Seaborn Tutorial

Distribution Plots

. dishplot-

. jointplot- helps to analysis bivariant analysis

. pairplot- helps to analysis 3d graph

we can draw diagram in 2d,3d and 4d. 4d is impossible f1= univariant analysis f1,f2= bivariant analysis this type analysis easily can done by seaborn Practise problem on IRIS Dataset

In [6]:
```
import seaborn as sns
```

In [7]:
```
import pandas as pd
```

In [8]:
```
df=pd.read_csv('tips.csv')
```

In [9]:
```
df.head()
```

Out[9]:

	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 [10]:
```
df.dtypes
```

Out[10]:
```
total_bill    float64
tip           float64
sex            object
smoker         object
day            object
time           object
size            int64
dtype: object
```

In [ ]:
```

```

Correlation with Heatmap

A correlation heatmap uses colored cells, typically in a monochromatic scale, to show a 2D correlation matrix (table) between two discrete dimensions or events types, it is very important in feature selection

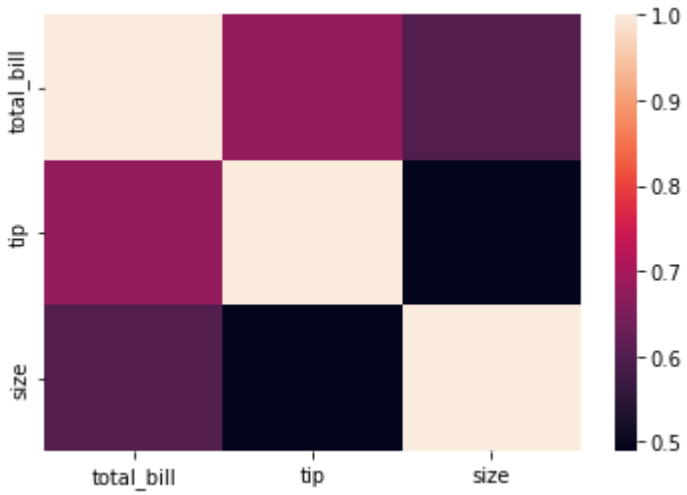
In [11]:
```
df.corr()
```

Out[11]:

	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 [12]:
```
sns.heatmap(df.corr())
```

Out[12]:
```
<AxesSubplot:>
```



Jointplot

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

Univariate Analysis

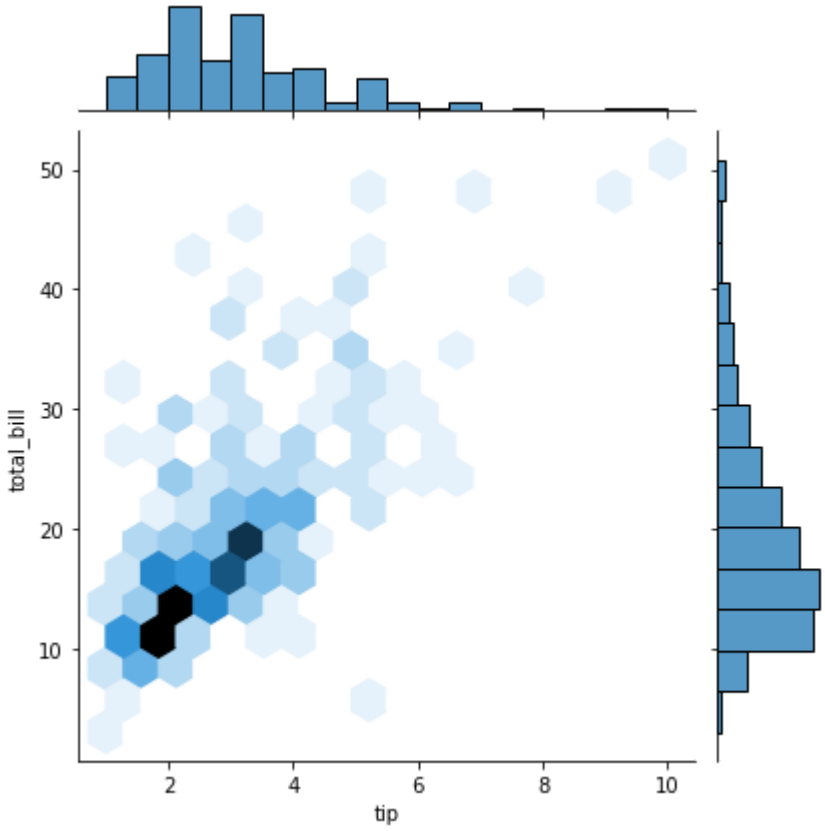
In [13]:
```
df.head()
```

Out[13]:

