5- Seaborn

November 25, 2025

0.1 Seaborn

0.2 Distribution plots

- distplot
- joinplot
- pairplot

[9]:
```
import seaborn as sns
```

[11]:
```
df=sns.load_dataset("tips")
```

[4]:
```
df.head()
```

[4]:

	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

0.2.1 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 event types. It is very important in Feature Selection

[45]:
```
df.corr()
```

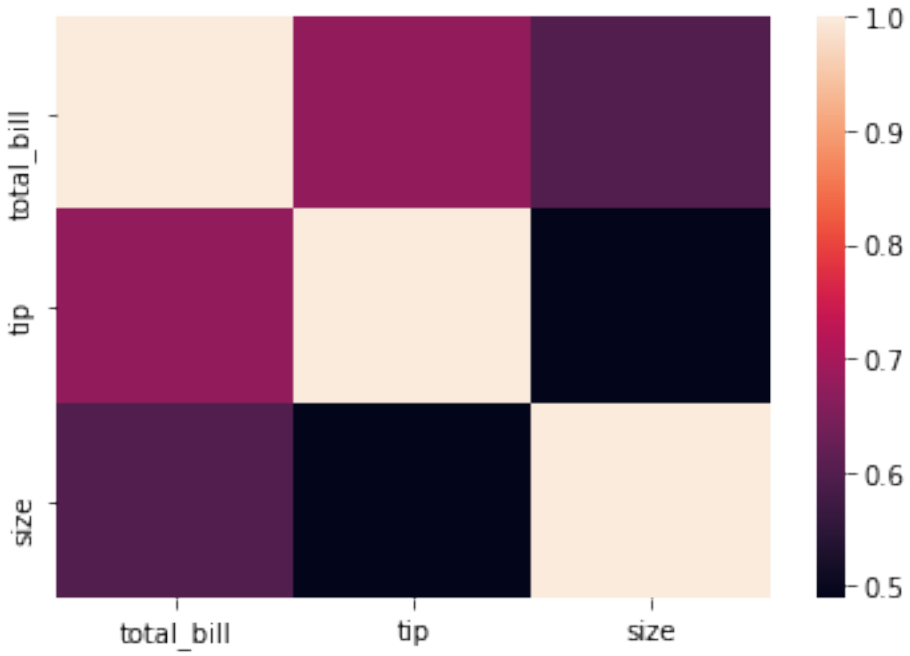
[45]:

	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

It calculates how strongly numerical columns in your DataFrame are related to each other. * Output: a table (matrix) * Values range from -1 to +1

