





```
import seaborn as sns
import matplotlib.pyplot as plt
df=sns.load_dataset("tips")
df
```



	total_bill	tip	sex	smoker	day	time	size	
0	16.99	1.01	Female	No	Sun	Dinner	2	
1	10.34	1.66	Male	No	Sun	Dinner	3	
2	21.01	3.50	Male	No	Sun	Dinner	3	
3	23.68	3.31	Male	No	Sun	Dinner	2	
4	24.59	3.61	Female	No	Sun	Dinner	4	
...	
239	29.03	5.92	Male	No	Sat	Dinner	3	
240	27.18	2.00	Female	Yes	Sat	Dinner	2	
241	22.67	2.00	Male	Yes	Sat	Dinner	2	
242	17.82	1.75	Male	No	Sat	Dinner	2	
243	18.78	3.00	Female	No	Thur	Dinner	2	

244 rows × 7 columns

Next steps:

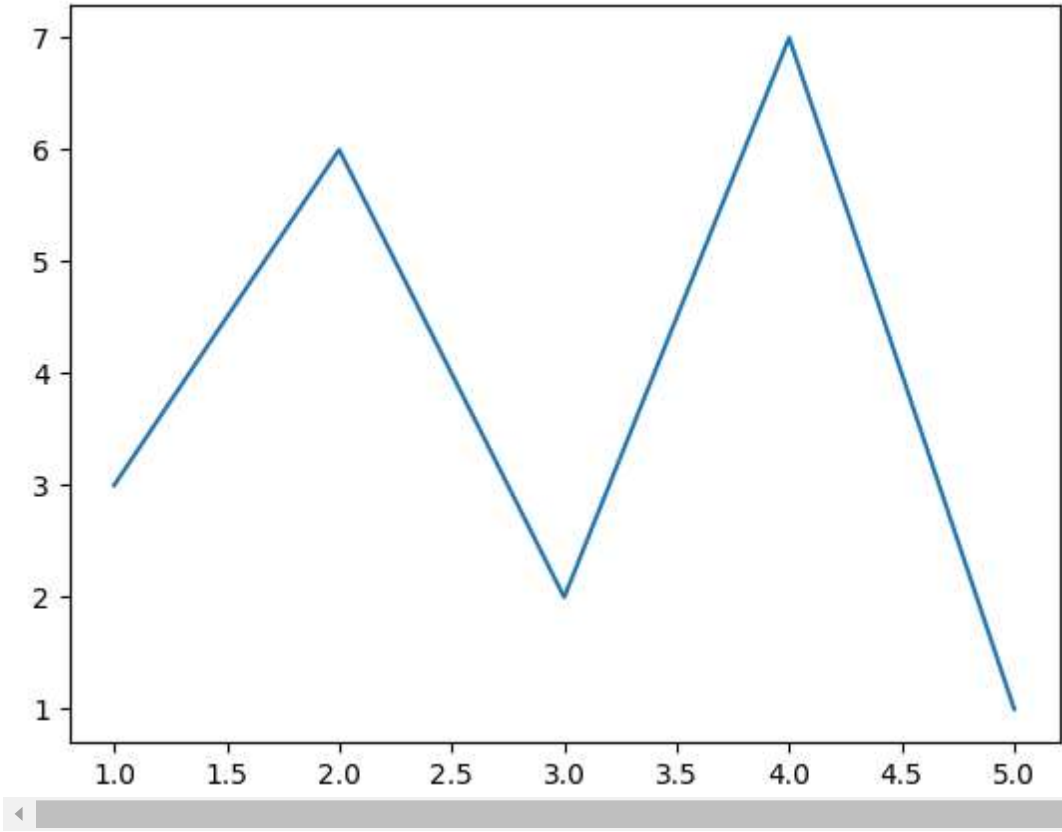
Generate code with df

 View recommended plots

New interactive sheet

```
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
x = [1, 2, 3, 4, 5]
y = [3, 6, 2, 7, 1]
sns.lineplot(x=x,y=y)
```

<Axes: >



Generate

a slider using jupyter widgets



Close

```
df = sns.load_dataset("tips")
df
```

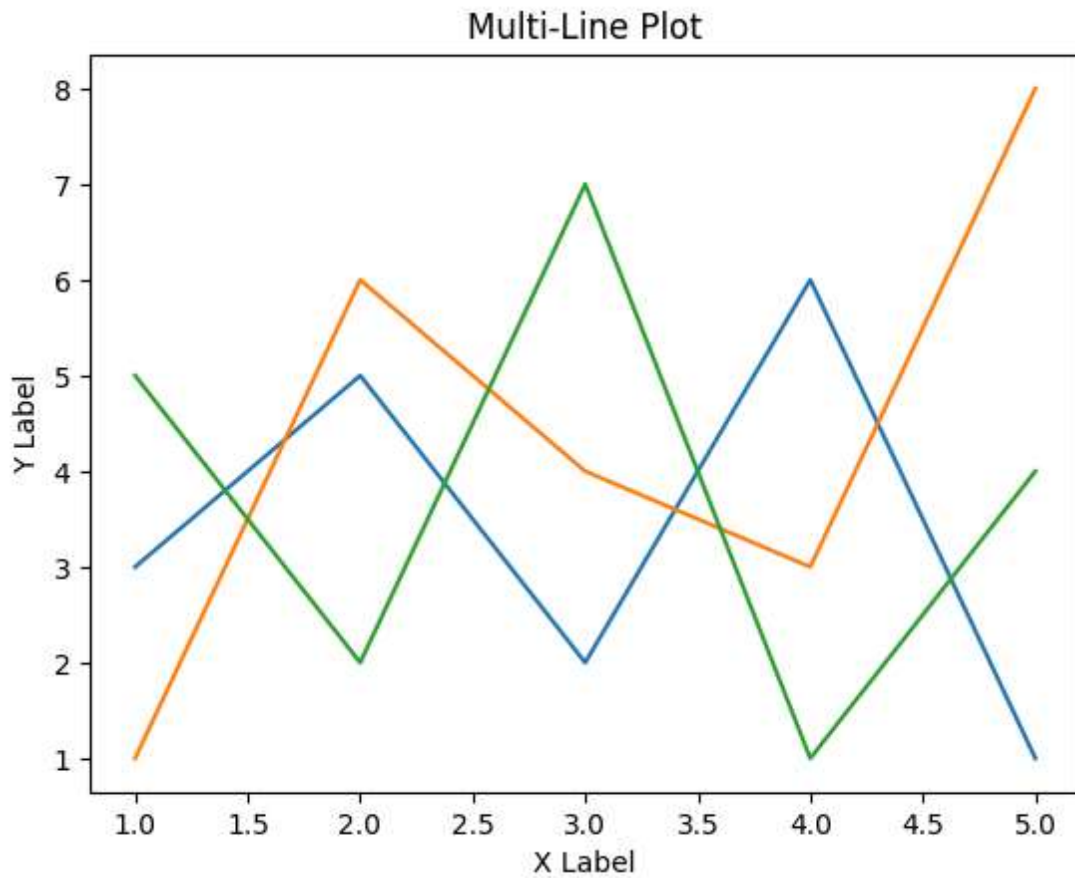
	total_bill	tip	sex	smoker	day	time	size	
0	16.99	1.01	Female	No	Sun	Dinner	2	
1	10.34	1.66	Male	No	Sun	Dinner	3	
2	21.01	3.50	Male	No	Sun	Dinner	3	
3	23.68	3.31	Male	No	Sun	Dinner	2	
4	24.59	3.61	Female	No	Sun	Dinner	4	
...	
239	29.03	5.92	Male	No	Sat	Dinner	3	
240	27.18	2.00	Female	Yes	Sat	Dinner	2	
241	22.67	2.00	Male	Yes	Sat	Dinner	2	
242	17.82	1.75	Male	No	Sat	Dinner	2	
243	18.78	3.00	Female	No	Thur	Dinner	2	