	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

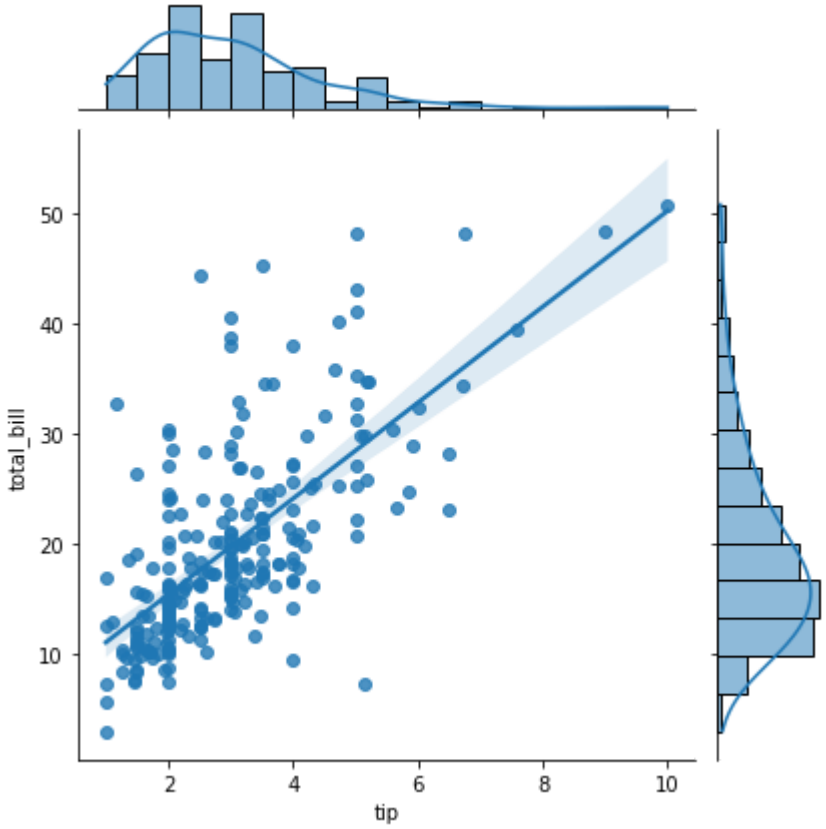
In [14]:
```
sns.jointplot(x='tip',y='total_bill',data=df,kind='hex')
```

