| |
|---|
| Name: Shivam Pandey |
| Roll no:28 |
| Experiment No. 4 |
| To explore the data visualization techniques |
| Date of Performance:13/02/2024 |
| Date of Submission:26/03/2024 |

**Experiment No. 4**

**1. Aim:** To explore the data visualization techniques.

**Dataset:** In this experiment, data visualization techniques are explored on seaborn tips dataset.

**2. Software used:** Google Colaboratory / Jupyter Notebook

**3. Theory :-** Visualizations make it easier to explore and extract relevant information from the data by identifying patterns, relationships, outliers, and much more. Seaborn is a statistical plotting library in Python and is an extended version of Matplotlib. It supports complex visualizations and makes the plotting of graphs simple and intuitive. It can be used in Python scripts, Jupyter notebook, and web application servers.Seaborn uses less syntax as compared to Matplotlib. Hence, it is easier to use.It is easier to customize themes and high-level interfaces in Seaborn to make the plots more attractive and readable.Seaborn is much more functional and organized than Matplotlib and is better integrated to work with Pandas DataFrames.

Seaborn provides different plots for different types of variables as follows:

a. Frequency Distribution - Categorical Variables

* countplot
* catplot

b. Distribution of the Numerical Variable

* distplot(histogram)
* kdeplot
* boxplot
* violinplot

c. Relationship between 2 Numerical Variables

* lineplot
* scatterplot
* relplot
* lmplot
* heatmap
* pairplot
* facetgrid

d. Relationship between Numerical and Categorical Variables

* pointplot
* barplot
* boxplot
* violinplot
* swarmplot
* catplot
* facetgrid

**4. Program:**

[112]:
```
import numpy as np
import pandas as pd
import seaborn as sns
from scipy import stats
import matplotlib.pyplot as plt
from sklearn.preprocessing import LabelEncoder
```

[113]:
```
tip = pd.read_csv('/content/tips-expt4.csv')
```

[114]:
```
tip.head()
```

[114]:

| | 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 |

[115]:
```
tip.isnull().sum()
```

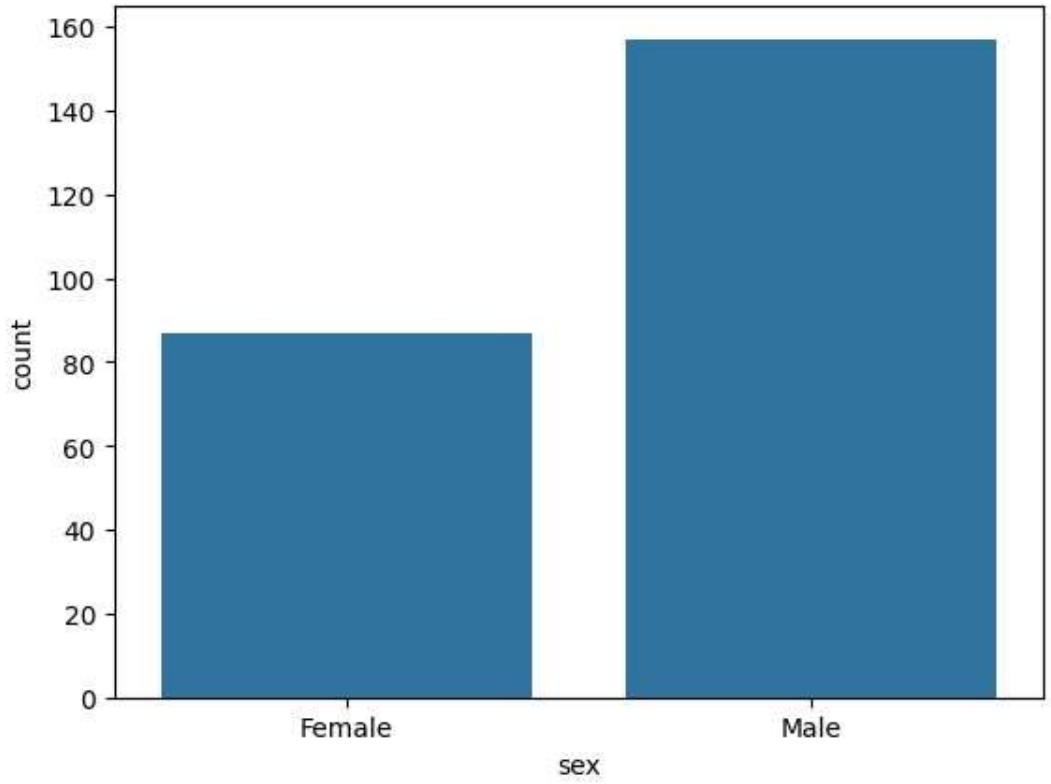
[115]:
```
total_bill
           0 tip 0 sex
           0 smoker
           0 day 0
      time 0 size
           0
      dtype: int64
```