244 rows × 7 columns

```
x=[1, 2, 3, 4, 5]
y1=[3, 5, 2, 6, 1]
y2=[1, 6, 4, 3, 8]
y3=[5, 2, 7, 1, 4]
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")
```

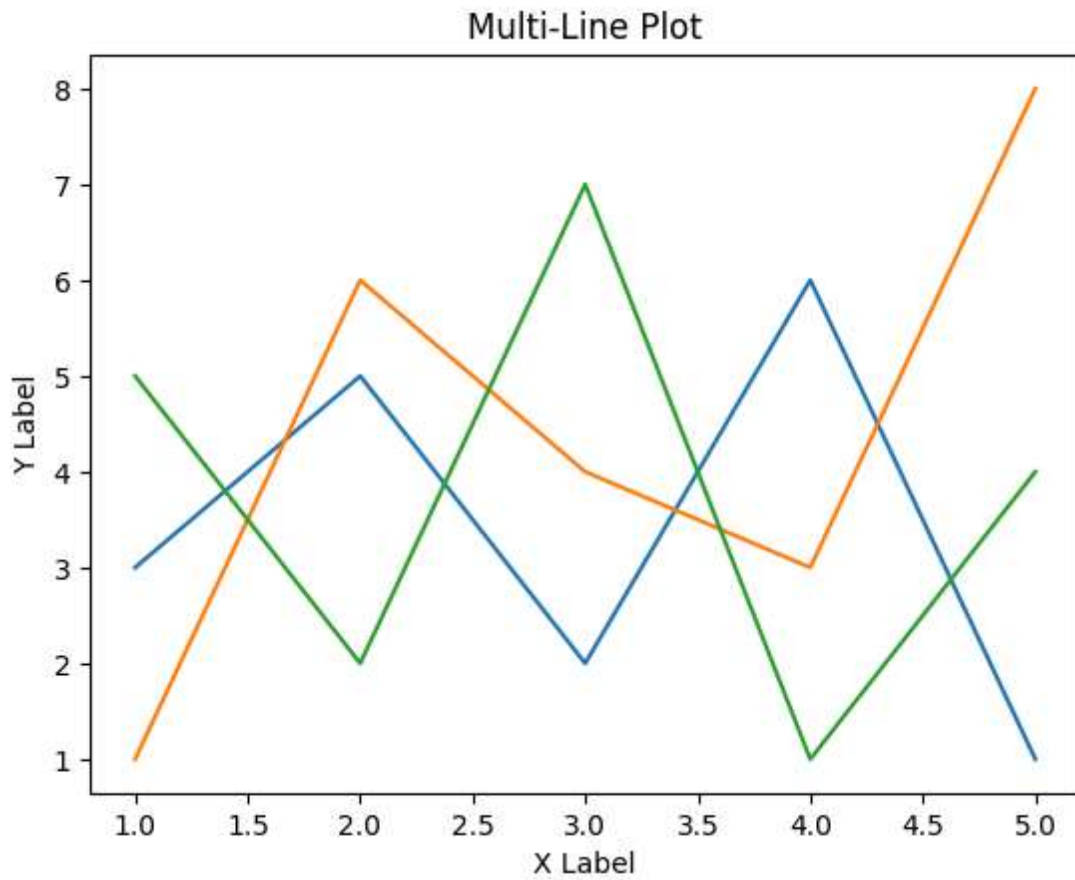


```
Text(0, 0.5, 'Y Label')
```



```
x=[1, 2, 3, 4, 5]
y1=[3, 5, 2, 6, 1]
y2=[1, 6, 4, 3, 8]
y3=[5, 2, 7, 1, 4]
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")
```

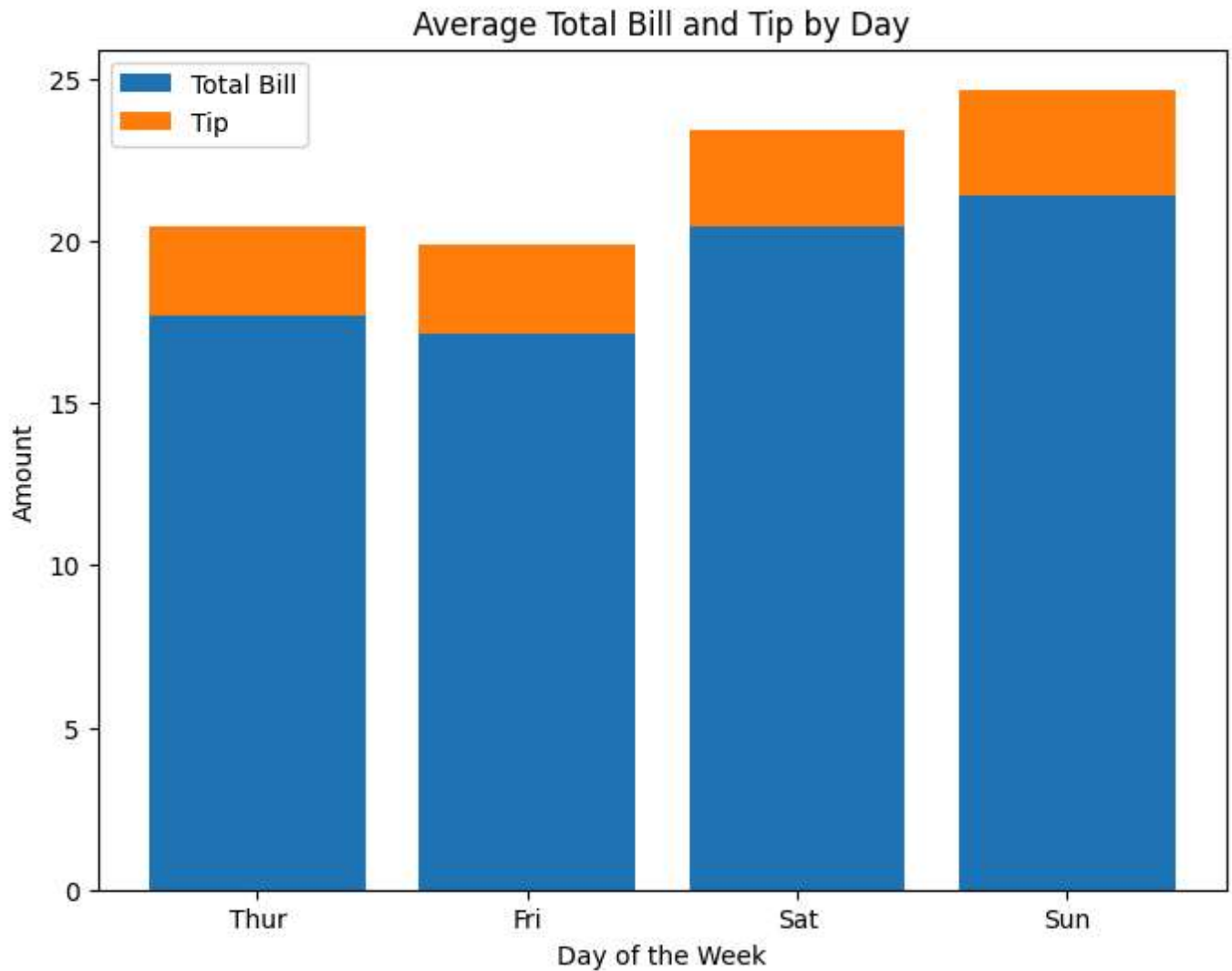
Text(0, 0.5, 'Y Label')



```
tips=sns.load_dataset('tips')
avg_total_bill = tips.groupby('day')['total_bill'].mean()
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')
plt.xlabel('Day of the Week')
plt.ylabel('Amount')
plt.title('Average Total Bill and Tip by Day')
plt.legend()
```



