EDA - Exploratory Data anlysis can be done using Univariate analysis, bivariate analysis and multivariate analysis

If we have any data, what we will ask it first and how we do EDA

Lets use Famous Titanic data

Step 1: Import libraries

In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

Step 2: Import data and ask atleast 7 Question to data

In [2]:
```
df=pd.read_csv(r"C:\Users\USER\Downloads\train.csv")
df.head()
```

Out[2]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

Question

In [3]:
```
# 1. How big is the data?     - Rows,Column
df.shape
```

Out[3]:
```
(891, 12)
```

In [4]:
```
# 2. How does the data look like?
df.head()
```

Out[4]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

In [5]:
```
df.tail()
```

Out[5]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.00	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.00	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.45	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.00	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.75	NaN	Q

In [6]:
```
df.sample(5)
```

Out[6]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
267	268	1	3	Persson, Mr. Ernst Ulrik	male	25.0	1	0	347083	7.7750	NaN	S
771	772	0	3	Jensen, Mr. Niels Peder	male	48.0	0	0	350047	7.8542	NaN	S
331	332	0	1	Partner, Mr. Austen	male	45.5	0	0	113043	28.5000	C124	S
494	495	0	3	Stanley, Mr. Edward Roland	male	21.0	0	0	A/4 45380	8.0500	NaN	S
857	858	1	1	Daly, Mr. Peter Denis	male	51.0	0	0	113055	26.5500	E17	S

In [7]:
```
# 3. What is the data type of cols?  - int means numerical, object means categorical, float means numerical wit
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  891 non-null    int64  
 1   Survived     891 non-null    int64  
 2   Pclass       891 non-null    int64  
 3   Name         891 non-null    object 
 4   Sex          891 non-null    object 
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64  
 7   Parch        891 non-null    int64  
 8   Ticket       891 non-null    object 
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object 
 11  Embarked     889 non-null    object 
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

In [8]:
```
# 3. What is the data type of cols?   - Null or zero values or not available
df.isnull()
```

Out[8]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	False	False	False	False	False	False	False	False	False	False	True	False
1	False	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	True	False
3	False	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	True	False
...	...	...	...	...	...	...	...	...	...	...	...	...
886	False	False	False	False	False	False	False	False	False	False	True	False
887	False	False	False	False	False	False	False	False	False	False	False	False
888	False	False	False	False	False	True	False	False	False	False	True	False
889	False	False	False	False	False	False	False	False	False	False	False	False
890	False	False	False	False	False	False	False	False	False	False	True	False

891 rows × 12 columns

In [9]:
```
df.isnull().sum()
```

Out[9]:
```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

In [10]:
```
df.isna().sum()
```

Out[10]:
```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

In [11]:
```
# 5. How does the data look statistically?
df.describe()
```

Out[11]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

In [12]:
```
# 6. Are there duplicate values?
df.duplicated().sum()
```

Out[12]:
```
0
```

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

Out[13]:
```
0      False
1      False
2      False
3      False
4      False
       ...
886    False
887    False
888    False
889    False
890    False
Length: 891, dtype: bool
```

In [14]:
```
# 7. How is the correlation between cols?
df.corr()    # this give detail correlation without focusing with single column against other column
```

```
C:\Users\USER\AppData\Local\Temp\ipykernel_1384\2604216460.py:2: FutureWarning: The default value of numeric_on
ly in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns or
specify the value of numeric_only to silence this warning.
  df.corr()    # this give detail correlation without focusing with single column against other column
```

Out[14]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
PassengerId	1.000000	-0.005007	-0.035144	0.036847	-0.057527	-0.001652	0.012658
Survived	-0.005007	1.000000	-0.338481	-0.077221	-0.035322	0.081629	0.257307
Pclass	-0.035144	-0.338481	1.000000	-0.369226	0.083081	0.018443	-0.549500
Age	0.036847	-0.077221	-0.369226	1.000000	-0.308247	-0.189119	0.096067
SibSp	-0.057527	-0.035322	0.083081	-0.308247	1.000000	0.414838	0.159651
Parch	-0.001652	0.081629	0.018443	-0.189119	0.414838	1.000000	0.216225
Fare	0.012658	0.257307	-0.549500	0.096067	0.159651	0.216225	1.000000

In [15]:
```
# using single column comparing with other
# what is relation of survived passenger with other column or what is relationship
df.corr()['Survived']
```

```
C:\Users\USER\AppData\Local\Temp\ipykernel_1384\2840766514.py:3: FutureWarning: The default value of numeric_on
ly in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns or
specify the value of numeric_only to silence this warning.
  df.corr()['Survived']
```

Out[15]:
```
PassengerId   -0.005007
Survived       1.000000
Pclass        -0.338481
Age           -0.077221
SibSp         -0.035322