Out[14]:
```
<seaborn.axisgrid.JointGrid at 0x28303cb3e50>
```



In [15]:
```
sns.jointplot(x='tip',y='total_bill',data=df,kind='reg')
```

Out[15]:
```
<seaborn.axisgrid.JointGrid at 0x28303e1e7f0>
```

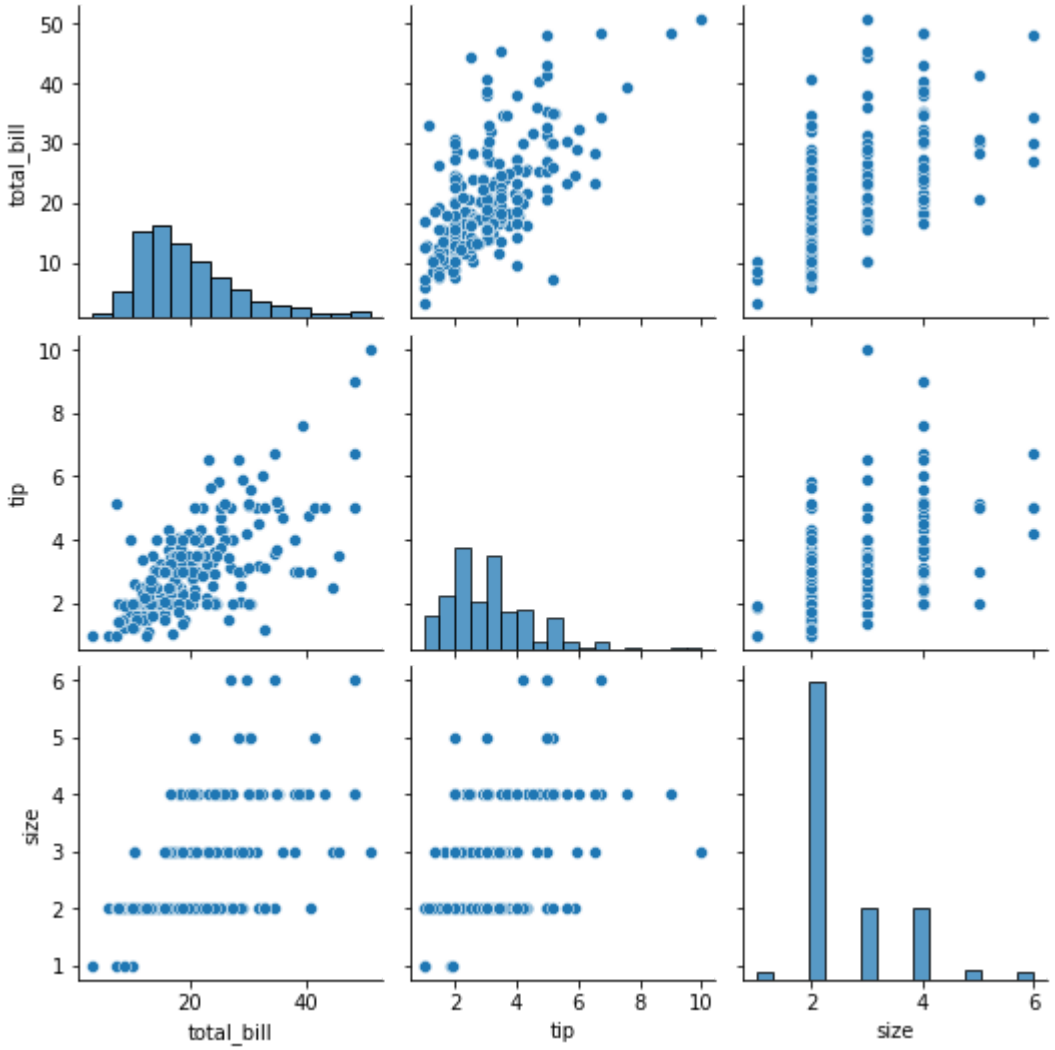


Pair Plot

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 elborations on this, showing all variables paired with all the other variables