[47]:
```
sns.heatmap(df.corr())
```

[47]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x26d290442e8>
```

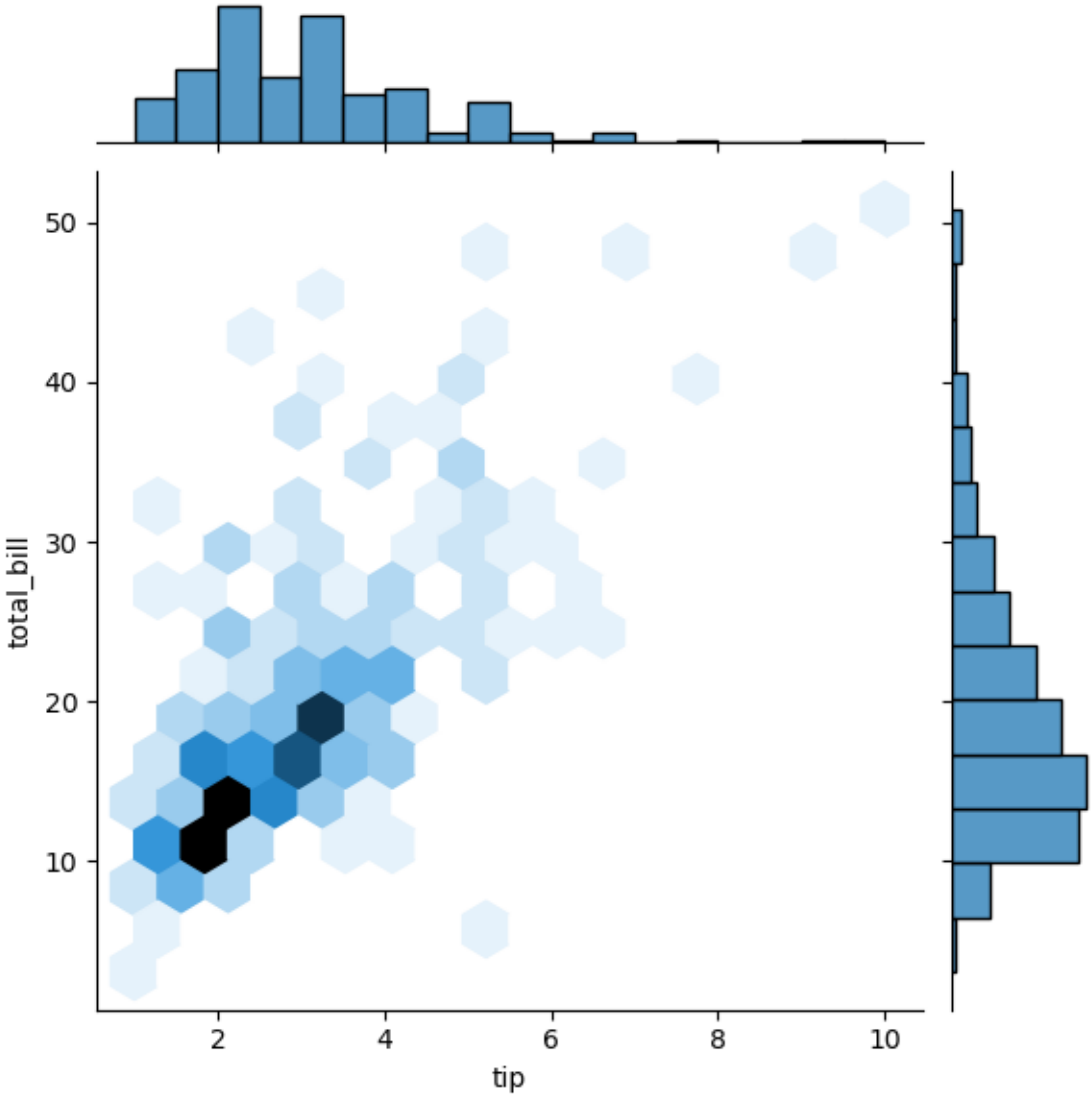


0.3 JoinPlot / scatterplot

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 ### Univariate Analysis

[21]:
```
sns.jointplot(x='tip',y='total_bill',data=df,kind='hex') #Other optionsfor kind:
↪ 'scatter', 'kde', 'reg', 'resid'.
```

[21]:
```
<seaborn.axisgrid.JointGrid at 0x16ab84c80>
```

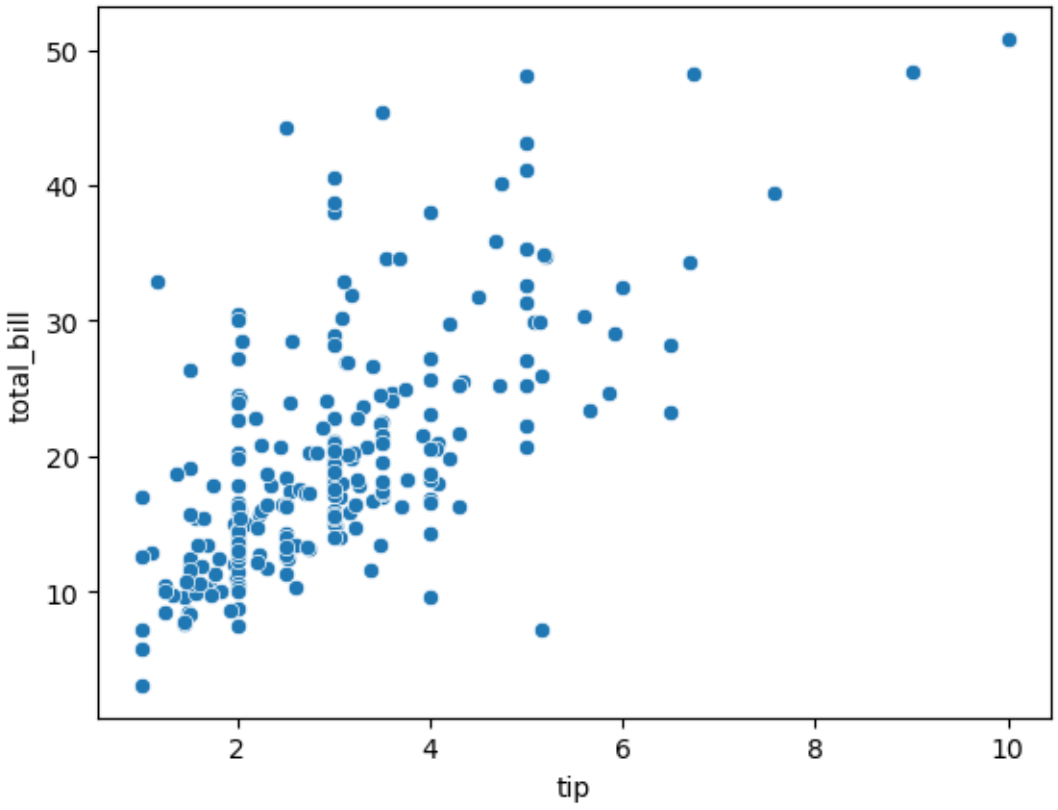


0.4 Features of a scatter plot:

- Each point represents one row in the dataset
- Shows the relationship between tip and total_bill

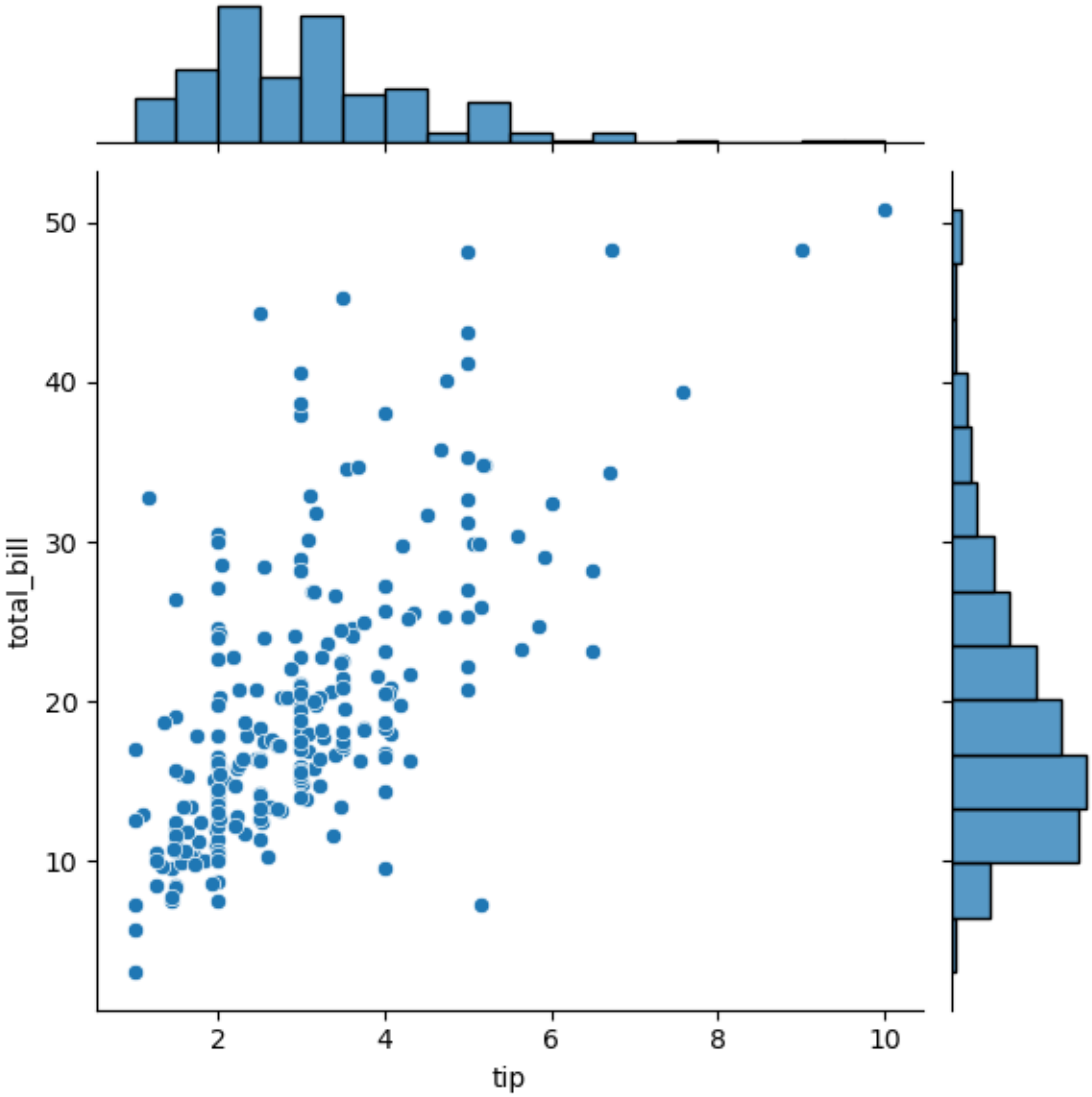
[17]:
```