#1.FREQUNCY DISTRIBITION

[116]:
```
tip['sex'].value_counts()
```

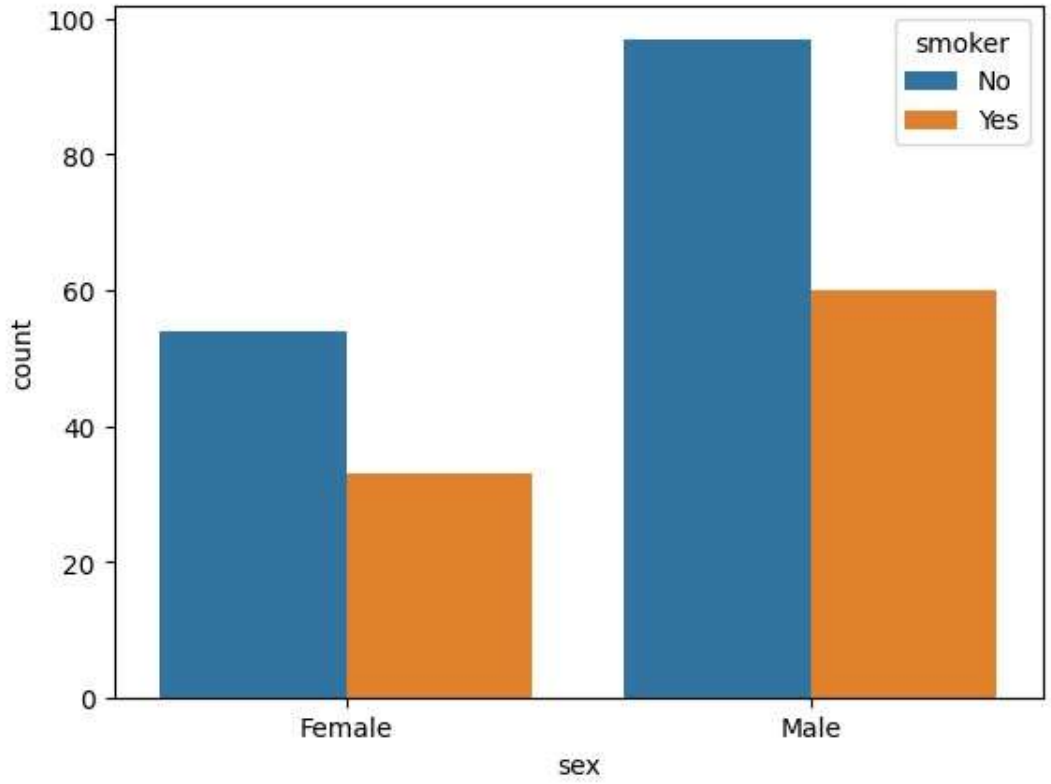
[116]:
```
Male 157 Female
87
Name: sex, dtype: int64
```

