# ads-exp4-1

February 6, 2025

Name : Ayush Panigrahi

Div : B

ID : 21102050

Sub : ADS

Batch : A2

[14]:
```
# Importing the seaborn library
import seaborn as sns
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from scipy import stats
%matplotlib inline
import warnings
warnings.simplefilter(action='ignore',category=FutureWarning)
```

[15]:
```
# Load the dataset
tips = sns.load_dataset('tips')
df = pd.DataFrame(tips)
df.head()
```

[15]:

| | 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 |

[16]:
```
df.isnull().sum()
```

[16]:
```
total_bill    0
tip           0
sex           0
smoker        0
day           0
time          0
```

```
size          0
dtype: int64
```

[17]:
```
df.describe()
```

[17]:
```
        total_bill         tip        size
count  244.000000  244.000000  244.000000
mean    19.785943    2.998279    2.569672
std      8.902412    1.383638    0.951100
min      3.070000    1.000000    1.000000
25%     13.347500    2.000000    2.000000
50%     17.795000    2.900000    2.000000
75%     24.127500    3.562500    3.000000
max     50.810000   10.000000    6.000000
```

[20]:
```
df.tip.describe()
```

[20]:
```
count    244.000000
mean       2.998279
std        1.383638
min        1.000000
25%        2.000000
50%        2.900000
75%        3.562500
max       10.000000
Name: tip, dtype: float64
```

[21]:
```
bill=df.total_bill

print ("Maximum Bill=", np.max (bill))
print("Minimum Bill= ",np.min (bill))
print ("Standard Deviation=",np.std(bill))
print("Median=", np.median (bill))
print ("Mean=",np.mean (bill))
```

```
Maximum Bill= 50.81
Minimum Bill=  3.07
Standard Deviation= 8.88415057777113
Median= 17.795
Mean= 19.78594262295082
```

[23]:
```
tip = df.tip
print("Maximum Tip =",np.max(tip))
print("Minimum Tip=",np.min(tip))
print ("Standard Deviation Tip =",np.std(tip))
print("Median Tip=" , np.median(tip))
print ("Mean Tip=", np.mean(tip))
```

```
print("Q1 =", np.quantile(tip,0.25))
print("Q2 =", np.quantile(tip,0.25))
print("Q3 =", np.quantile(tip,0.25))
```

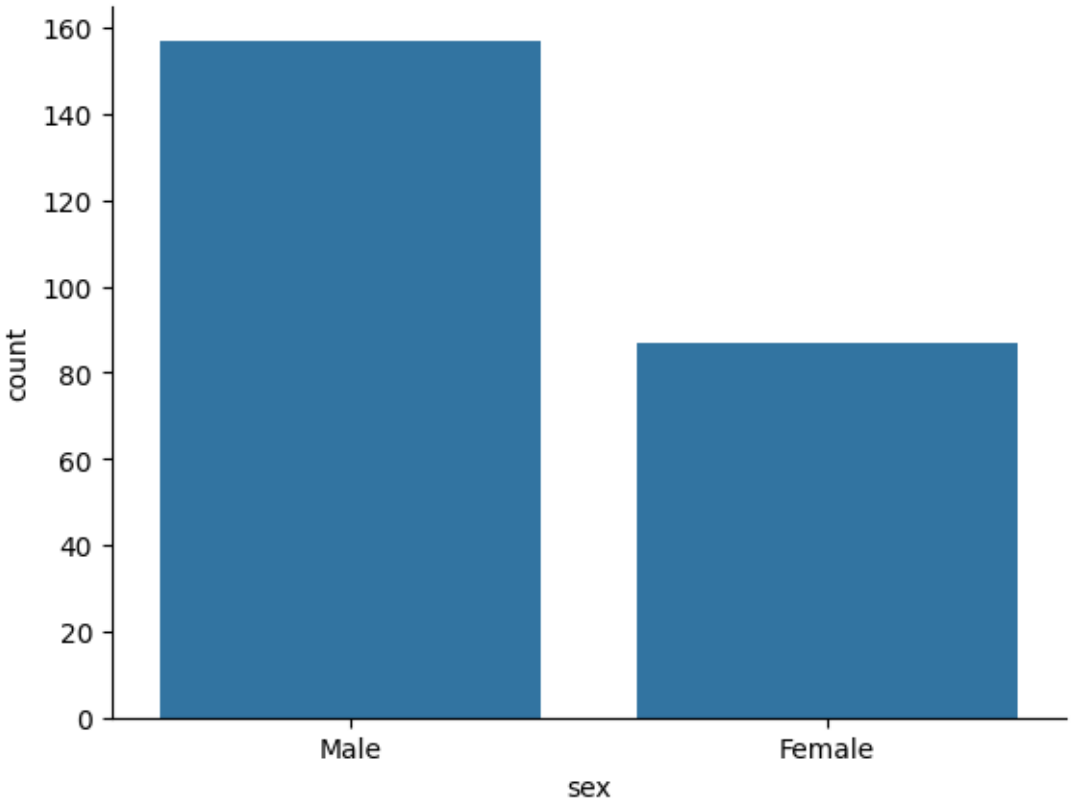
```
Maximum Tip = 10.0
Minimum Tip= 1.0
Standard Deviation Tip = 1.3807999538298958
Median Tip= 2.9
Mean Tip= 2.99827868852459
Q1 = 2.0
Q2 = 2.0
Q3 = 2.0
```

Exploratory data analysis

To explore if there is any dependency between the variable “Tip” and rest of the variables

[ ]:
```
sns.countplot(x='sex', data=tips)
sns.despine()#no top and right axes spine

print(tips.sex.value_counts())
```

