

# Data Visualization

## Importing Libraries

```
In [1]: import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

## stop warnings

```
In [2]: import warnings
warnings.filterwarnings("ignore")
```

## Load Dataset

```
In [3]: df = pd.read_csv("tips.csv")
df.head(3)
```

Out[3]:

	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

# Data Understanding

```
In [4]: df.shape
```

Out[4]: (244, 7)

```
In [5]: df.columns
```

Out[5]: Index(['total\_bill', 'tip', 'sex', 'smoker', 'day', 'time', 'size'], dtype='object')

In [6]: df.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 244 entries, 0 to 243
Data columns (total 7 columns):
#   Column      Non-Null Count  Dtype
---  -
0   total_bill  244 non-null    float64
1   tip         244 non-null    float64
2   sex         244 non-null    object
3   smoker      244 non-null    object
4   day         244 non-null    object
5   time       244 non-null    object
6   size        244 non-null    int64
dtypes: float64(2), int64(1), object(4)
memory usage: 13.5+ KB
```

In [7]: df.describe()

Out[7]:

	total_bill	tip	size
<b>count</b>	244.000000	244.000000	244.000000
<b>mean</b>	19.785943	2.998279	2.569672
<b>std</b>	8.902412	1.383638	0.951100
<b>min</b>	3.070000	1.000000	1.000000
<b>25%</b>	13.347500	2.000000	2.000000
<b>50%</b>	17.795000	2.900000	2.000000
<b>75%</b>	24.127500	3.562500	3.000000
<b>max</b>	50.810000	10.000000	6.000000

```
In [8]: df.describe(include="all")
```

```
Out[8]:
```

	total_bill	tip	sex	smoker	day	time	size
<b>count</b>	244.000000	244.000000	244	244	244	244	244.000000
<b>unique</b>	NaN	NaN	2	2	4	2	NaN
<b>top</b>	NaN	NaN	Male	No	Sat	Dinner	NaN
<b>freq</b>	NaN	NaN	157	151	87	176	NaN
<b>mean</b>	19.785943	2.998279	NaN	NaN	NaN	NaN	2.569672
<b>std</b>	8.902412	1.383638	NaN	NaN	NaN	NaN	0.951100
<b>min</b>	3.070000	1.000000	NaN	NaN	NaN	NaN	1.000000
<b>25%</b>	13.347500	2.000000	NaN	NaN	NaN	NaN	2.000000
<b>50%</b>	17.795000	2.900000	NaN	NaN	NaN	NaN	2.000000
<b>75%</b>	24.127500	3.562500	NaN	NaN	NaN	NaN	3.000000
<b>max</b>	50.810000	10.000000	NaN	NaN	NaN	NaN	6.000000

```
In [9]: df.dtypes
```

```
Out[9]: total_bill    float64
tip                float64
sex                object
smoker             object
day                object
time               object
size               int64
dtype: object
```

```
In [10]: df.isnull().sum()
```

```
Out[10]: total_bill    0
tip                0
sex                0
smoker             0
day                0
time               0
size               0
dtype: int64
```

```
In [11]: df["sex"].value_counts()
```

```
Out[11]: Male        157
Female         87
Name: sex, dtype: int64
```

# EDA

# plots

## Univariant Analysis

- Histogram
- Box Plot

## Bivariant Analysis

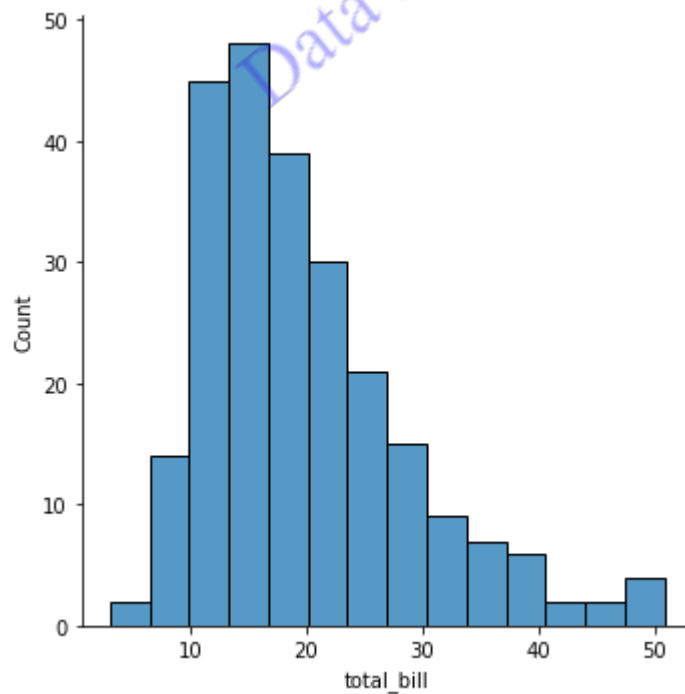
- Scatter Plot
- Line Plot
- Joint Plot
- Violin Plot