[117]:
```
<Axes: xlabel='sex', ylabel='count'>
```



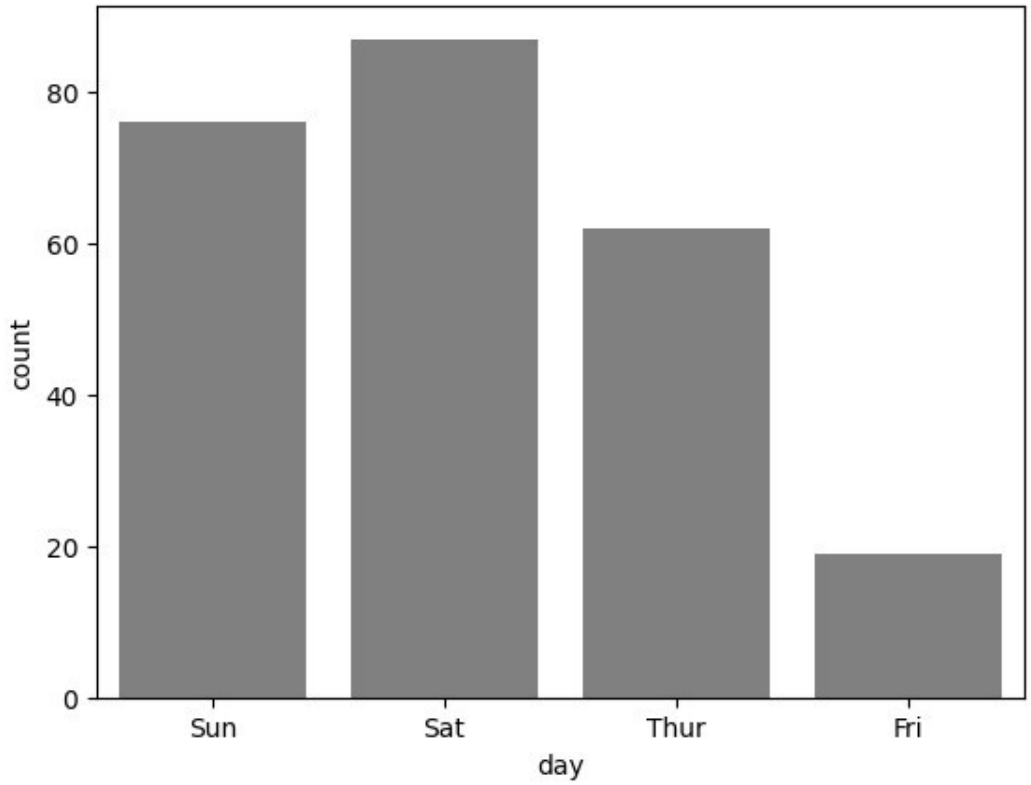

[118]:
```
sns.countplot(tip,x='sex',hue='smoker')
```

[118]:
```
<Axes: xlabel='sex', ylabel='count'>
```



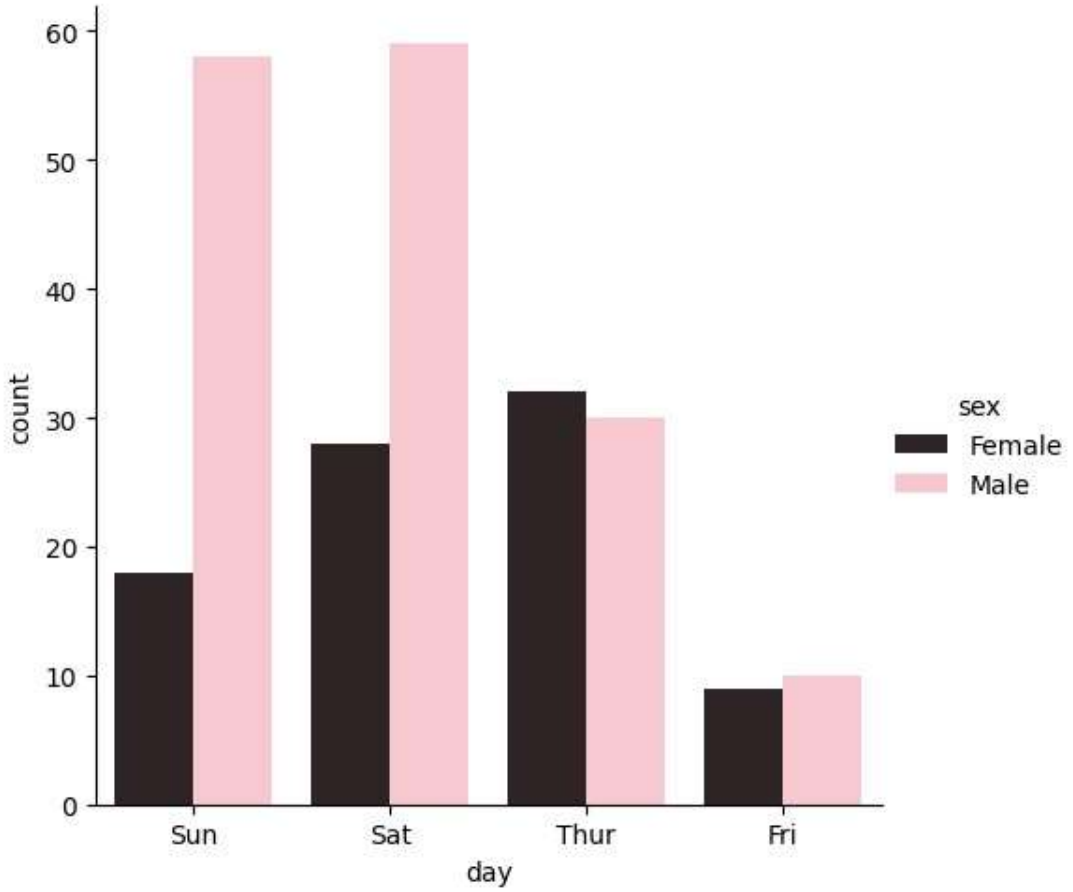

[119]:
```
sns.countplot(tip,x='day',color='grey')
```

[119]: <Axes: xlabel='day', ylabel='count'>




[120]:
```
sns.catplot(tip,x='day',hue='sex',kind='count',color='pink')
```

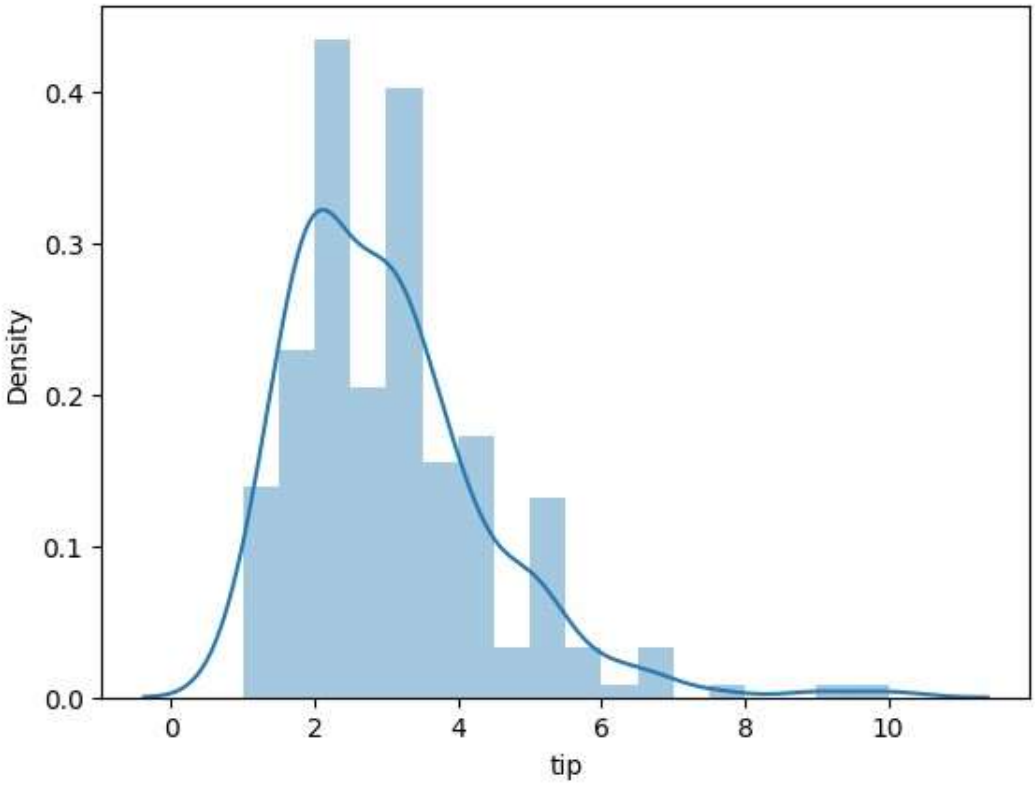
[120]: <seaborn.axisgrid.FacetGrid at 0x7d67d9cd64d0>




#2.Numerical Value

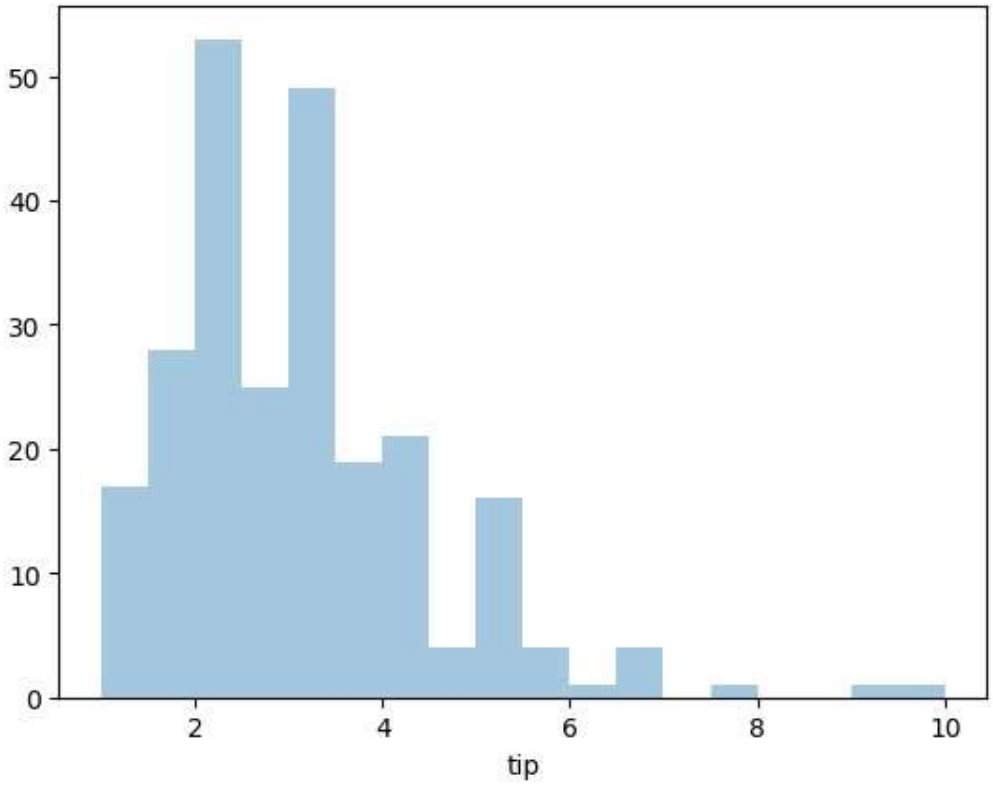
```
[121]: sns.distplot(tip['tip'])
```

```
[121]: <Axes: xlabel='tip', ylabel='Density'>
```



