In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

import seaborn as sns
```

In [2]:

```
df = pd.read_csv("tips.csv")
df.head()
```

Out[2]:

	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

In [3]:

```
continous= ['total_bill','tip']
discrete = ['sex','smoker','day','time','size']
```

Exploratory Data Analysis

In [4]:

```
# for concontinous variables
df[continous].describe()              # df[['total_bill','tip']].describe()
```

Out[4]:

	total_bill	tip
count	244.000000	244.000000
mean	19.785943	2.998279
std	8.902412	1.383638
min	3.070000	1.000000
25%	13.347500	2.000000
50%	17.795000	2.900000
75%	24.127500	3.562500
max	50.810000	10.000000

In [5]:

```
# for discrete variables
df.describe(include='object')
```

Out[5]:

	sex	smoker	day	time
count	244	244	244	244
unique	2	2	4	2
top	Male	No	Sat	Dinner
freq	157	151	87	176

In [6]:

```
# for all types of variables
df.describe(include='all')
```

Out[6]:

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

In [7]:

```
df['sex'].value_counts()
```

Out[7]:

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

In [8]:

```
1 df['smoker'].value_counts()
```

Out[8]:

```
No     151
Yes     93
Name: smoker, dtype: int64
```

In [9]:

```
1 df['day'].value_counts()
```

Out[9]:

```
Sat     87
Sun     76
Thur    62
Fri     19
Name: day, dtype: int64
```

In [10]:

```
1 df['time'].value_counts()
```

Out[10]:

```
Dinner    176
Lunch      68
Name: time, dtype: int64
```

In [11]:

```
1 df['size'].value_counts()
```

Out[11]:

```
2    156
3     38
4     37
5      5
1      4
6      4
Name: size, dtype: int64
```

In [12]:

```
1 df.isnull().sum()
```

Out[12]:

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

Plot's for Continous Data

1. Univariate (Single Variable)

- Histogram
- Boxplot

2. Bivariate (plot between two Variables)

- Scatter plot
- Line plot
- Joint plot
- Violin plot

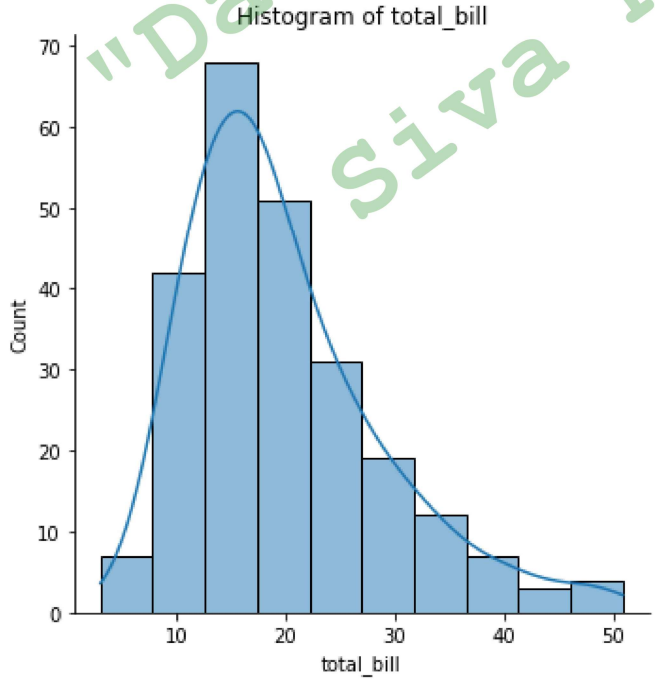
3. Multivariate (More than 2 Variables)

- Pair plot
- Heatmap

Histogram/Distribution plot

In [14]:

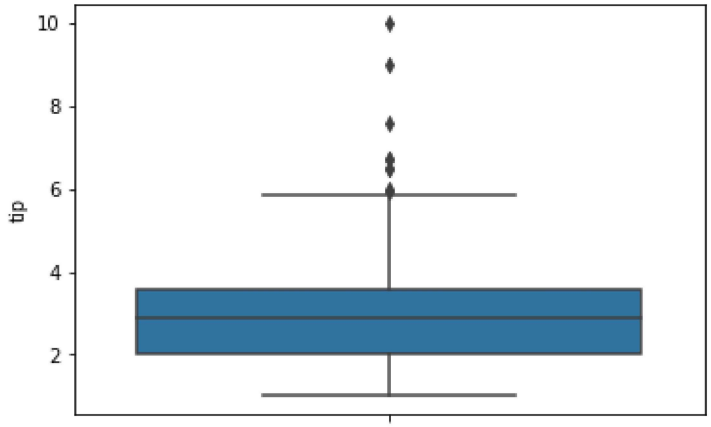
```
1 sns.displot(df['total_bill'],bins=10,kde=True)
2 plt.show()
```



Box plot

In [15]:

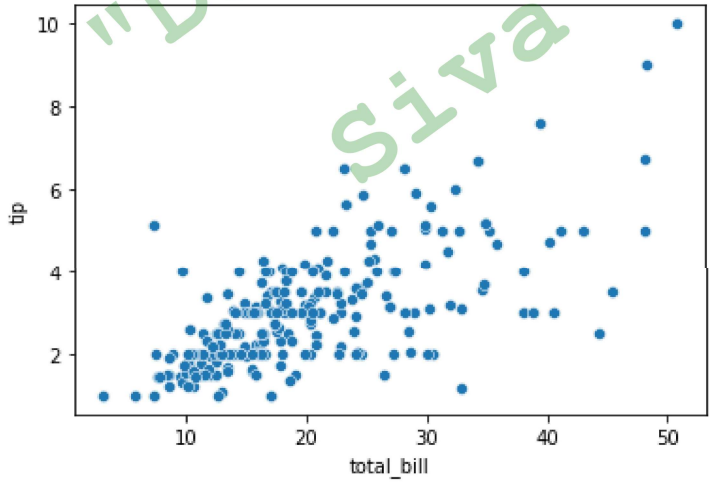
```
1 sns.boxplot(y=df["tip"])
2 plt.show()
```



Scatter Plot

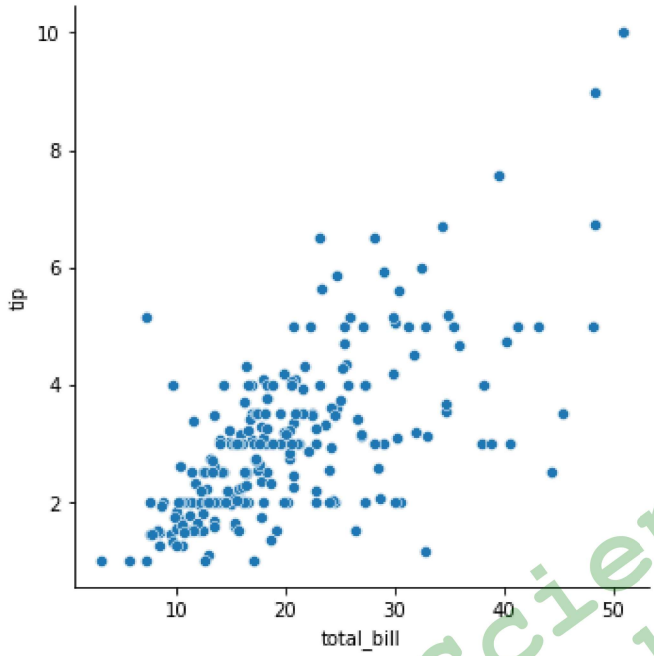
In [16]:

```
1 sns.scatterplot(x=df['total_bill'],y=df['tip'])
2 plt.show()
```



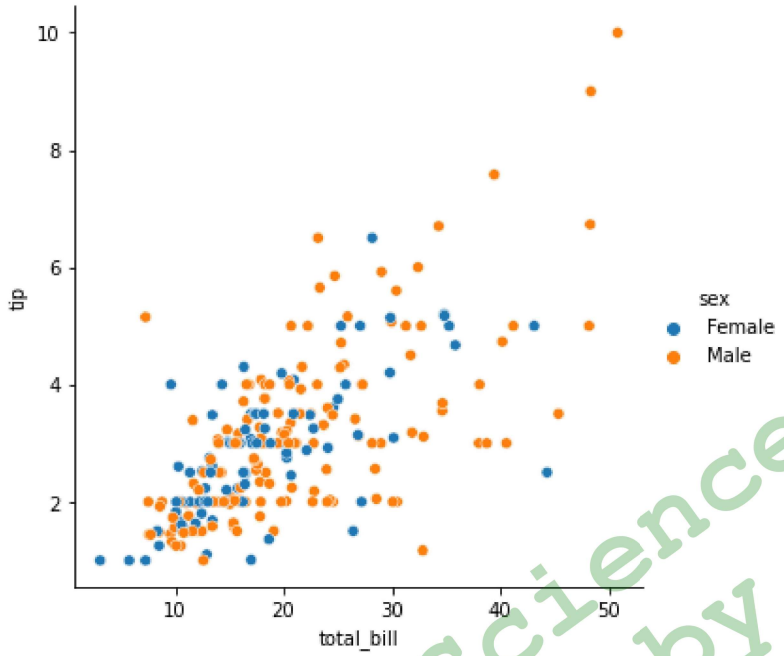
In [17]:

```
1 sns.relplot(x=df['total_bill'],y=df['tip'])
2 plt.show()
```



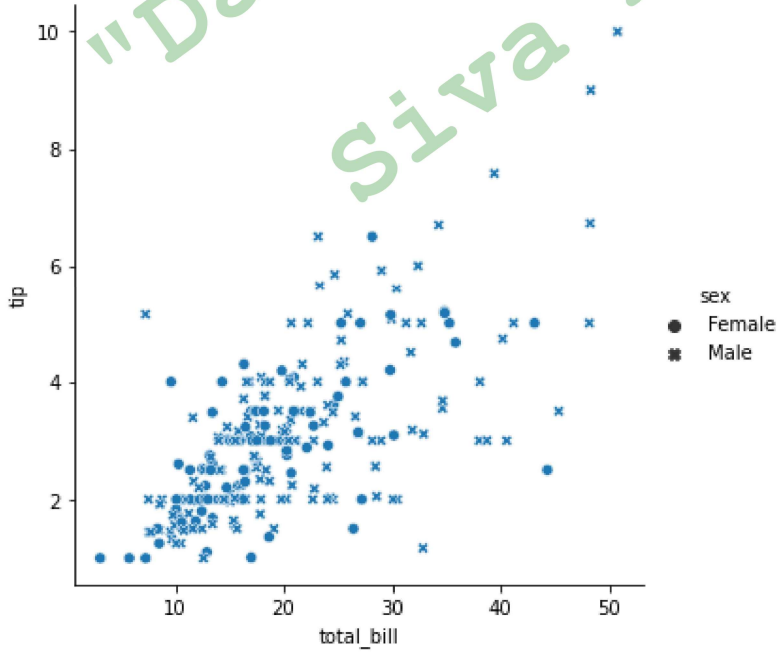
In [18]:

```
1 sns.relplot(x='total_bill',y='tip',data = df,hue='sex')
2 plt.show()
```



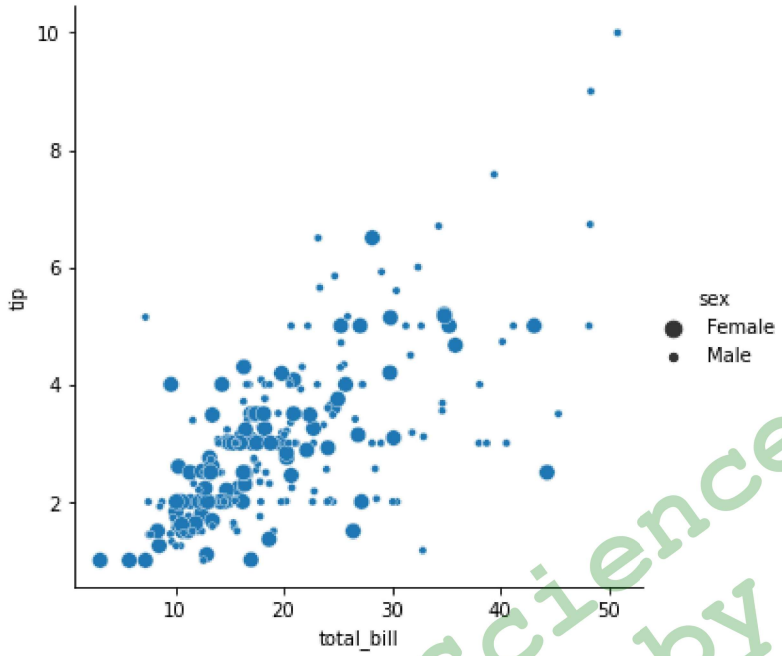
In [19]:

```
1 sns.relplot(x='total_bill',y='tip',data = df,style='sex')
2 plt.show()
```



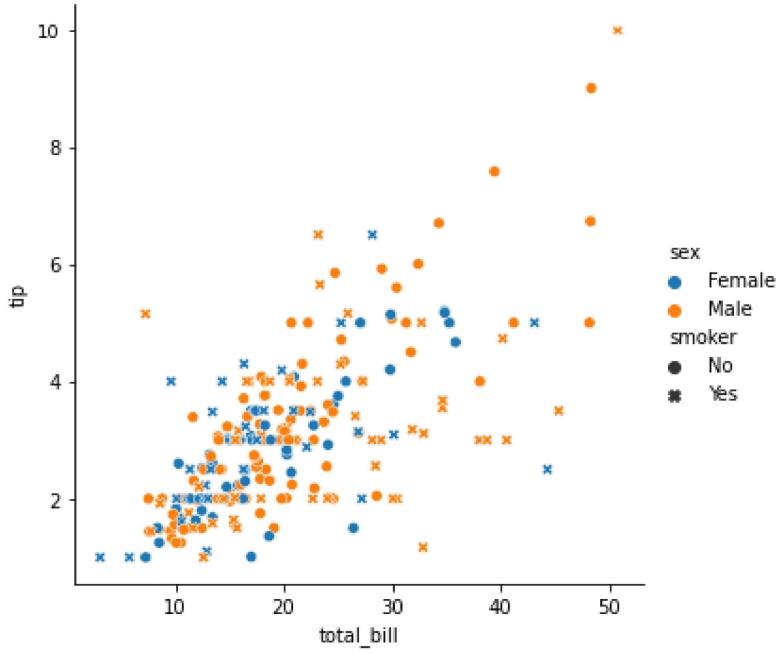
In [20]:

```
1 sns.relplot(x='total_bill',y='tip',data = df,size='sex')
2 plt.show()
```



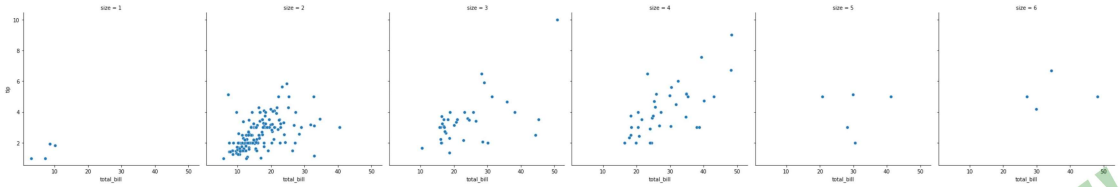
In [21]:

```
1 sns.relplot(x='total_bill',y='tip',data = df,hue='sex',style="smoker")
2 plt.show()
```



In [22]:

```
1 sns.relplot(x='total_bill',y='tip',data = df,col='size')
2 plt.show()
```



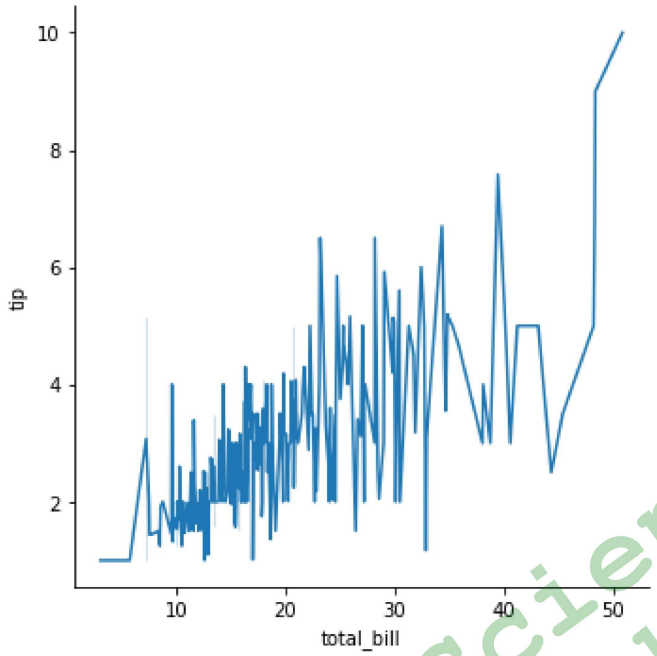
In [23]:

```
1 sns.relplot(x='total_bill',y='tip',data = df,hue="sex",col='size',col_wrap=2)
2 plt.show()
```

Line Plot

In [24]:

```
sns.relplot(x='total_bill',y='tip',data = df,kind='line')
plt.show()
```



In [25]:

```
df['sno'] = pd.DataFrame(np.arange(1,245))
df
```

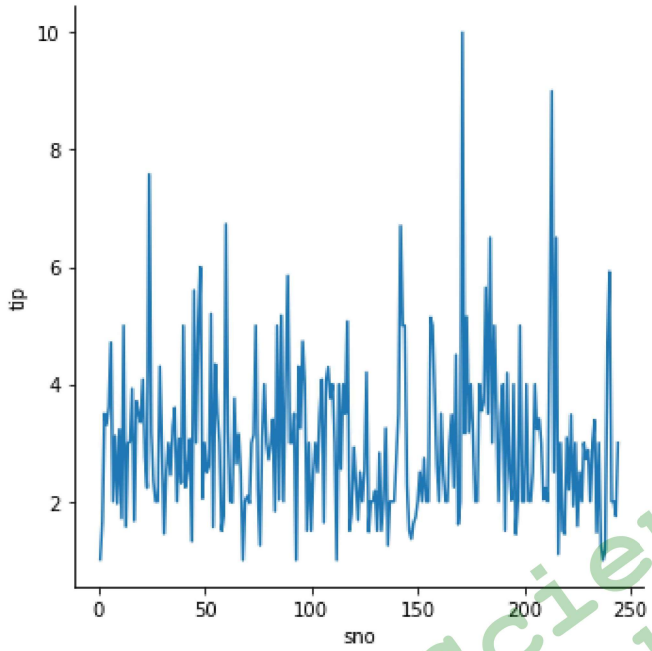
Out[25]:

	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 [26]:

```
1 sns.relplot(x = 'sno', y = 'tip', kind = 'line',data = df)
2 plt.show()
```



In [27]:

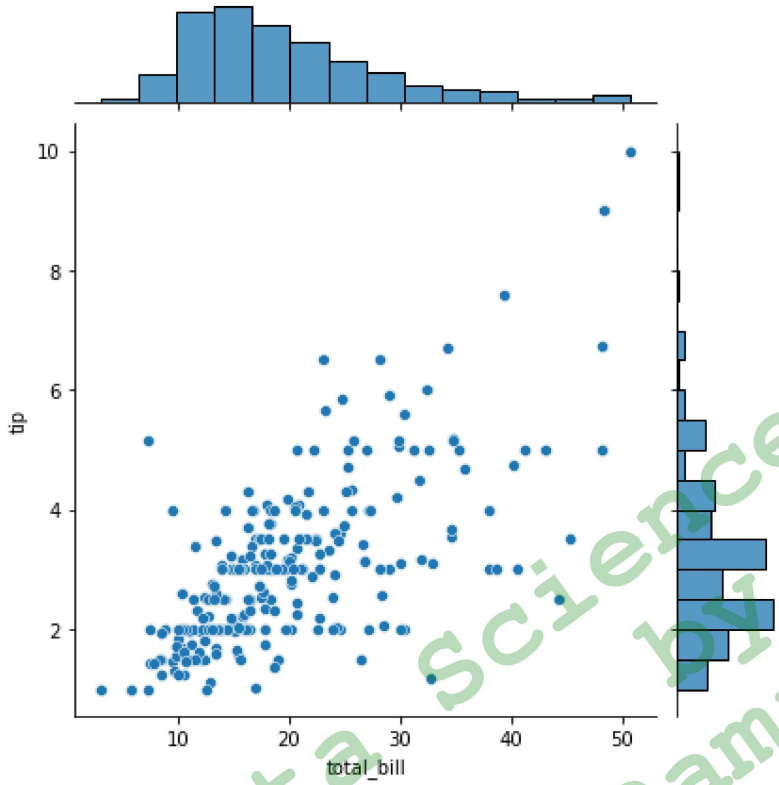
```
1 df.drop("sno",axis=1,inplace=True)
```


JoinPlot

A join plot allows to study the relationship between 2 numeric variables. The central chart display their correlation. It is usually a scatterplot, a hexbin plot, a 2D histogram or a 2D density plot

In [28]:

```
1 sns.jointplot(x='total_bill',y='tip',data=df)
2 plt.show()
```

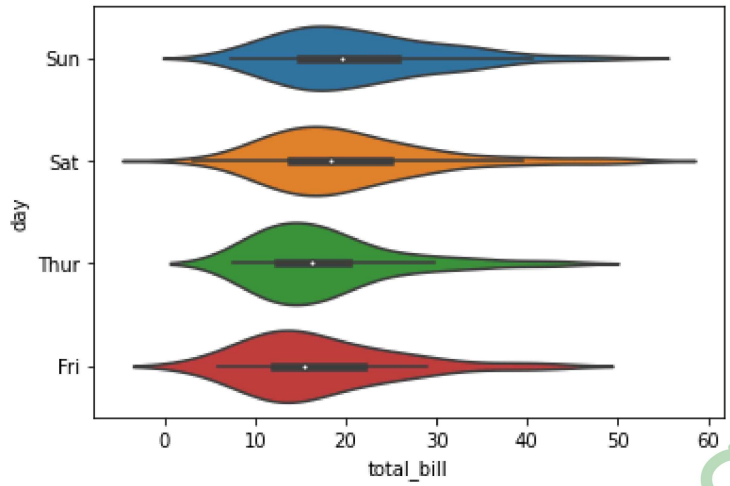


Violin Plot

Violin plot helps us to see both the distribution of data in terms of Kernel density estimation and the box plot

In [29]:

```
1 sns.violinplot(x="total_bill", y="day",data=df)
2 plt.show()
```

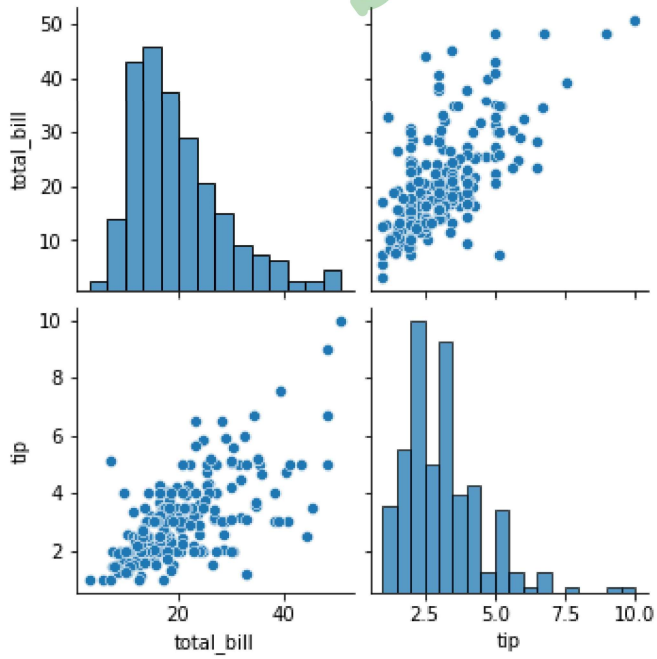


Pair plot - Multiple continous variables

A “pairs plot” is also known as a scatterplot, in which one variable in the same data row is matched with another variable's value, like this: Pairs plots are just elaborations on this, showing all variables paired with all the other variables

In [30]:

```
1 sns.pairplot(df,vars=continous)
2 plt.show()
```



Heat Map

A heatmap uses colored cells to represent relation between variables

Heatmap (for correlation)

- A correlation heatmap uses colored cells to show a 2D correlation matrix (table) between two numeric dimensions.
- It is very important in Feature Selection

Correlation (r) = $\frac{COV(x,y)}{\sigma x \sigma y}$

In [31]:

```
c_m=df[["total_bill","tip"]].corr()
c_m
```

Out[31]:

	total_bill	tip
total_bill	1.000000	0.675734
tip	0.675734	1.000000

In [32]:

```
sns.heatmap(c_m,annot=True)
plt.show()
```

Plot's for Discrete Data

1. Univariate (Single Variable)
    - Pie plot
    - Bar plot
    - Countplot
2. Bivariate (plot between two Variables)
    - Boxplot --> one discrete variable & one continous variable

CountPlot

In [33]:

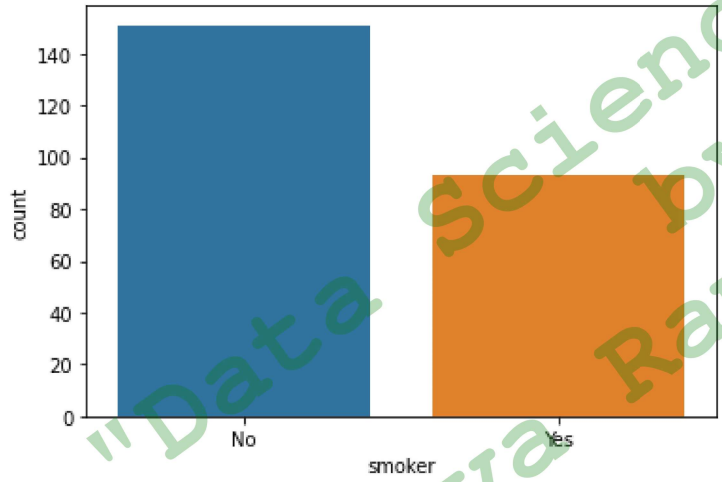
```
1 df['smoker'].value_counts()
```

Out[33]:

```
No     151
Yes     93
Name: smoker, dtype: int64
```

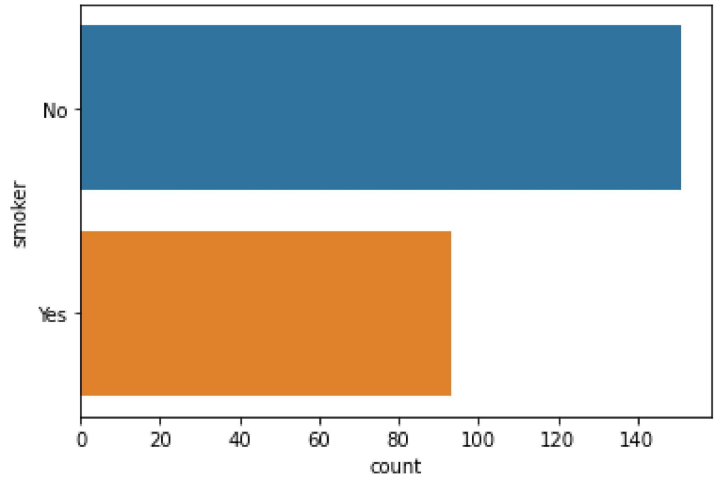
In [34]:

```
1 sns.countplot(x=df['smoker'])
2 plt.show()
```



In [35]:

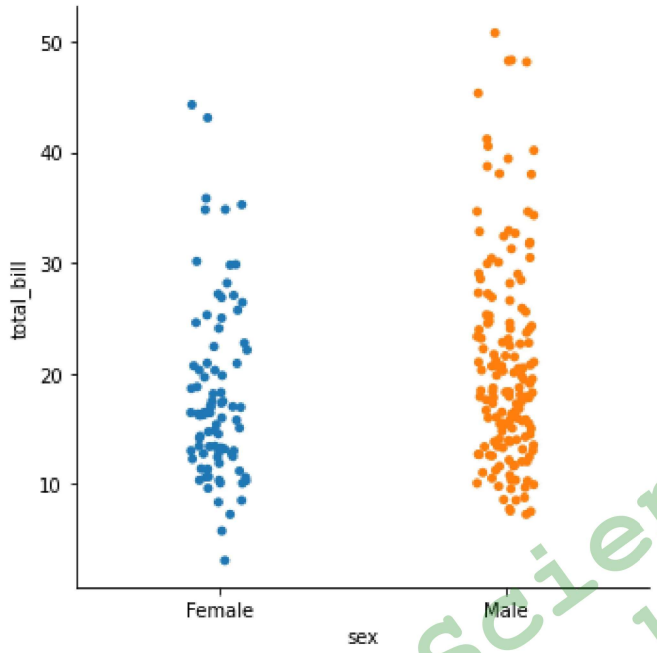
```
1 sns.countplot(y='smoker',data=df)
2 plt.show()
```



Bar plot

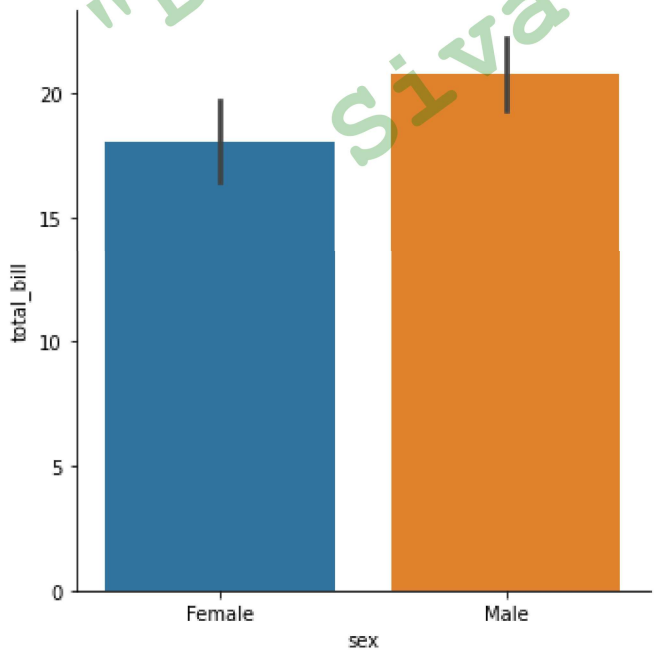
In [36]:

```
1 sns.catplot(x='sex',y='total_bill',data=df)
2 plt.show()
```



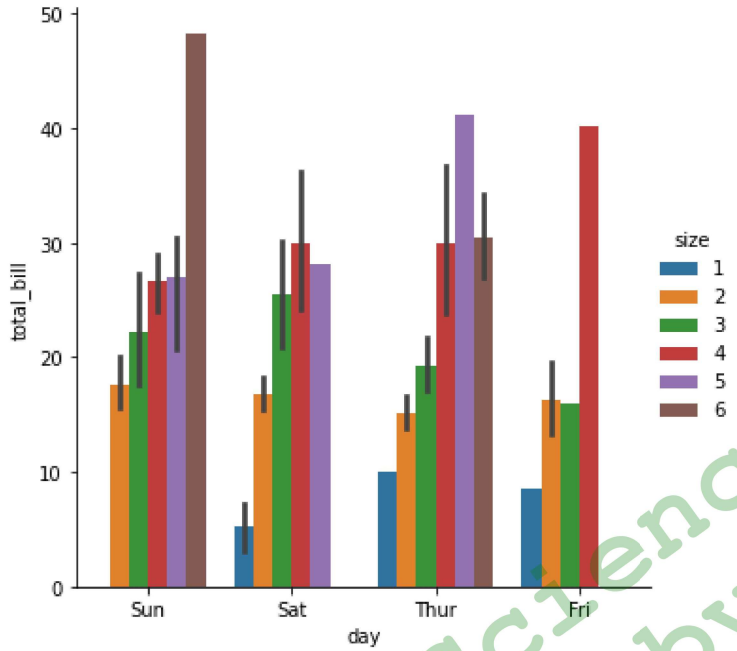
In [37]:

```
1 sns.catplot(x='sex',y='total_bill',data=df, kind="bar")
2 plt.show()
```



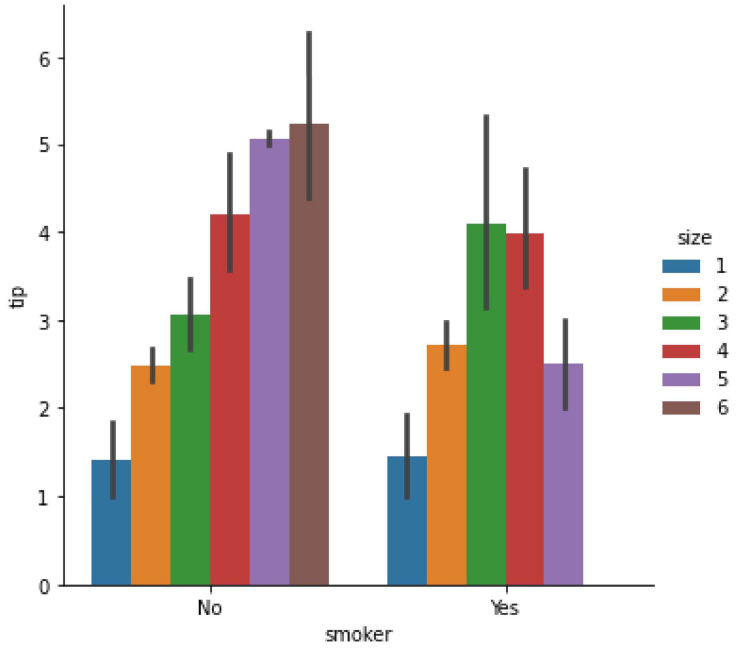
In [38]:

```
sns.catplot(x = 'day', y = 'total_bill', data = df, kind="bar", hue = 'size')
plt.show()
```



In [39]:

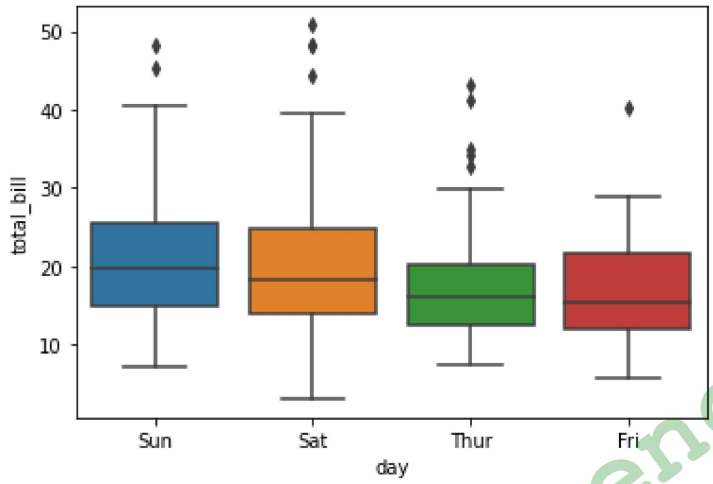
```
sns.catplot(x = 'smoker', y = 'tip', data = df, kind="bar",hue="size")
plt.show()
```



Box plot

In [40]:

```
1 sns.boxplot(x="day", y="total_bill", data=df)
2 plt.show()
```



In [41]:

```
1 sns.catplot(x="day", y="total_bill", hue="smoker", data=df, kind = 'box')
2 plt.show()
```