Seaborn By Virat Tiwari

October 18, 2023

1 Seaborn - Seaborn is used for data Visualization or Graphical representation of data , it is almost similar to the Matplotlib . We can easily understand the HIDDEN PATTERNS of data with the help of graphs and we get the INSIGHTS of data as well

```
[2]: # We can easily " import seaborn " with alias " sns "

     import seaborn as sns
```

Note - Seaborn provide some built-in Datasets

```
[3]: # sns.load_dataset ( ) function is used for importing the dataset in python

     # iris is built-in dataset that is provided by seaborn

     sns.load_dataset("iris")
```

```
[3]:      sepal_length  sepal_width  petal_length  petal_width    species
     0             5.1          3.5           1.4          0.2     setosa
     1             4.9          3.0           1.4          0.2     setosa
     2             4.7          3.2           1.3          0.2     setosa
     3             4.6          3.1           1.5          0.2     setosa
     4             5.0          3.6           1.4          0.2     setosa
     ..            …          …           …          …        …
     145           6.7          3.0           5.2          2.3  virginica
     146           6.3          2.5           5.0          1.9  virginica
     147           6.5          3.0           5.2          2.0  virginica
     148           6.2          3.4           5.4          2.3  virginica
     149           5.9          3.0           5.1          1.8  virginica

     [150 rows x 5 columns]
```

```
[24]: # We can store dataset in variable iris

      iris=sns.load_dataset("iris")
```

[37]:

```
# to_csv ( "here we write name of file that we want to save with extention" ) -␣
 ↪This function is used for saving the data file in our local system

iris.to_csv("iris.csv")
```

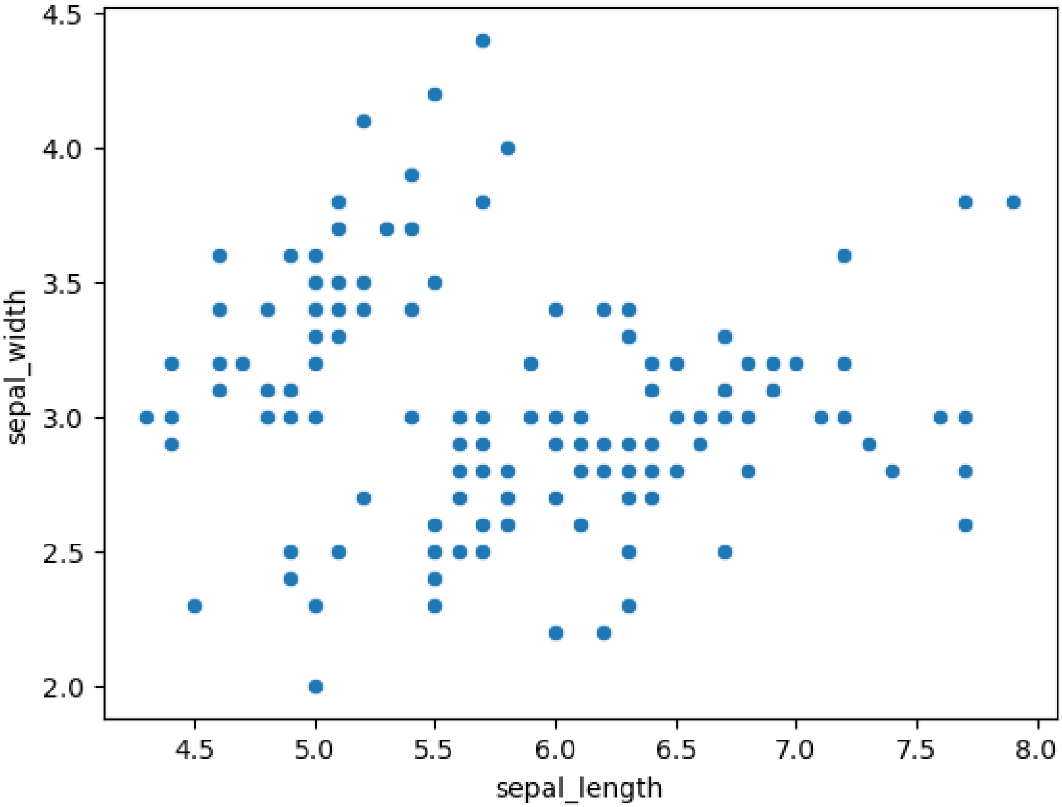
[25]:

```
# sns.scatter ( ) function is used for represent the data in scatter form or␣
 ↪graph

sns.scatterplot(x=iris.sepal_length,y=iris.sepal_width)
```