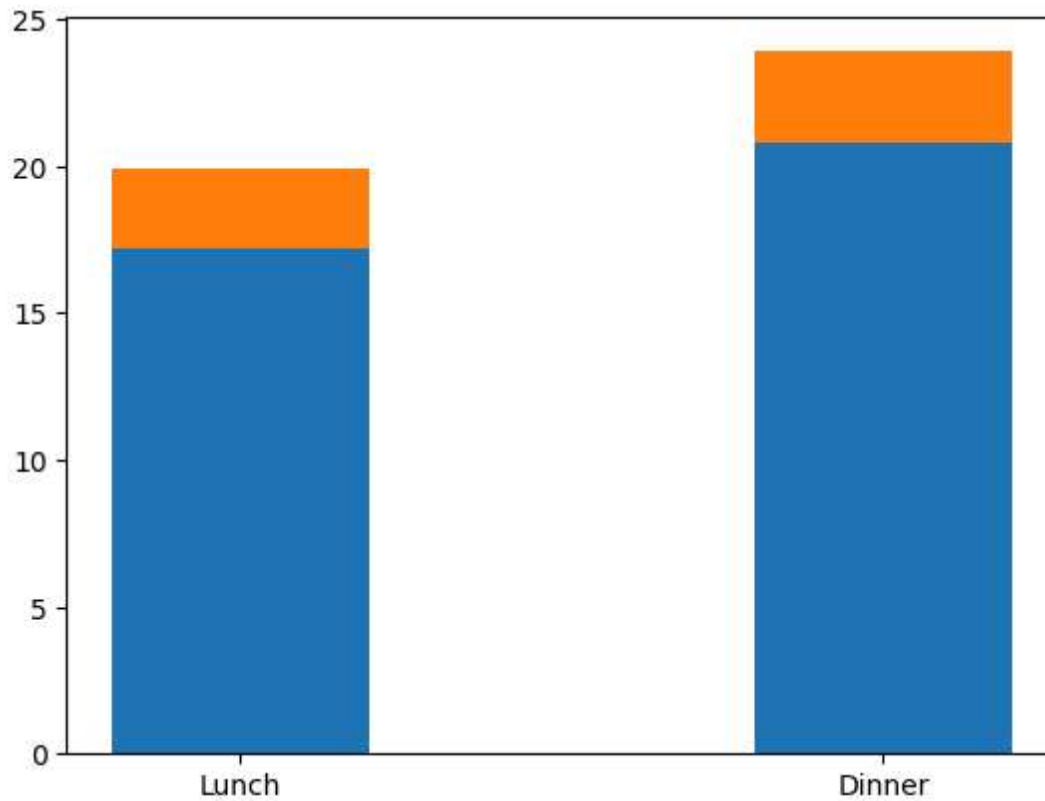
```
<ipython-input-6-92cb48b741d4>:2: FutureWarning: The default of observed=False is deprecated and w
    avg_total_bill = tips.groupby('day')['total_bill'].mean()
<ipython-input-6-92cb48b741d4>:3: FutureWarning: The default of observed=False is deprecated and w
    avg_tip = tips.groupby('day')['tip'].mean()
<matplotlib.legend.Legend at 0x79efd6104280>
```



```
avg_total_bill = tips.groupby('time')['total_bill'].mean()
avg_tip=tips.groupby('time') ['tip'].mean()
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)
```

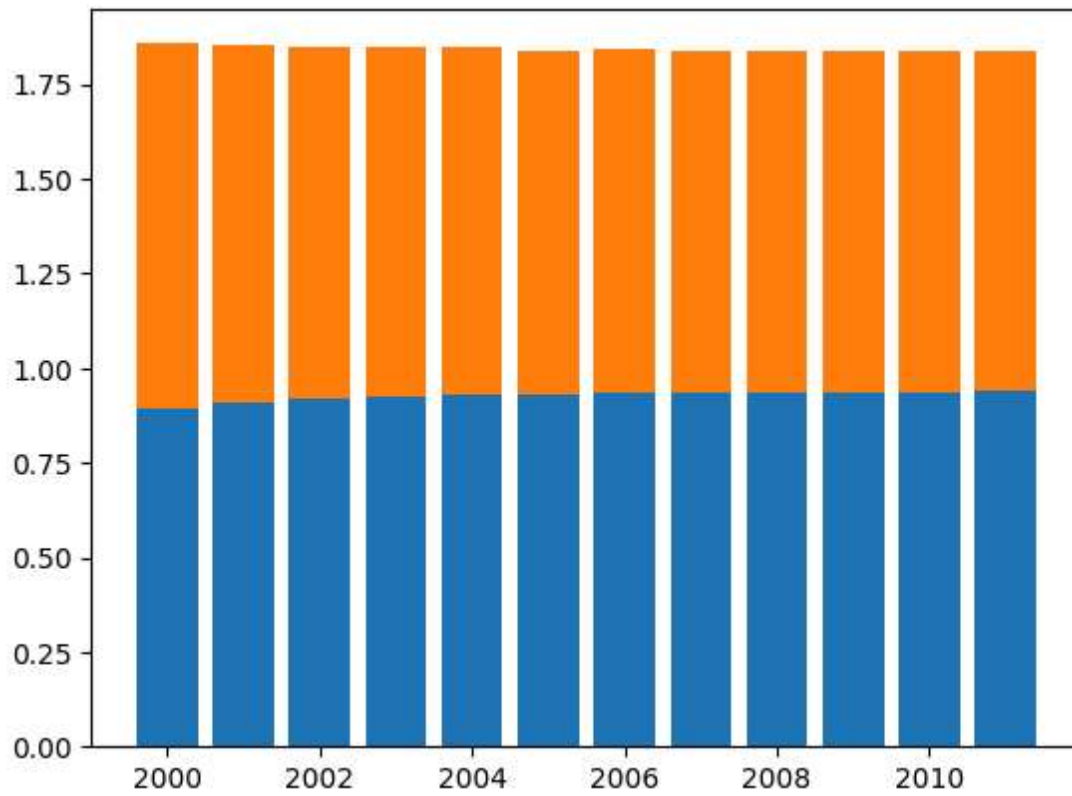


```
<ipython-input-7-a1bf10520fa8>:1: FutureWarning: The default of observed=False is deprecated and will be removed in a future version. To silence this warning, use observed=False. This will become the default in a future version.  
avg_total_bill = tips.groupby('time')['total_bill'].mean()  
<ipython-input-7-a1bf10520fa8>:2: FutureWarning: The default of observed=False is deprecated and will be removed in a future version. To silence this warning, use observed=False. This will become the default in a future version.  
avg_tip=tips.groupby('time') ['tip'].mean()
```