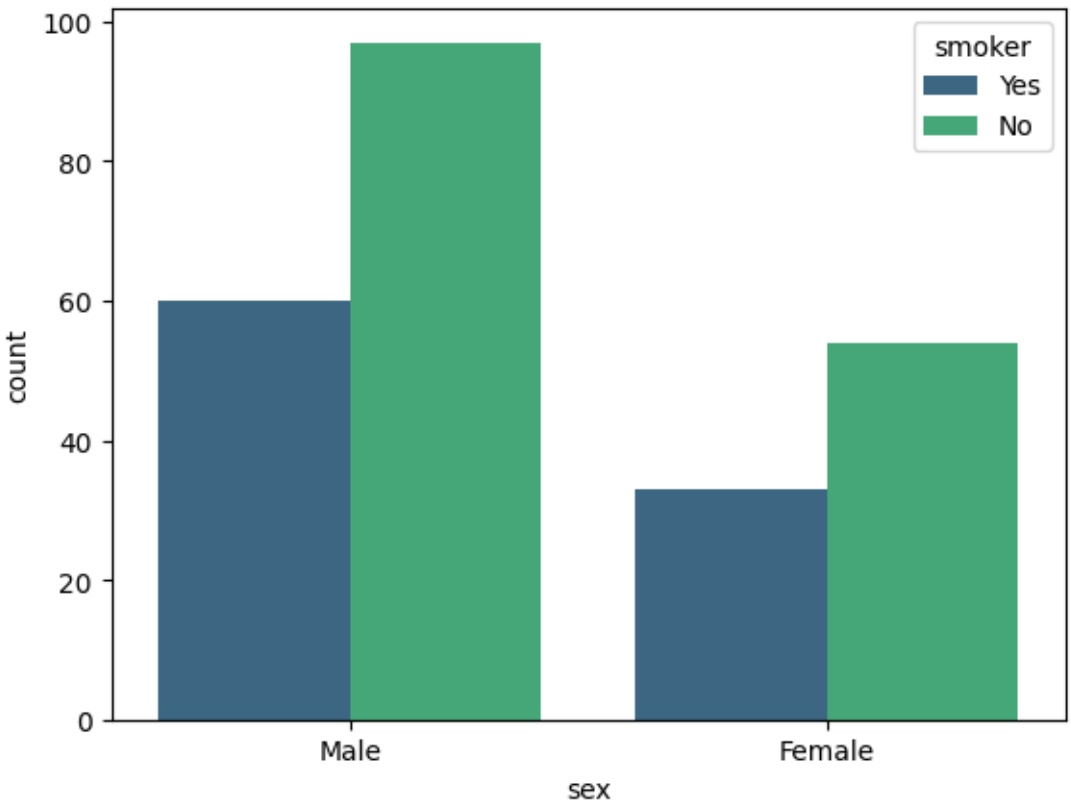
```
sex
Male      157
Female     87
Name: count, dtype: int64
```




Male customers have given more tip than female customer

[ ]:
```
sns.countplot(x='sex', data=tips, hue='smoker', palette='viridis')
```

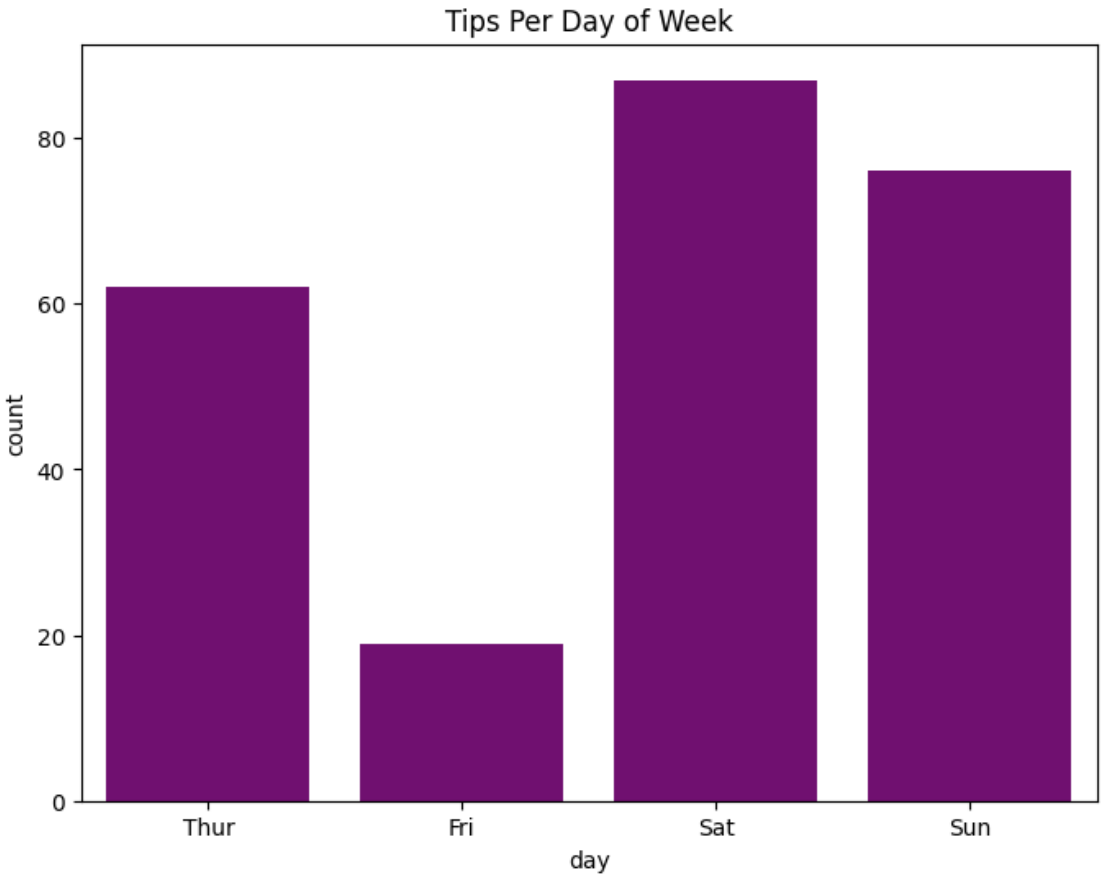
[ ]:
```
<Axes: xlabel='sex', ylabel='count'>
```




non smoker gives more tips as compare to smoker

[ ]:
```
plt.figure(figsize=(8,6))
plt.title("Tips Per Day of Week")
sns.countplot(x=tips['day'],color='purple')
```