# Scatter plot
sns.scatterplot(x='tip', y='total_bill', data=df)
```

[17]:
```
<Axes: xlabel='tip', ylabel='total_bill'>
```



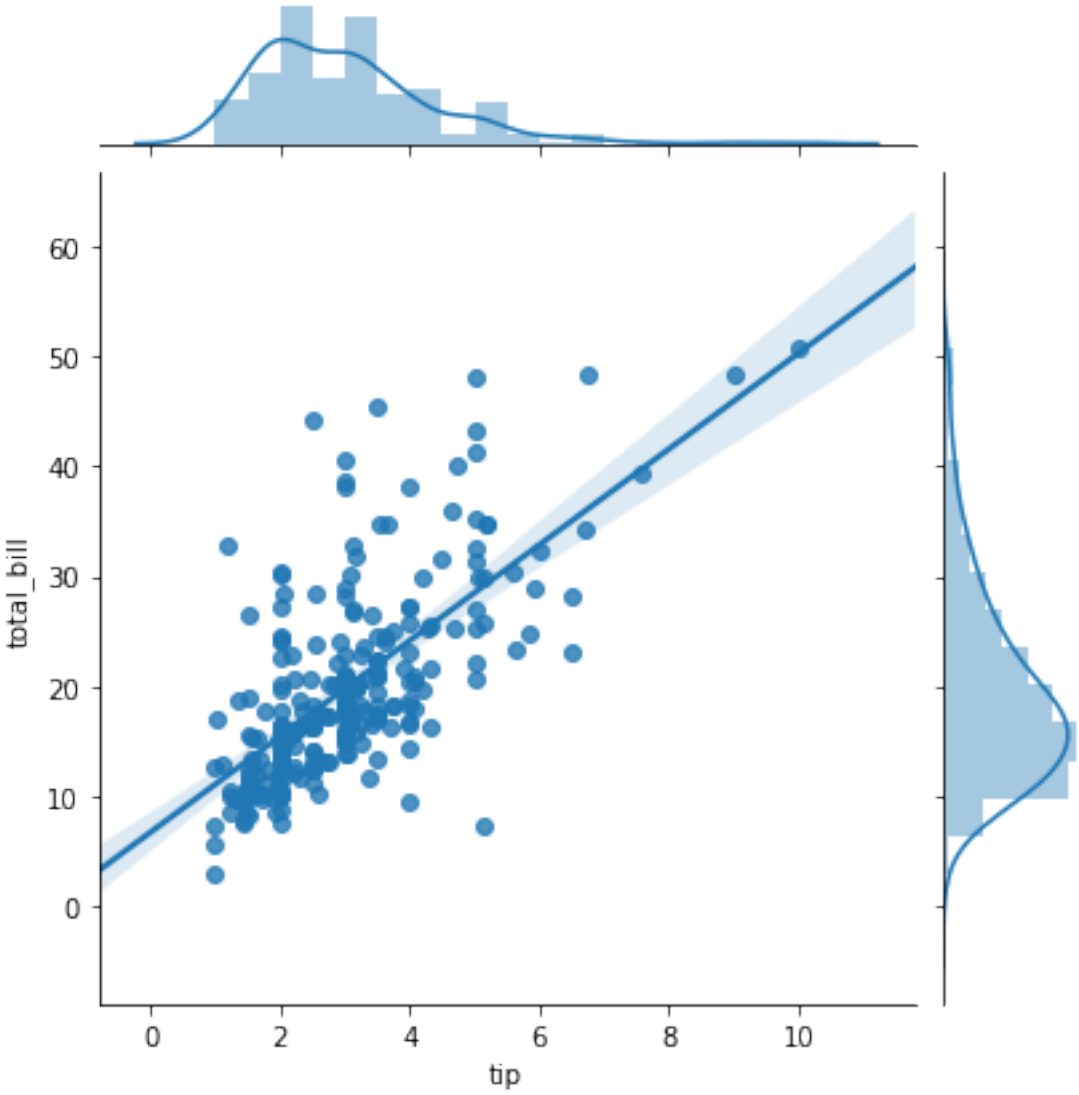
[19]:
```
sns.jointplot(x='tip', y='total_bill', data=df, kind='scatter')
```

[19]: <seaborn.axisgrid.JointGrid at 0x16a98fe60>



[21]:
```
sns.jointplot(x='tip',y='total_bill',data=df,kind='reg')
```

[21]: <seaborn.axisgrid.JointGrid at 0x26d268a2780>

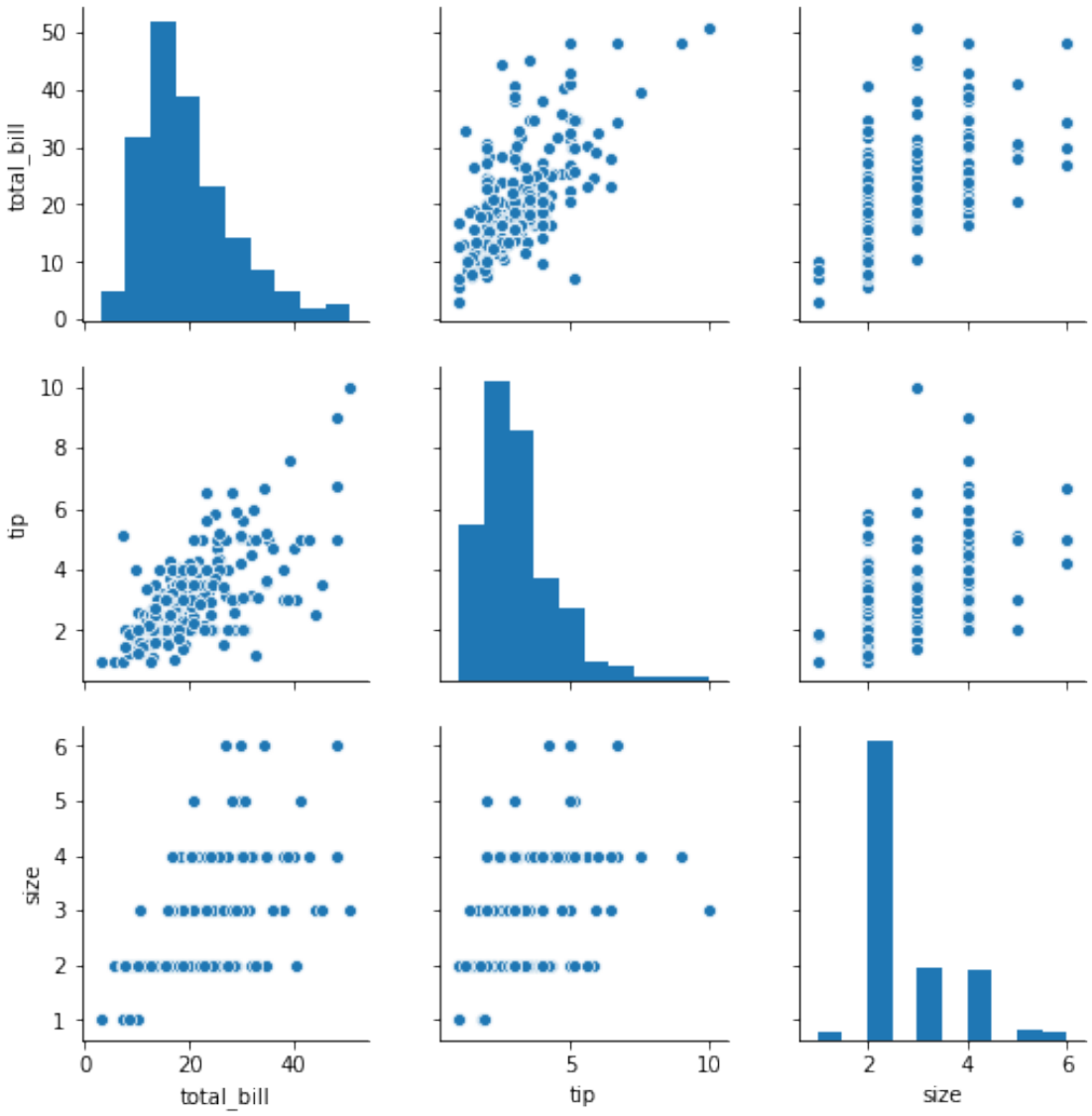


0.5 Pair plot

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

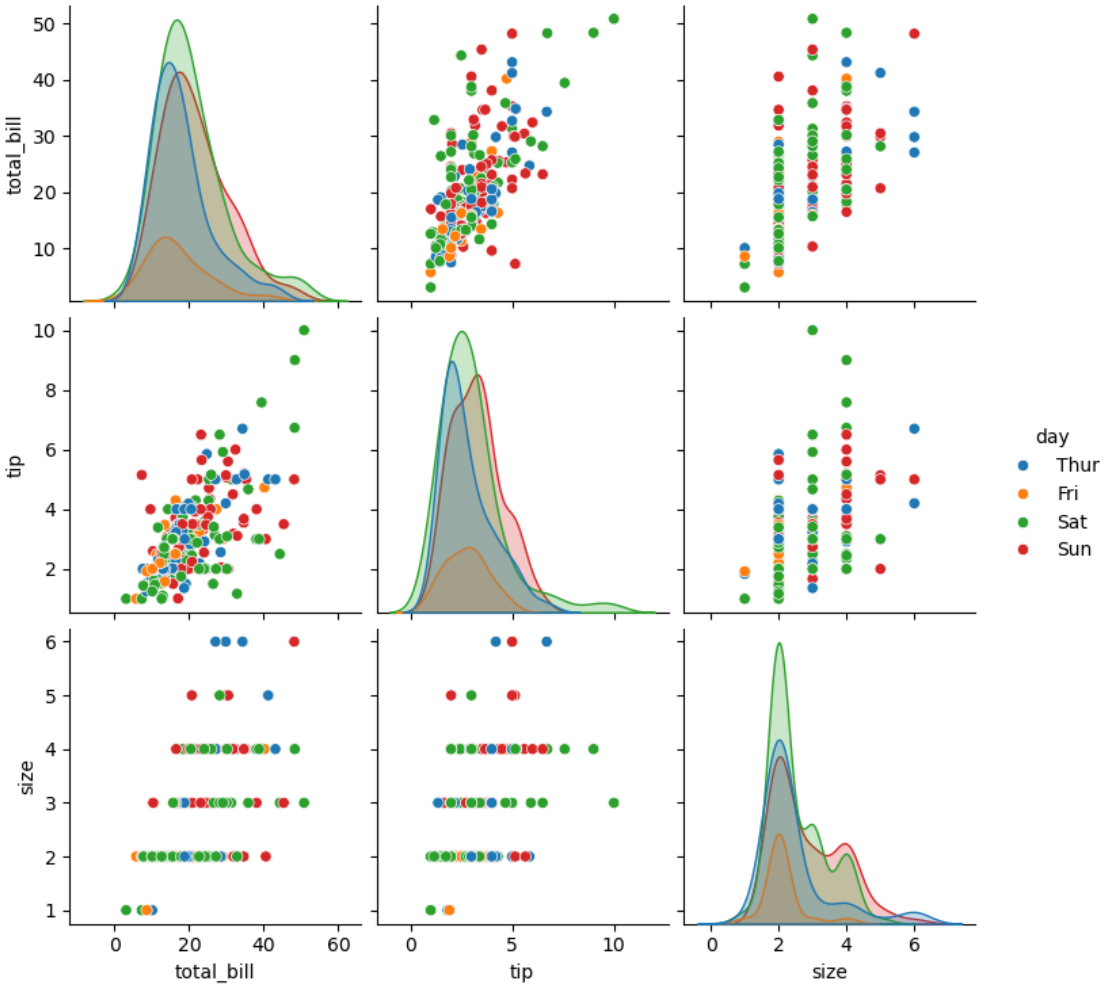
[8]:
```
sns.pairplot(df)
```

[8]:
```
<seaborn.axisgrid.PairGrid at 0x26d24010630>
```



[26]:
```
sns.pairplot(df,hue='day')