[25]:

```
<AxesSubplot: xlabel='sepal_length', ylabel='sepal_width'>
```

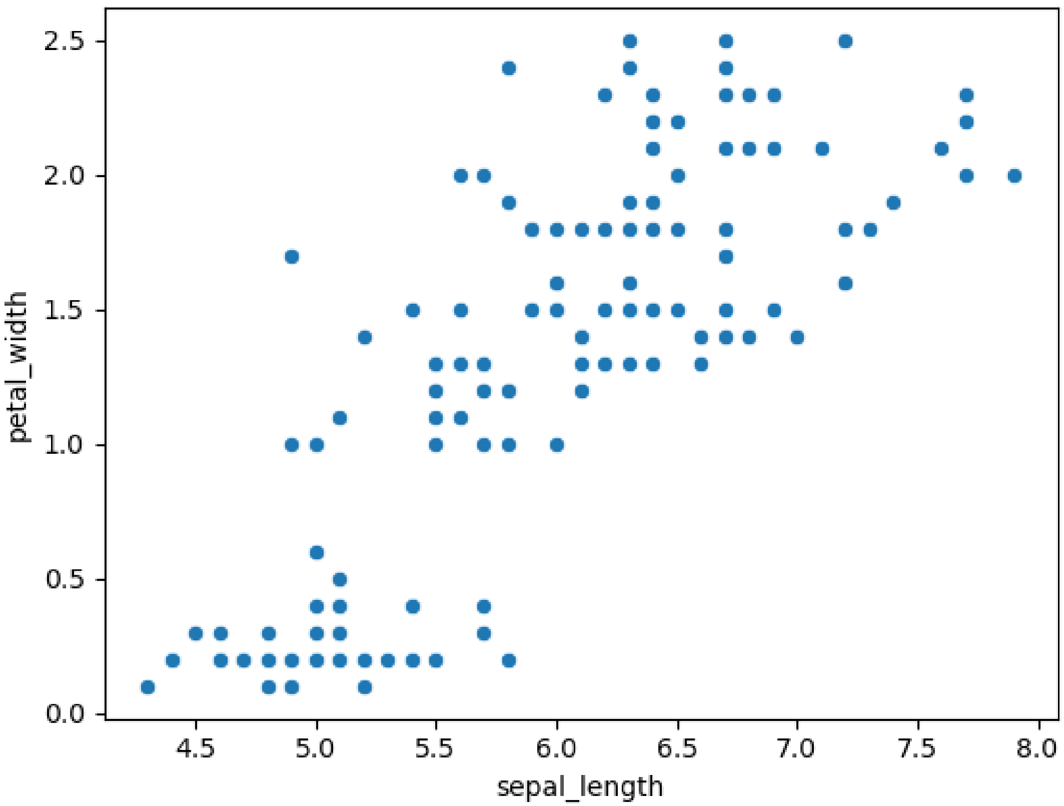


[26]:

```
sns.scatterplot(x=iris.sepal_length,y=iris.petal_width)
```

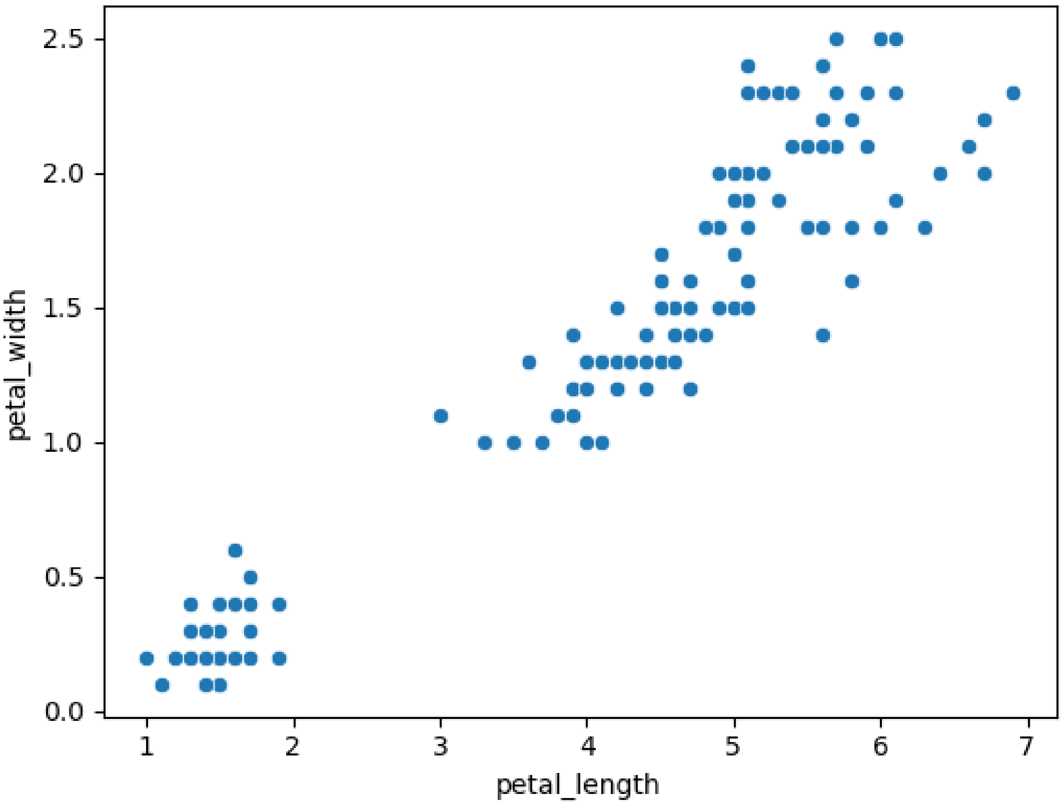
[26]:

```
<AxesSubplot: xlabel='sepal_length', ylabel='petal_width'>
```



```
[27]: sns.scatterplot(x=iris.petal_length,y=iris.petal_width)
```