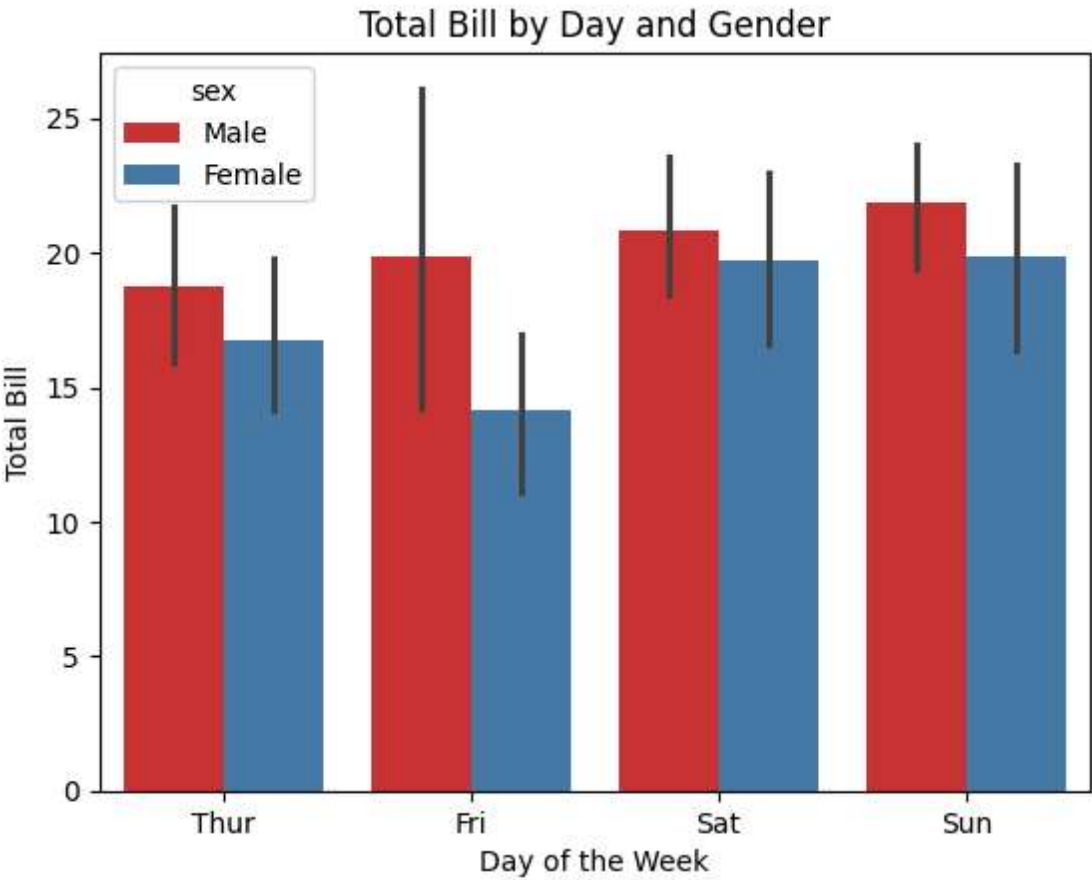
```
years=range(2000, 2012)  
apples=[0.895, 0.91, 0.919, 0.926, 0.929, 0.931, 0.934, 0.936, 0.937, 0.9375, 0.9372, 0.939]  
oranges = [0.962, 0.941, 0.930, 0.923, 0.918, 0.908, 0.907, 0.904, 0.901, 0.898, 0.9, 0.896, ]  
plt.bar(years, apples)  
plt.bar(years, oranges, bottom=apples)
```

↔ <BarContainer object of 12 artists>



```
import seaborn as sns
dt= sns.load_dataset('tips')
sns.barplot(x='day', y='total_bill', hue='sex', data=dt, palette='Set1')
plt.xlabel('Day of the Week')
plt.ylabel("Total Bill")
plt.title('Total Bill by Day and Gender')
```

```
Text(0.5, 1.0, 'Total Bill by Day and Gender')
```



```
tit=pd.read_csv("titanic_dataset.csv")
tit
```




	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN
...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN

891 rows × 12 columns

Next steps:

Generate code with `tit`



View recommended plots

New interactive sheet

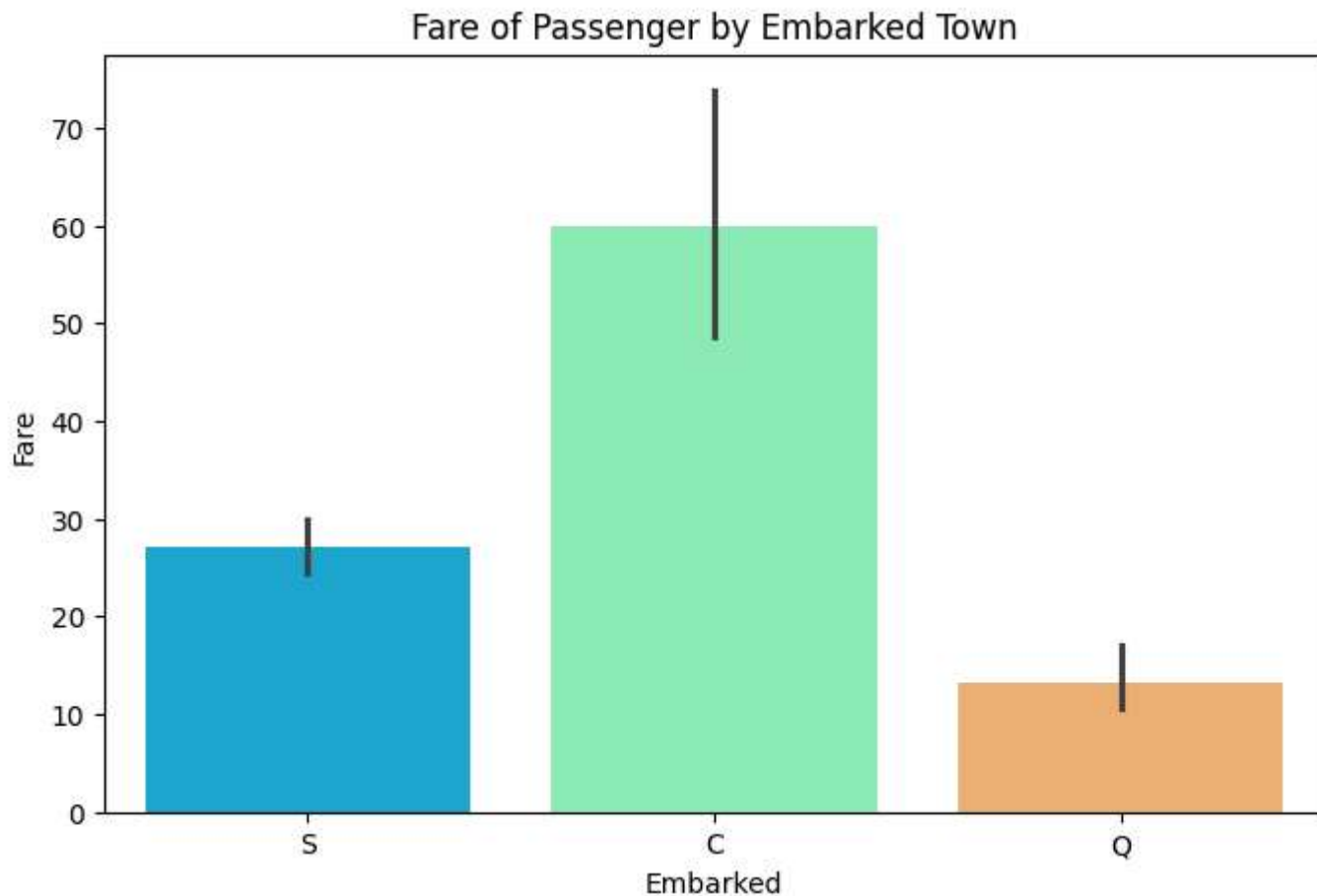
✓ New Section

```
plt.figure(figsize=(8,5))
sns.barplot(x='Embarked', y='Fare', data=tit, palette='rainbow')
plt.title("Fare of Passenger by Embarked Town")
```

↗ <ipython-input-12-a78cb65e3ba7>:2: FutureWarning:

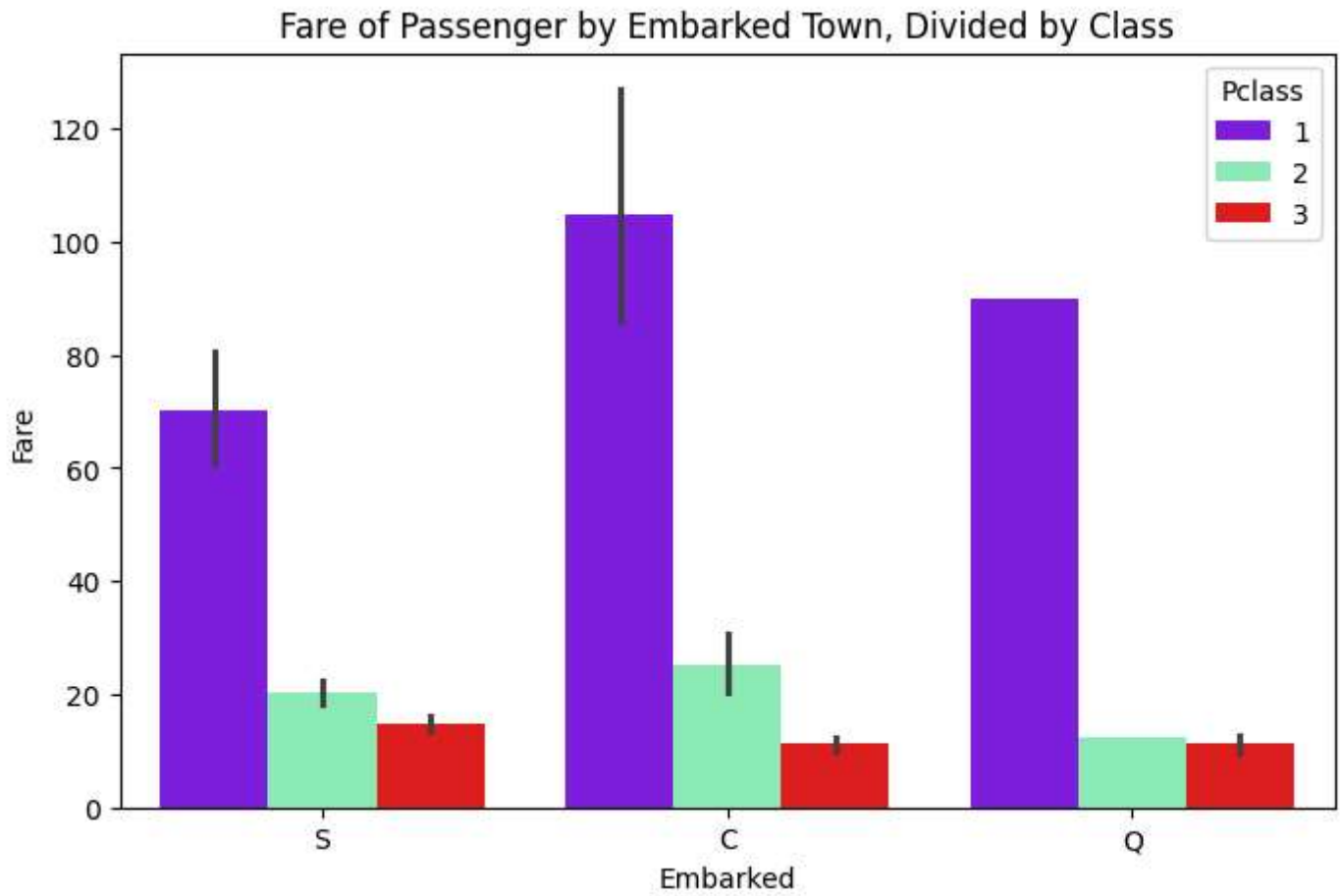
Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the

```
sns.barplot(x='Embarked', y='Fare', data=tit, palette='rainbow')
Text(0.5, 1.0, 'Fare of Passenger by Embarked Town')
```