Parch          0.081629
Fare           0.257307
Name: Survived, dtype: float64
```

UNIVARIATE ANALYSIS

First of all we have to think how we can simply look categorical and numerical data

Categorical data

In [16]:
```
df.head()
```

Out[16]:

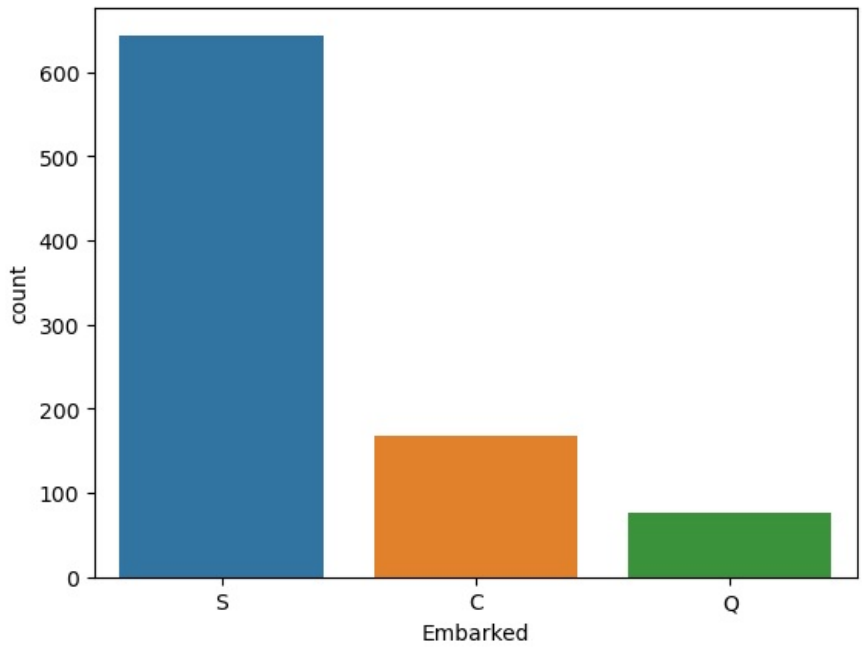
	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

In [17]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  891 non-null    int64
 1   Survived     891 non-null    int64
 2   Pclass       891 non-null    int64
 3   Name         891 non-null    object
 4   Sex          891 non-null    object
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64
 7   Parch        891 non-null    int64
 8   Ticket       891 non-null    object
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object
 11  Embarked     889 non-null    object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

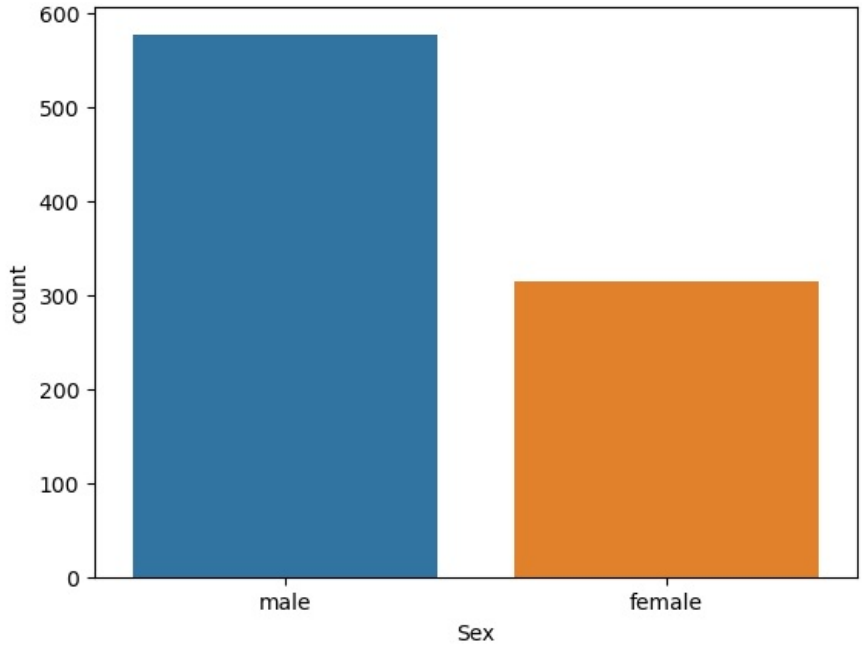
In [18]:
```
sns.countplot(x ='Embarked', data = df)  # Embarked is from which station passenger took the ship
```

Out[18]:
```
<Axes: xlabel='Embarked', ylabel='count'>
```



In [19]:
```
sns.countplot(x ='Sex', data = df)
```

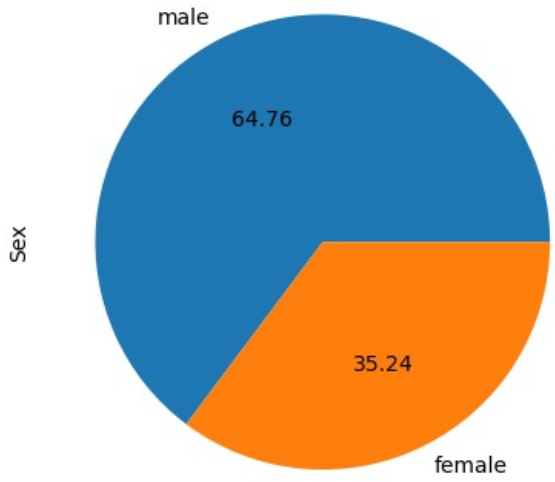
Out[19]:
```
<Axes: xlabel='Sex', ylabel='count'>
```



Categorical data - pie chart

In [20]:
```
df['Sex'].value_counts().plot(kind='pie',autopct='%.2f')
```

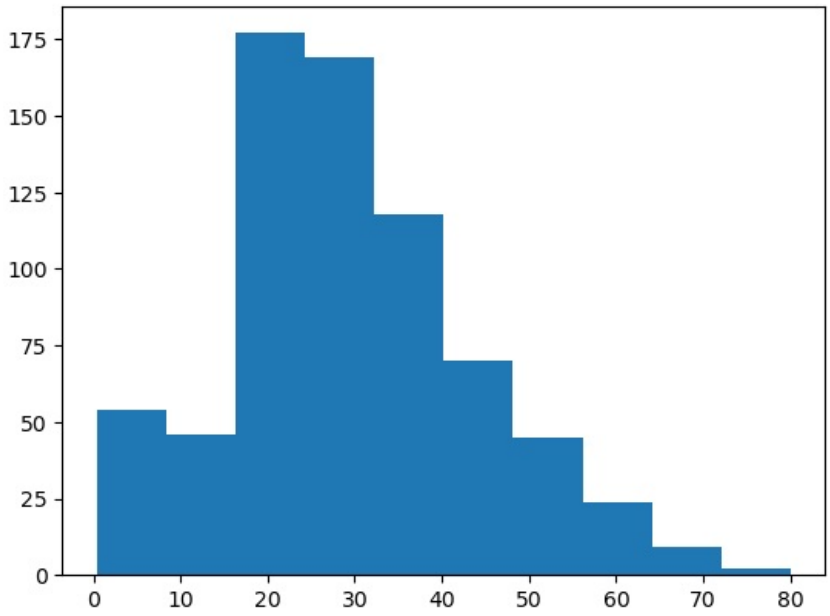
Out[20]:
```
<Axes: ylabel='Sex'>
```



Numerical column - Hiistogram and distplot

In [21]:
```
plt.hist(df['Age'])
```

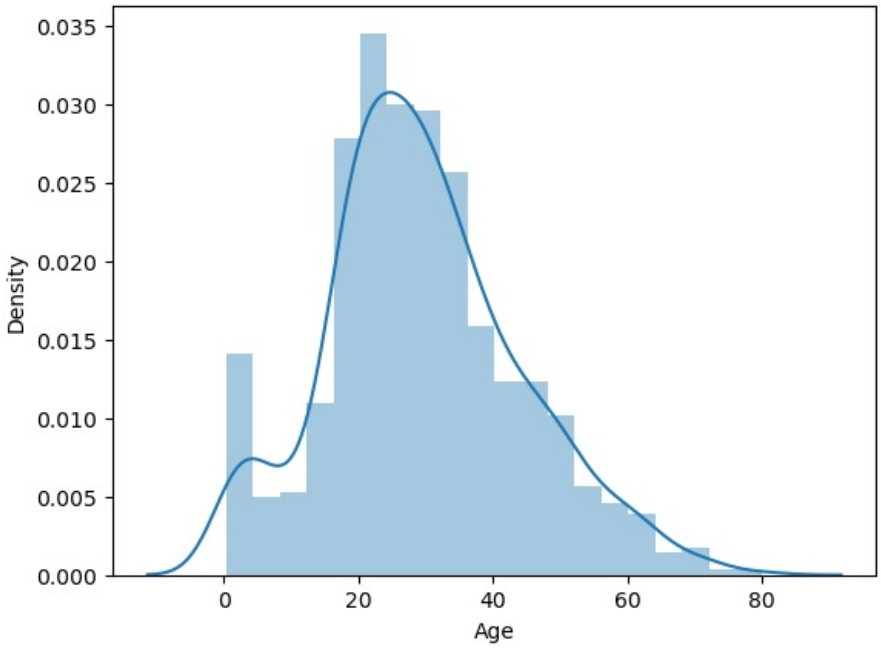
Out[21]:
```
(array([ 54.,  46., 177., 169., 118.,  70.,  45.,  24.,   9.,   2.]),
 array([ 0.42 ,  8.378, 16.336, 24.294, 32.252, 40.21 , 48.168, 56.126,
        64.084, 72.042, 80.   ]),
 <BarContainer object of 10 artists>)
```



In [22]:
```
sns.distplot(df['Age'])
```