In [16]:
```
sns.pairplot(df)
```

Out[16]:
```
<seaborn.axisgrid.PairGrid at 0x28303f85f70>
```



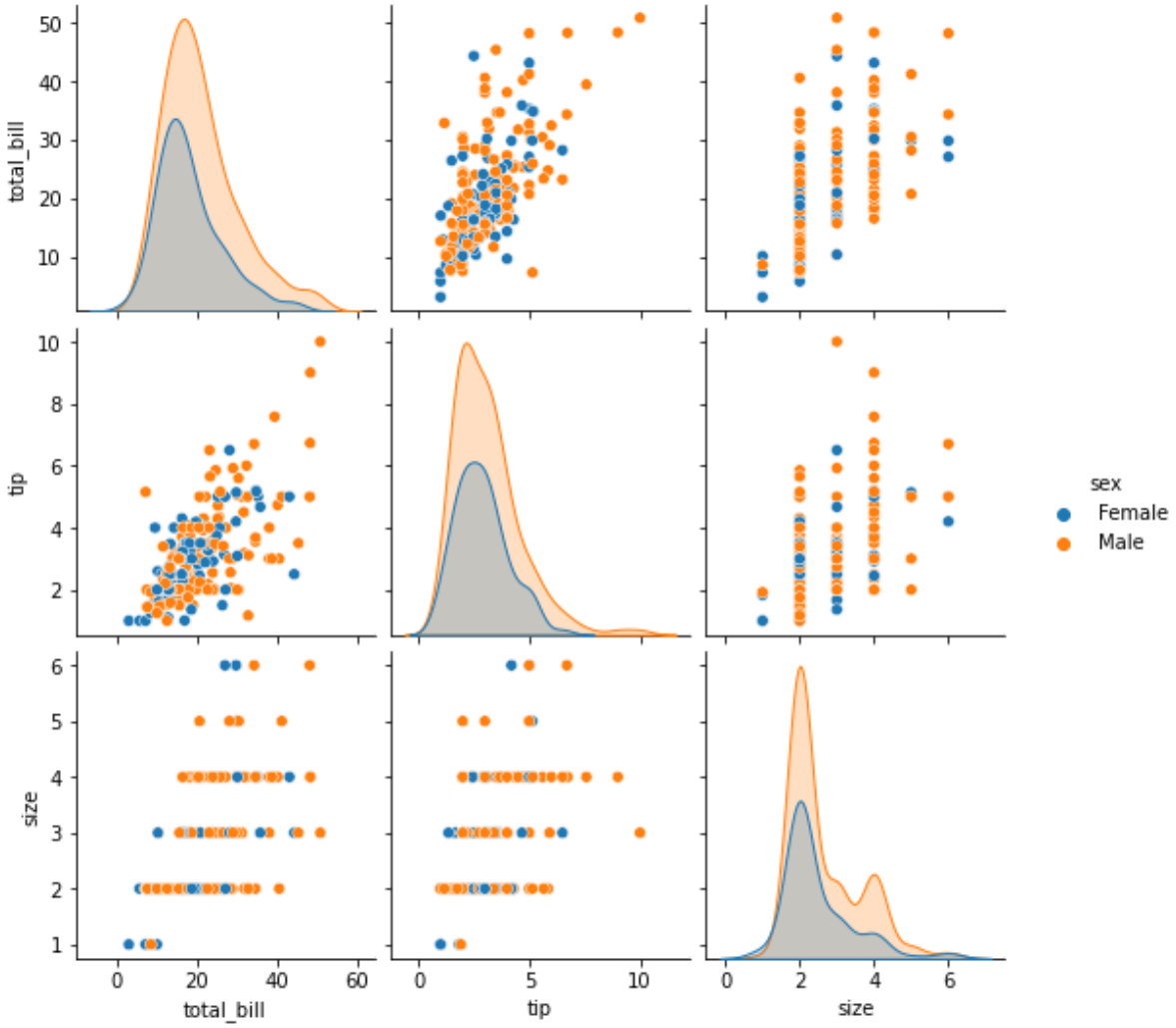
In [17]:
```
df.head(2)
```

Out[17]:

	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

In [18]:
```
sns.pairplot(df,hue='sex')
```

Out[18]:
```
<seaborn.axisgrid.PairGrid at 0x28303f85310>
```

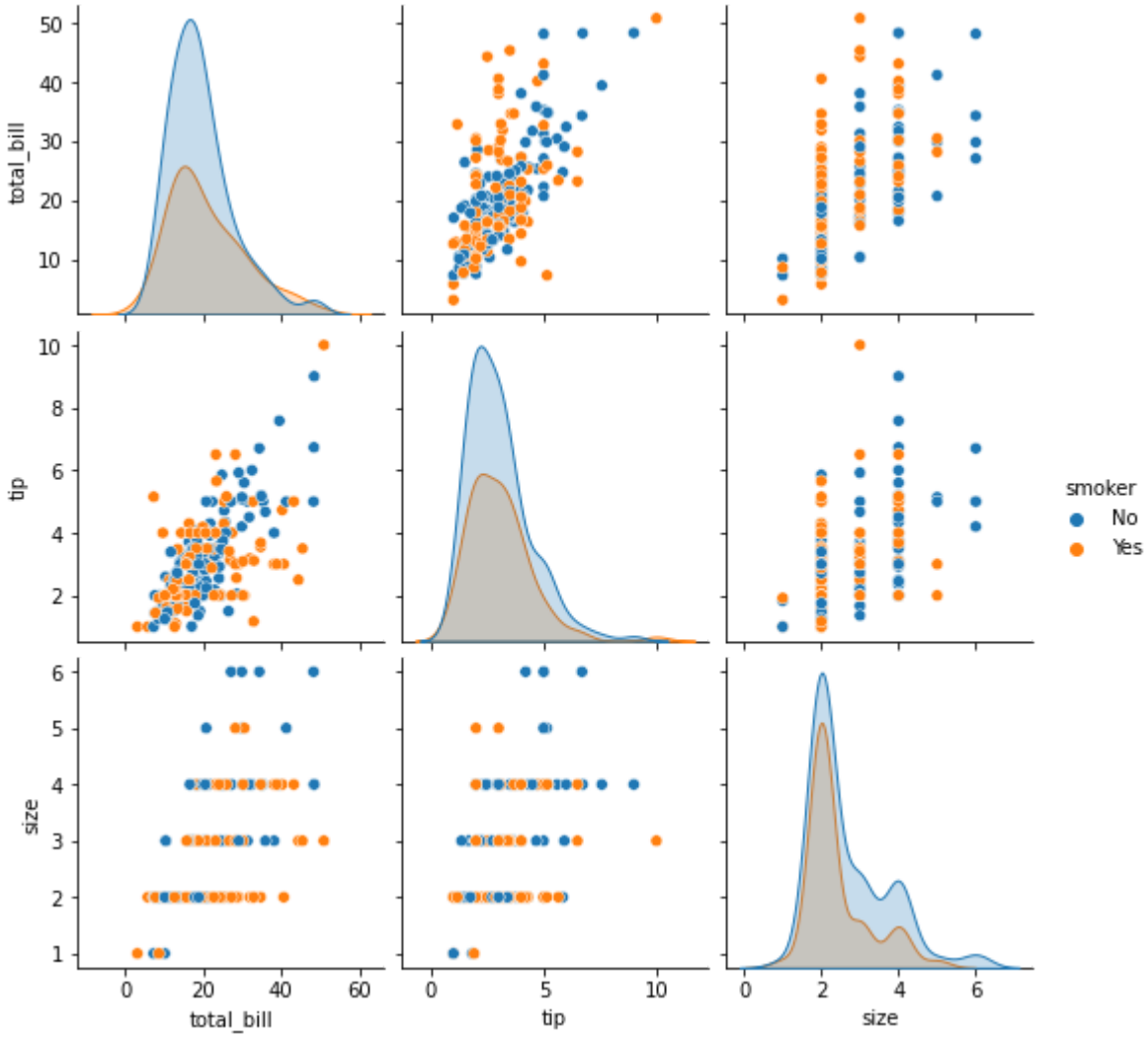


In [19]:
```
df['smoker'].value_counts()
```

Out[19]:
```
No     151
Yes     93
Name: smoker, dtype: int64
```

In [20]:
```
sns.pairplot(df,hue='smoker')
```

Out[20]:
```
<seaborn.axisgrid.PairGrid at 0x283065fcf40>
```



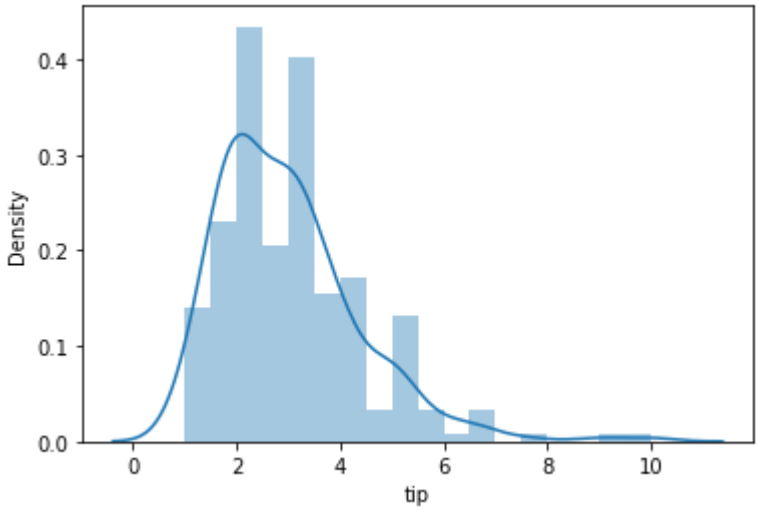
Dish Plot

Dish plot helps us to check the distribution of the columns features

In [21]:
```
sns.distplot(df['tip'])
```

```
E:\anaconda\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `
distplot` is a deprecated function and will be removed in a future version. P
lease adapt your code to use either `displot` (a figure-level function with s
imilar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

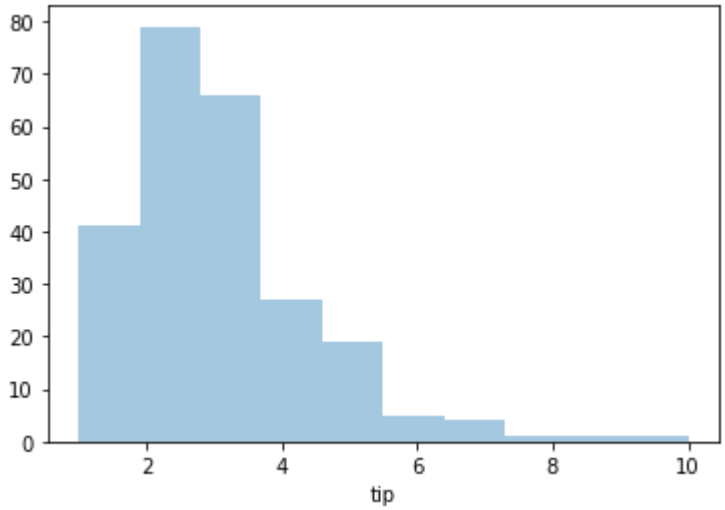
Out[21]:
```
<AxesSubplot:xlabel='tip', ylabel='Density'>
```



In [22]:
```
sns.distplot(df['tip'],kde=False,bins=10)  ## kernal density estimation
```

```
E:\anaconda\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `
distplot` is a deprecated function and will be removed in a future version. P
lease adapt your code to use either `displot` (a figure-level function with s
imilar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[22]:
```
<AxesSubplot:xlabel='tip'>
```



Categorial Plots Seaborn also helps us in doing the analysis on categorical Data points. In this section we will discuss about