## Multivariant Analysis

- Pair Plot
- HeatMap

Histogram

```
In [12]: sns.displot(df["total_bill"])  
plt.show()
```



```
In [13]: from matplotlib import colors  
from matplotlib.ticker import PercentFormatter
```

```
In [14]: x = df["total_bill"]

# Creating histogram
fig, axs = plt.subplots(1, 1,
figsize =(10, 7),
tight_layout = True)

# Remove axes splines
for s in ['top', 'bottom', 'left', 'right']:
    axs.spines[s].set_visible(False)

# Remove x, y ticks
axs.xaxis.set_ticks_position('none')
axs.yaxis.set_ticks_position('none')

# Add padding between axes and labels
axs.xaxis.set_tick_params(pad = 5)
axs.yaxis.set_tick_params(pad = 10)

# Add x, y gridlines
axs.grid(b = True, color = 'grey',
linestyle = '-.', linewidth = 0.5, alpha = 0.6)

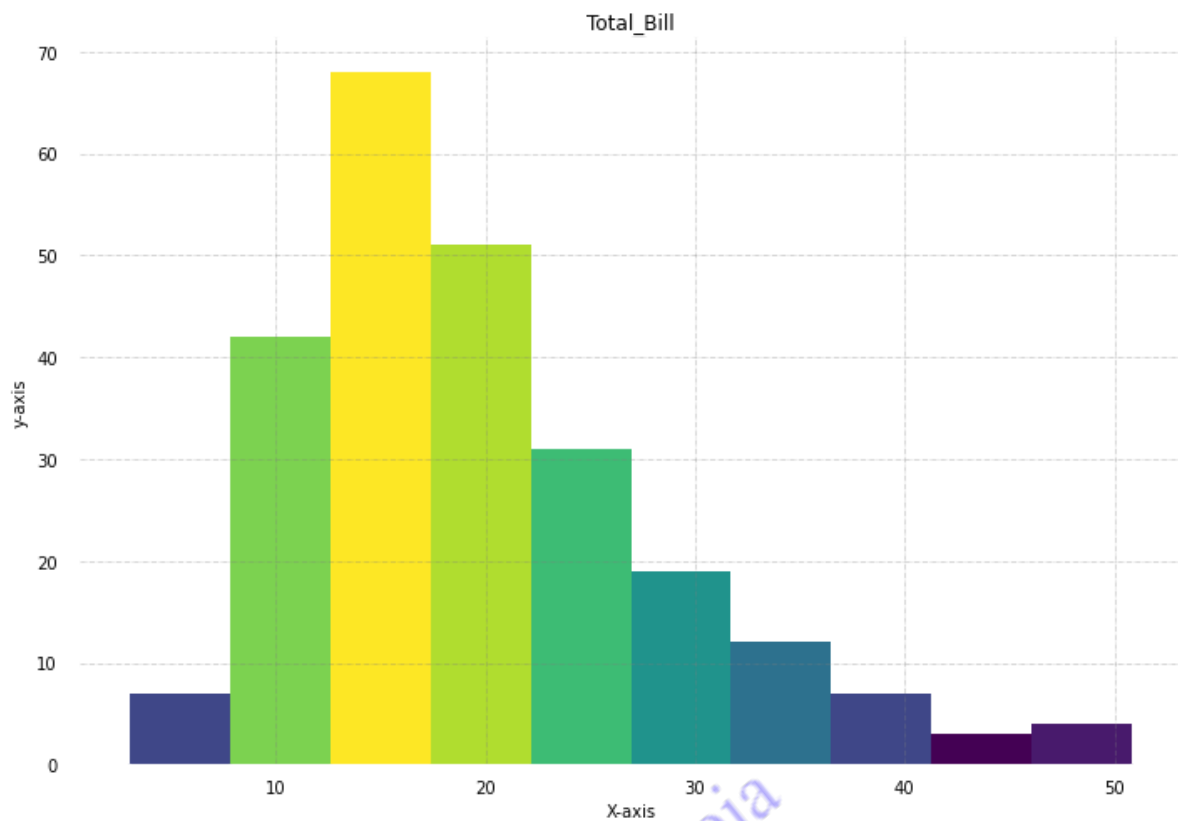
# Creating histogram
N, bins, patches = axs.hist(x)

# Setting color
fracs = ((N*(1 / 5)) / N.max())
norm = colors.Normalize(fracs.min(), fracs.max())

for thisfrac, thispatch in zip(fracs, patches):
    color = plt.cm.viridis(norm(thisfrac))
    thispatch.set_facecolor(color)

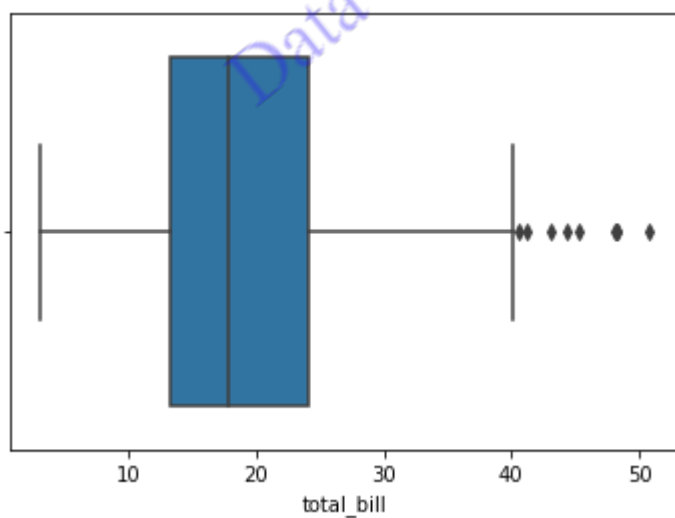
# Adding extra features
plt.xlabel("X-axis")
plt.ylabel("y-axis")
plt.title('Total_Bill')

# Show plot
plt.show()
```