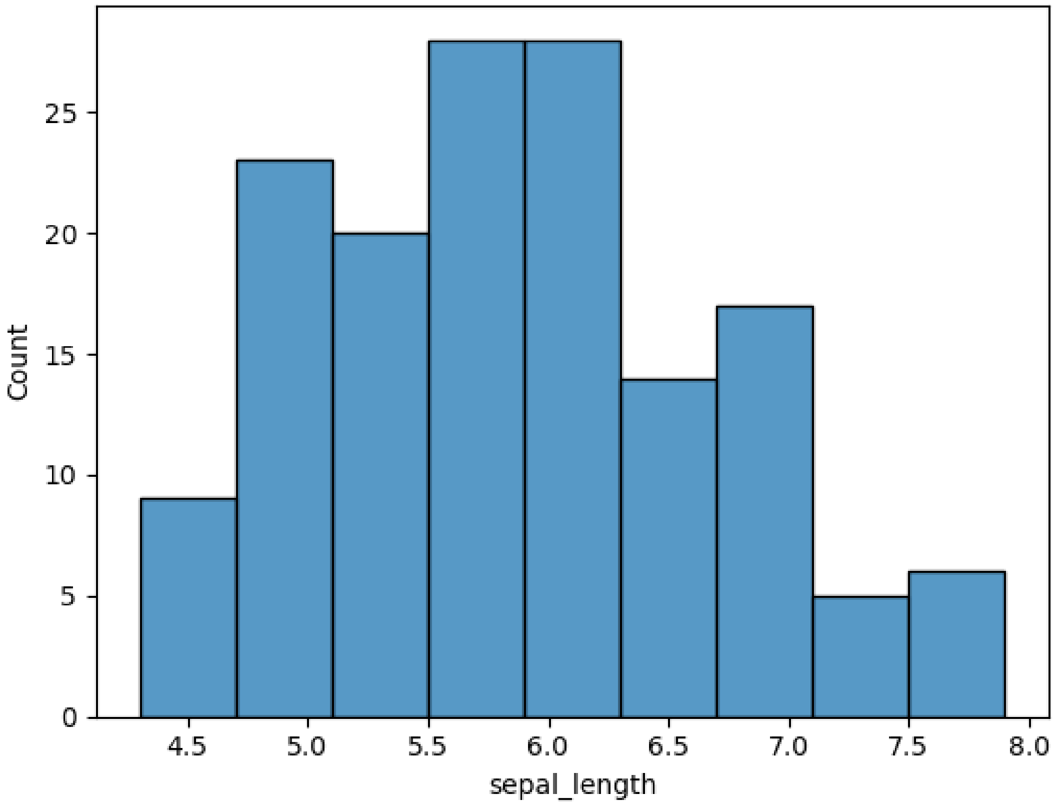
```
[27]: <AxesSubplot: xlabel='petal_length', ylabel='petal_width'>
```



[29]:
```
# histplot ( ) function is used for histogram graphs and histplot is equivalent␣
↪to distplot(distribution plot)

sns.histplot(iris["sepal_length"])
```

[29]:
```
<AxesSubplot: xlabel='sepal_length', ylabel='Count'>
```



[30]:
```
# Tips is built-in dataset that is provided by seaborn

# sns.load_dataset ( ) fucntion is used for importing the dataset , we pass␣
↪name of datset inside the load_dataset dunction

tips=sns.load_dataset("tips")
```