```

[26]: <seaborn.axisgrid.PairGrid at 0x16ab84a10>



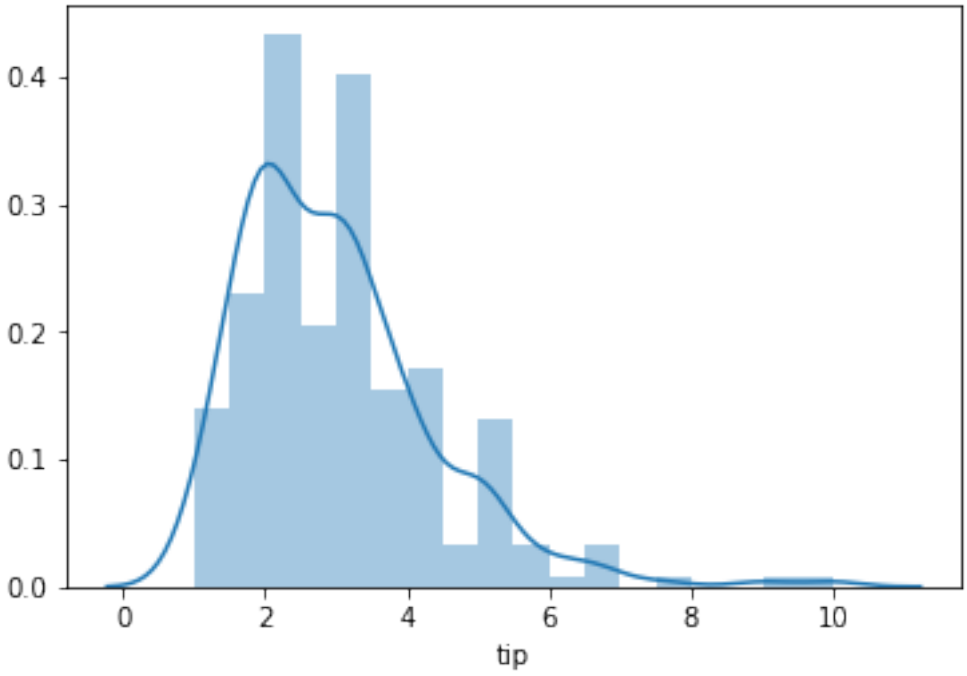
0.6 Dist plot

Dist plot helps us to check the distribution of the columns feature

[11]:
```
sns.distplot(df['tip'])
```

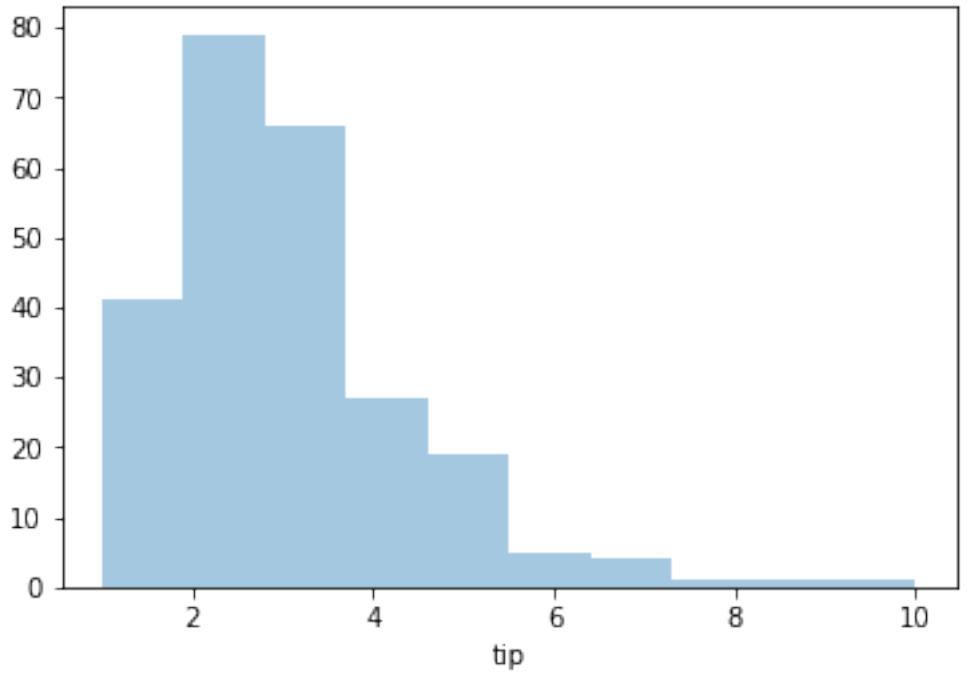
```
C:\Users\krish.naik\AppData\Local\Continuum\anaconda3\envs\myenv\lib\site-
packages\scipy\stats\stats.py:1713: FutureWarning: Using a non-tuple sequence
for multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of
`arr[seq]`. In the future this will be interpreted as an array index,
`arr[np.array(seq)]`, which will result either in an error or a different
result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```