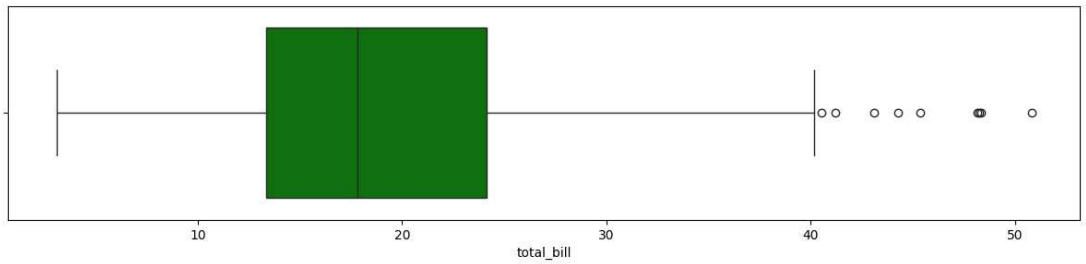

[122]:
```
sns.distplot(tip.tip,kde=False)
```

[122]: <Axes: xlabel='tip'>



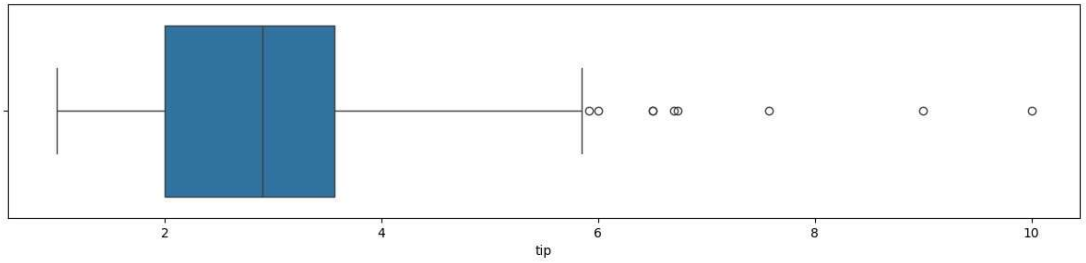

[123]:
```
plt.figure(figsize=(15,3))
sns.boxplot(x='total_bill',data= tip,color='g')
```

[123]: <Axes: xlabel='total_bill'>



[124]:
```
plt.figure(figsize=(15,3))
sns.boxplot(x='tip',data= tip)
```

[124]: <Axes: xlabel='tip'>




[125]:
```

```
tip

[125]:

| | 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 x 7 columns]

[126]:
```
bill_tips = pd.DataFrame(tip,columns=['total_bill','tip','size'])
```

[127]:
```
bill_tips
```

[127]:

| | total_bill | tip | size |
|---|---|---|---|
| 0 | 16.99 | 1.01 | 2 |
| 1 | 10.34 | 1.66 | 3 |
| 2 | 21.01 | 3.50 | 3 |
| 3 | 23.68 | 3.31 | 2 |
| 4 | 24.59 | 3.61 | 4 |
| .. | … | … | … |
| 239 | 29.03 | 5.92 | 3 |
| 240 | 27.18 | 2.00 | 2 |
| 241 | 22.67 | 2.00 | 2 |
| 242 | 17.82 | 1.75 | 2 |
| 243 | 18.78 | 3.00 | 2 |

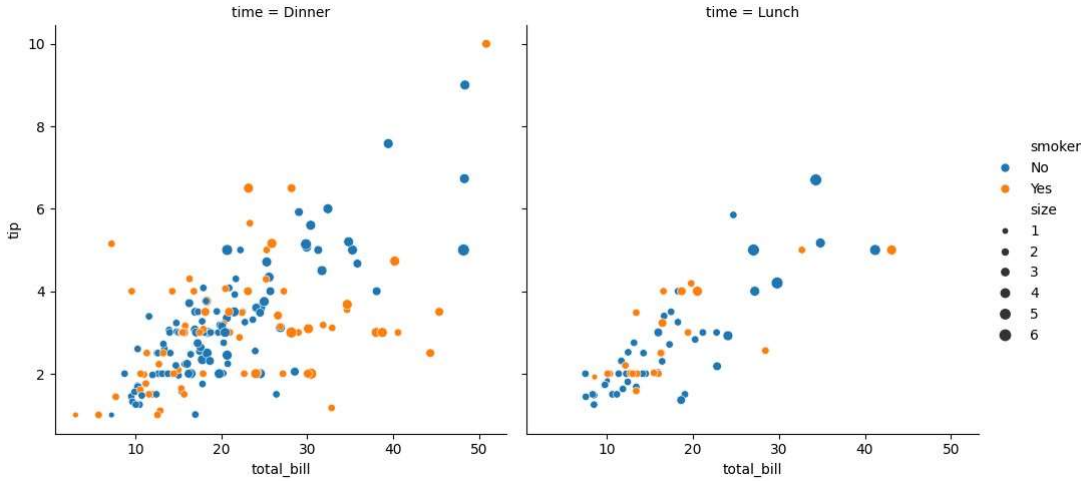
[244 rows x 3 columns]

[128]:
```
print("IQR FOR TOTAL BILL :
",stats.iqr(bill_tips['total_bill'])) print("IQR FOR TIPS :
",stats.iqr(bill_tips['tip']))
```

```
IQR FOR TOTAL BILL : 10.779999999999998
IQR FOR TIPS : 1.5625
```