[31]:
```
tips
```

[31]:
```
     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
```

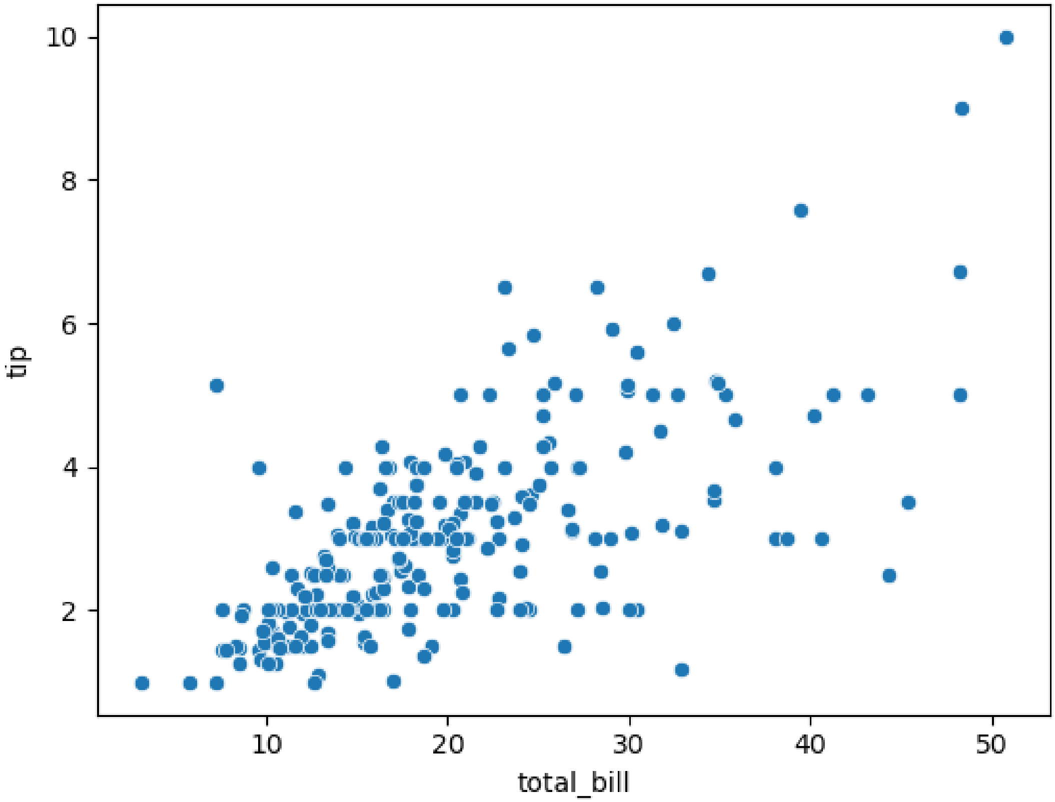
```
[244 rows x 7 columns]
```

[35]:
```
# This is how we save dataset in our local file system or in our computer

tips.to_csv("Tips.csv")
```

[38]:
```
sns.scatterplot(x=tips.total_bill,y=tips.tip)
```

[38]:
```
<AxesSubplot: xlabel='total_bill', ylabel='tip'>
```



[39]:
```
# head ( ) function give the initial 5 data from the dataset

tips.head()
```

[39]:
```
   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
```

NOTE- IMPORTANT FUNCTIONS -

[40]:
```
# value_counts ( ) function give the total values of numbers from the dataset

# tips [ "here we pass the column name" ]

# Here we find out that how many people smoke or how many are not smoke

tips["smoker"].value_counts()
```

[40]:
```
No     151
Yes     93
Name: smoker, dtype: int64
```

[41]:
```
# value_counts ( ) function give the total values of numbers from the dataset

# tips [ "here we pass the column name" ]

# Here we find out that how many people Male or how many are Female


tips["sex"].value_counts()
```

[41]:
```
Male      157
Female     87
Name: sex, dtype: int64
```

[42]:
```
# value_counts ( ) function give the total values of numbers from the dataset

# tips [ "here we pass the column name" ]

# Here we find out that how many total days of dinner and lunch

tips["time"].value_counts()
```

[42]:
```
Dinner    176
Lunch      68
Name: time, dtype: int64
```

[43]:
```
# value_counts ( ) function give the total values of numbers from the dataset

# tips [ "here we pass the column name" ]

# Here we find out that how many total days of sun,mon,thur,fri in numbers from␣
 ↪the dataset

tips["day"].value_counts()
```

[43]:
```
Sat     87
Sun     76
Thur    62
Fri     19
Name: day, dtype: int64
```

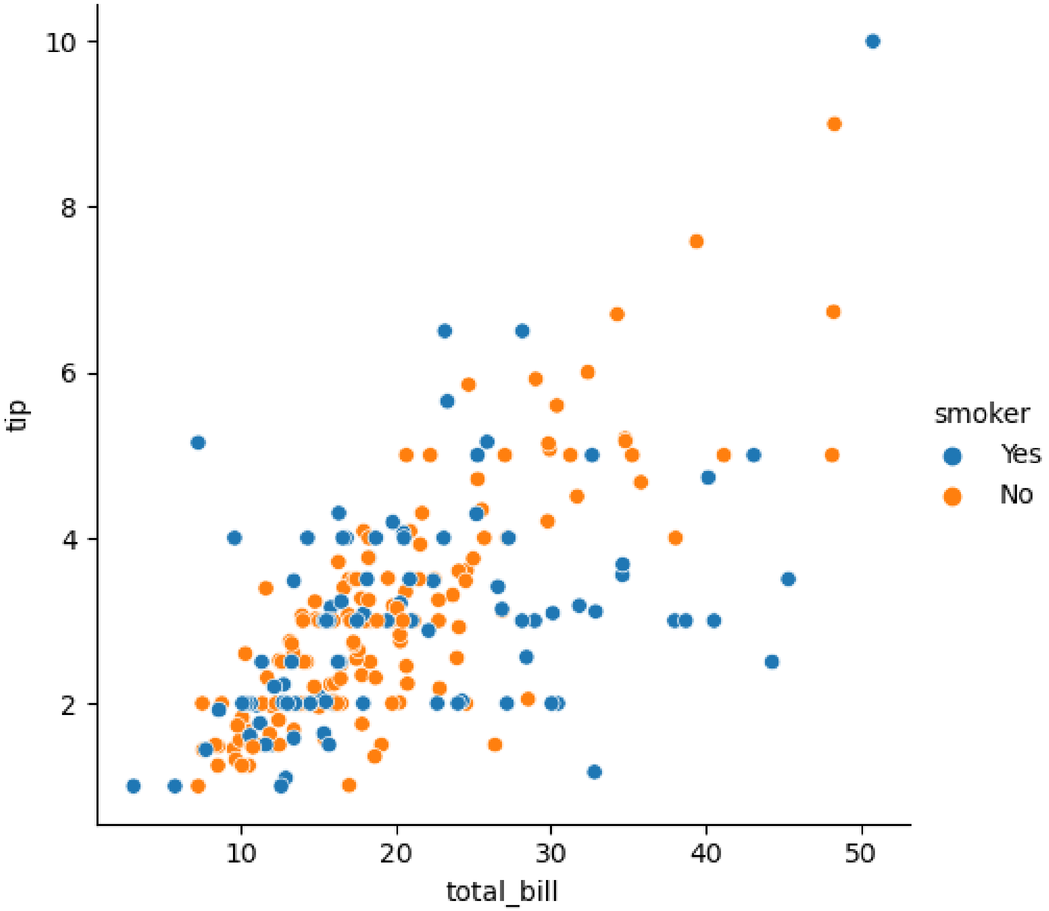
[48]:
```
# NOTE- IMPORTANT

# relplot ( ) is used for representing data for analysis purpose

# hue is nothing but define a colour

sns.relplot(x=tips.total_bill,y=tips.tip,data=tips,hue="smoker")
```