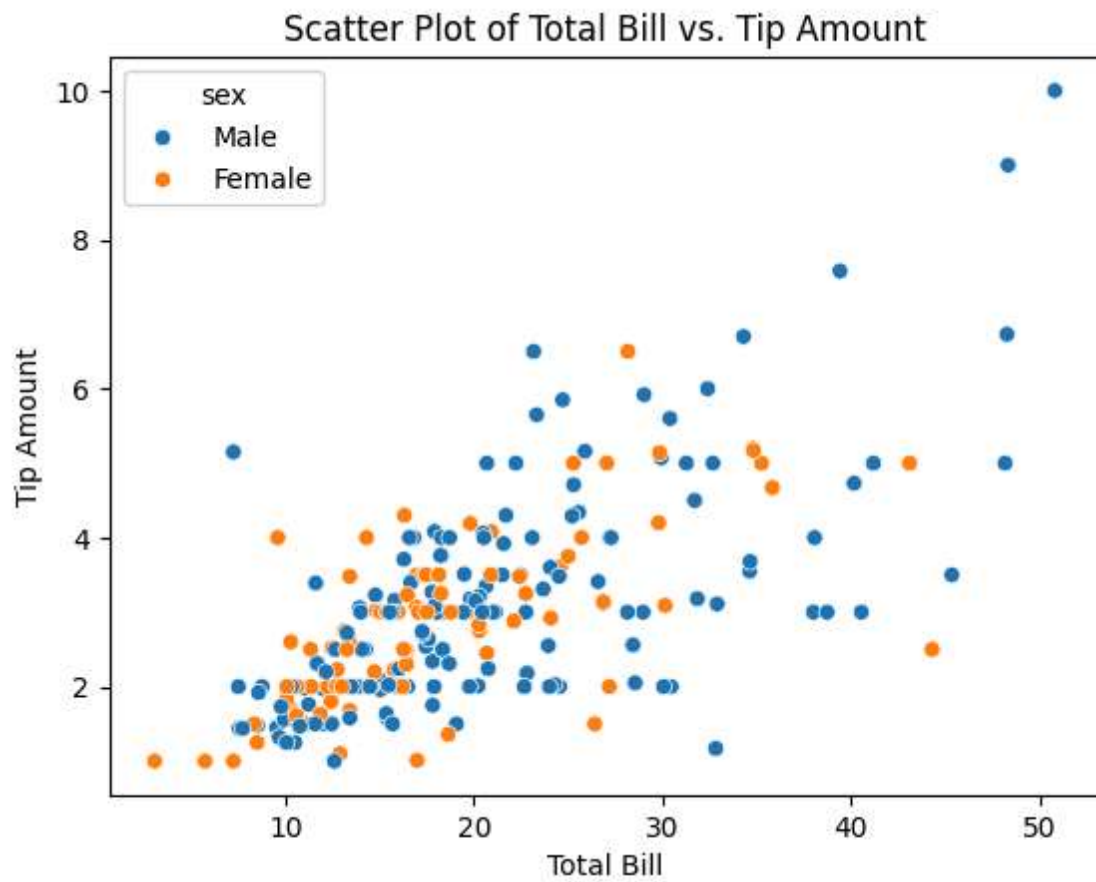
```
plt.figure(figsize=(8,5))
sns.barplot(x='Embarked', y='Fare', data=tit, palette='rainbow', hue='Pclass')
plt.title("Fare of Passenger by Embarked Town, Divided by Class")
```

Text(0.5, 1.0, 'Fare of Passenger by Embarked Town, Divided by Class')



```
tips=sns.load_dataset('tips')
sns.scatterplot(x='total_bill', y='tip', hue='sex', data=tips)
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')



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



Numerical variable

0	-0.357577
1	-1.410512
2	0.366449
3	-0.703565
4	-1.109753
...	...
995	0.180852
996	-1.590946
997	-0.833726
998	-0.057623
999	0.973418

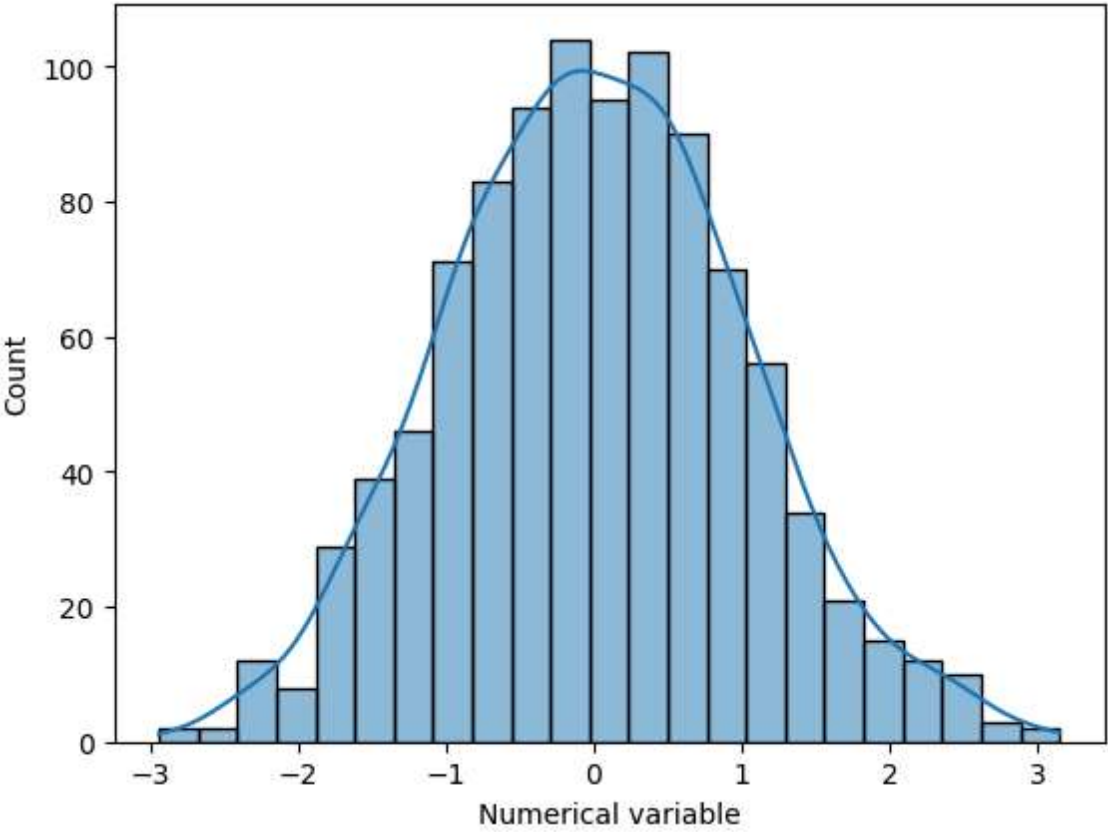
1000 rows × 1 columns

dtype: float64