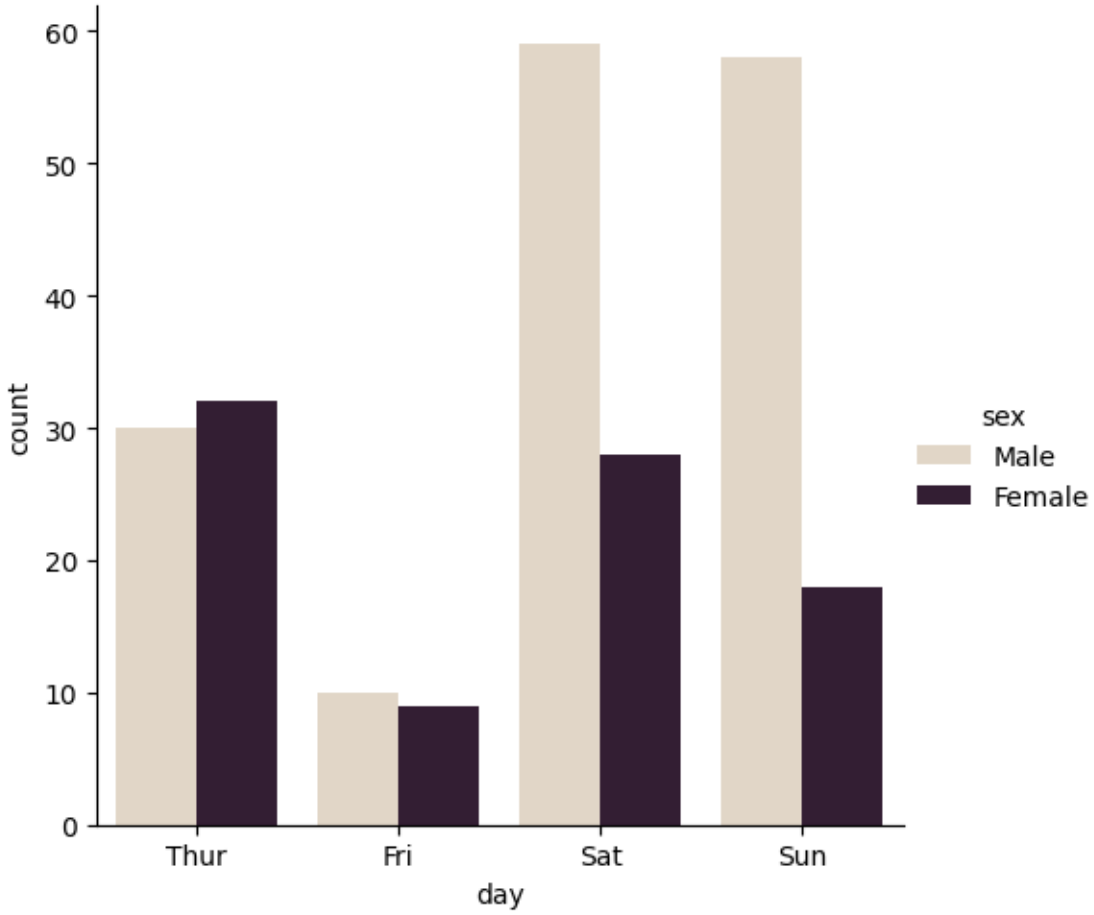
[ ]:
```
<Axes: title={'center': 'Tips Per Day of Week'}, xlabel='day', ylabel='count'>
```




Higher amount of tips are given on weekends

[ ]:
```
sns.catplot(x='day', data=tips, hue='sex', palette='ch:.25', kind='count')
```

[ ]:
```
<seaborn.axisgrid.FacetGrid at 0x7cbb5922d110>
```

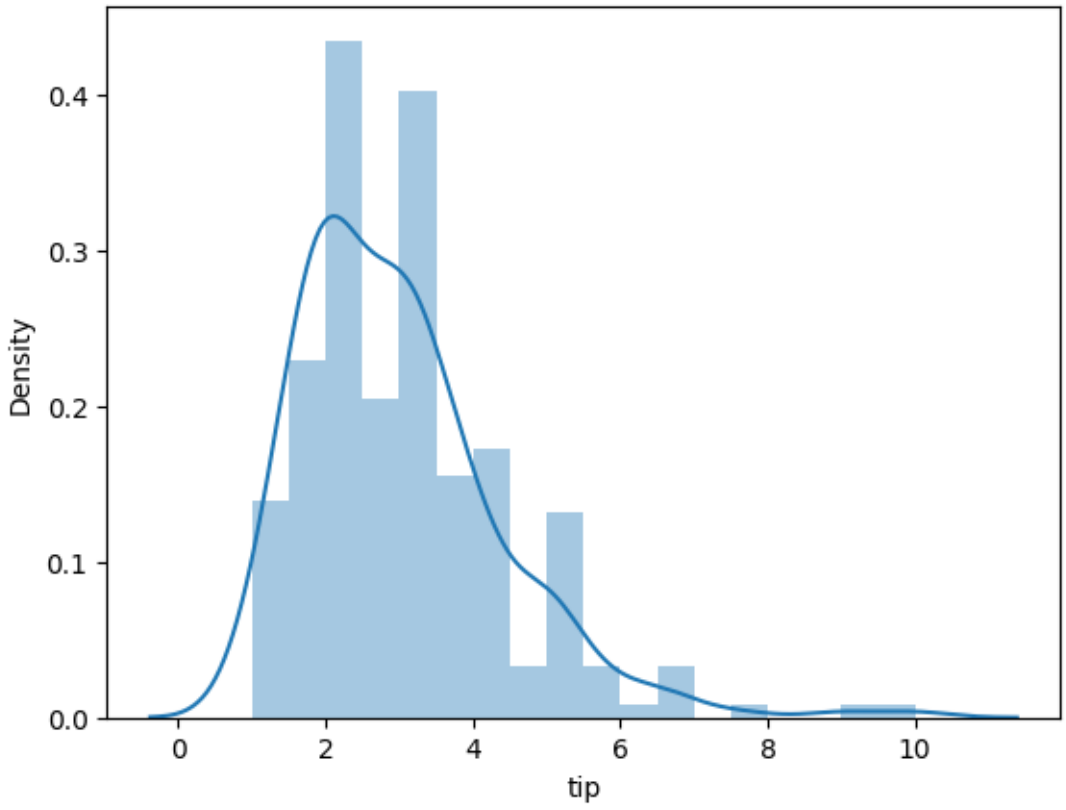



Over the weekends male customer are giving more tips

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

```
<ipython-input-22-3f4491b3d128>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df['tip'])
```

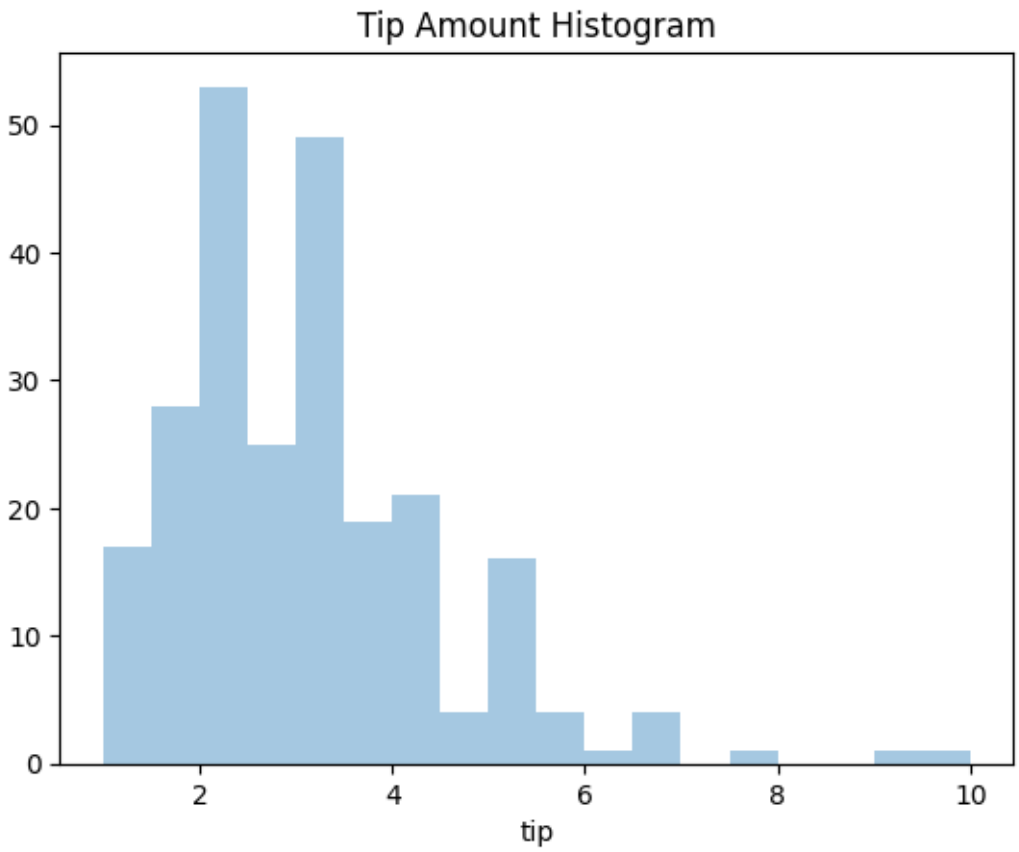
[ ]:
```
<Axes: xlabel='tip', ylabel='Density'>
```




The distribution is a right skewed distribution with Outliers between 6$ to 10$

[ ]:
```
g = sns.distplot (tips.tip,kde=False)
g.set_title('Tip Amount Histogram');
```

```
<ipython-input-24-e5706eb2b7da>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  g = sns.distplot (tips.tip,kde=False)
```

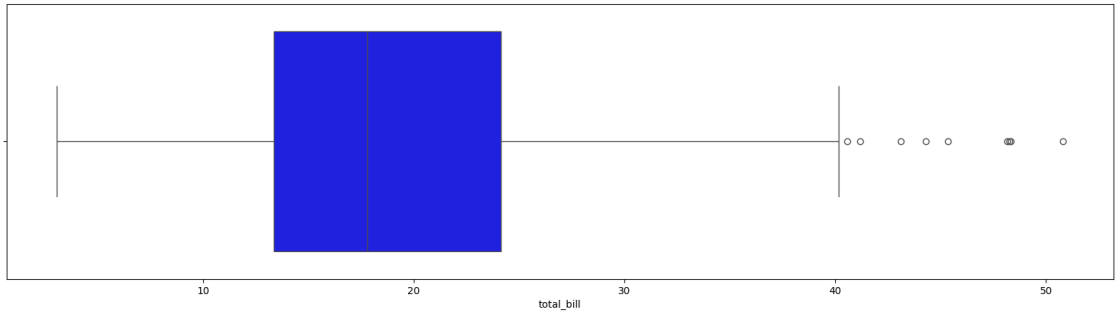



The distribution is a right skewed distribution with Outliers between 6 to 10 dollars without kde(kernal density)

Find the outliers for bill and tip

[ ]:
```
plt.figure(figsize=(20,5))
sns.boxplot(x=bill, color='b')
```

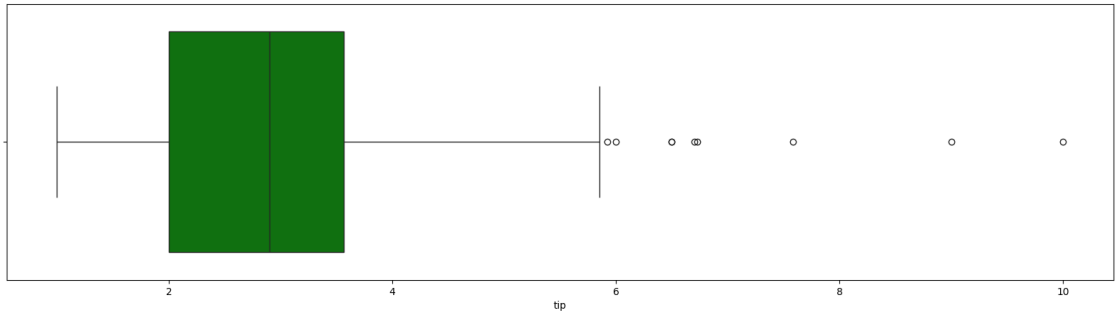
[ ]:
```
<Axes: xlabel='total_bill'>
```



Majority of customers have bill between 13 to 25 and outliers are after 40 $

[ ]:
```
plt.figure(figsize=(20,5))
sns.boxplot (x=tip, color='g')
```

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



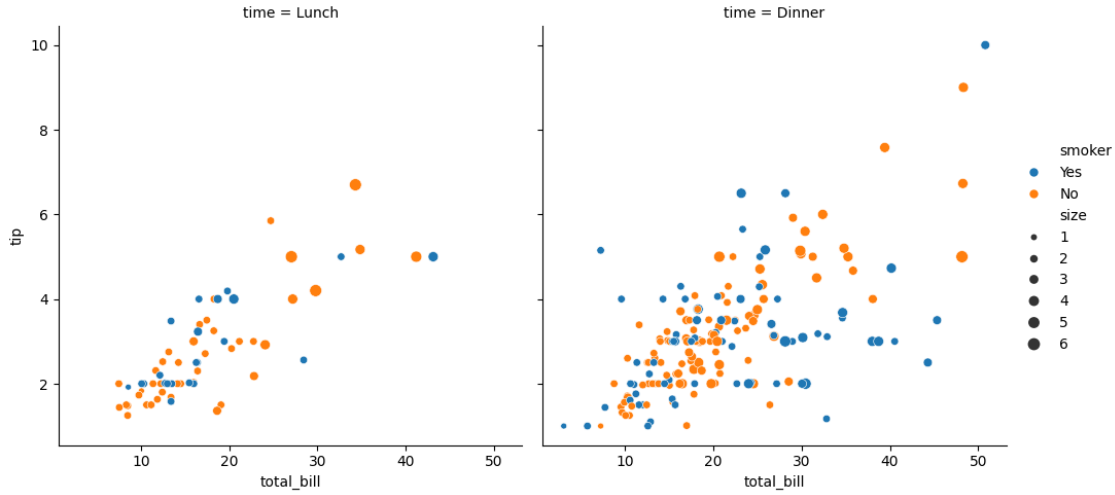

majority of the customer has tip between 2 to 3.8 dollars and tip after 6 $ are outliers

[ ]:
```
bill_tip = pd.DataFrame (df,columns=['total_bill', 'tip', 'size'])
print (bill_tip)

print("IQR For Total Bill: ",stats.iqr(bill))
print("IQR For Tip: ", stats.iqr(tip))
```

```
     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]
IQR For Total Bill:  10.779999999999998
IQR For Tip:  1.5625
```

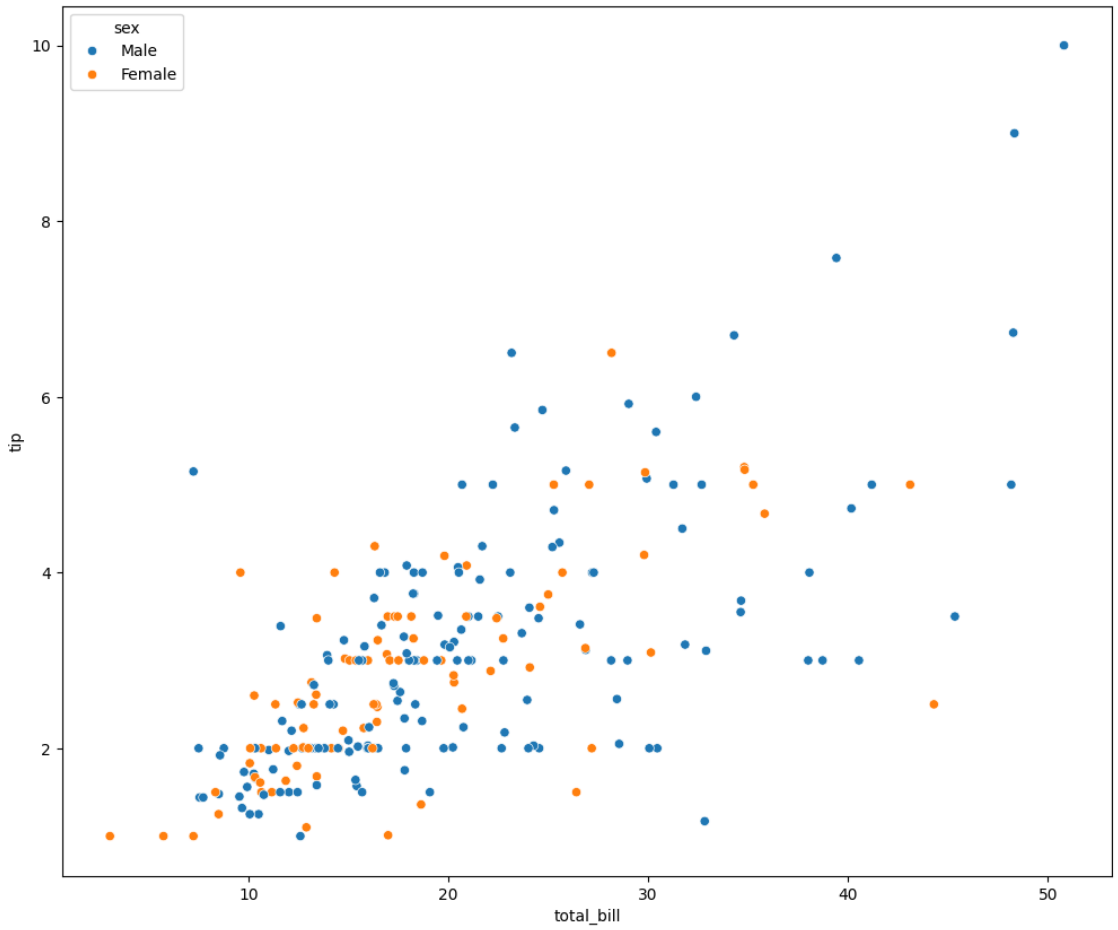
[ ]:
```
sns.relplot(x='total_bill',y='tip',data=df,col='time', hue␣
↪='smoker',size='size')
```

[ ]:
```
<seaborn.axisgrid.FacetGrid at 0x7cbb58de2890>
```




plot based on lunch and dinner based on whether a person is a smoker or not.can see a linear pattern i.e as total bill increases tip also increases. replot and implot are used for visualizing linear relationship. more no.of tips are given on dinner.Bubble size shows the group of people, bigger the size bigger the group.

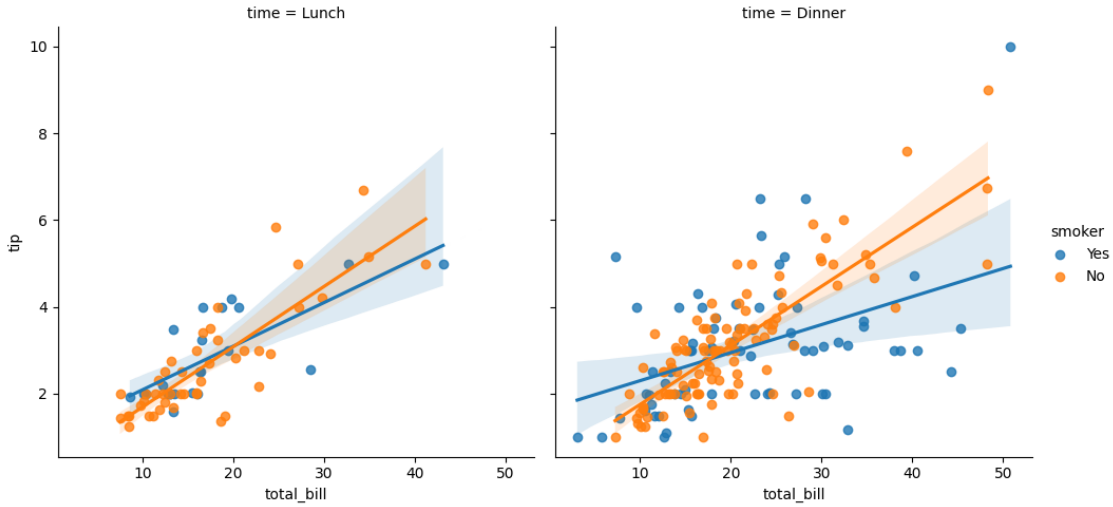
[ ]:
```
plt.figure(figsize=(12,10))
sns.scatterplot (data=df,x="total_bill",y="tip", hue="sex");
```




we can see a linear pattern

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

[ ]:
```
<seaborn.axisgrid.FacetGrid at 0x7cbb54d59150>
```

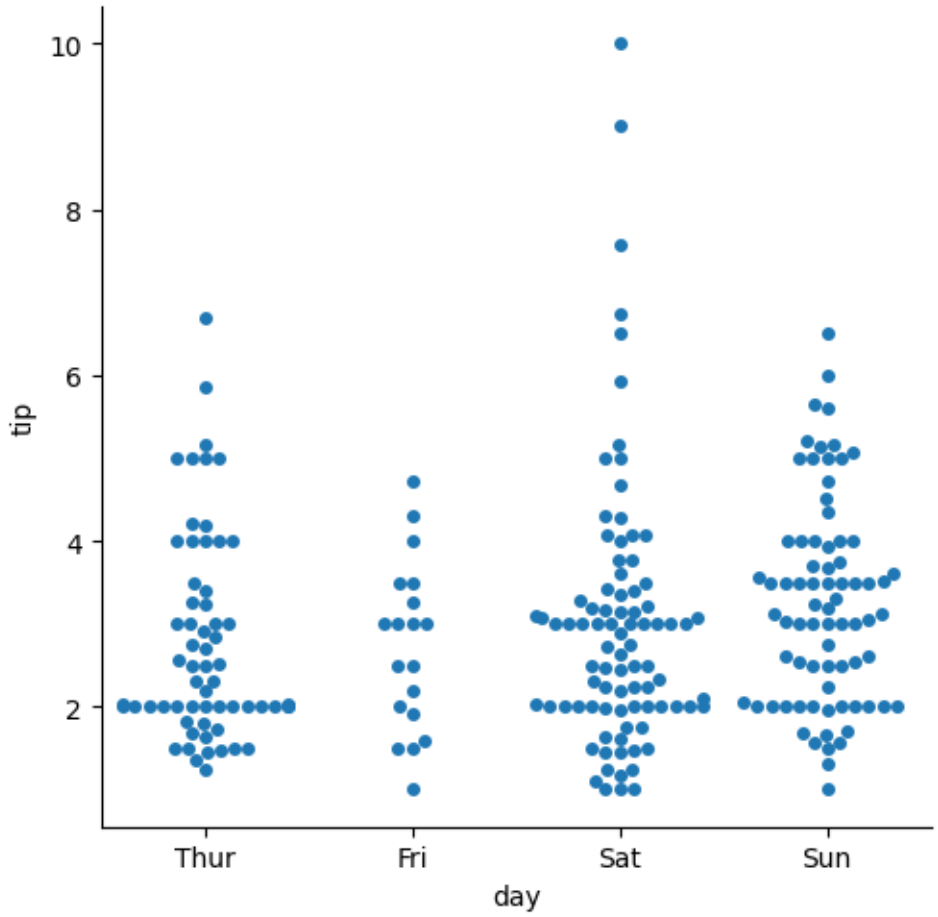



If bill amount is less, smokers give more tip than non smokers but they give less tips if the bill amount is high. So it is better to serve a non smoker if the bill amount is high.

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

```
/usr/local/lib/python3.11/dist-packages/seaborn/categorical.py:3399:
UserWarning: 8.1% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
```

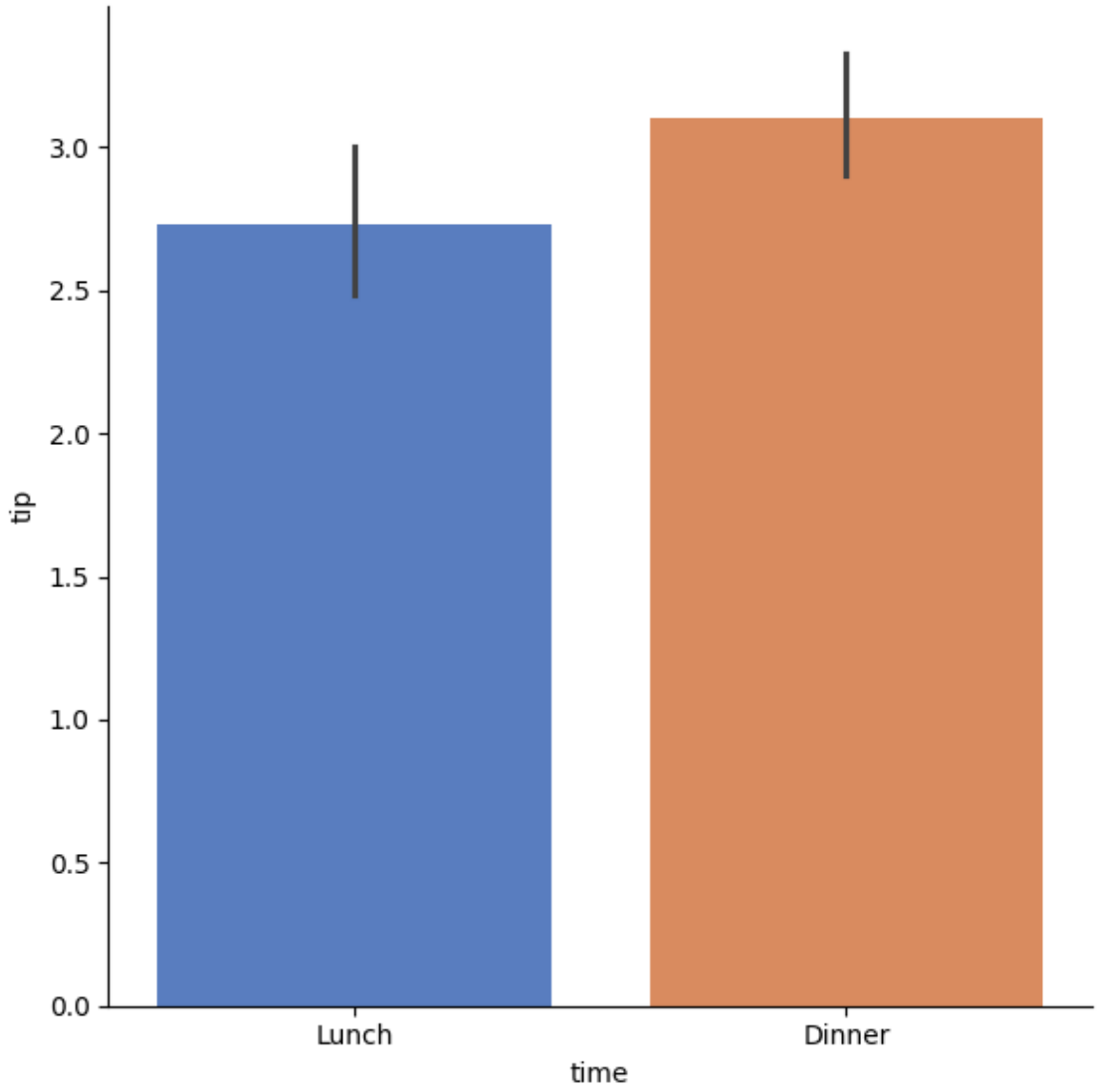
[ ]:
```
<seaborn.axisgrid.FacetGrid at 0x7cbb56ca2410>
```