[48]:
```
<seaborn.axisgrid.FacetGrid at 0x7f2b18266620>
```



[49]:
```
tips.head()
```

[49]:

```
    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
```

Note - In the below grapgh , We find out or simply analysis that who paid the highest bill and that person came single or came in group

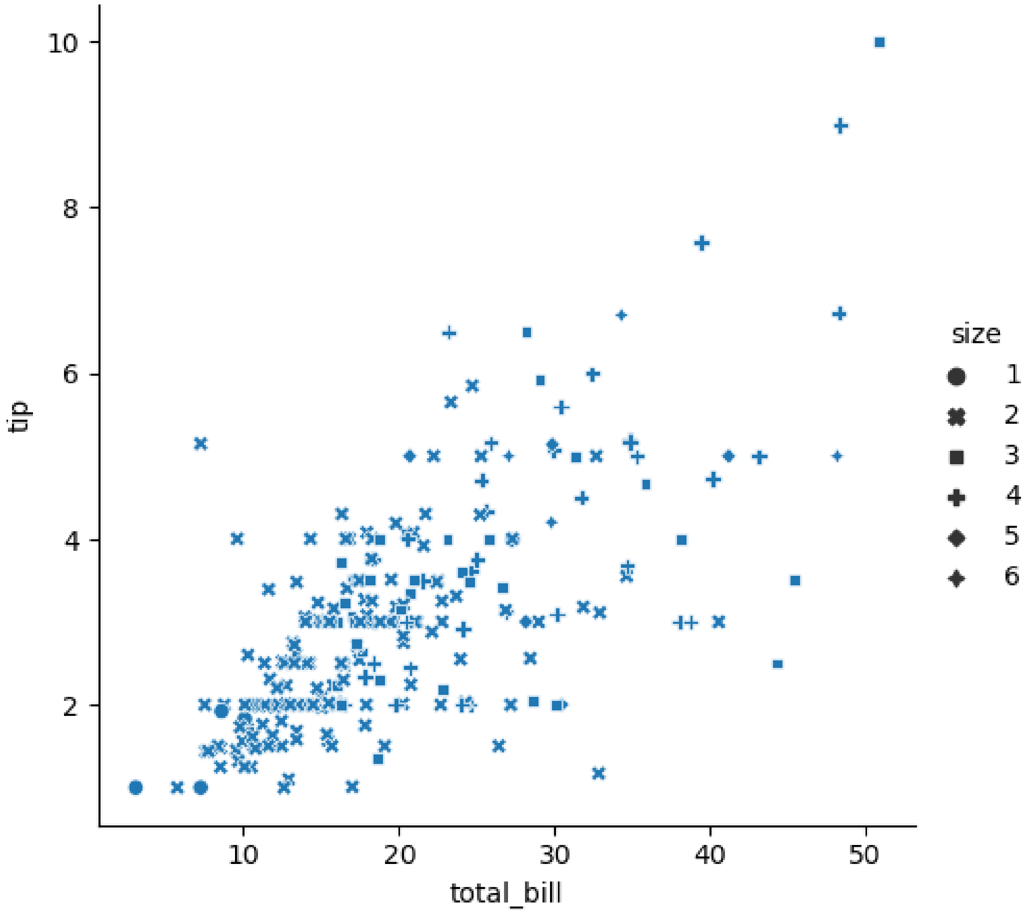
[52]:

```
# style present the data in different forms

sns.relplot(x=tips.total_bill,y=tips.tip,data=tips,style="size")
```

[52]:

```
<seaborn.axisgrid.FacetGrid at 0x7f2b183dd030>
```



[51]:

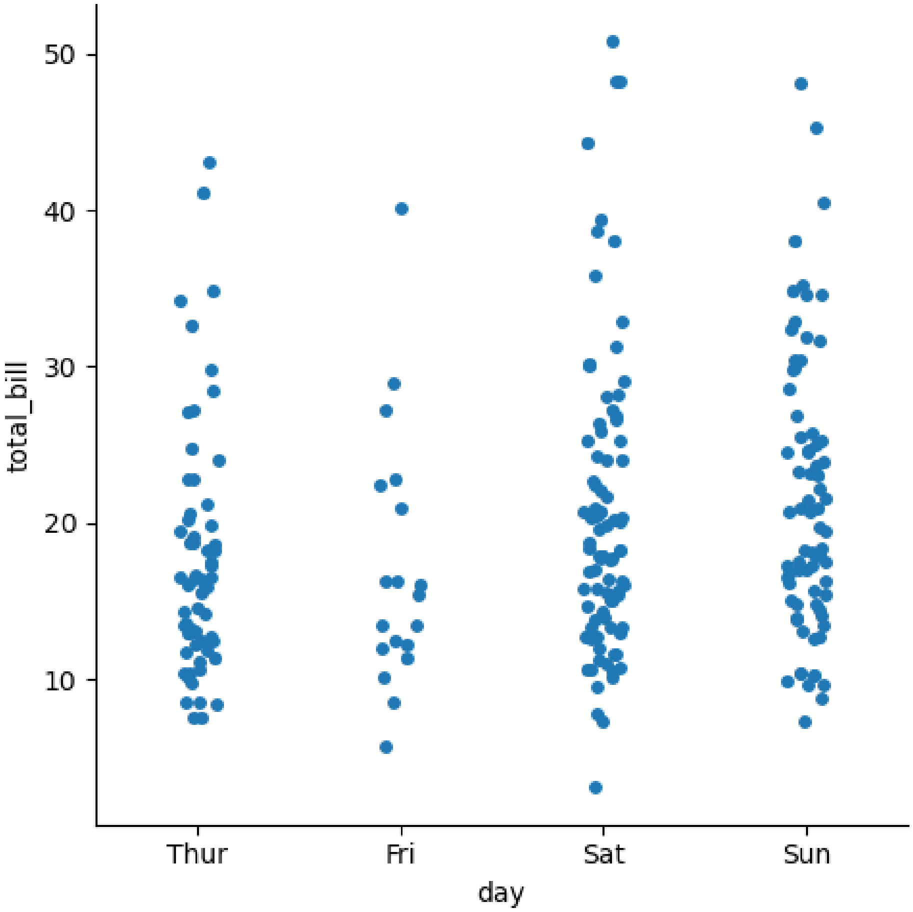
```
tips["size"].value_counts()
```

[51]:
```
2    156
3     38
4     37
5      5
1      4
6      4
Name: size, dtype: int64
```

In the given below graph , we have seen that on which day maximum engagement generate on the restrourent so we found that maximum people came on Saturday

[53]:
```
sns.catplot(x="day",y="total_bill",data=tips)
```

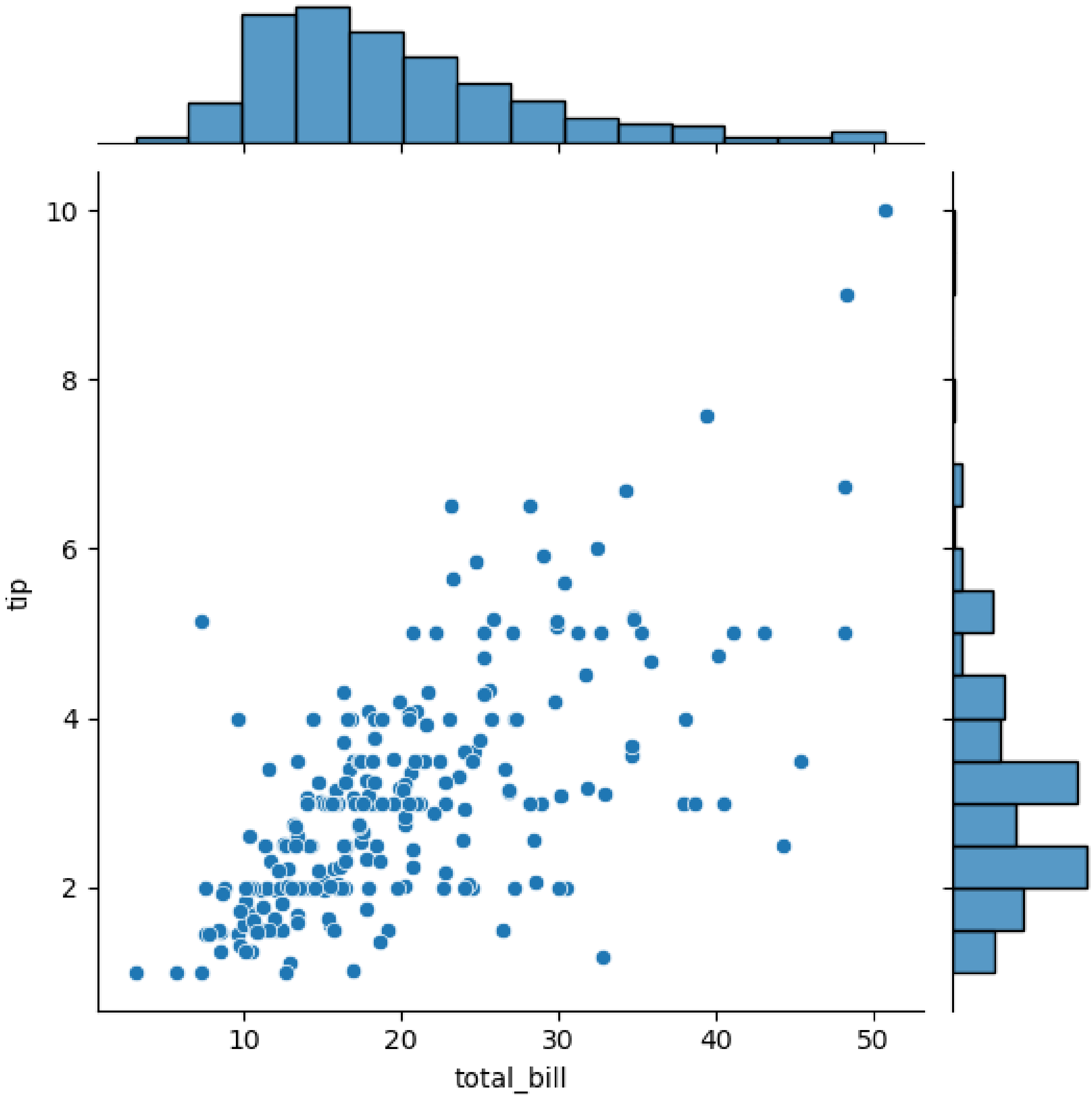
[53]:
```
<seaborn.axisgrid.FacetGrid at 0x7f2b183dd2a0>
```