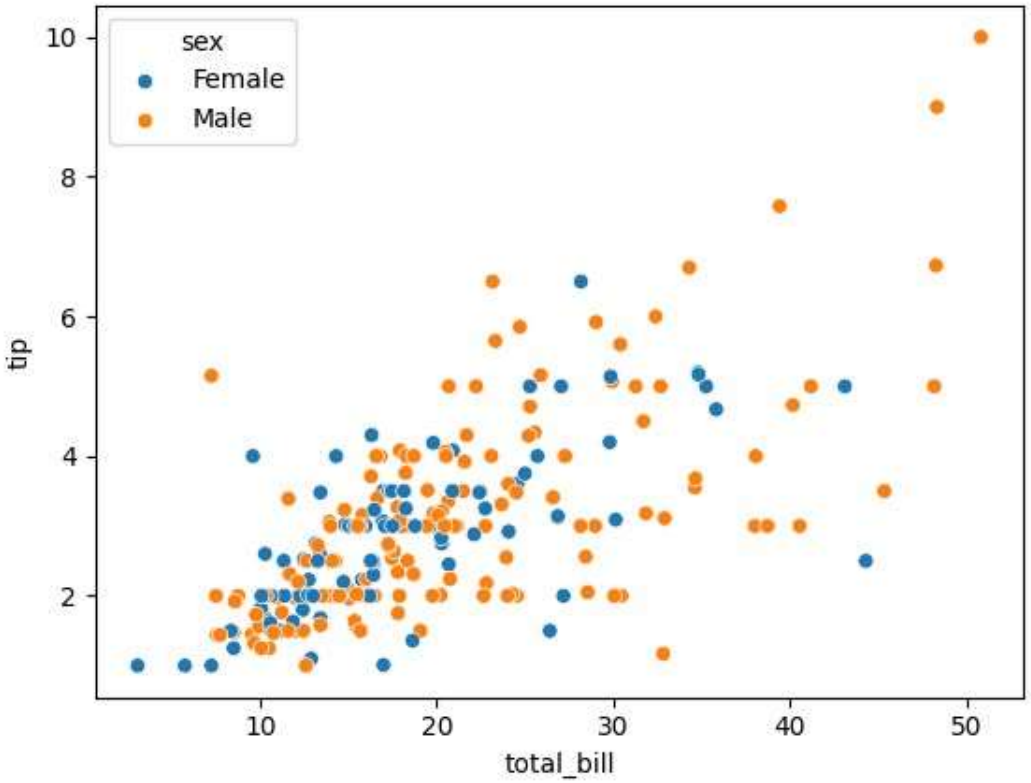
[129]:
```
sns.relplot(x='total_bill',y='tip',data=tip,col='time',hue='smoker',size='s
ize')
```