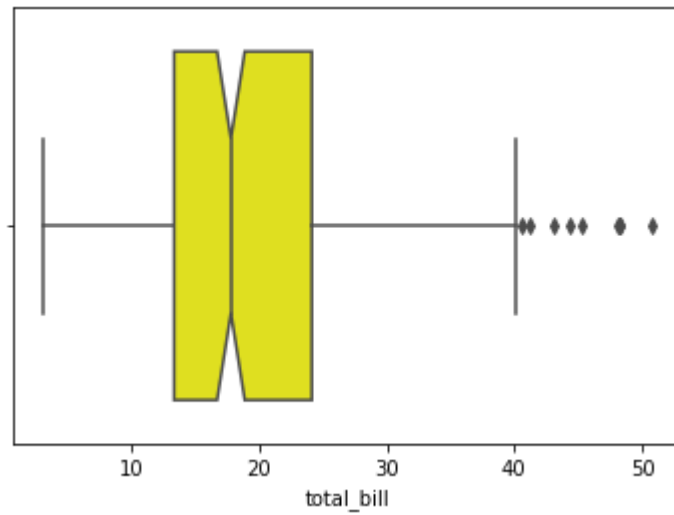


Box Plot

```
In [15]: sns.boxplot(x=df["total_bill"])  
plt.show()
```

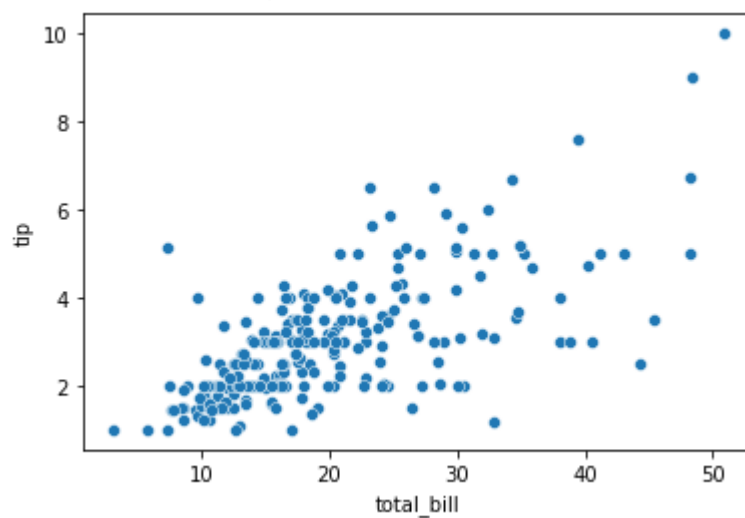


```
In [16]: sns.boxplot(x=df["total_bill"],color="yellow",notch=True)  
plt.show()
```