```
sns.histplot(data = num_var, kde = True)
```



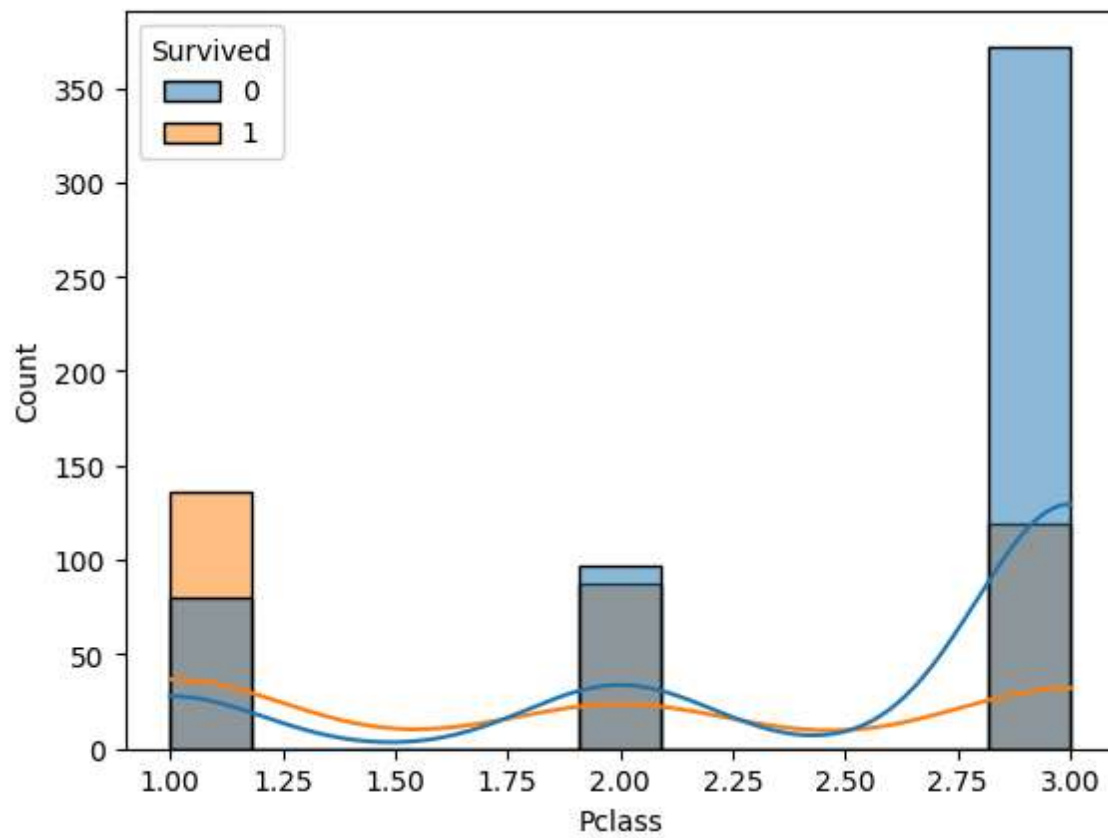
<Axes: xlabel='Numerical variable', ylabel='Count'>