```
C:\Users\USER\AppData\Local\Temp\ipykernel_1384\3255828239.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

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

Out[22]:
```
<Axes: xlabel='Age', ylabel='Density'>
```



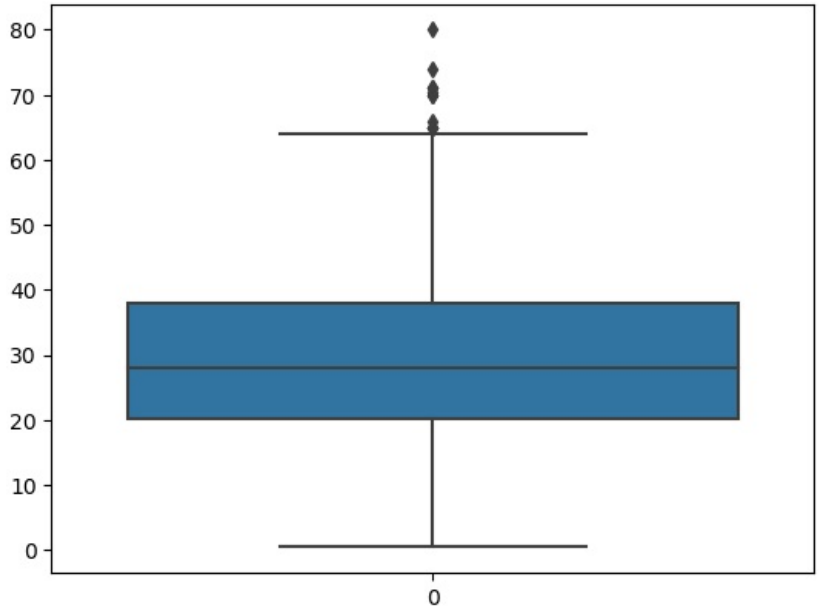
So, what is difference between histogram and distplot? what it says?

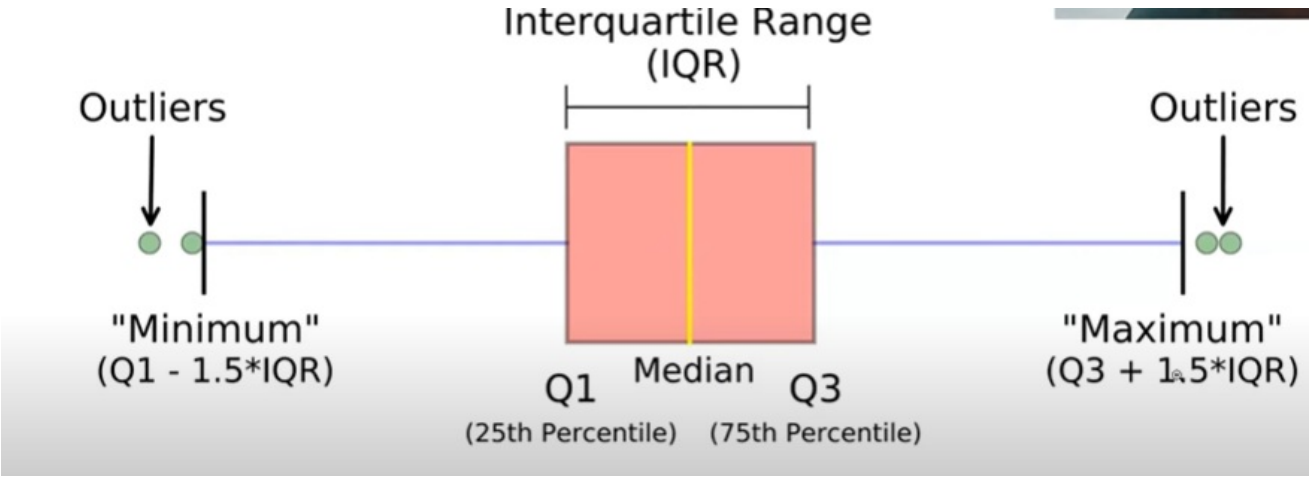
So, what is difference between histogram and distplot? What it says?

- In distplot we can see Kde too - curve like (Kernel density estimation) : distplot actually shows the probability, suppose there is probability of 0.5% that Titanic ship contain people with age 60. Histogram provide actual number like in that range how many people was there - like if we see above figure there is approx 20 people with age 60.