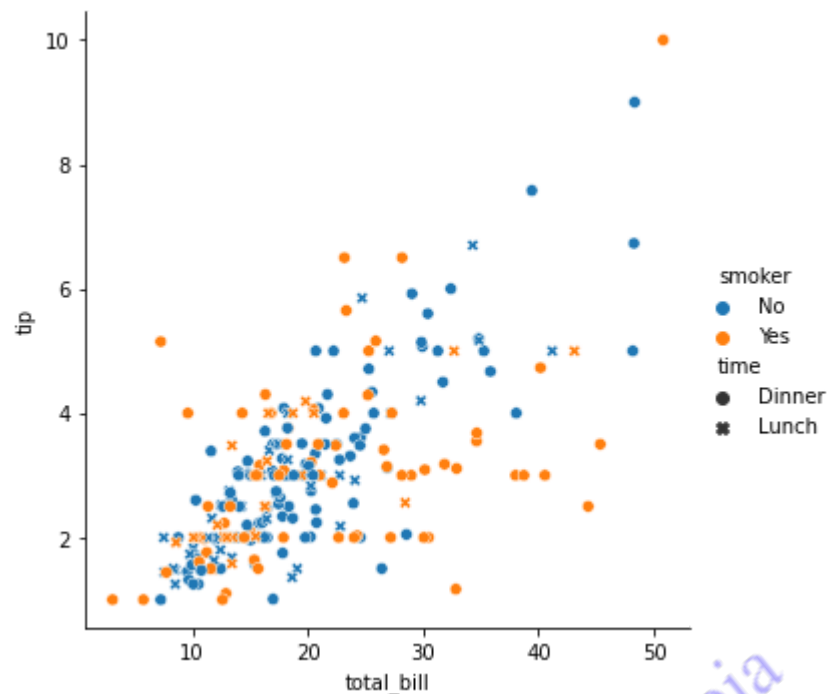


*Scatter Plot*

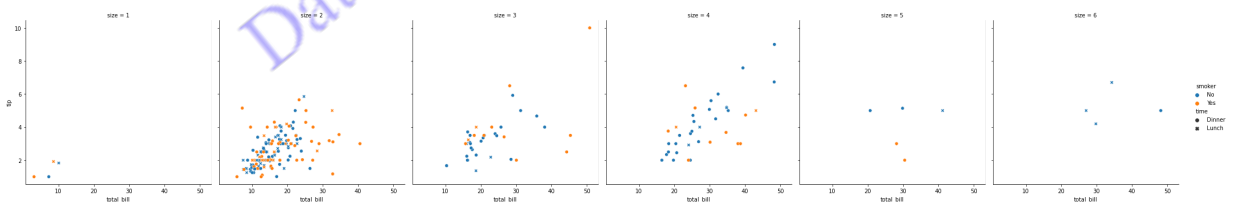
```
In [17]: sns.scatterplot(x=df["total_bill"],y=df["tip"])  
plt.show()
```



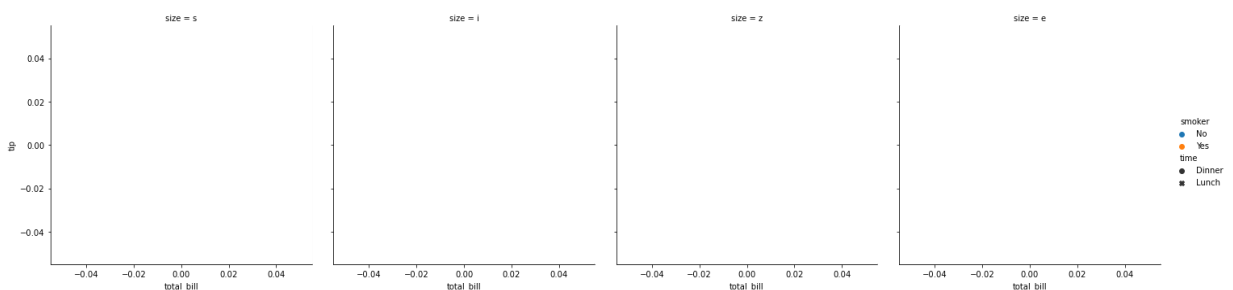
```
In [18]: sns.relplot(x='total_bill',y='tip',data=df,hue='smoker',style='time')
plt.show()
```



```
In [19]: sns.relplot(x='total_bill',y='tip',data=df,hue='smoker',style='time',col='size')
plt.show()
```



```
In [20]: sns.relplot(x='total_bill',y='tip',data=df,hue='smoker',style='time',col='size',
plt.show())
```