```
df=pd.read_csv("titanic_dataset.csv")
```

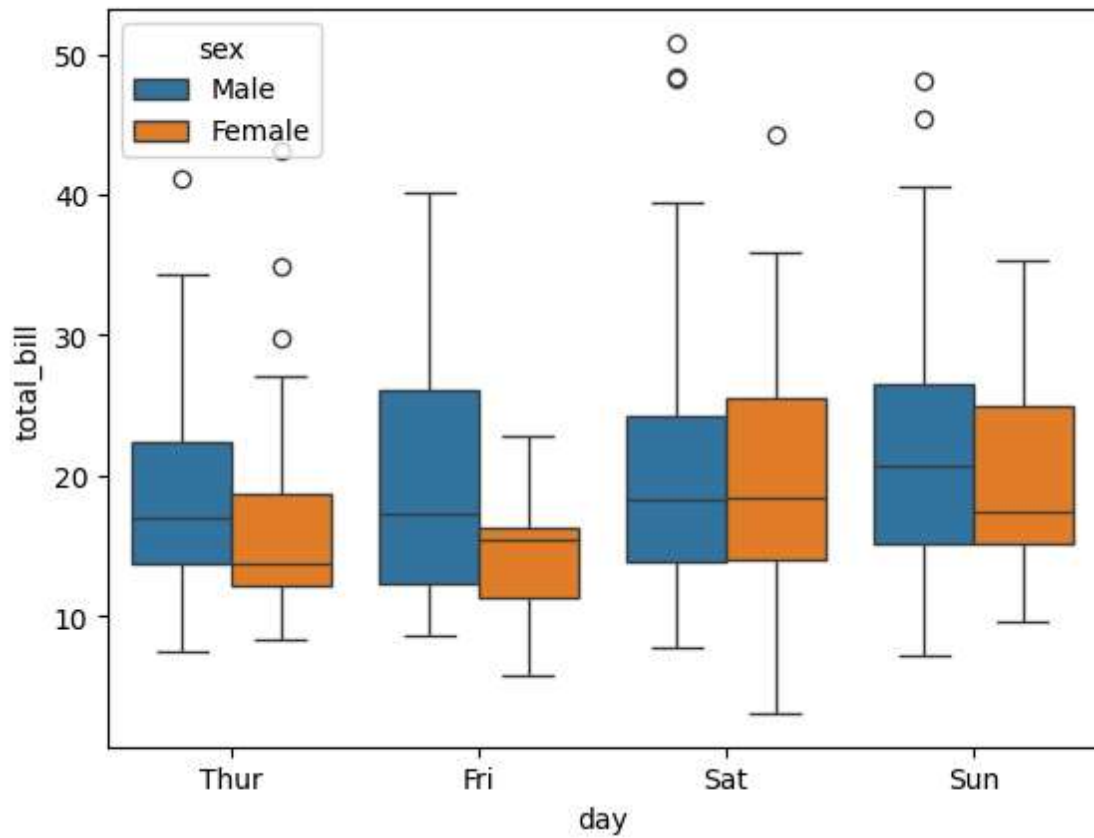
```
sns.histplot(data=df,x="Pclass", hue="Survived", kde=True)
```

↔ <Axes: xlabel='Pclass', ylabel='Count'>



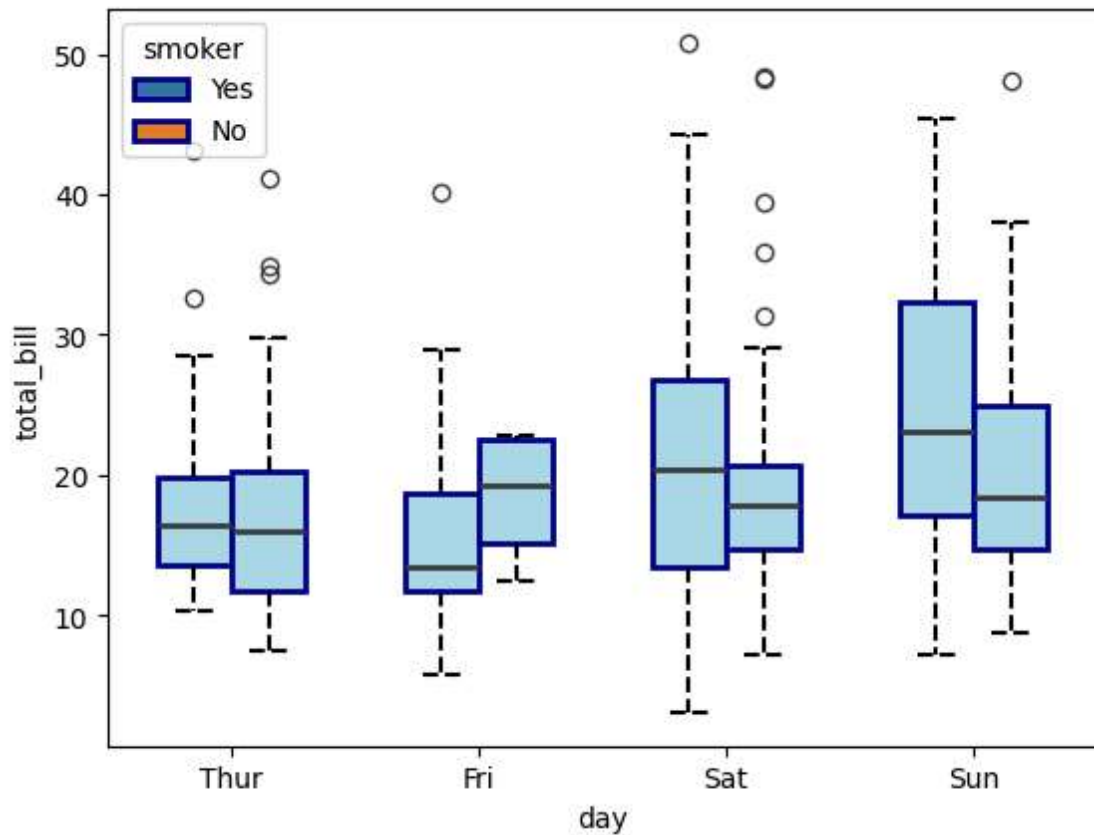
```
tips=sns.load_dataset('tips')  
sns.boxplot(x=tips['day'], y=tips ['total_bill'], hue=tips['sex'])
```

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



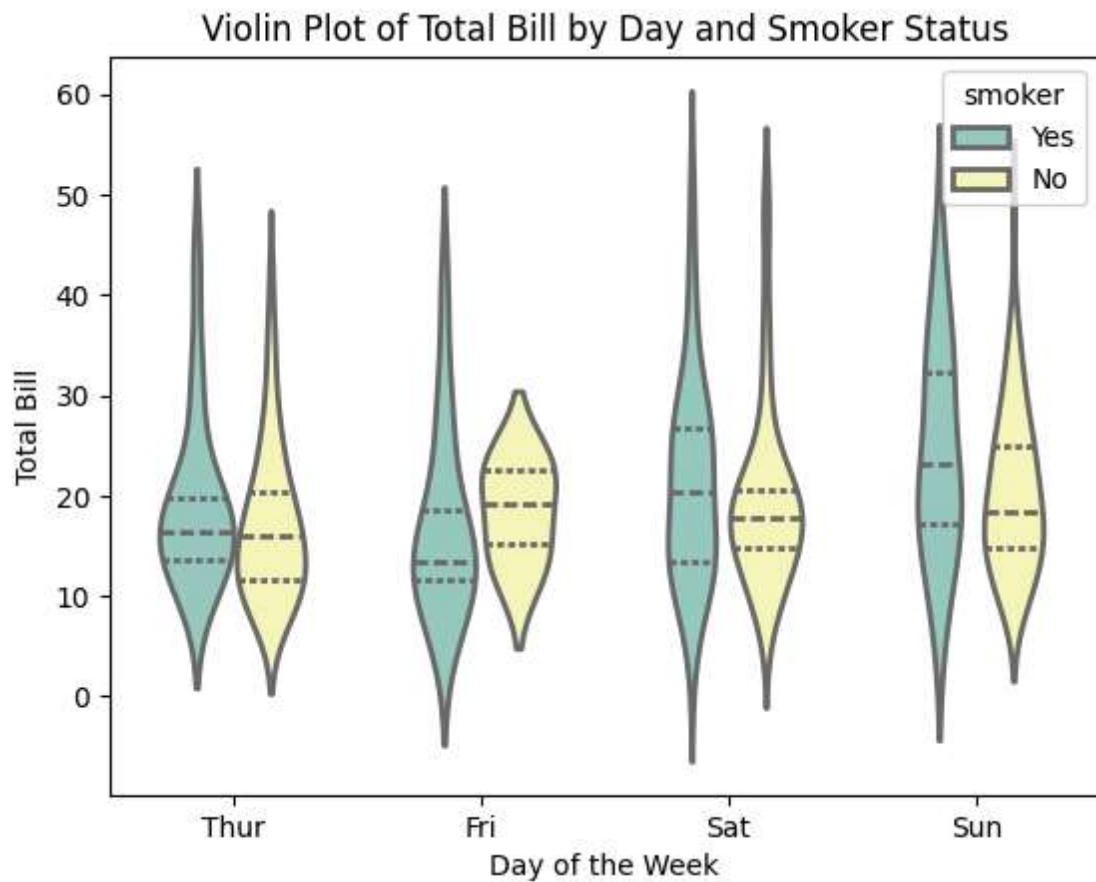
```
sns.boxplot(x="day", y="total_bill", hue="smoker", data=tips, linewidth=2, width=0.6, boxprops={"facecolor": "lightblue", "stroke": "darkblue", "strokewidth": 2}, whiskerprops={"color": "black", "linestyle": "--", "linewidth": 1.5}, capprops={"color": "black", "linestyle": "solid", "linewidth": 1.5})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

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



```
sns.violinplot(x="day", y="total_bill", hue="smoker", data=tips, linewidth=2, width=0.6, palette="Set3")
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')



```
mart=pd.read_csv("titanic_dataset.csv")
mart
```




	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN
...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN

891 rows × 12 columns


Next steps:

Generate code with [mart](#)

☒ View recommended plots

[New interactive sheet](#)

```
mart=mart[['PassengerId', 'Survived', 'Age', 'Name', 'Ticket', 'Embarked']]
mart.head(10)
```



	PassengerId	Survived	Age	Name	Ticket	Embarked
0	1	0	22.0	Braund, Mr. Owen Harris	A/5 21171	S
1	2	1	38.0	Cumings, Mrs. John Bradley (Florence Briggs Th...	PC 17599	C
2	3	1	26.0	Heikkinen, Miss. Laina	STON/O2. 3101282	S
3	4	1	35.0	Futrelle, Mrs. Jacques Heath (Lily May Peel)	113803	S
4	5	0	35.0	Allen, Mr. William Henry	373450	S
5	6	0	NaN	Moran, Mr. James	330877	Q
6	7	0	54.0	McCarthy, Mr. Timothy J	17463	S
7	8	0	2.0	Palsson, Master. Gosta Leonard	349909	S
8	9	1	27.0	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	347742	S
9	10	1	14.0	Nasser, Mrs. Nicholas (Adele Achem)	237736	G