[11]: `<matplotlib.axes._subplots.AxesSubplot at 0x26d24a32be0>`



[13]:
```
sns.distplot(df['tip'],kde=False,bins=10)
```

[13]: <matplotlib.axes._subplots.AxesSubplot at 0x26d24aec668>



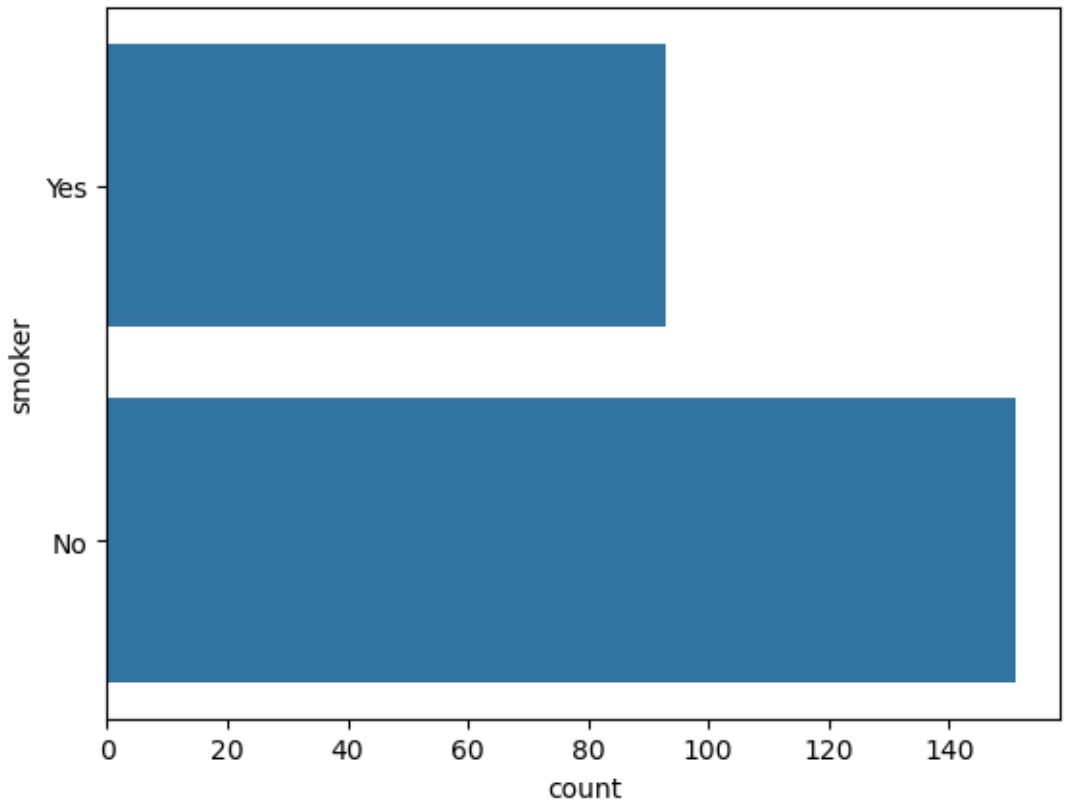
0.7 Categorical Plots

Seaborn also helps us in doing the analysis on Categorical Data points.

- boxplot
- violinplot
- countplot
- bar plot

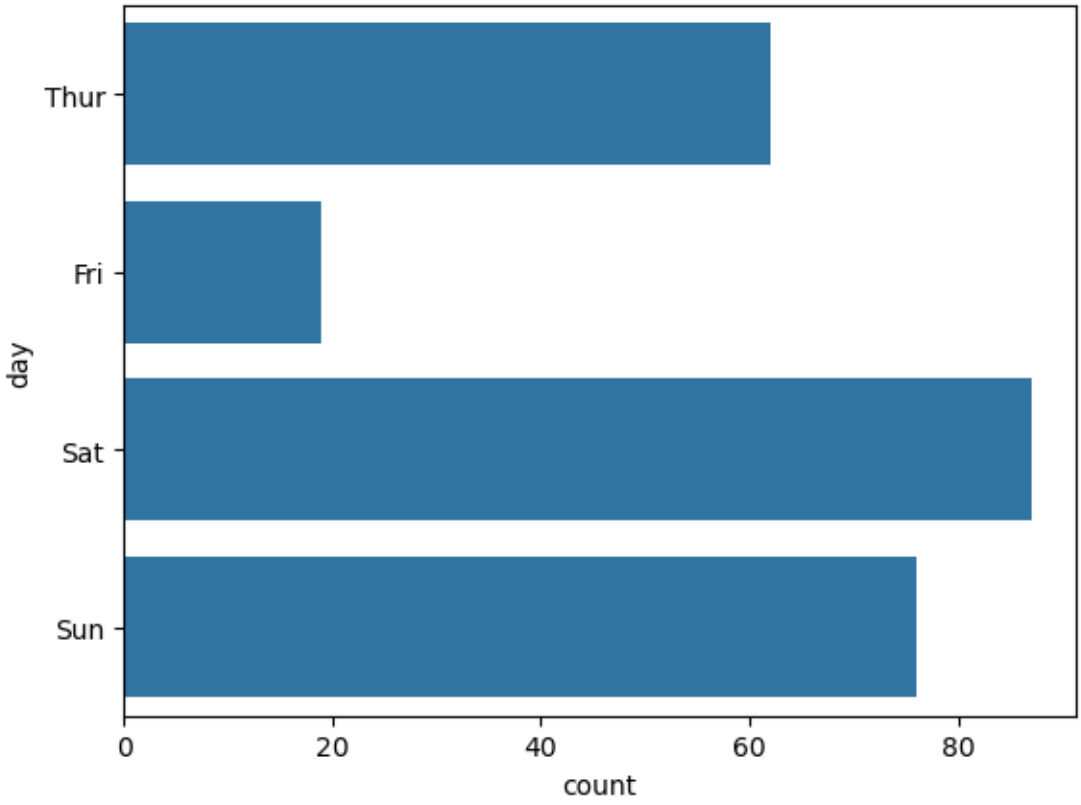
[38]:
```
## Count plot

sns.countplot(y='smoker',data=df)
```

[38]:
```
<Axes: xlabel='count', ylabel='smoker'>
```



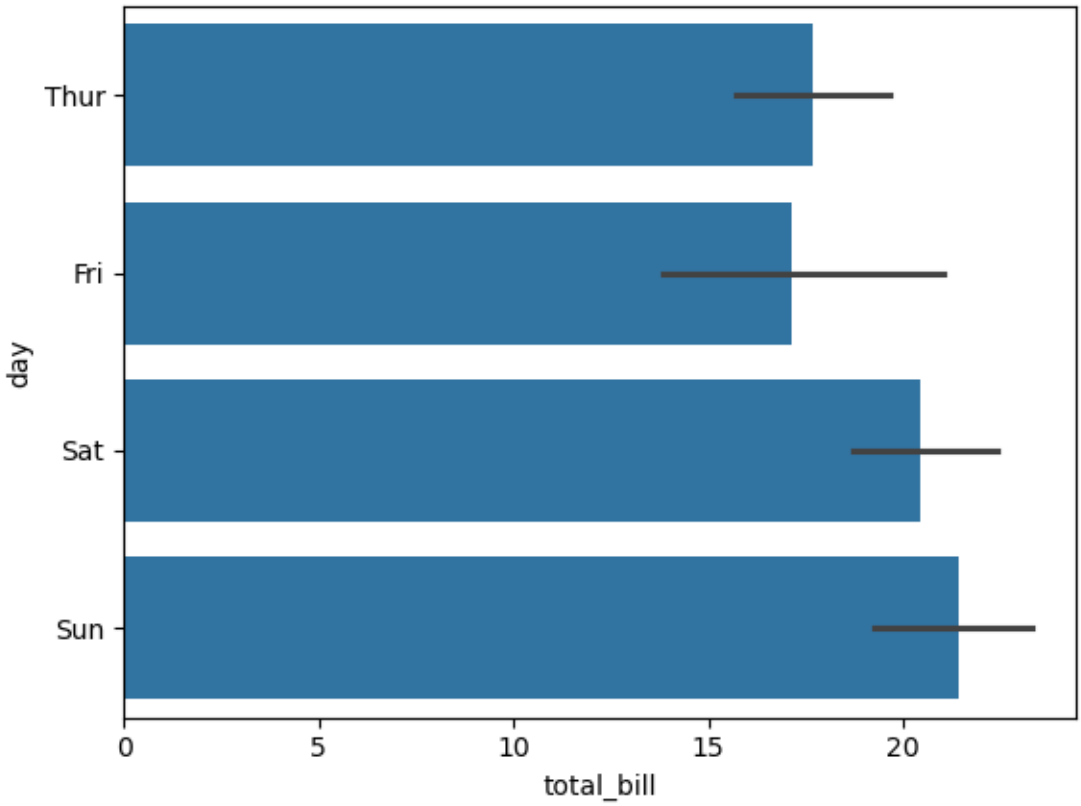
[34]:
```
## Count plot

sns.countplot(y='day',data=df)
```