In [23]:
```
# Box plot - It give 5 number summary - it give median (50% percentile), 25% value (Q1),75% value (Q3),
# Maximum and minimum value

sns.boxplot(df['Age'])
```

Out[23]:
```
<Axes: >
```





$$Q1 = \text{Data point at } \frac{n+1}{4}$$

$$Q2 = \text{Data point at } \frac{n+1}{2}$$

$$Q3 = \text{Data point at } \frac{3(n+1)}{4}$$

EDA using Bivariate and Multivariate

Lets load required datasets and libraries, I have loaded 5 datasets just to look Bivariate and multivariate analysis in different dataset

In [24]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

In [25]:
```
titanic=pd.read_csv(r"C:\Users\USER\Downloads\train.csv")
iris=pd.read_csv(r"C:\Users\USER\Downloads\iris.csv")
tips=pd.read_csv(r"C:\Users\USER\Downloads\tips.csv")
airport=pd.read_csv(r"C:\Users\USER\Downloads\Los_Angeles_International_Airport_-_Passenger_Traffic_By_Terminal
```

Lets look galnce of datasets

In [26]:
```
titanic.head()
```

Out[26]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

In [27]:
```
iris.tail()
```

Out[27]:

	sepal.length	sepal.width	petal.length	petal.width	variety
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

In [28]:
```
tips.sample(5)
```

Out[28]:

	total_bill	tip	sex	smoker	day	time	size
129	22.82	2.18	Male	No	Thur	Lunch	3
102	44.30	2.50	Female	Yes	Sat	Dinner	3
68	20.23	2.01	Male	No	Sat	Dinner	2
37	16.93	3.07	Female	No	Sat	Dinner	3
242	17.82	1.75	Male	No	Sat	Dinner	2

In [29]:
```
airport.head() # its about LA airport
```

Out[29]:

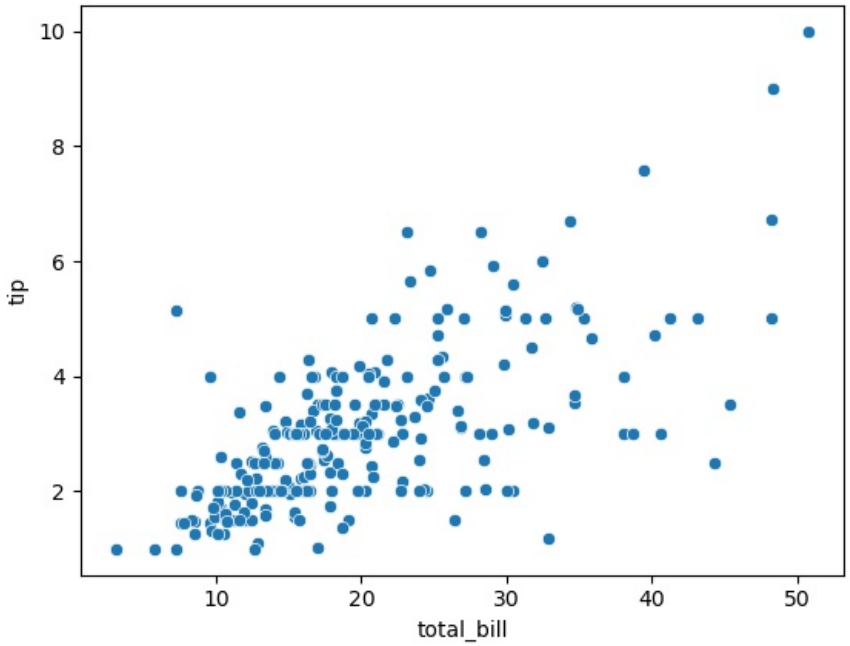
	DataExtractDate	ReportPeriod	Terminal	Arrival_Departure	Domestic_International	Passenger_Count
0	05/10/2021 06:01:09 AM	04/01/2021 12:00:00 AM	T1	Departure	Domestic	160413
1	05/03/2021 03:08:02 PM	03/01/2021 12:00:00 AM	T5	Departure	Domestic	223866
2	05/27/2021 03:16:34 PM	04/01/2021 12:00:00 AM	T5	Departure	Domestic	266035
3	07/10/2021 06:01:27 AM	06/01/2021 12:00:00 AM	T6	Arrival	International	6195
4	05/10/2021 06:01:09 AM	04/01/2021 12:00:00 AM	T8	Arrival	Domestic	54925

Scatterplot - Numerical/Numerical

Bivariate Analysis - Analysis between two column

In [30]:
```
sns.scatterplot(x=tips['total_bill'],y=tips['tip'])
```

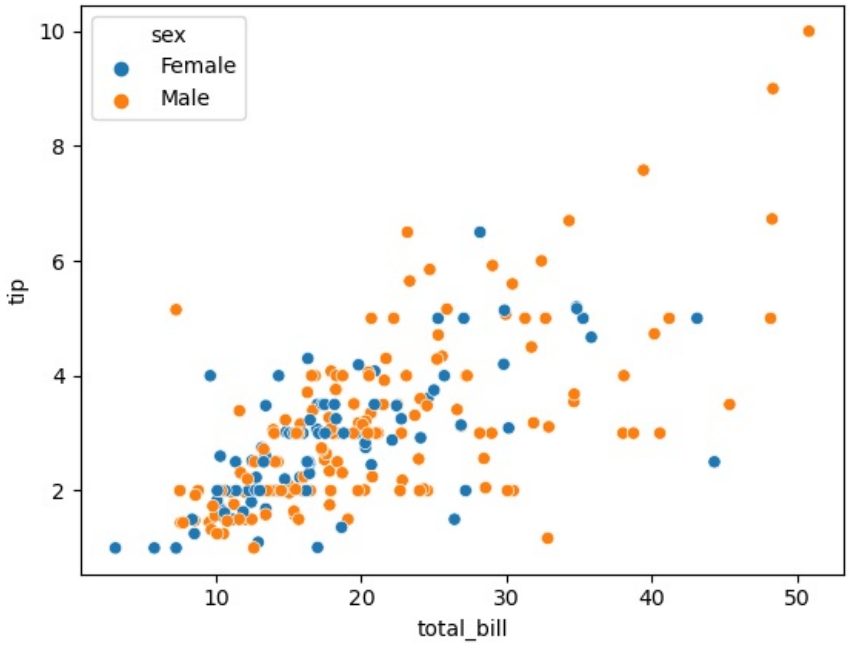
Out[30]:
```
<Axes: xlabel='total_bill', ylabel='tip'>
```



Above Scatterplot showing somewhat linear relationship between tip and total bill - high total bill high tip.