[129]: <seaborn.axisgrid.FacetGrid at 0x7d67d8ec3730>



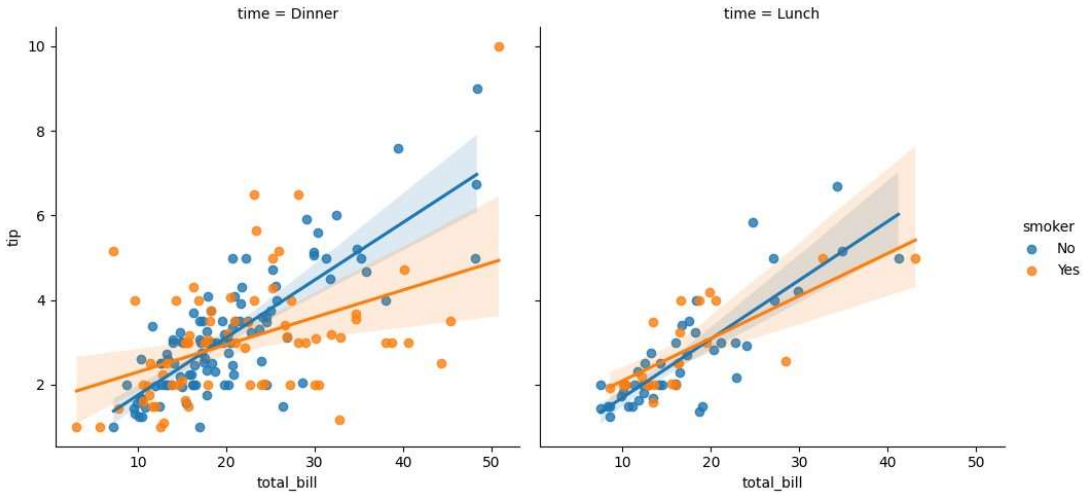

[130]:
```
sns.scatterplot(data=tip,x='total_bill',y='tip',hue='sex')
```

[130]: <Axes: xlabel='total_bill', ylabel='tip'>



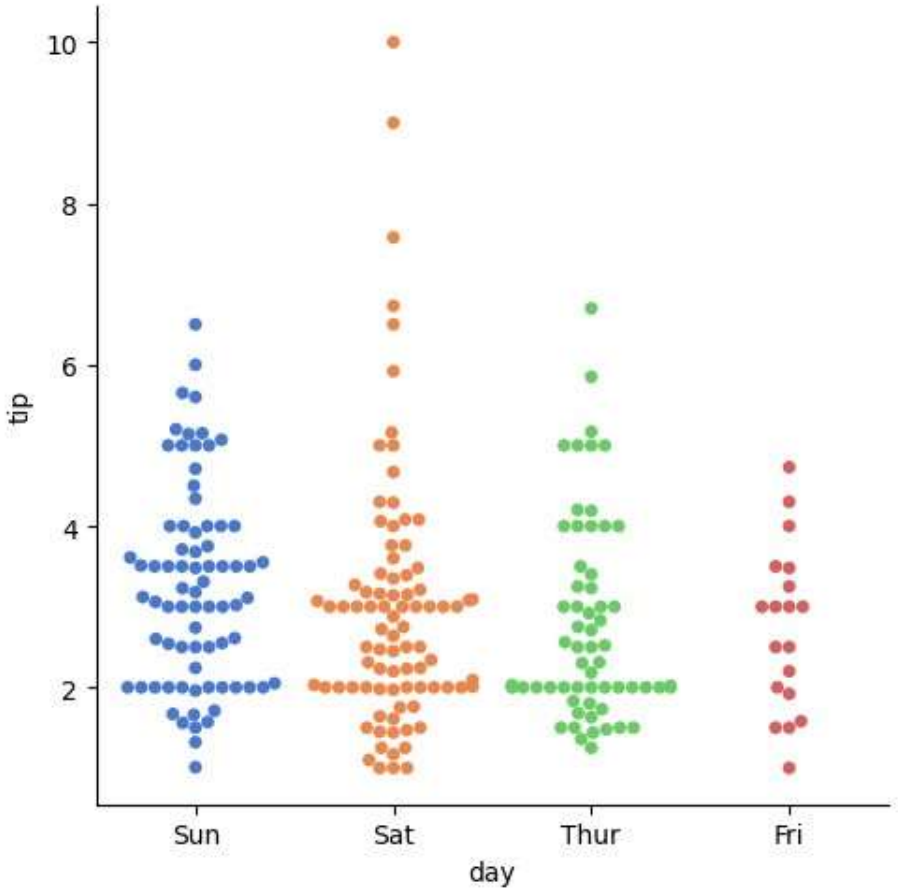

[131]:
```
sns.lmplot(x='total_bill',y='tip',data=tip,col='time',hue='smoker')
```

[131]: <seaborn.axisgrid.FacetGrid at 0x7d67d8e3b610>



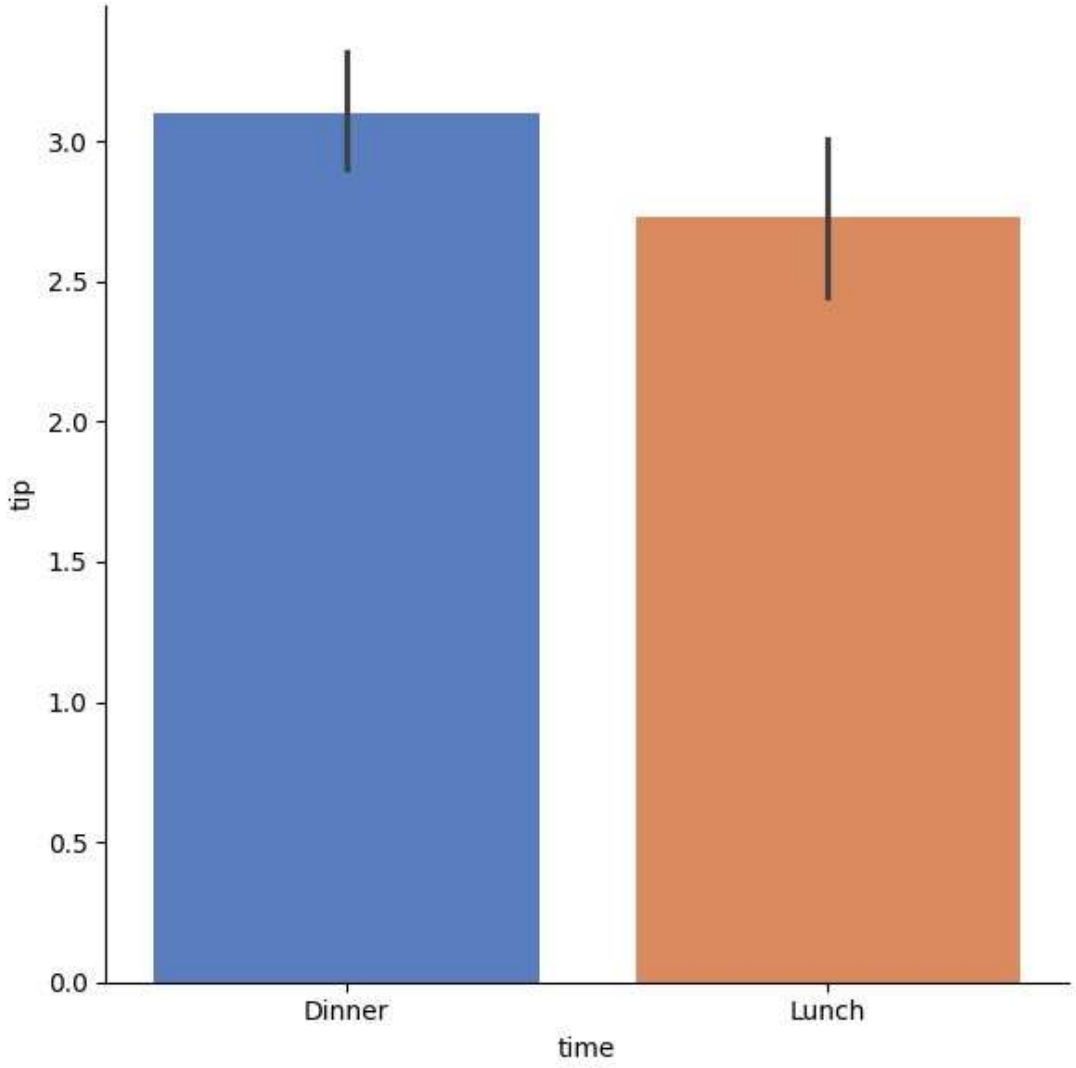
[132]:
```
sns.catplot(x='day',y='tip',data=tip,kind='swarm',palette='muted')
```

[132]: <seaborn.axisgrid.FacetGrid at 0x7d67d8dcbd60>



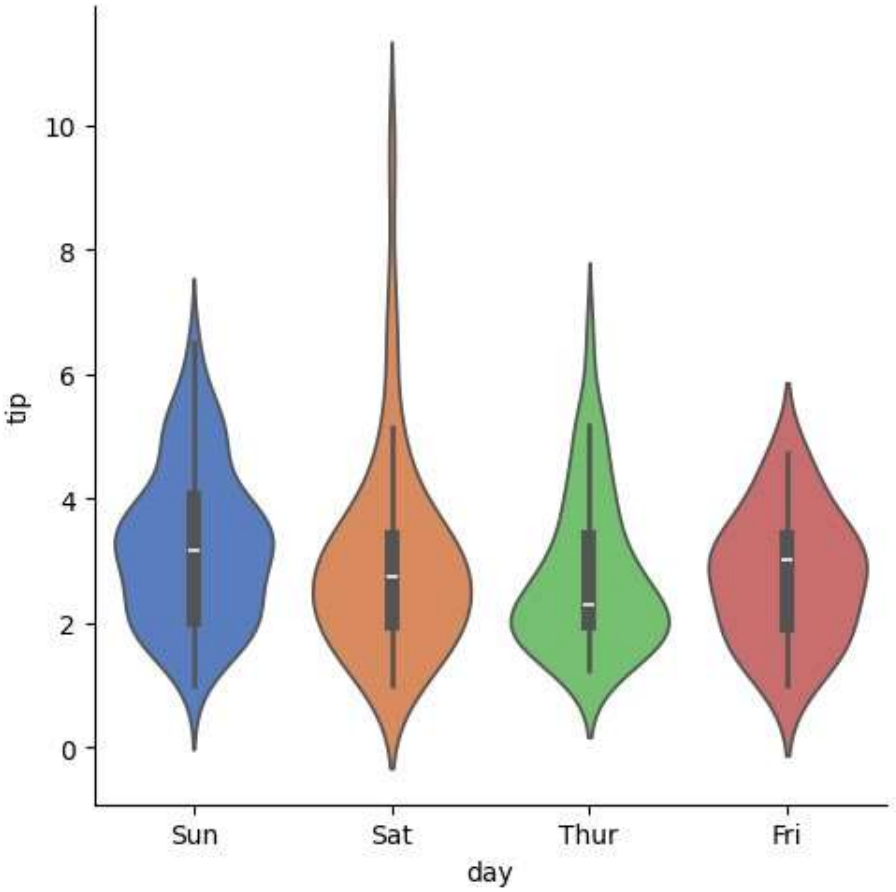

[133]:
```
sns.catplot(x='time',y='tip',data=tip,height=6,kind='bar',palette='muted')
```

[133]: <seaborn.axisgrid.FacetGrid at 0x7d67d8bd6f20>



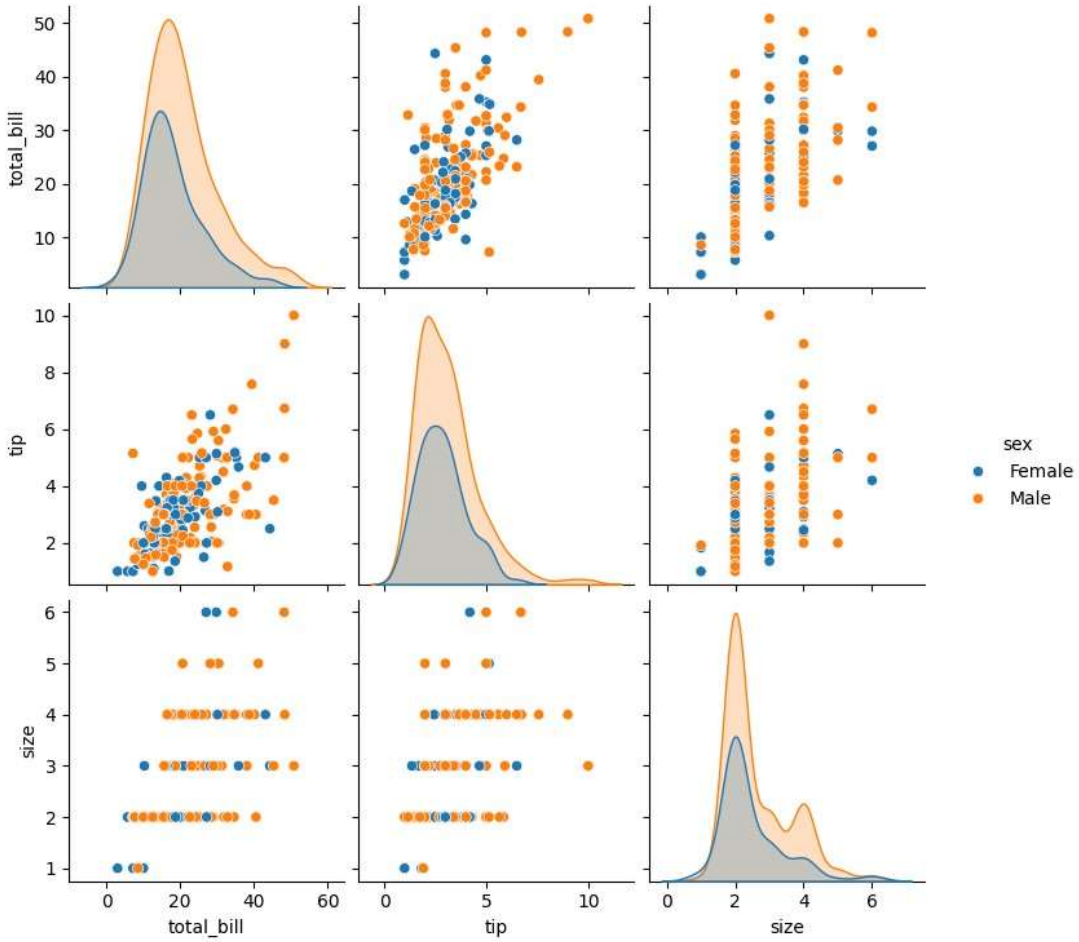

[134]:
```
sns.catplot(x='day',y='tip',data=tip,kind='violin',palette='muted'
)
```