[34]: <Axes: xlabel='count', ylabel='day'>



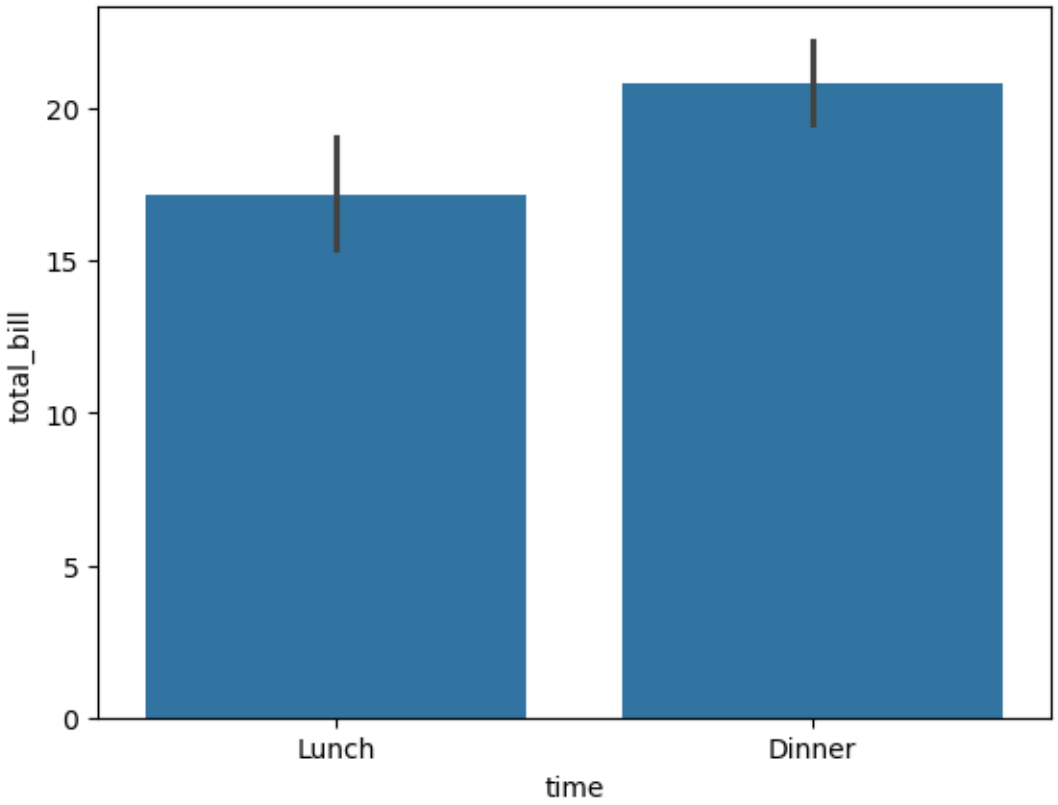
[40]:
```
## Bar plot
sns.barplot(x='total_bill',y='day',data=df)
```

[40]: <Axes: xlabel='total_bill', ylabel='day'>



[44]:
```
## Bar plot
sns.barplot(x='time',y='total_bill',data=df)
```

[44]:
```
<Axes: xlabel='time', ylabel='total_bill'>
```



[30]:
```
df.head()
```

[30]:

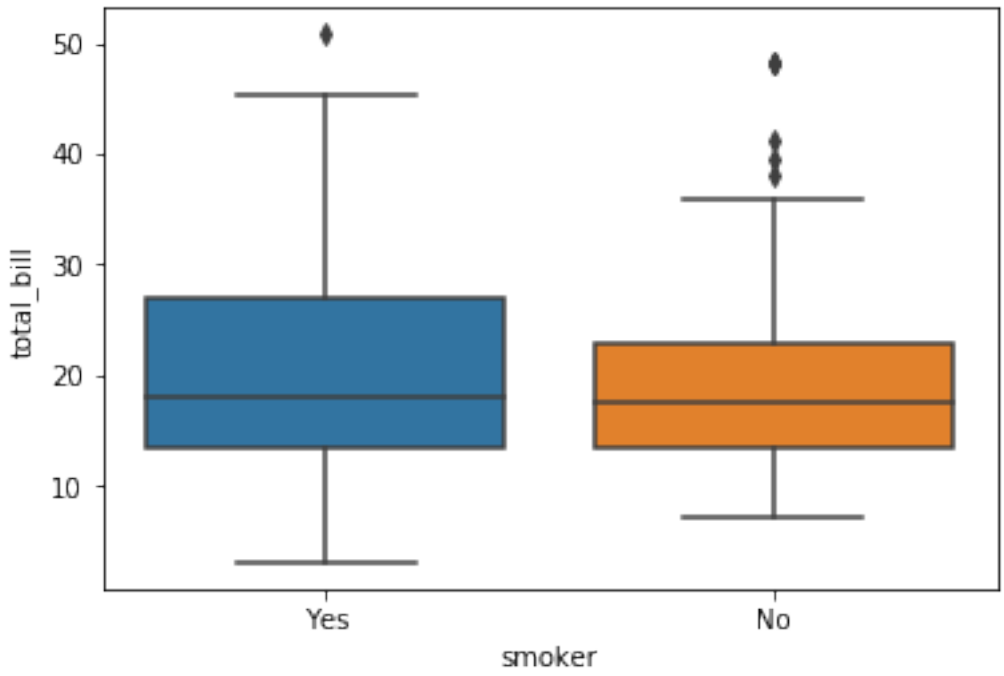
	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

0.8 Box plot

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

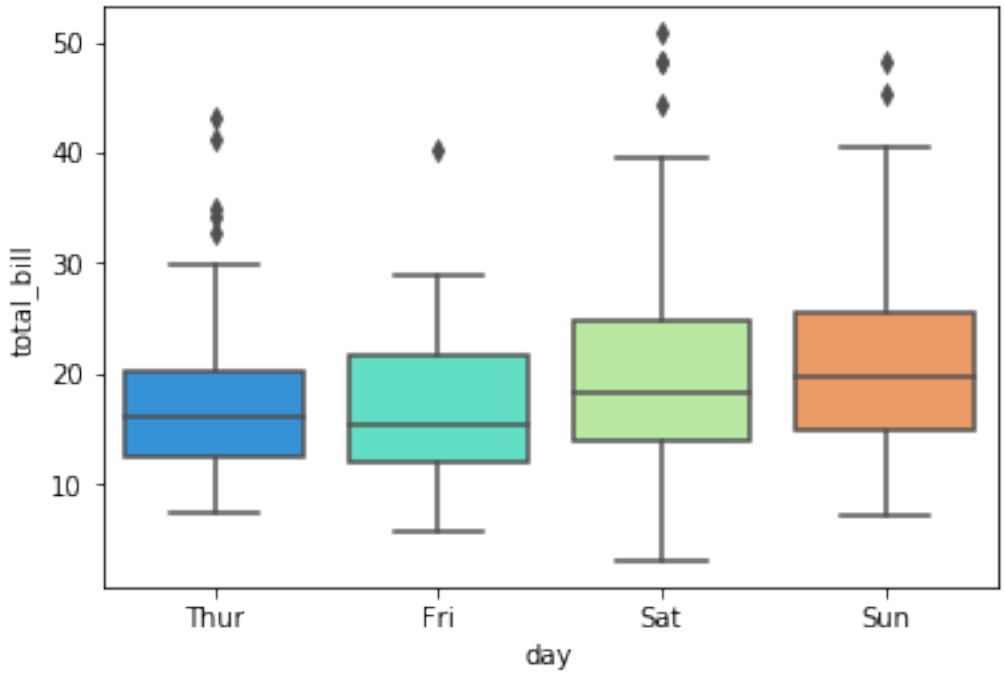
[31]:
```
sns.boxplot('smoker','total_bill', data=df)
```