More no of tip is given on saturday and 2 $ is given the most as a tip

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

[ ]:
```
<seaborn.axisgrid.FacetGrid at 0x7cbb5b9db6d0>
```

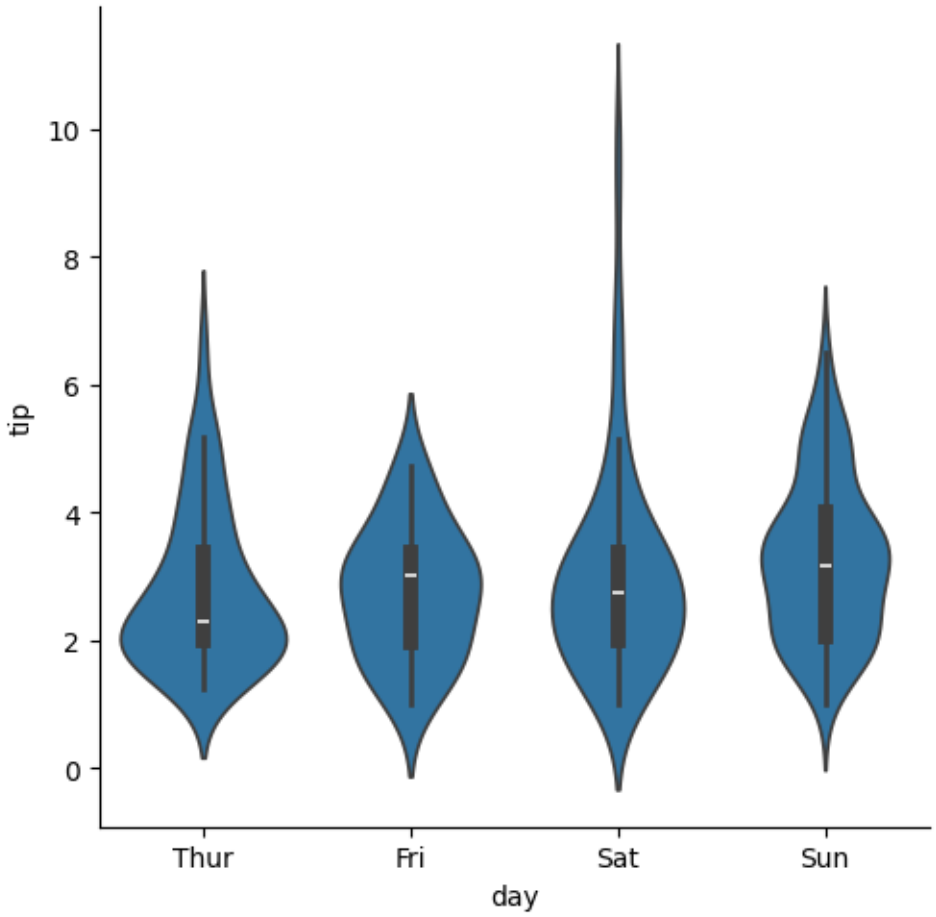



Conclusion

max tips are given on dinner

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

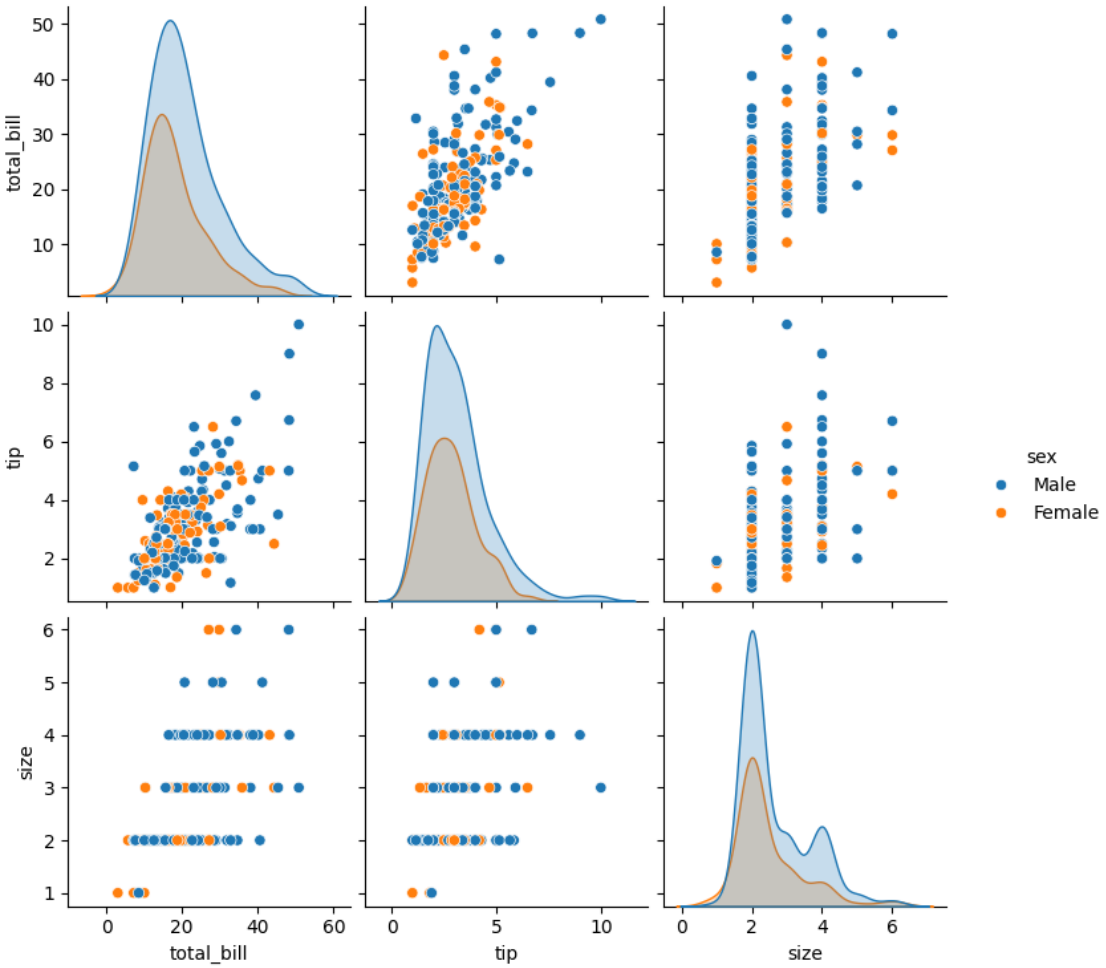
[ ]:
```
<seaborn.axisgrid.FacetGrid at 0x7cbb56c0dad0>
```