[54]:
```
# joinplot ( ) function create a scatter + hsitogram graph together

sns.jointplot(x=tips.total_bill,y=tips.tip)
```

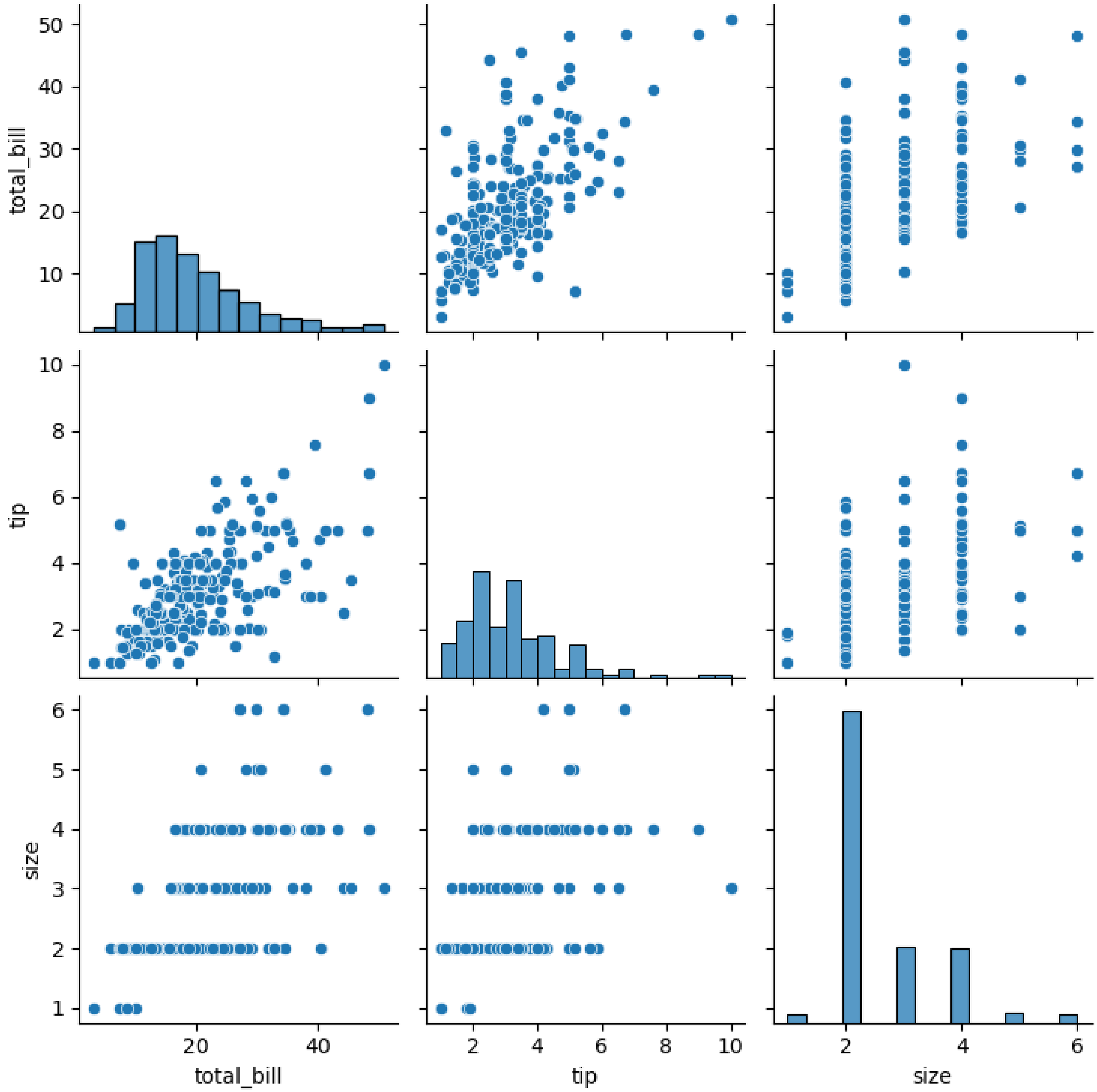
```
[54]: <seaborn.axisgrid.JointGrid at 0x7f2b1883d5a0>
```



```
[55]: # pairplot ( ) function is used for generate all graphs so it will easy to␣
       ↪compare with each others

      sns.pairplot(tips)
```