**Line Plot**



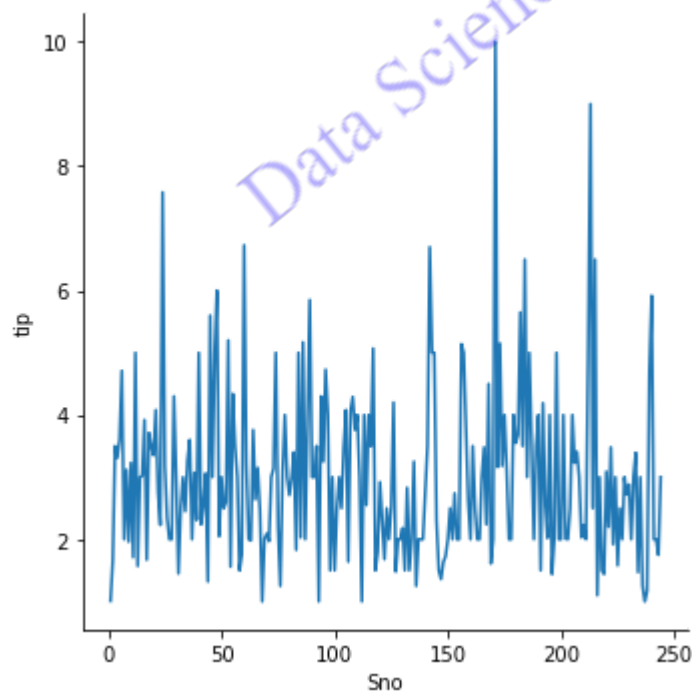
```
In [21]: df["Sno"] = pd.DataFrame(np.arange(1,245))
df
```

Out[21]:

	total_bill	tip	sex	smoker	day	time	size	Sno
0	16.99	1.01	Female	No	Sun	Dinner	2	1
1	10.34	1.66	Male	No	Sun	Dinner	3	2
2	21.01	3.50	Male	No	Sun	Dinner	3	3
3	23.68	3.31	Male	No	Sun	Dinner	2	4
4	24.59	3.61	Female	No	Sun	Dinner	4	5
...	...	...	...	...	...	...	...	...
239	29.03	5.92	Male	No	Sat	Dinner	3	240
240	27.18	2.00	Female	Yes	Sat	Dinner	2	241
241	22.67	2.00	Male	Yes	Sat	Dinner	2	242
242	17.82	1.75	Male	No	Sat	Dinner	2	243
243	18.78	3.00	Female	No	Thur	Dinner	2	244

244 rows × 8 columns

```
In [22]: sns.relplot(x='Sno',y='tip',kind='line',data=df)
plt.show()
```



```
In [23]: df.drop("Sno",axis=1,inplace=True)
```

In [24]: df

Out[24]:

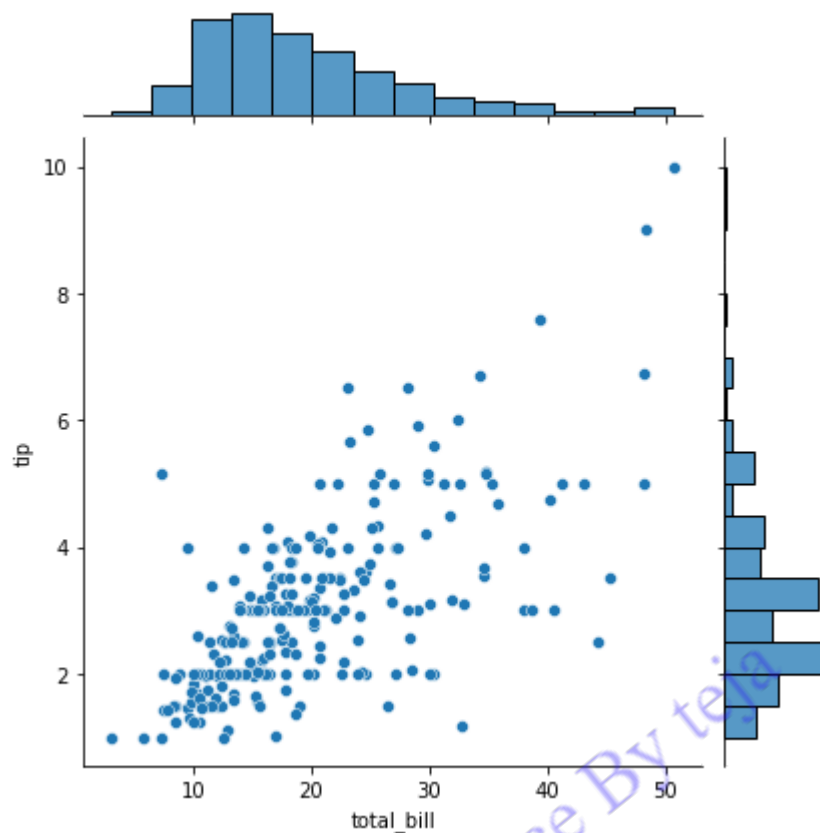
	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4
...	...	...	...	...	...	...	...
239	29.03	5.92	Male	No	Sat	Dinner	3
240	27.18	2.00	Female	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