. boxplot

. violinplot

. countplot

. bar plot

In [24]:
```
df.head(2)
```

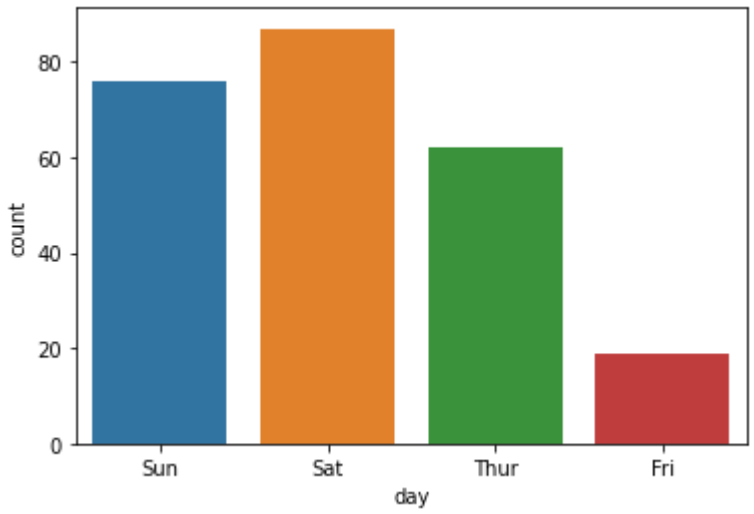
Out[24]:

	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

In [26]:
```
## count plot

sns.countplot('day',data=df)   ## can be used sex,smoker day inplace of day
```

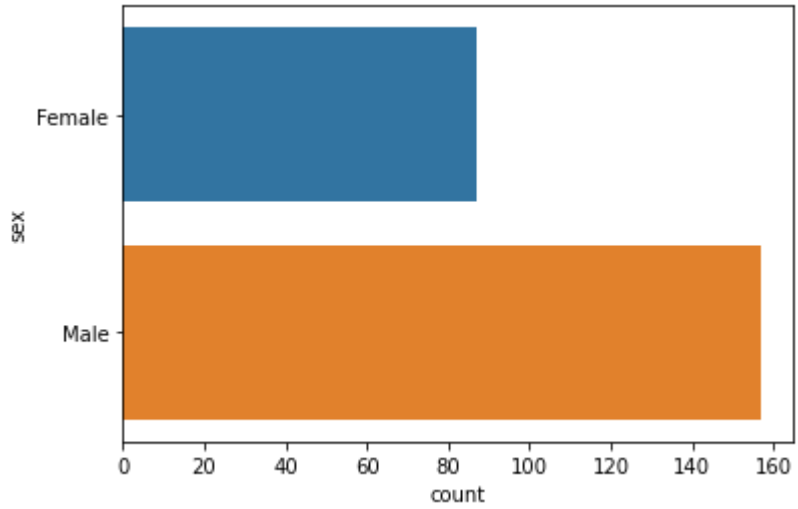
```
E:\anaconda\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass
the following variable as a keyword arg: x. From version 0.12, the only valid
positional argument will be `data`, and passing other arguments without an ex
plicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[26]:
```
<AxesSubplot:xlabel='day', ylabel='count'>
```



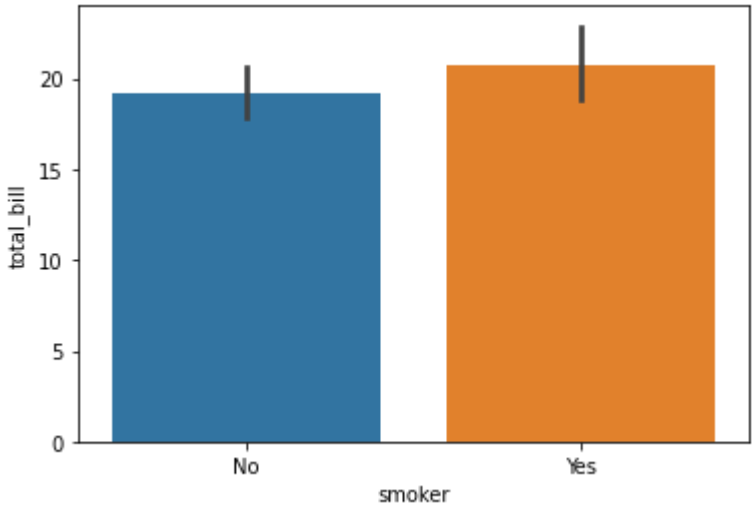
In [29]:
```
sns.countplot(y='sex',data=df)
```

Out[29]:
```
<AxesSubplot:xlabel='count', ylabel='sex'>
```



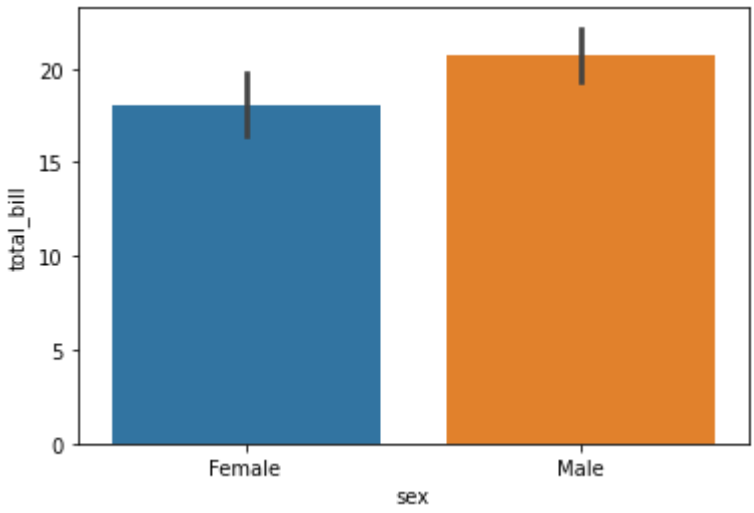
In [32]:
```
## Bar plot

