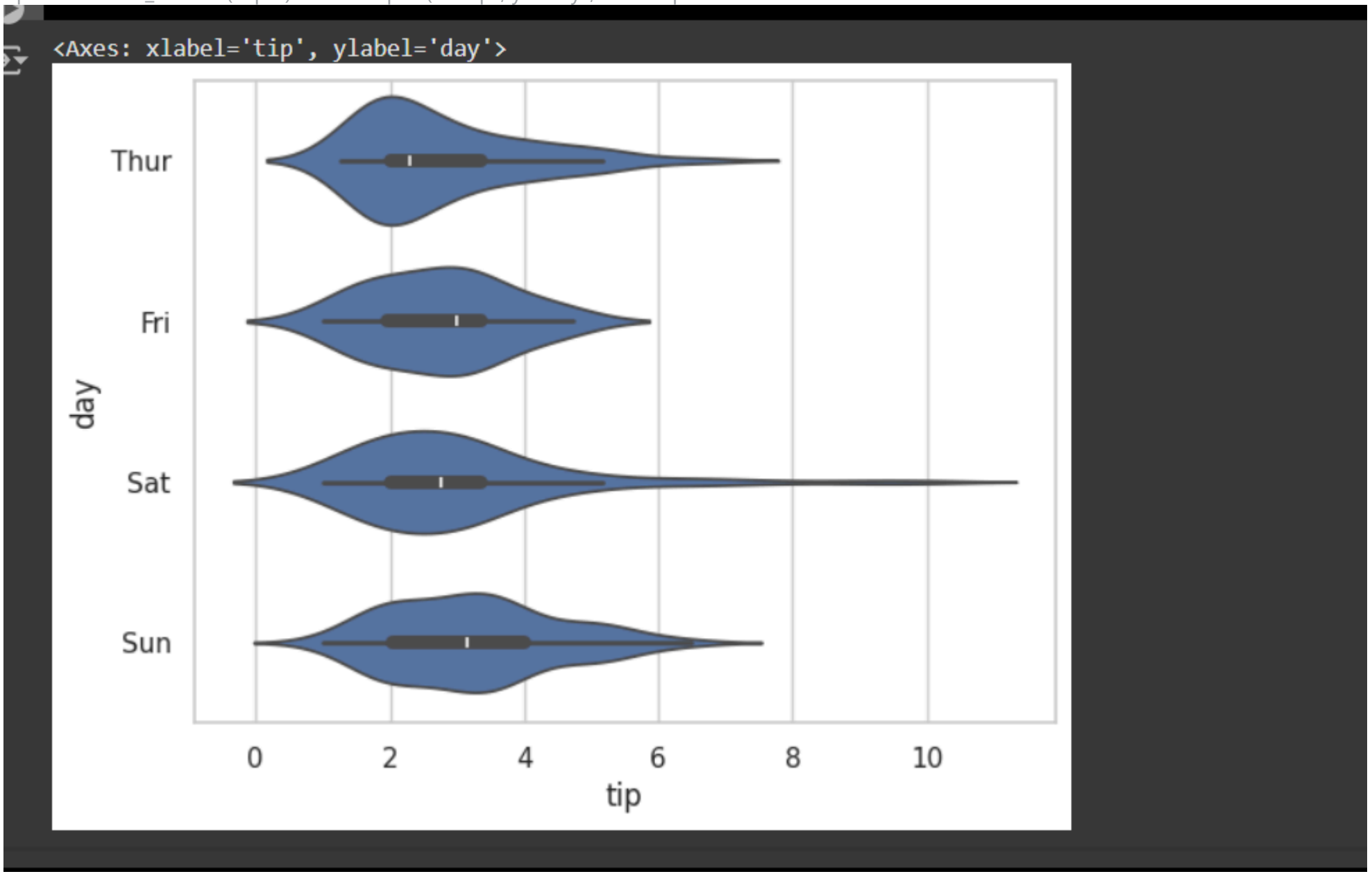
import seaborn as sns

use to set style of background of plot

sns.set(style="whitegrid")

loading data-set

tips = sns.load_dataset("tips") sns.violinplot(x="tip", y="day", data=tip



import pandas as pd import numpy as np import matplotlib.pyplot as plt import seaborn as sns

```
data = np.random.randint(low = 1, high = 100, size = (10,10)) print("The data to be plotted: \n") print(data)
```

```
The data to be plotted:

[[33 71 86 32 14 72 57 25 80 42]
 [19 41 55 80 12 39 94  2 96 45]
 [89 25 68 83  4 77 36 87 62 70]
 [88 44 33 12 85 11 55 38 29  3]
 [28 84 90 24 54 52 47 21 54 30]
 [68 36 40 10 74 42 24  4 47 91]
 [51  4 32 10 11 28 46 72 40 62]
 [86 98 45 35 35 89 34  6 37  1]
 [76 35 70 54 81 63  9 62  2 82]
 [36 92 41 37 49 26 68 36 31 30]]
```

Result:

The abpve code is executed successfully.