*Join Plot*

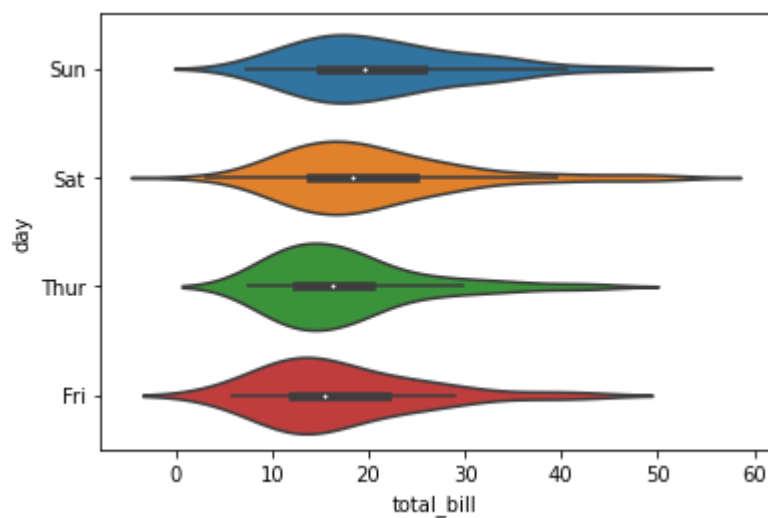
Data Science By teja

```
In [25]: sns.jointplot(y='tip',x='total_bill',data=df)
plt.show()
```