sns.barplot(y='total_bill',x='smoker',data=df)  ## x and y used for horizontal
```

Out[32]:
```
<AxesSubplot:xlabel='smoker', ylabel='total_bill'>
```



In [33]:
```
## Bar plot
sns.barplot(x='sex',y='total_bill',data=df)
```

Out[33]:
```
<AxesSubplot:xlabel='sex', ylabel='total_bill'>
```



In [34]:
```
df.head(2)
```

Out[34]:

	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

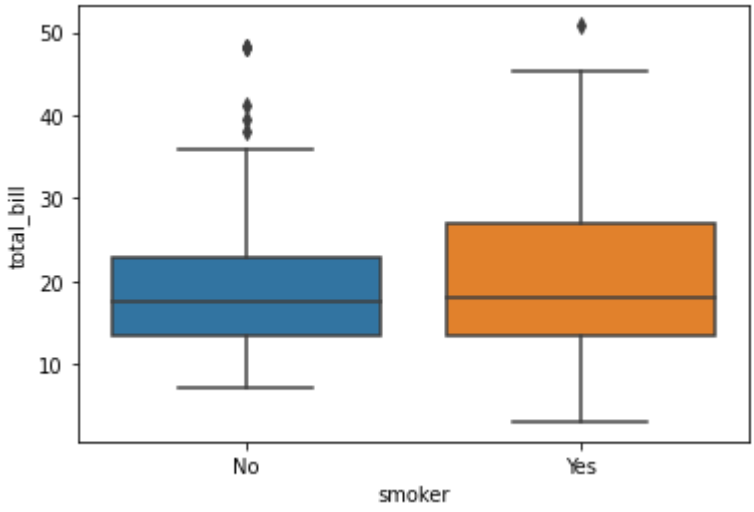
Box plot

A box and whisker plot(sometimes called a boxplot) is a graph that presents information from a five-number summary.

In [36]:
```
sns.boxplot('smoker','total_bill',data=df)  ## smoker x-axis and total_bill y-
```

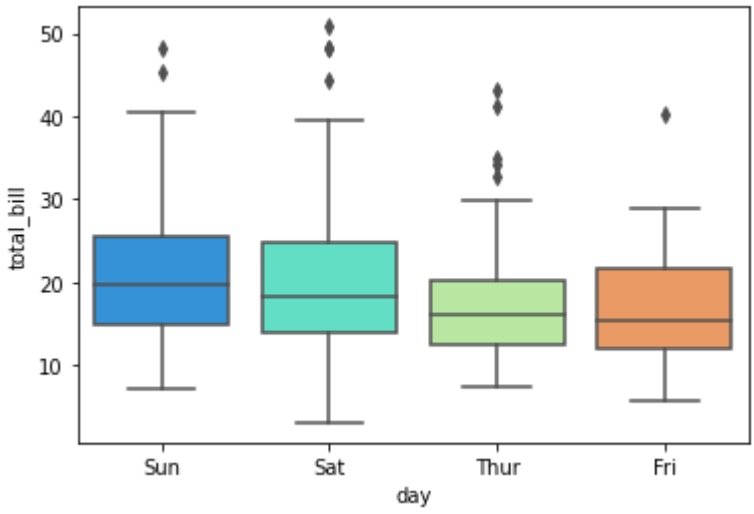
```
E:\anaconda\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass
the following variables as keyword args: x, y. From version 0.12, the only va
lid positional argument will be `data`, and passing other arguments without a
n explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[36]: <AxesSubplot:xlabel='smoker', ylabel='total_bill'>