More no of tip is given on saturday and 2 $ is given the most as a tip

[ ]:
```
sns.pairplot(df, hue='sex')
```

[ ]:
```
<seaborn.axisgrid.PairGrid at 0x7cbb58daf3d0>
```

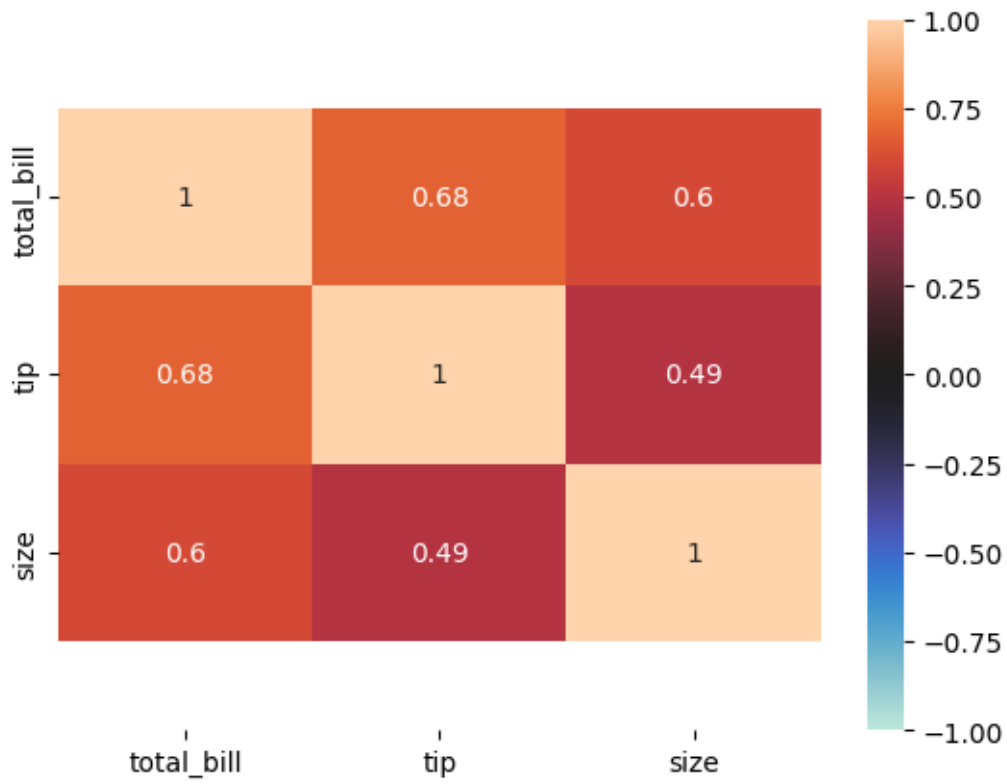



Conclusion

The pairplot shows how the variable in the dataset relate to each other. for example it shows that higher bill tends to get higher tips. It also helps to see that there are diffrence between how men and women tip

[ ]:
```
corr_matrix=df.corr(numeric_only=True)
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)
```

[ ]:
```
(3.5, -0.5)
```




Corelation Matrix

Converting categorical variables into numerical values so that the machine learning model can understand

the most corelation is between total bill and tip

[6]:
```
from IPython import get_ipython
from IPython.display import display
# %%
# Importing the seaborn library
import seaborn as sns
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from scipy import stats
%matplotlib inline
import warnings
warnings.simplefilter(action='ignore',category=FutureWarning)
# %%
# Load the dataset
tips = sns.load_dataset('tips')
df = pd.DataFrame(tips) # Define df here
```

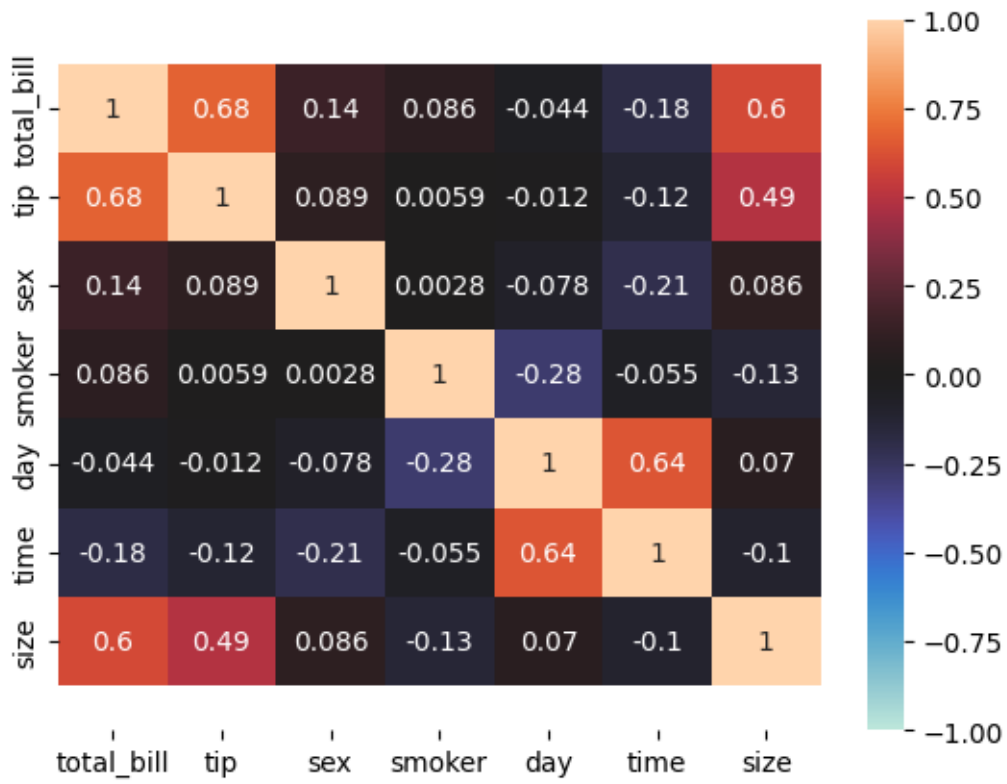
```
df.head()
# %%
# ... rest of your code ...

from sklearn.preprocessing import LabelEncoder
labelencoder_df=LabelEncoder()
df['sex']=labelencoder_df.fit_transform(df['sex'])
df['smoker']=labelencoder_df.fit_transform(df['smoker'])
df['day']=labelencoder_df.fit_transform(df['day'])
df['time']=labelencoder_df.fit_transform(df['time'])
df.head()
```

[6]:

| | 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 |

[7]:
```
corr_matrix = df.corr(numeric_only=True)
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)
```

[7]: (7.5, -0.5)




Conclusion Similar correlations as the rst heatmap, with an additional moderate positive correlation between tip and size, indicating larger groups also give higher tips. Overall, total_bill is a key factor inuencing tip amount, with group size also playing a role.