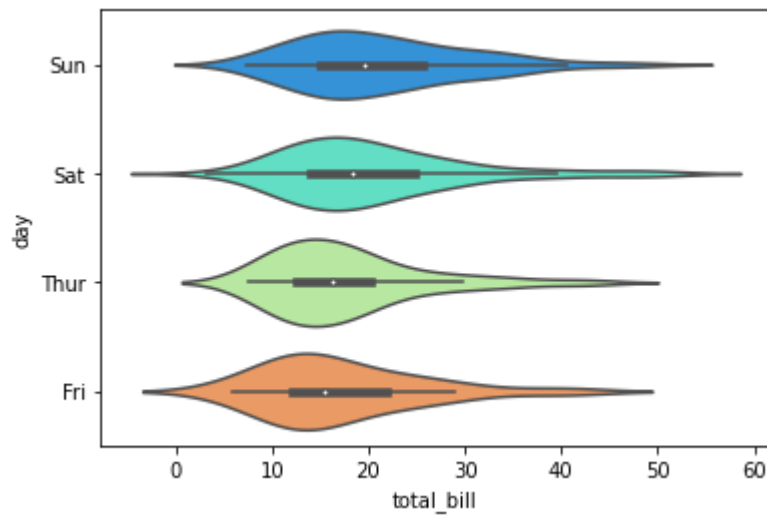


**Violin Plot**

```
In [26]: sns.violinplot(x='total_bill',y='day',data=df)
plt.show()
```



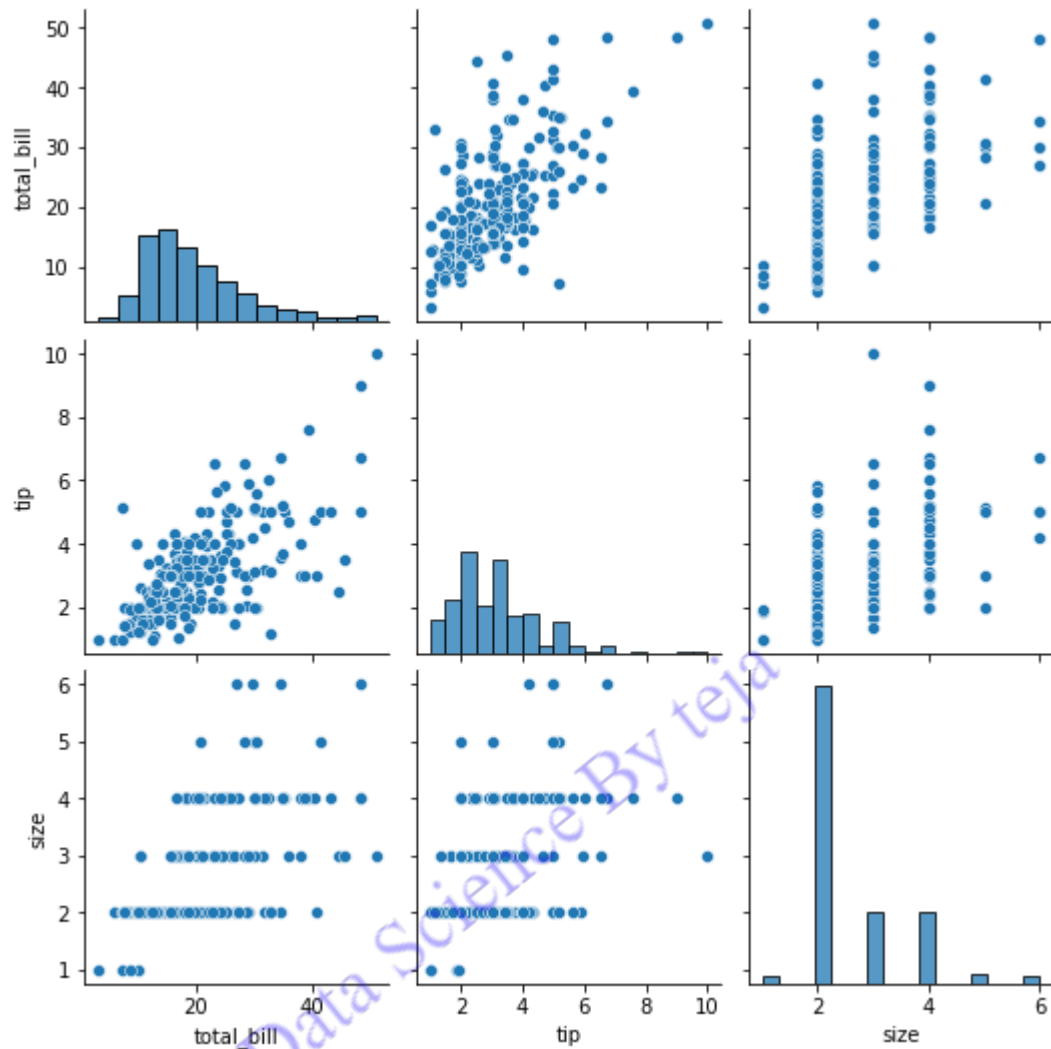
```
In [27]: sns.violinplot(x='total_bill',y='day',data=df,palette='rainbow')  
plt.show()
```



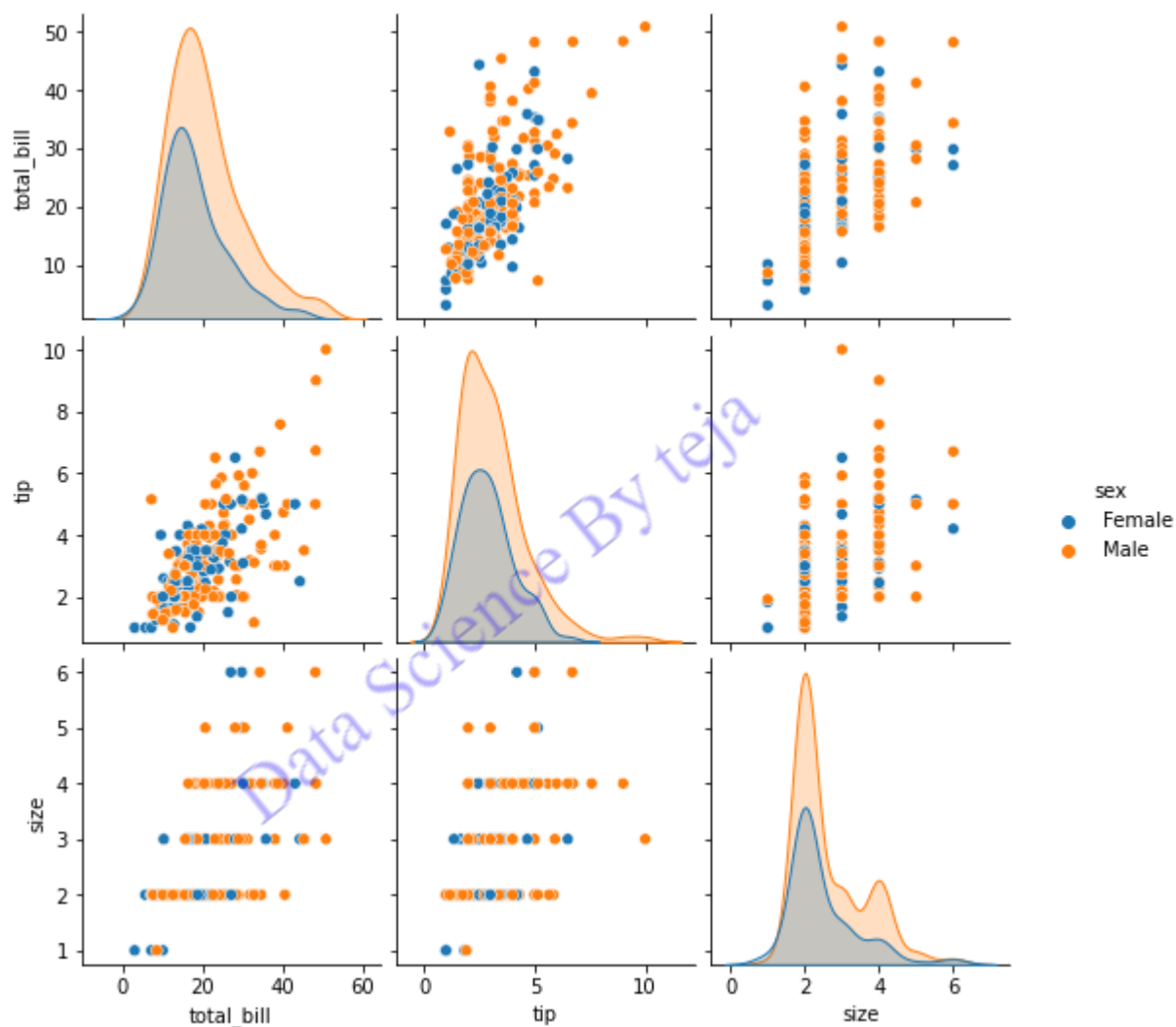
Pair Plot

Data Science By teja

```
In [28]: sns.pairplot(df)  
plt.show()
```



```
In [29]: sns.pairplot(df,hue='sex')
plt.show()
```



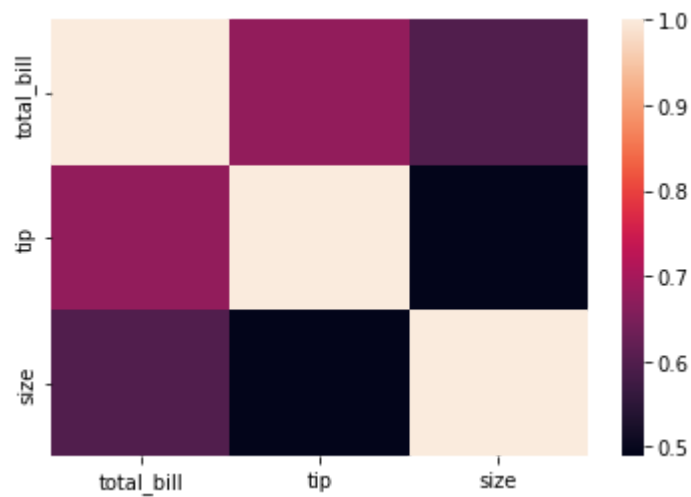
*Heat Map*

```
In [30]: df.corr()
```

Out[30]:

	total_bill	tip	size
total_bill	1.000000	0.675734	0.598315
tip	0.675734	1.000000	0.489299
size	0.598315	0.489299	1.000000

```
In [31]: sns.heatmap(df.corr())  
plt.show()
```



```
In [ ]:
```

Data Science By teja