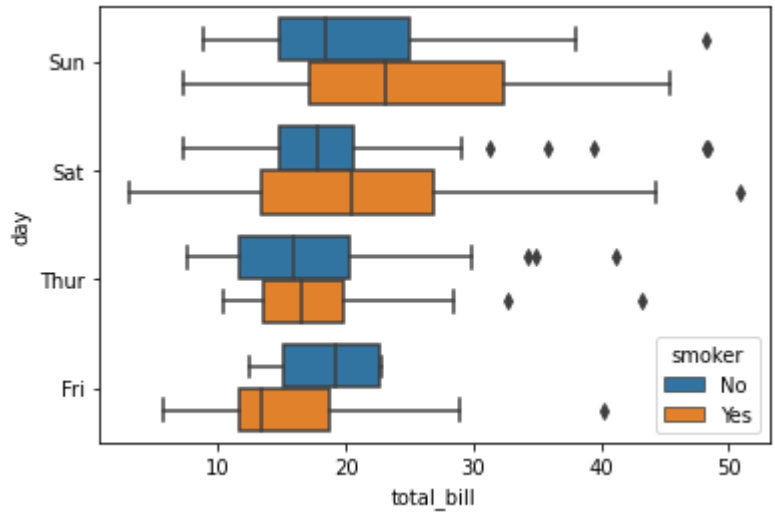
In [39]:
```
sns.boxplot(x='day',y='total_bill',data=df,palette='rainbow')
```

Out[39]: <AxesSubplot:xlabel='day', ylabel='total_bill'>



In [40]:
```
# categorize my data based on some other categories

sns.boxplot(x='total_bill',y='day',hue='smoker',data=df)
```

Out[40]:
```
<AxesSubplot:xlabel='total_bill', ylabel='day'>
```

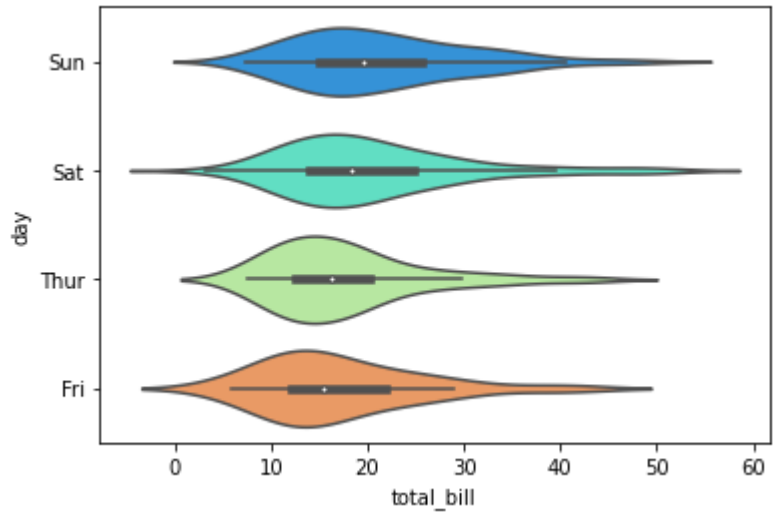


Violin plot

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

In [41]:
```
sns.violinplot(x='total_bill',y='day',data=df,palette='rainbow')
```

Out[41]:
```
<AxesSubplot:xlabel='total_bill', ylabel='day'>
```



In [42]:
```
## Practice homework

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

```
---------------------------------------------------------------------------
TimeoutError                              Traceback (most recent call last)
File E:\anaconda\lib\urllib\request.py:1346, in AbstractHTTPHandler.do_open
(self, http_class, req, **http_conn_args)
   1345 try:
-> 1346     h.request(req.get_method(), req.selector, req.data, headers,
   1347               encode_chunked=req.has_header('Transfer-encoding'))
   1348 except OSError as err: # timeout error

File E:\anaconda\lib\http\client.py:1285, in HTTPConnection.request(self, m
ethod, url, body, headers, encode_chunked)
   1284 """Send a complete request to the server."""
-> 1285 self._send_request(method, url, body, headers, encode_chunked)

File E:\anaconda\lib\http\client.py:1331, in HTTPConnection._send_request(s
elf, method, url, body, headers, encode_chunked)
   1330     body = _encode(body, 'body')
-> 1331 self.endheaders(body, encode_chunked=encode_chunked)
```

In [ ]: