Name : Kundan Patil
Email id: patilkundan.1718@gmail.com (mailto:patilkundan.1718@gmail.com)
contact: +91 8485010139
linkedIn: www.linkedin.com/in/kundanpatilds (http://www.linkedin.com/in/kundanpatilds)
Github: https://github.com/patilkundan?tab=repositories (https://github.com/patilkundan?tab=repositories)

Q.6 Write Python Programming for given Dataset “tips.csv” file.

Expected Output:

1. Need to use required libraries
2. Perform pre-processing techniques
3. Apply visualization modules like Box Plot, Scatter plot and explain your understanding.

In [1]:

```
#Let's start with importing required libraries
import sklearn
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

In [2]:

```
df = pd. read_csv ("tips.csv") # Reading the Data
df.head()
```

Out[2]:

	Unnamed: 0	total_bill	tip	sex	smoker	day	time	size
0	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

In [3]:

```
df.shape
```

Out[3]:

```
(244, 8)
```

In [4]:

```
df.dtypes # df.info() also this way can be done
```

Out[4]:

```
Unnamed: 0      int64
total_bill    float64
tip           float64
sex            object
smoker         object
day            object
time           object
size            int64
dtype: object
```

In [5]:

```
df.columns
```

Out[5]:

```
Index(['Unnamed: 0', 'total_bill', 'tip', 'sex', 'smoker', 'day', 'time',
       'size'],
      dtype='object')
```

In [6]:

```
duplicate = df[df.duplicated()]
print("Duplicate Rows :")
```

```
Duplicate Rows :
```

In [7]:

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

Out[7]:

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

In [8]:

```
df.describe()
```

Out[8]:

	Unnamed: 0	total_bill	tip	size
count	244.000000	244.000000	244.000000	244.000000
mean	121.500000	19.785943	2.998279	2.569672
std	70.580923	8.902412	1.383638	0.951100
min	0.000000	3.070000	1.000000	1.000000
25%	60.750000	13.347500	2.000000	2.000000
50%	121.500000	17.795000	2.900000	2.000000
75%	182.250000	24.127500	3.562500	3.000000
max	243.000000	50.810000	10.000000	6.000000

In [9]:

```
df.drop (columns = ['Unnamed: 0'],inplace=True,)
```

In [11]:

```
df['day'].unique()
```

Out[11]:

```
array(['Sun', 'Sat', 'Thur', 'Fri'], dtype=object)
```

In [12]:

```
df.nunique()
```

Out[12]:

```
total_bill    229
tip           123
sex             2
smoker          2
day             4
time            2
size            6
dtype: int64
```

In [13]:

```
df.dtypes
```

Out[13]:

```
total_bill    float64
tip           float64
sex            object
smoker         object
day            object
time           object
size            int64
dtype: object
```

univariate analysis of continuous variables

total_bill

In [14]:

```
1 df['total_bill'].describe()
```

Out[14]:

```
count    244.000000
mean      19.785943
std        8.902412
min        3.070000
25%       13.347500
50%       17.795000
75%       24.127500
max       50.810000
Name: total_bill, dtype: float64
```

In [15]:

```
1 df['total_bill'].skew()
```

Out[15]:

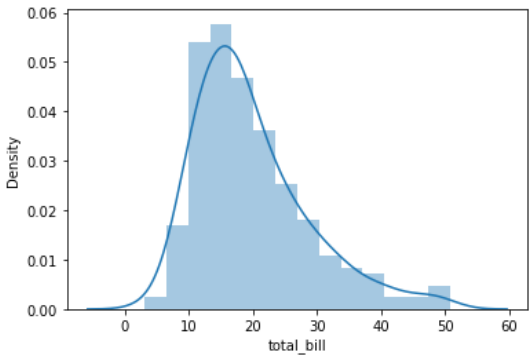
```
1.1332130376158205
```

In [16]:

```
1 sns.distplot(df['total_bill'])
```

Out[16]:

```
<AxesSubplot:xlabel='total_bill', ylabel='Density'>
```

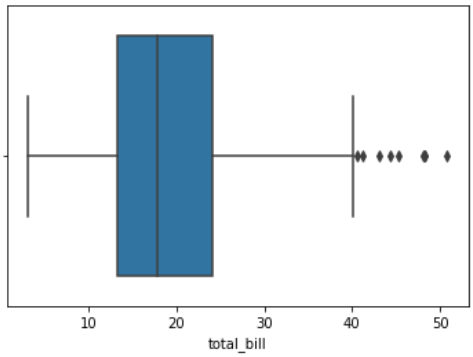


In [18]:

```
1 sns.boxplot(df['total_bill'])
```

Out[18]:

```
<AxesSubplot:xlabel='total_bill'>
```



tip

In [19]:

```
1 df['tip'].describe()
```

Out[19]:

```
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
```

In [20]:

```
1 df['tip'].skew()
```

Out[20]:

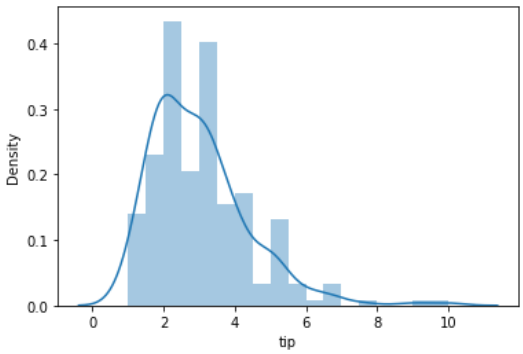
```
1.4654510370979401
```

In [21]:

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

Out[21]:

```
<AxesSubplot:xlabel='tip', ylabel='Density'>
```

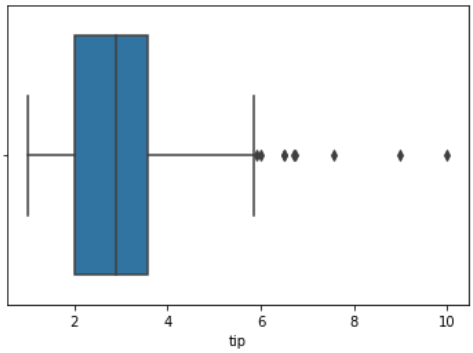


In [22]:

```
1 sns.boxplot(df['tip'])
```

Out[22]:

```
<AxesSubplot:xlabel='tip'>
```



size

In [23]:

```
print('unique categories--->',df['size'].unique())
#counting the uniques
print('value_counts for each unique categories---->',df['size'].value_counts())
print('Null value--->',df['size'].isnull().sum())
```

```
unique categories---> [2 3 4 1 6 5]
value_counts for each unique categories----> 2    156
3     38
4     37
5      5
1      4
6      4
Name: size, dtype: int64
Null value---> 0
```

In [24]:

```
df['size'].describe()
```

Out[24]:

```
count    244.000000
mean       2.569672
std        0.951100
min        1.000000
25%        2.000000
50%        2.000000
75%        3.000000
max        6.000000
Name: size, dtype: float64
```

In [25]:

```
df['size'].skew()
```

Out[25]:

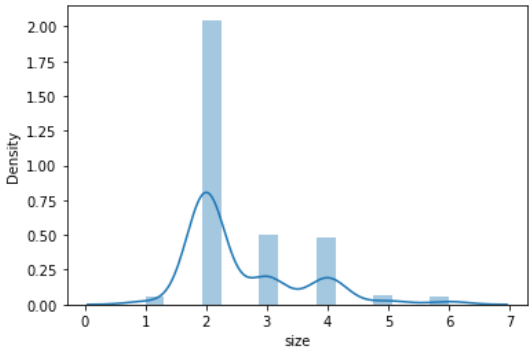
```
1.4478815386834785
```

In [27]:

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

Out[27]:

```
<AxesSubplot:xlabel='size', ylabel='Density'>
```

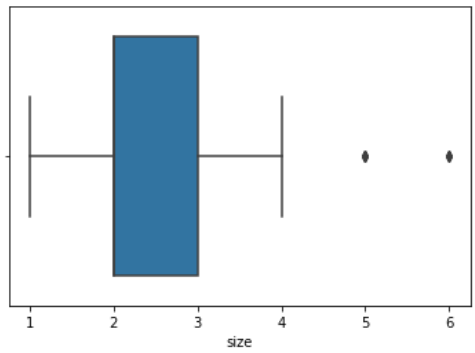


In [26]:

```
1 sns.boxplot(df['size'])
```

Out[26]:

<AxesSubplot:xlabel='size'>

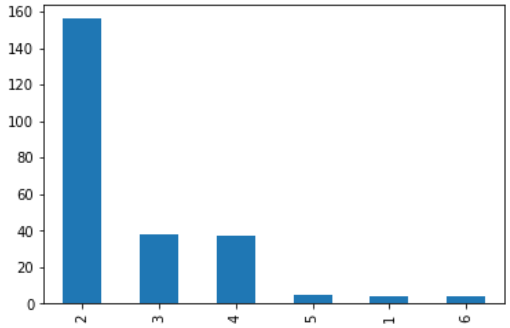


In [28]:

```
1 df['size'].value_counts().plot(kind='bar')
```

Out[28]:

<AxesSubplot:>



sex

In [29]:

```
1 print('unique categories--->',df['sex'].unique())
2 #counting the uniques
3 print('value_counts for each unique categories---->',df['sex'].value_counts())
4 print('Null value--->',df['sex'].isnull().sum())
```

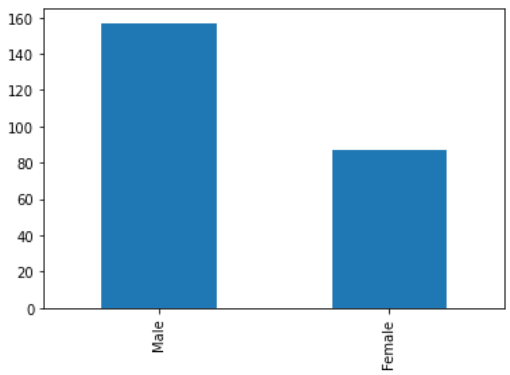
```
unique categories---> ['Female' 'Male']
value_counts for each unique categories----> Male      157
Female     87
Name: sex, dtype: int64
Null value---> 0
```

In [30]:

```
1 df['sex'].value_counts().plot(kind='bar')
```

Out[30]:

<AxesSubplot:>



smoker

In [31]:

```
print('unique categories--->',df['smoker'].unique())
#counting the uniques
print('value_counts for each unique categories---->',df['smoker'].value_counts())
print('Null value--->',df['smoker'].isnull().sum())
```

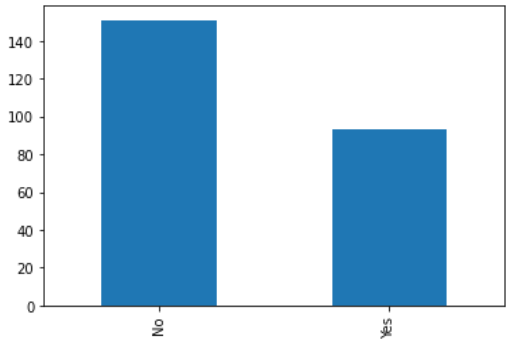
```
unique categories---> ['No' 'Yes']
value_counts for each unique categories----> No     151
Yes     93
Name: smoker, dtype: int64
Null value---> 0
```

In [32]:

```
df['smoker'].value_counts().plot(kind='bar')
```

Out[32]:

```
<AxesSubplot:>
```



day

In [33]:

```
print('unique categories--->',df['day'].unique())
#counting the uniques
print('value_counts for each unique categories---->',df['day'].value_counts())
print('Null value--->',df['day'].isnull().sum())
```

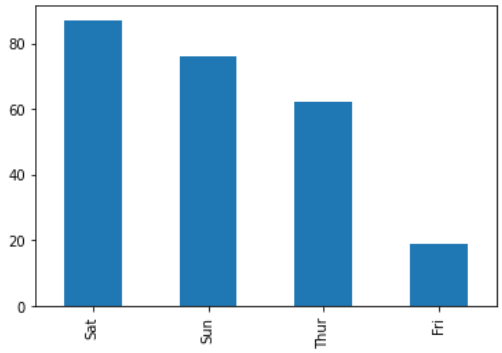
```
unique categories---> ['Sun' 'Sat' 'Thur' 'Fri']
value_counts for each unique categories----> Sat     87
Sun     76
Thur    62
Fri     19
Name: day, dtype: int64
Null value---> 0
```

In [34]:

```
df['day'].value_counts().plot(kind='bar')
```

Out[34]:

```
<AxesSubplot:>
```



time

In [36]:

```
print('unique categories--->',df['time'].unique())
#counting the uniques
print('value_counts for each unique categories---->',df['time'].value_counts())
print('Null value--->',df['time'].isnull().sum())
```

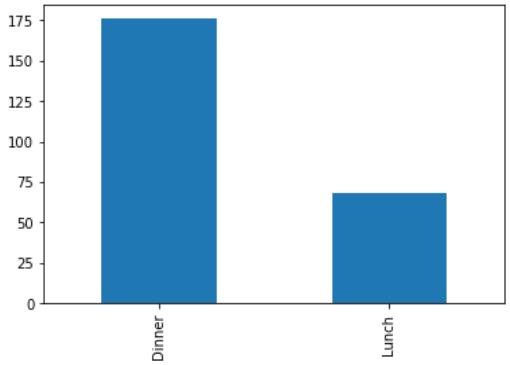
```
unique categories---> ['Dinner' 'Lunch']
value_counts for each unique categories----> Dinner    176
Lunch      68
Name: time, dtype: int64
Null value---> 0
```

In [37]:

```
df['time'].value_counts().plot(kind='bar')
```

Out[37]:

```
<AxesSubplot:>
```



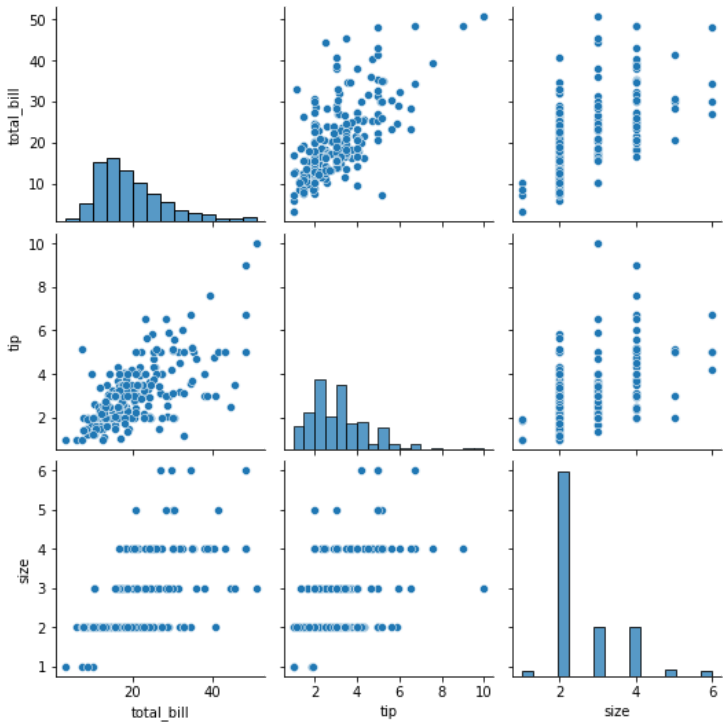
multivariate analysis

In [38]:

```
sns.pairplot(df)
```

Out[38]:

```
<seaborn.axisgrid.PairGrid at 0x1b3dfdd8a30>
```



In [ ]:

```

```

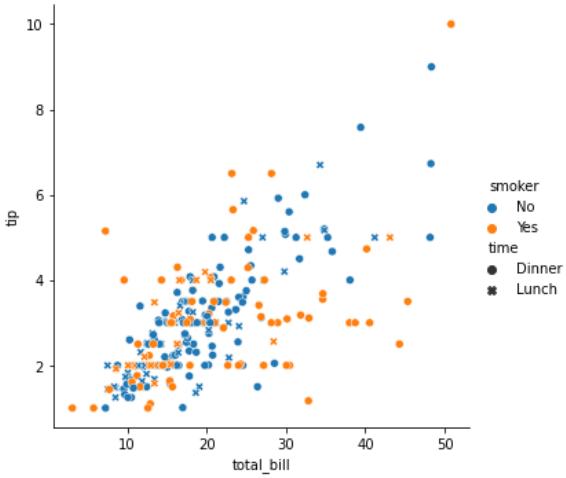
Type *Markdown* and LaTeX: α^2

Type *Markdown* and LaTeX: α^2

Type *Markdown* and LaTeX: α^2

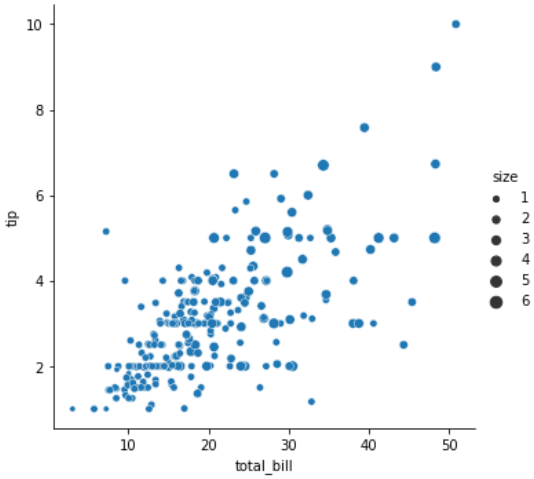
In [41]:

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



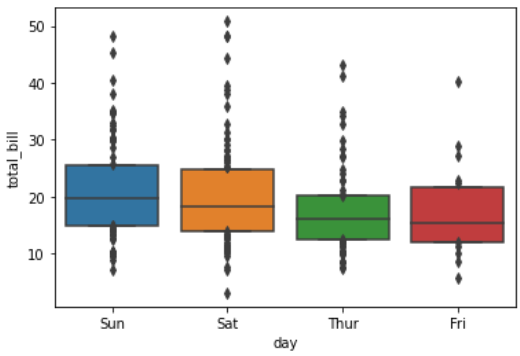
In [43]:

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



In [45]:

```
sns.boxplot(x='day',y='total_bill',data=df,whis=False)
plt.show()
```



In [47]:

```
x=pd.DataFrame(pd.pivot_table(df,index=['sex','smoker'],aggfunc='count')['total_bill'])
```

In [48]:

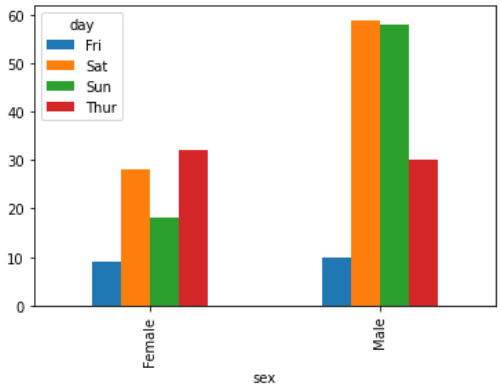
```
1 x
```

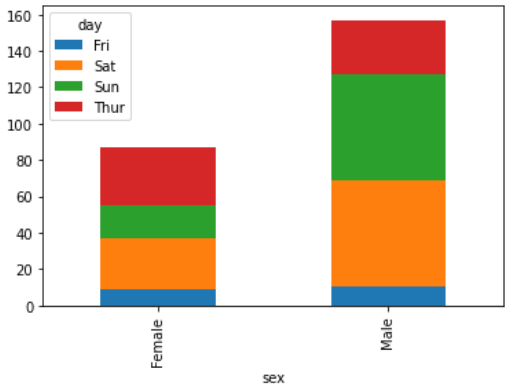
Out[48]:

		total_bill
sex	**smoker**	
Female	**No**	54
	Yes	33
Male	**No**	97
	Yes	60

In [50]:

```
1 pd.crosstab(df['sex'],df['day']).plot(kind='bar')
2 plt.show()
3 pd.crosstab(df['sex'],df['day']).plot(kind='bar',stacked=True)
4 plt.show()
```



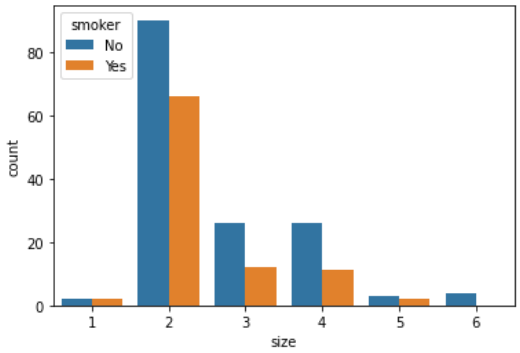


In [53]:

```
1 sns.countplot(
2     x='size', hue='smoker',data=df)
```

Out[53]:

```
<AxesSubplot:xlabel='size', ylabel='count'>
```



Maximum pepole are coming in size of 2 then followed by 3 , 4

maximum people prefer to go outside on Saturday and Sunday

Those people who are visiting single have an equal chance to smoke after lunch or dinner

people mostly go outise for the dinner

Based on the scatter plot, it's possible to see if there is a positive correlation between the total bill and the tip and size amount, meaning that as the total bill increases, so does the tip amount, which could suggest that customers tend to tip a higher percentage of the total bill for higher bill amounts. Or it could also indicate that when customers spend more, they tend to tip more.

Based on the scatter plot, it's possible to see if there is a positive correlation between the total bill and the party size, meaning that as the size of the party increases, so does the total bill, which could suggest that customers tend to spend more when there are more people in their party.

Based on the scatter plot, it's possible to see if there is a positive correlation between the tip amount and the party size, meaning that as the size of the party increases, so does the tip amount, which could suggest that customers tend to tip more when there are more people in their party.

In [ ]:

```
1
```