[31]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x26d287f7d30>
```



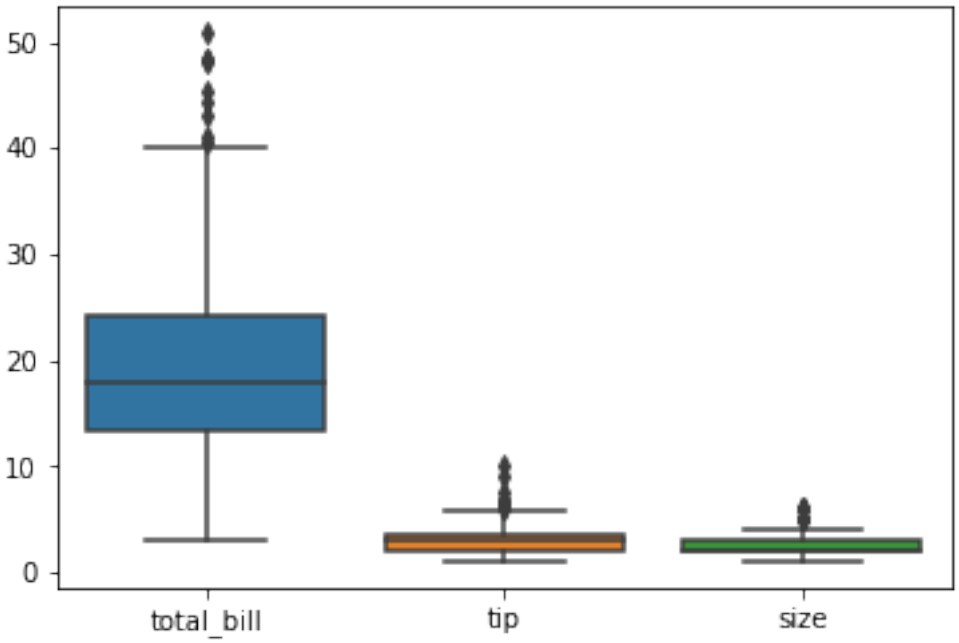
[33]:
```
sns.boxplot(x="day", y="total_bill", data=df,palette='rainbow')
```

[33]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x26d2885f940>
```



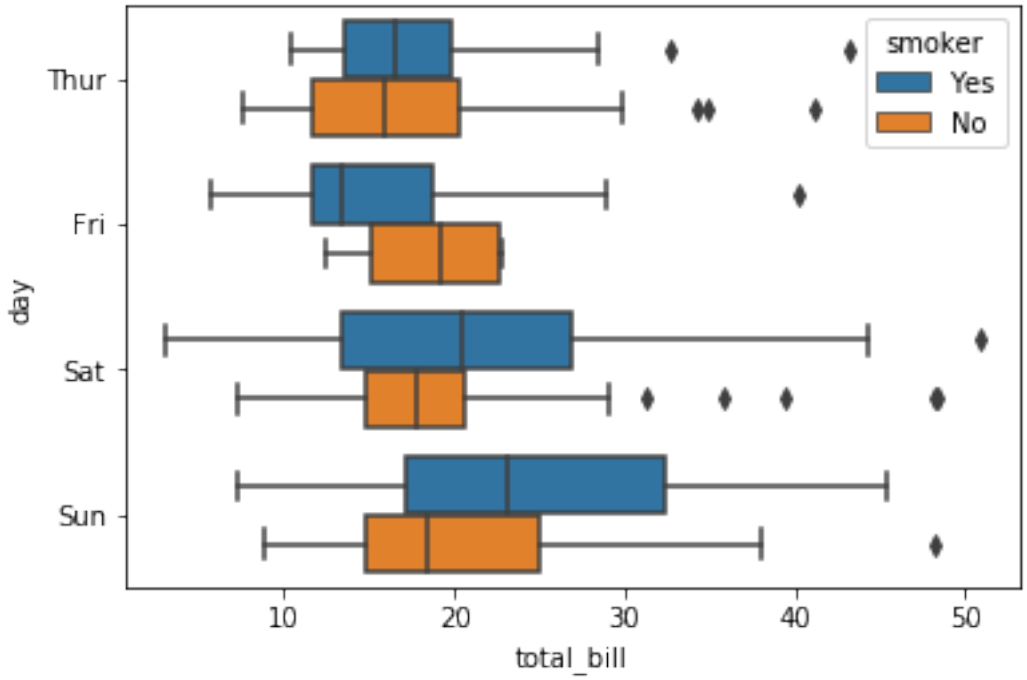
[37]:
```
sns.boxplot(data=df,orient='v')
```

[37]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x26d28b23748>
```



[39]:
```
# categorize my data based on some other categories

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

[39]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x26d28c6ecf8>
```

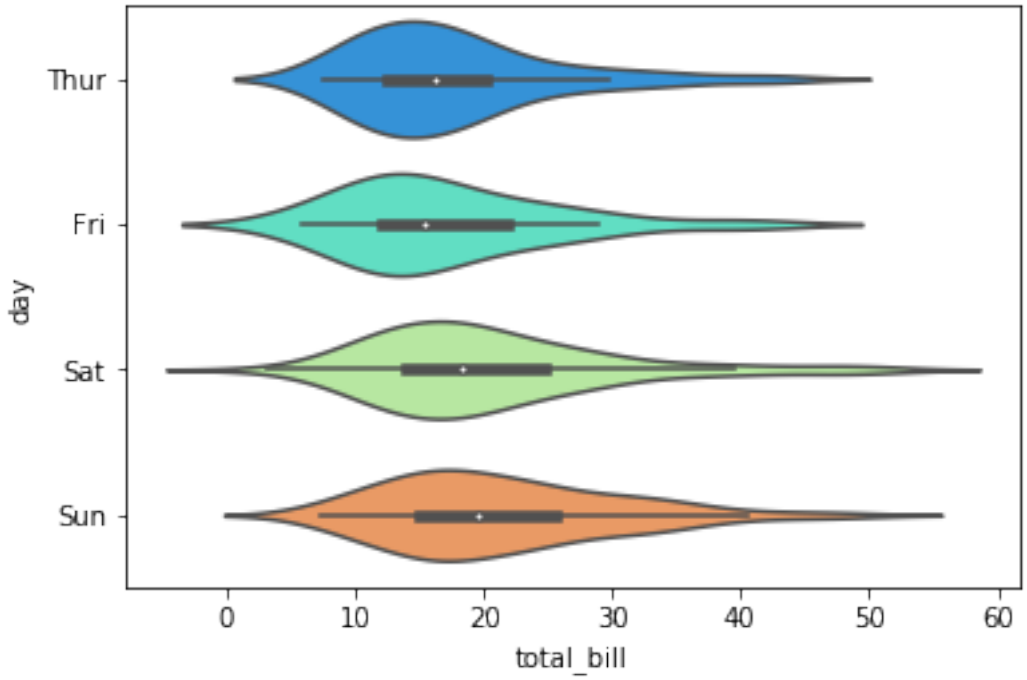


0.9 Violin Plot

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

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

[41]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x26d28d5ca20>
```



[ ]:
```
## Practise Homework: use iris dataset and plot all graphs

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

[ ]:
```

```