[134]: <seaborn.axisgrid.FacetGrid at 0x7d67d8c72440>



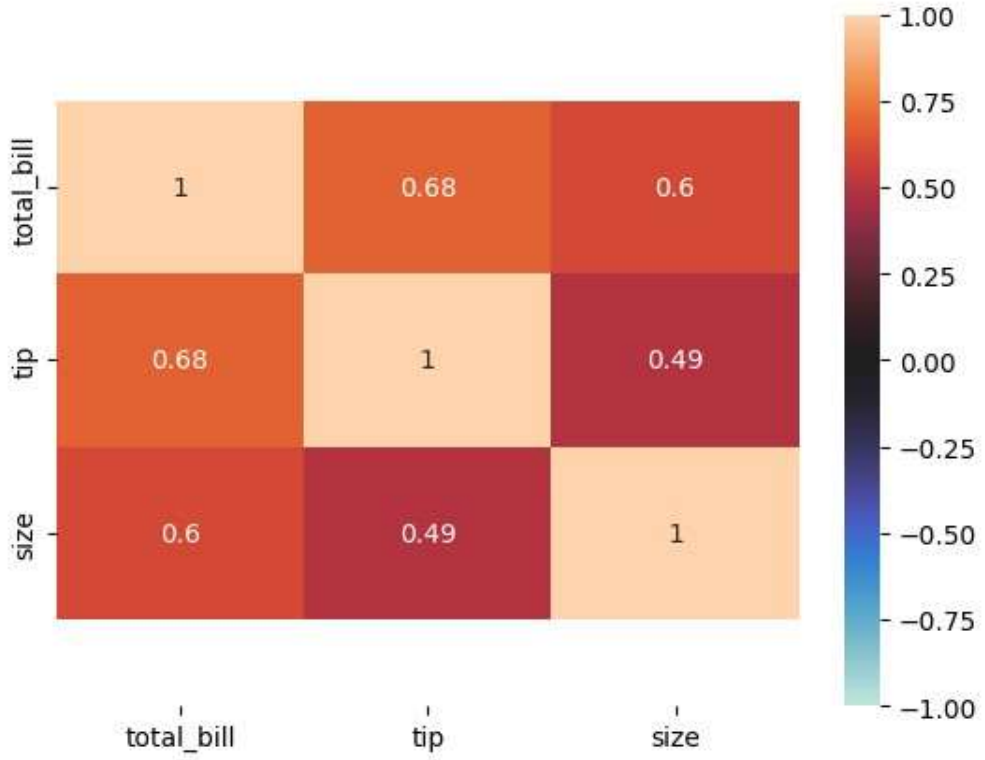

[135]:
```
sns.pairplot(data=tip,hue='sex')
```

[135]:
```
<seaborn.axisgrid.PairGrid at 0x7d67d8dcadd0>
```




[136]:
```
corr_matrix = tip.corr() ax =
sns.heatmap(data=corr_matrix,annot=True,vmax=1,vmin=-
1,center=0) bottom,top = ax.get_ylim()
ax.set_ylim(bottom+0.5,top-0.5)
```

[136]: (3.5, -0.5)




[137]:
```
LE = LabelEncoder()
tip['sex'] = LE.fit_transform(tip['sex'])
tip['smoker'] = LE.fit_transform(tip['smoker'])
tip['day'] = LE.fit_transform(tip['day'])
tip['time'] = LE.fit_transform(tip['time'])
tip.head()
```

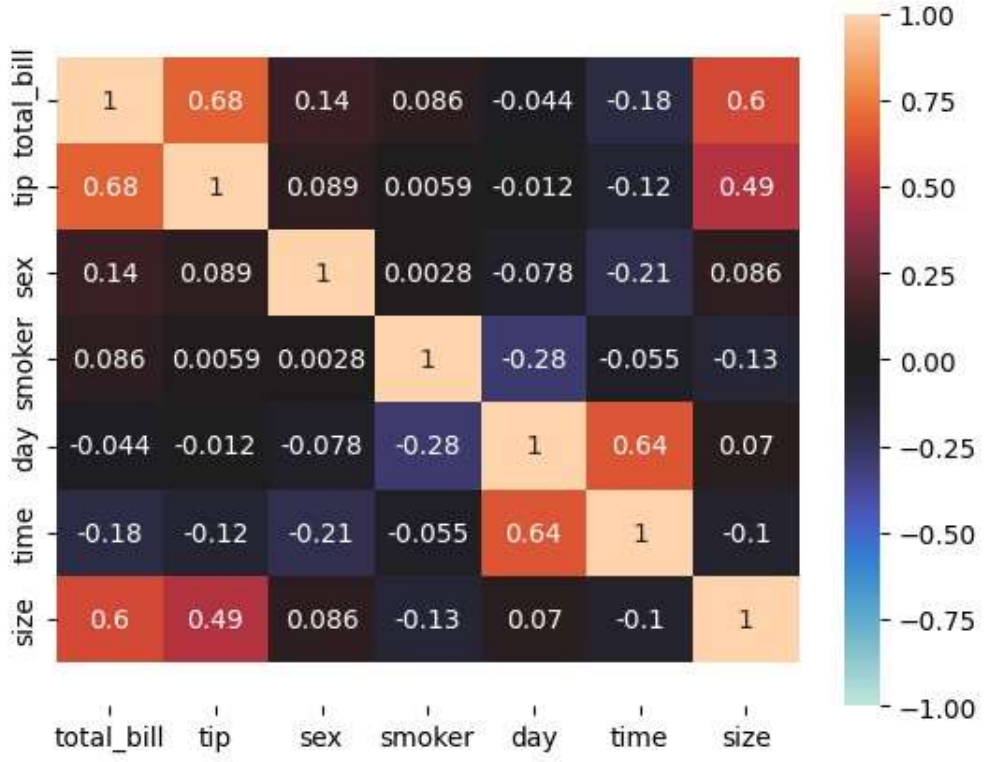
[137]:

| | total_bill | tip | sex | smoker | day | time | size |
|---|---|---|---|---|---|---|---|
| 0 | 16.99 | 1.01 | 0 | 0 | 2 | 0 | 2 |
| 1 | 10.34 | 1.66 | 1 | 0 | 2 | 0 | 3 |
| 2 | 21.01 | 3.50 | 1 | 0 | 2 | 0 | 3 |
| 3 | 23.68 | 3.31 | 1 | 0 | 2 | 0 | 2 |
| 4 | 24.59 | 3.61 | 0 | 0 | 2 | 0 | 4 |

[138]:
```
corr_matrix = tip.corr() ax =
sns.heatmap(data=corr_matrix,annot=True,vmax=1,vmin=-
```

```
    1,center=0) bottom,top = ax.get_ylim()
    ax.set_ylim(bottom+0.5,top-0.5)
```

```
[138]: (7.5, -0.5)
```




**5.Conclusion :-** Data visualization is done on the tips dataset of Seaborn using plots for different types of variables and inferences are made about the relationship between total bill, tip, day, time, gender, smoker or non-smoker etc.