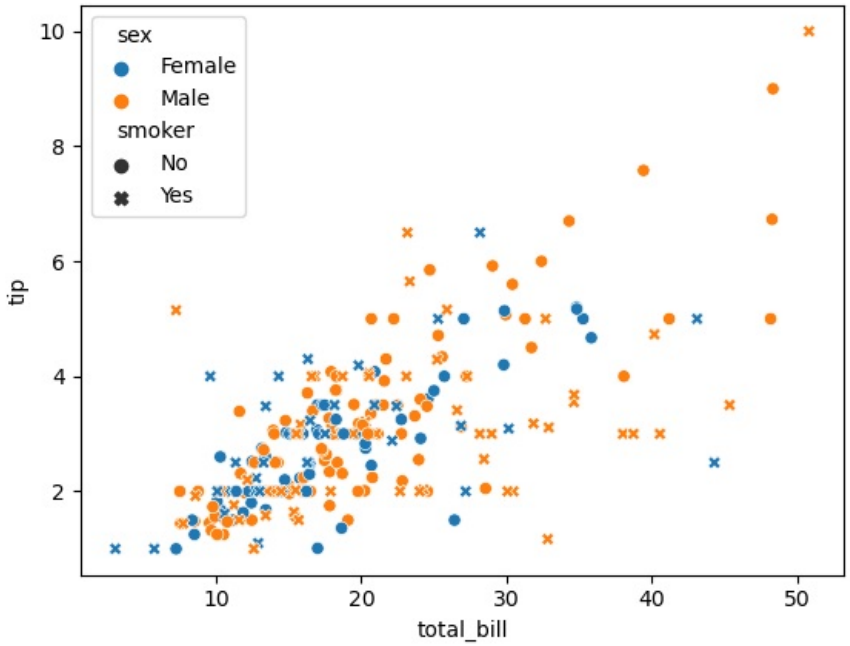
In [31]:
```
# Now lets see multivariate analysis in same datasets, lets find out which customer is male and which is female

sns.scatterplot(x=tips['total_bill'],y=tips['tip'], hue=tips['sex'])
```

Out[31]:
```
<Axes: xlabel='total_bill', ylabel='tip'>
```



In [32]:
```
#lets look another parameter too

sns.scatterplot(x=tips['total_bill'],y=tips['tip'], hue=tips['sex'], style=tips['smoker'])
```

Out[32]:
```
<Axes: xlabel='total_bill', ylabel='tip'>
```



Barplot - Numerical/Categorical

In [33]:
```
# lets use titanic dataset, but first lets look column
titanic.info()
```

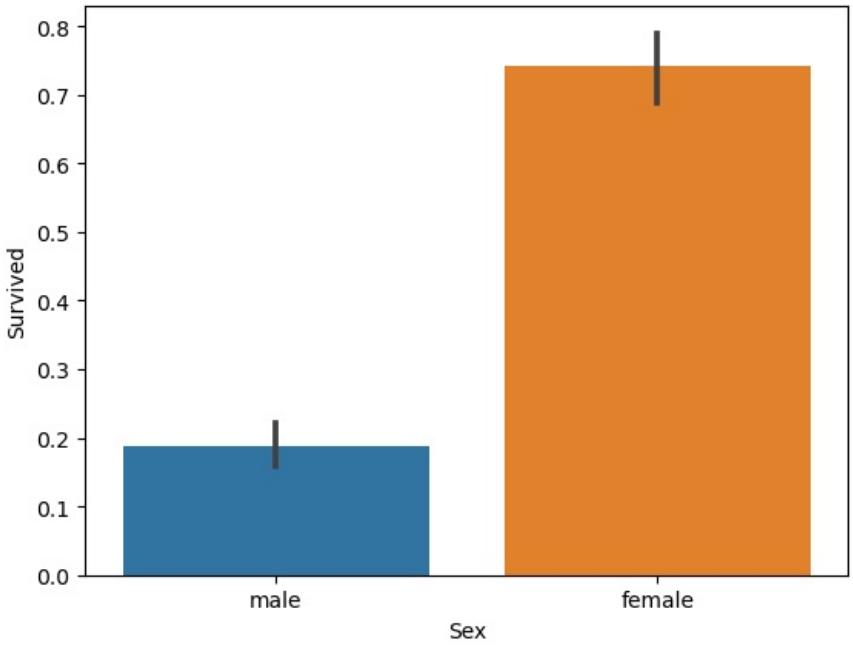
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  891 non-null    int64
 1   Survived     891 non-null    int64
 2   Pclass       891 non-null    int64
 3   Name         891 non-null    object
 4   Sex          891 non-null    object
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64
 7   Parch        891 non-null    int64
 8   Ticket       891 non-null    object
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object
 11  Embarked     889 non-null    object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

In [34]:
```
#so from above info, we can know that Survived is numerical and sex is object/categorical
```

Mainly, in barplot in x axis we put categorical

In [35]:
```
sns.barplot(x=titanic['Sex'],y=titanic['Survived'])
```

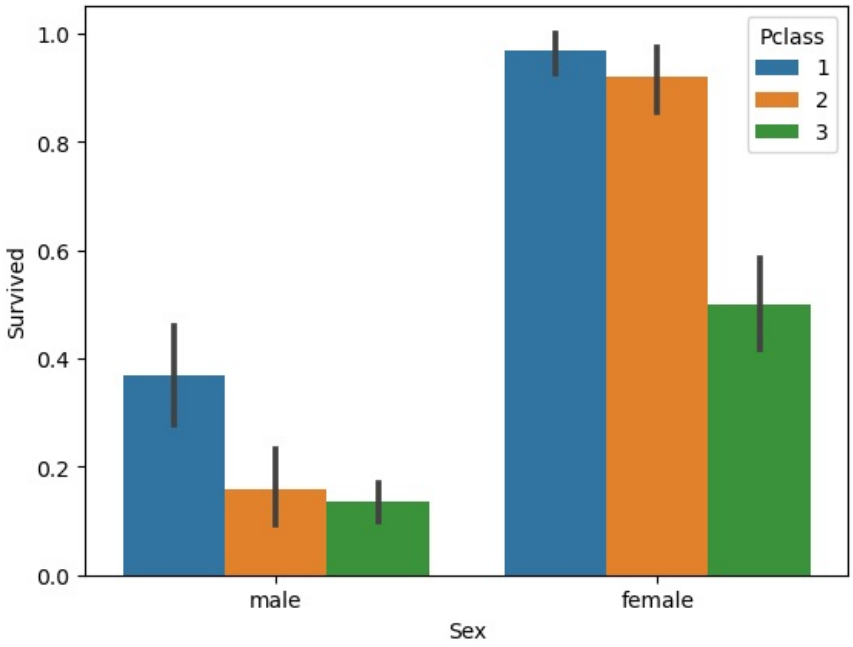
Out[35]:
```
<Axes: xlabel='Sex', ylabel='Survived'>
```



NOTE: Black color rod type in barplot shows confidence interval

In [36]:
```
# We can use hue too here in barplot as multivariate analysis. In above we just saw how many male and female su

sns.barplot(x=titanic['Sex'],y=titanic['Survived'],hue=titanic['Pclass'])
```

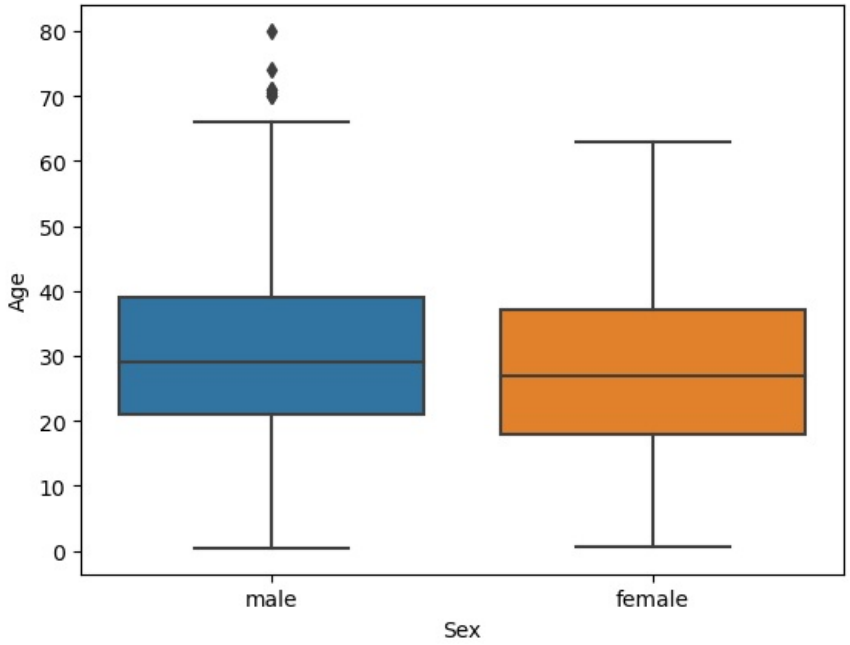
Out[36]:
```
<Axes: xlabel='Sex', ylabel='Survived'>
```



Boxplot - Numerical/Categorical

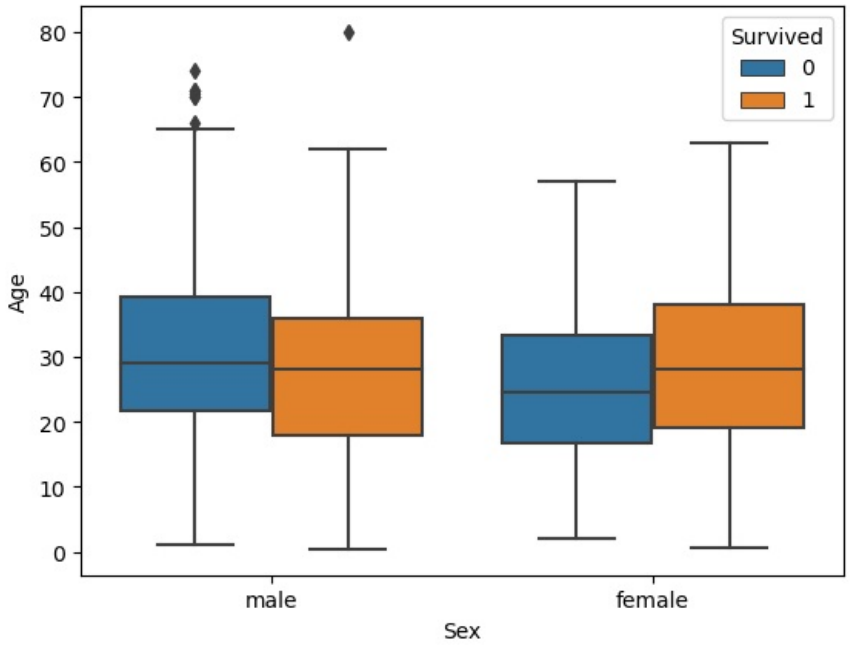
In [37]:
```
sns.boxplot(x=titanic['Sex'],y=titanic['Age'])          #Bivariate Box plot analysis
```

Out[37]:
```
<Axes: xlabel='Sex', ylabel='Age'>
```



In [38]:
```
# Box plot multivariate analysis

sns.boxplot(x=titanic['Sex'],y=titanic['Age'],hue=titanic['Survived'])
```

Out[38]:
```
<Axes: xlabel='Sex', ylabel='Age'>
```

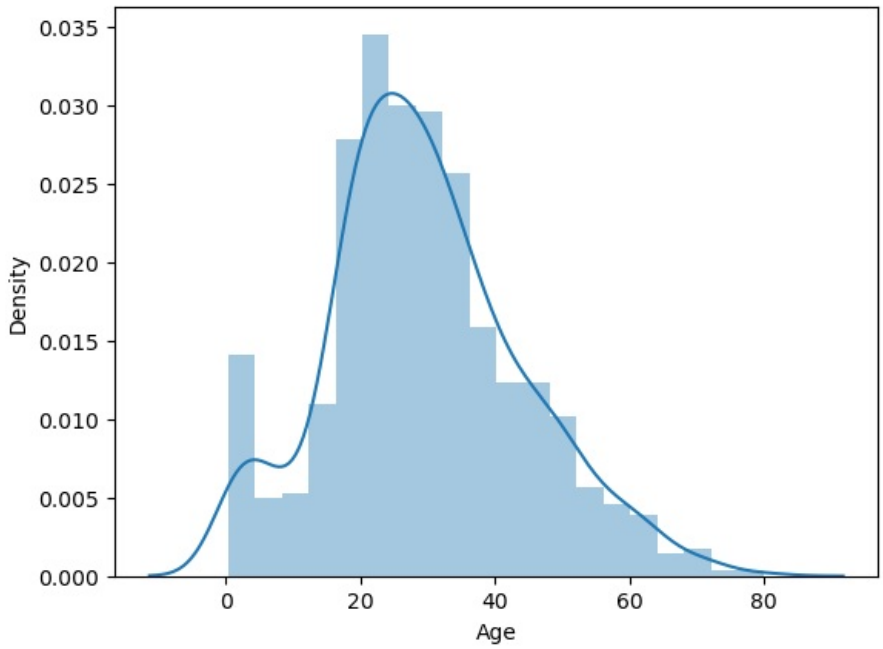


Distplot - Numerical/Categorical

In [39]:
```
sns.distplot(titanic['Age'])
```

```
C:\Users\USER\AppData\Local\Temp\ipykernel_1384\3677708691.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(titanic['Age'])
```

Out[39]:
```
<Axes: xlabel='Age', ylabel='Density'>
```



Another method if we upload by sns

In [40]:
```
d = sns.load_dataset("titanic")
```

In [41]:
```
d.head()
```

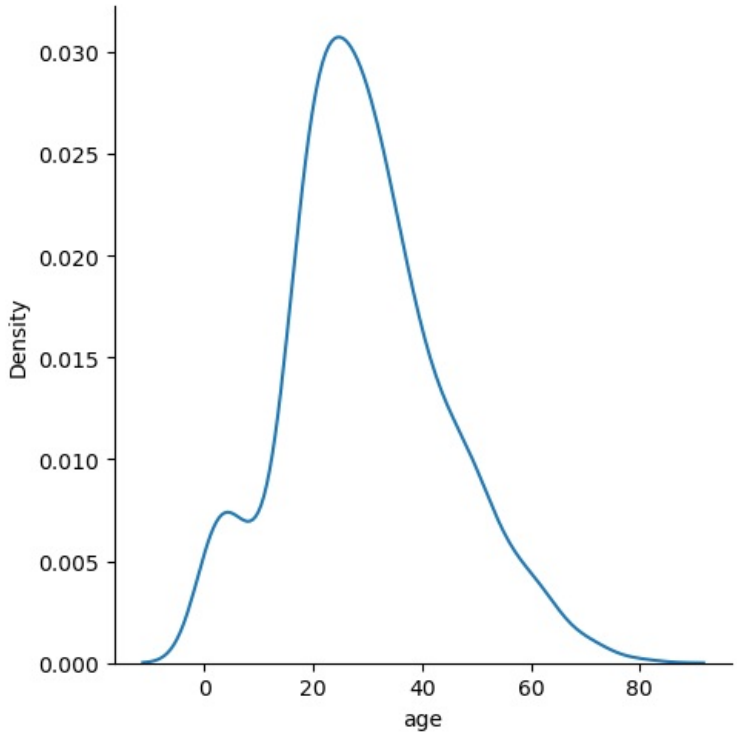
Out[41]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town	alive	alone
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	NaN	Southampton	no	False
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	C	Cherbourg	yes	False
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	NaN	Southampton	yes	True
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	C	Southampton	yes	False
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	NaN	Southampton	no	True

In [42]:
```
sns.displot(data=d, x="age", kind="kde")
```

```
C:\Users\USER\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning: The figure layout has changed t
o tight
  self._figure.tight_layout(*args, **kwargs)
```

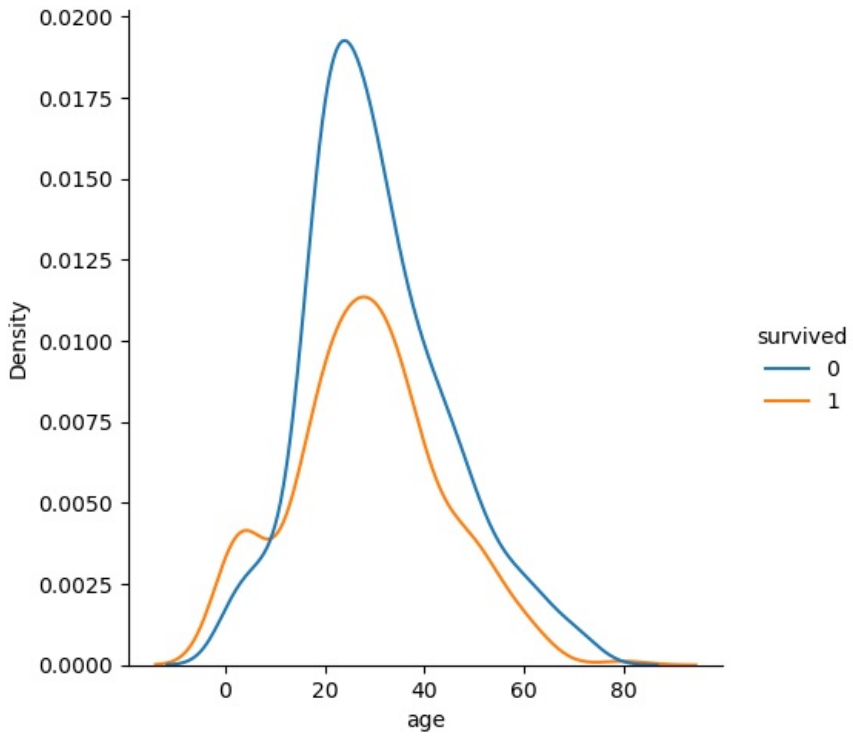
Out[42]:
```
<seaborn.axisgrid.FacetGrid at 0x197447e1910>
```



In [43]:
```
sns.displot(data=d, x="age", kind="kde", hue='survived')
```

```
C:\Users\USER\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning: The figure layout has changed t
o tight
  self._figure.tight_layout(*args, **kwargs)
```

Out[43]:
```
<seaborn.axisgrid.FacetGrid at 0x197447cd450>
```



Heatmap - Categorical/categorical

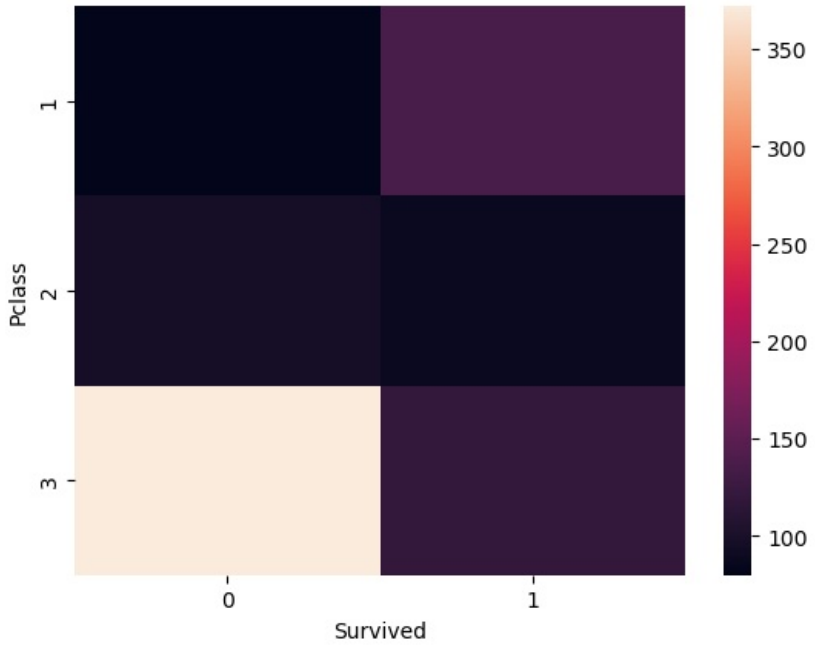
In [44]:
```
pd.crosstab(titanic['Pclass'],titanic['Survived'])   #this is showing how many passenger died & survived and in
```

Out[44]:

Survived	0	1
Pclass		
1	80	136
2	97	87
3	372	119

In [45]:
```
#if we want to see it as graphical

sns.heatmap(pd.crosstab(titanic['Pclass'],titanic['Survived']))
```

Out[45]:
```
<Axes: xlabel='Survived', ylabel='Pclass'>
```



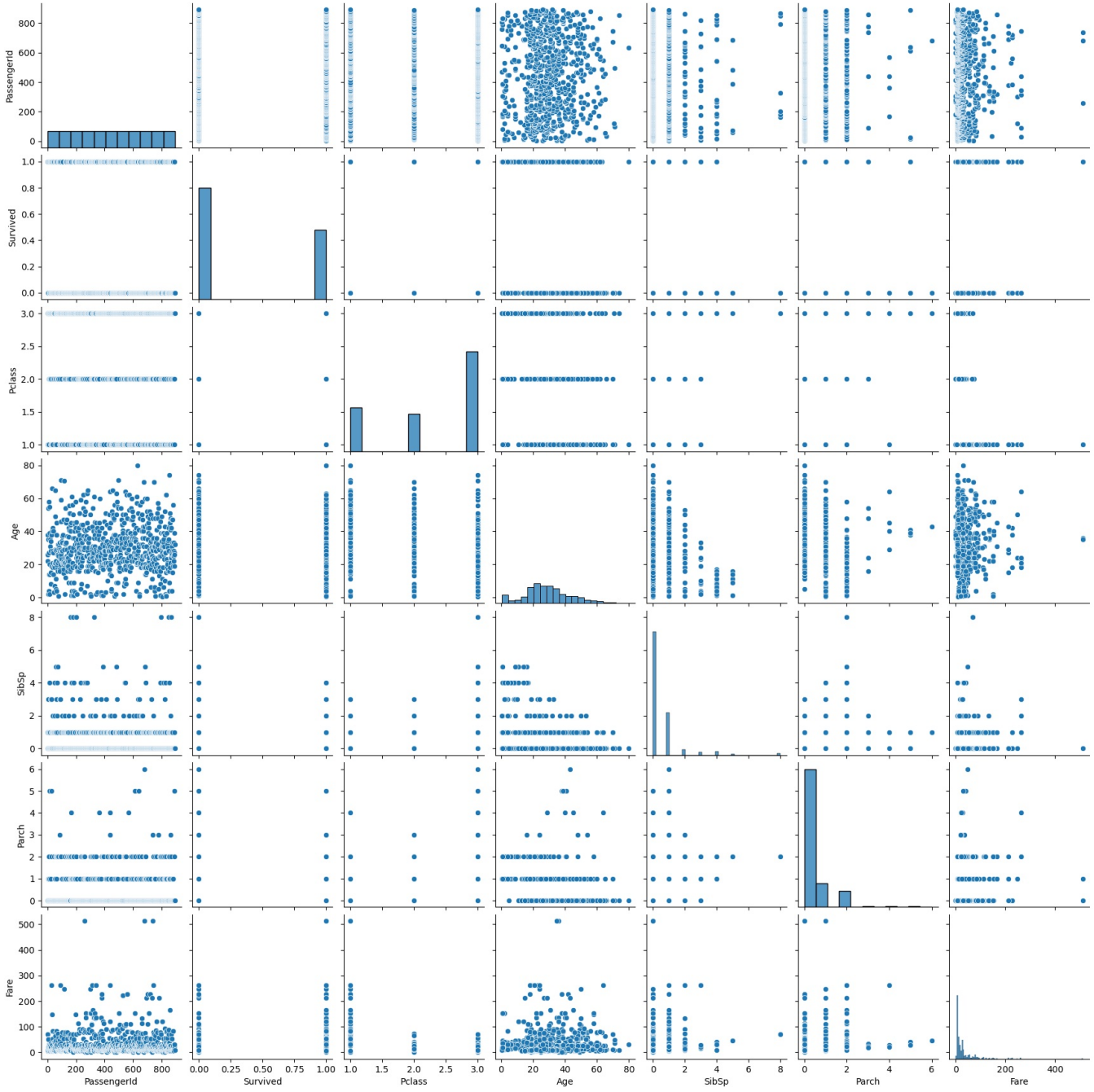
Pairplot - Mainly Numerical/Numerical But can use Categorical in hue

Pairplot gives scatterplot of all column except its own, it will give hist plot of its own.

In [47]:
```
sns.pairplot(titanic)
```

```
C:\Users\USER\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning: The figure layout has changed t
o tight
  self._figure.tight_layout(*args, **kwargs)
```

Out[47]:
```
<seaborn.axisgrid.PairGrid at 0x197485be9d0>
```



In [48]:
```
# If we see, its giving output numerical against numerical.
titanic.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  891 non-null    int64
 1   Survived     891 non-null    int64
 2   Pclass       891 non-null    int64
 3   Name         891 non-null    object
 4   Sex          891 non-null    object
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64
 7   Parch        891 non-null    int64
 8   Ticket       891 non-null    object
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object
 11  Embarked     889 non-null    object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

In [49]:
```
#But we can do multivariate analysis using hue by passing categorical data
```

In [49]:
```
#But we can do multivariate analysis using hue by passing categorical data
sns.pairplot(titanic, hue='Sex')
```

```
C:\Users\USER\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning: The figure layout has changed t
o tight
  self._figure.tight_layout(*args, **kwargs)
```

Out[49]:
```
<seaborn.axisgrid.PairGrid at 0x1974a6d8550>
```