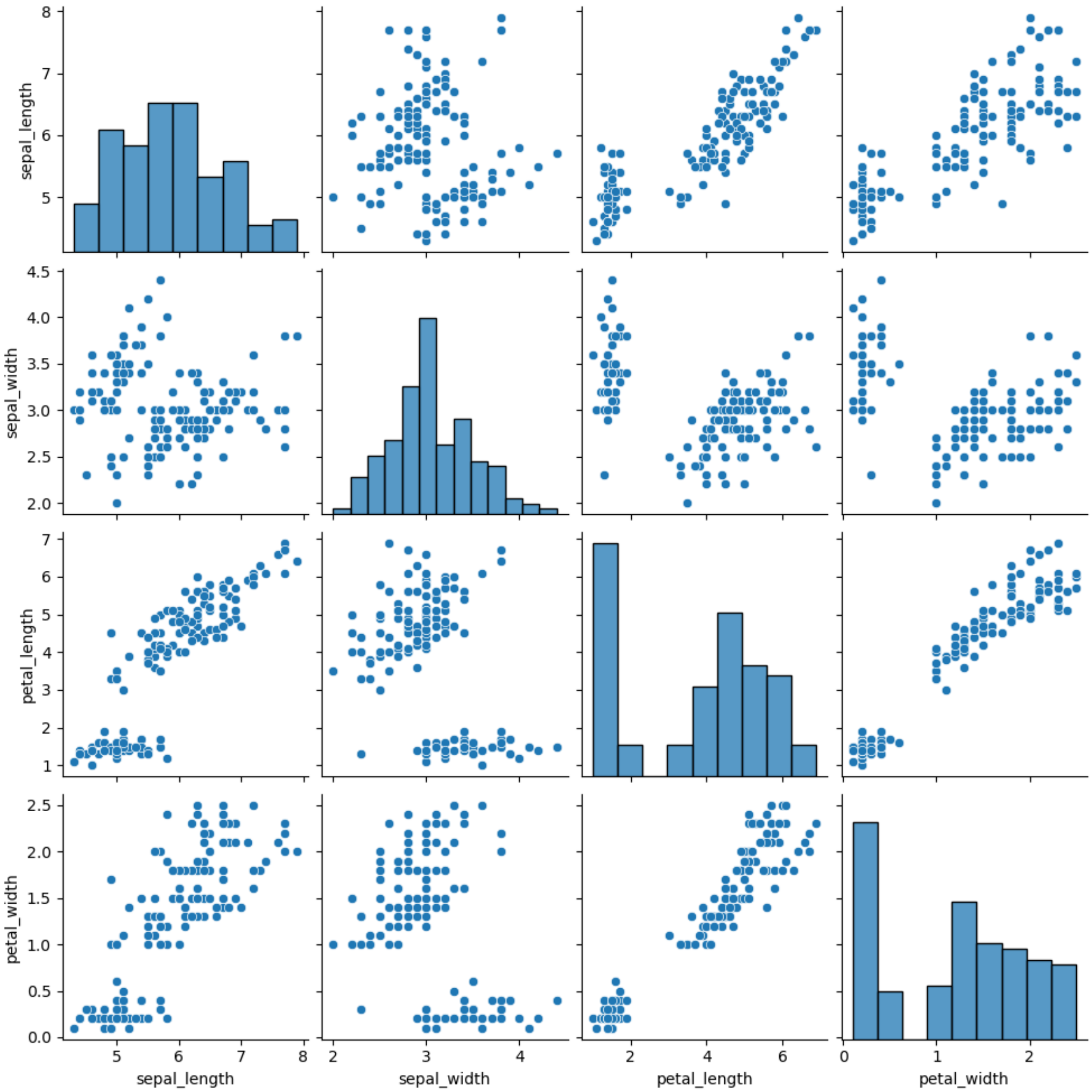
```
[55]: <seaborn.axisgrid.PairGrid at 0x7f2b19379990>
```



```
[57]: # pairplot ( ) function show all vs all , it simply plot all the pair of graphs

      sns.pairplot(iris)
```

```
[57]: <seaborn.axisgrid.PairGrid at 0x7f2b18f85cf0>
```



THANK YOU SO MUCH !!

YOURS VIRAT TIWARI :)