EDA in Univariate Analysis

Import Library

import pandas as pd

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

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

In [10]:
```
df.sample(5)
```

Out[10]:

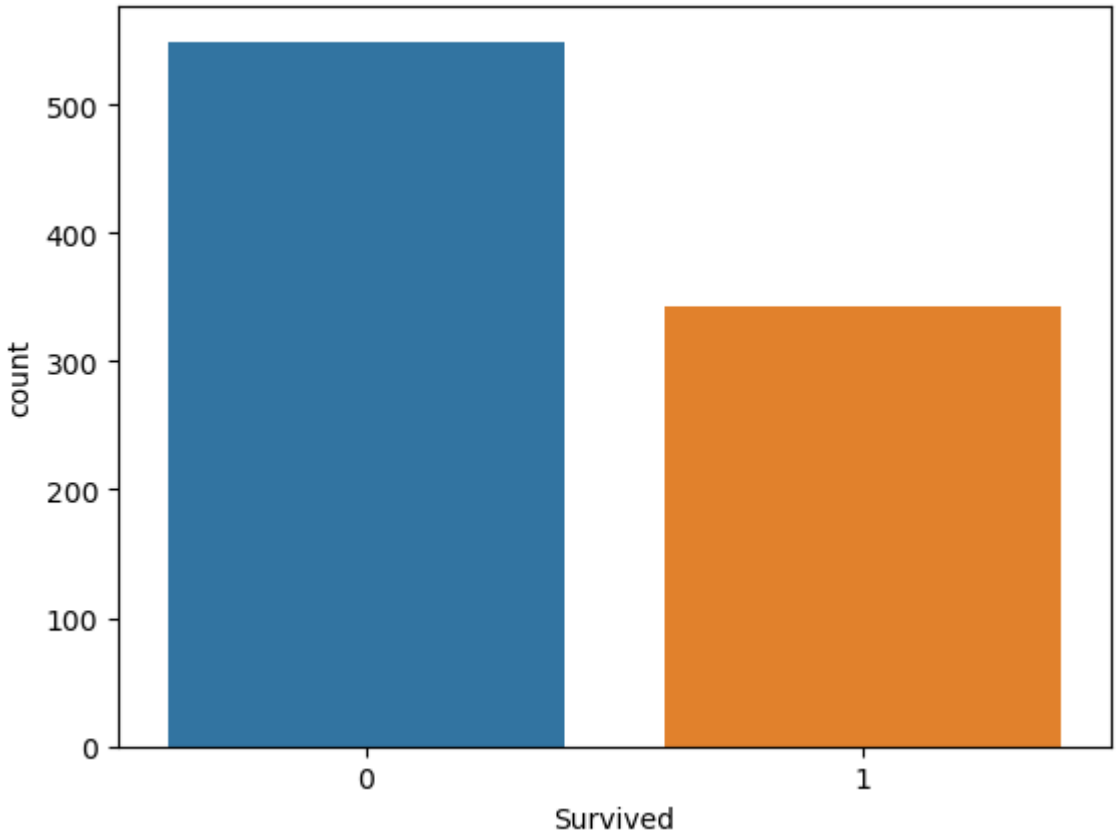
	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	E
628	629	0	3	Bostandyeff, Mr. Guentcho	male	26.0	0	0	349224	7.8958	NaN	
847	848	0	3	Markoff, Mr. Marin	male	35.0	0	0	349213	7.8958	NaN	
204	205	1	3	Cohen, Mr. Gurshon "Gus"	male	18.0	0	0	A/5 3540	8.0500	NaN	
146	147	1	3	Andersson, Mr. August Edvard ("Wennerstrom")	male	27.0	0	0	350043	7.7958	NaN	
655	656	0	2	Hickman, Mr. Leonard Mark	male	24.0	2	0	S.O.C. 14879	73.5000	NaN	

1. Categorical Data

a. Countplot

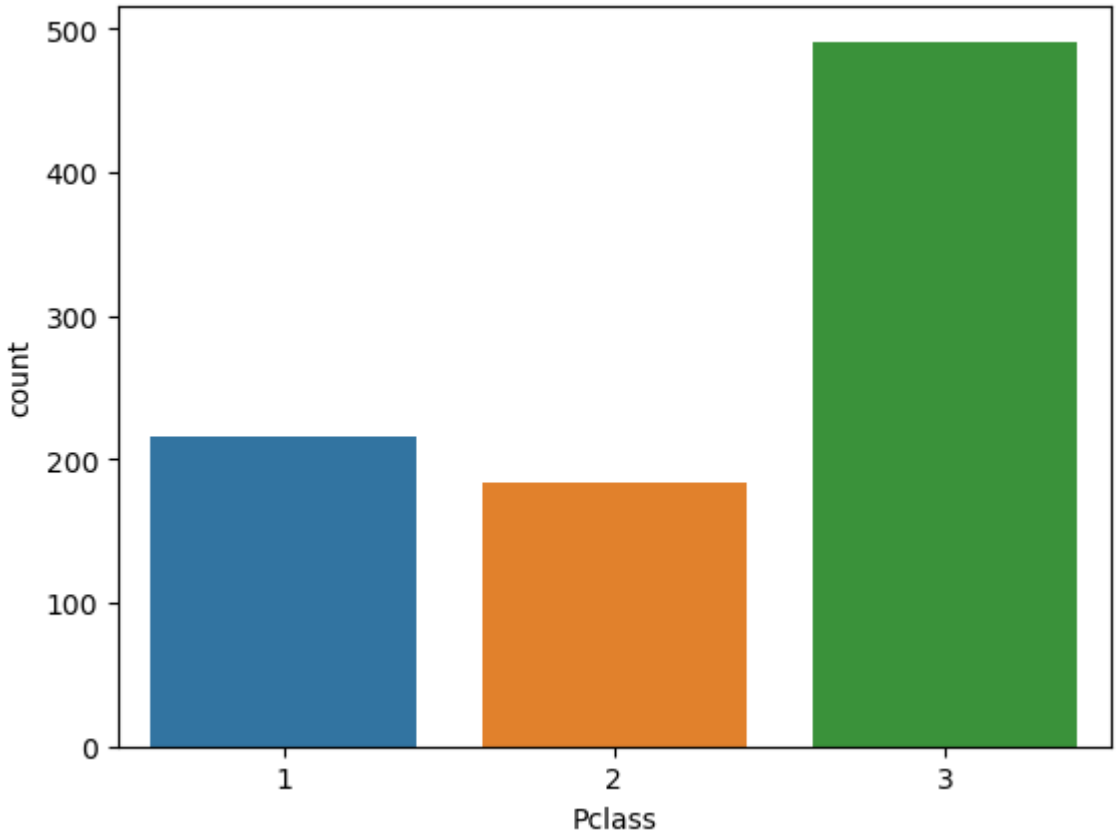
In [12]:
```
sns.countplot(x='Survived', data=df)
df['Survived'].value_counts()
```

Out[12]:
```
Survived
0    549
1    342
Name: count, dtype: int64
```



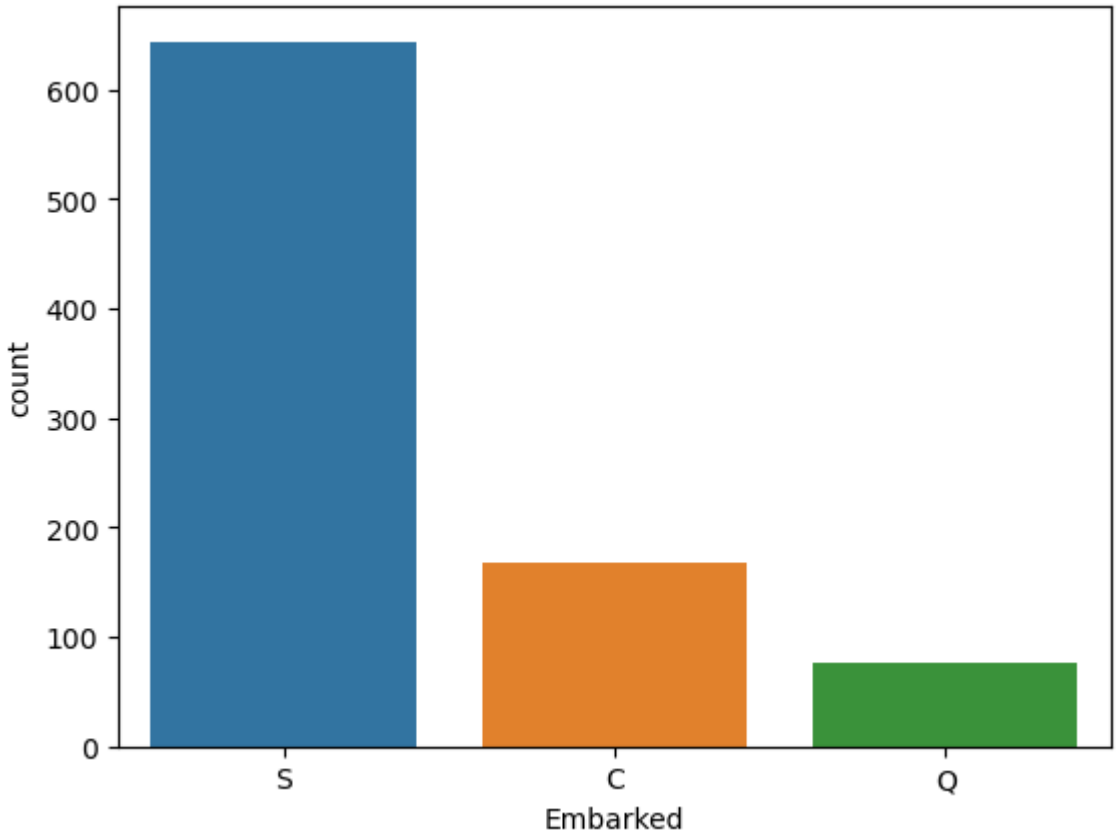
In [13]:
```
sns.countplot(x='Pclass', data=df)
df['Pclass'].value_counts()
```

Out[13]:
```
Pclass
3    491
1    216
2    184
Name: count, dtype: int64
```



In [14]:
```
sns.countplot(x='Embarked', data=df)
df['Embarked'].value_counts()
```

Out[14]:
```
Embarked
S    644
C    168
Q     77
Name: count, dtype: int64
```

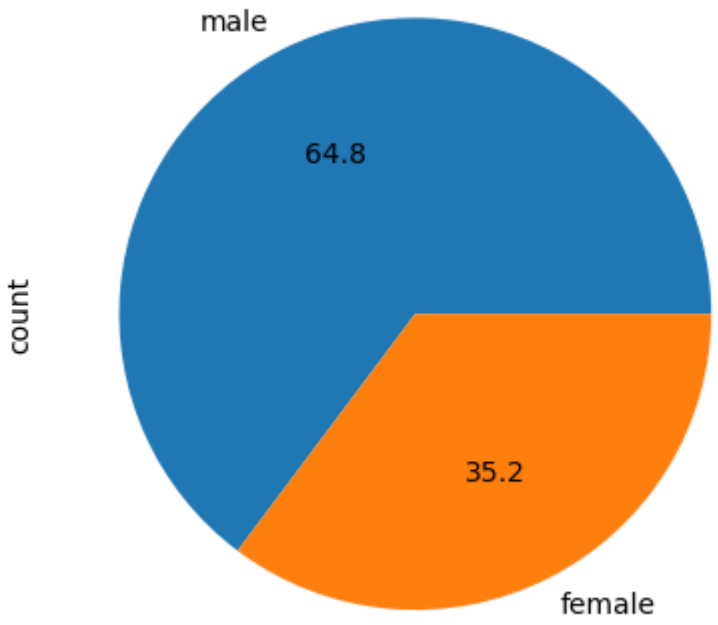


Type *Markdown* and LaTeX: α^2

b. Piechart

In [19]:
```
df['Sex'].value_counts().plot(kind='pie',autopct='%.1f')
```

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



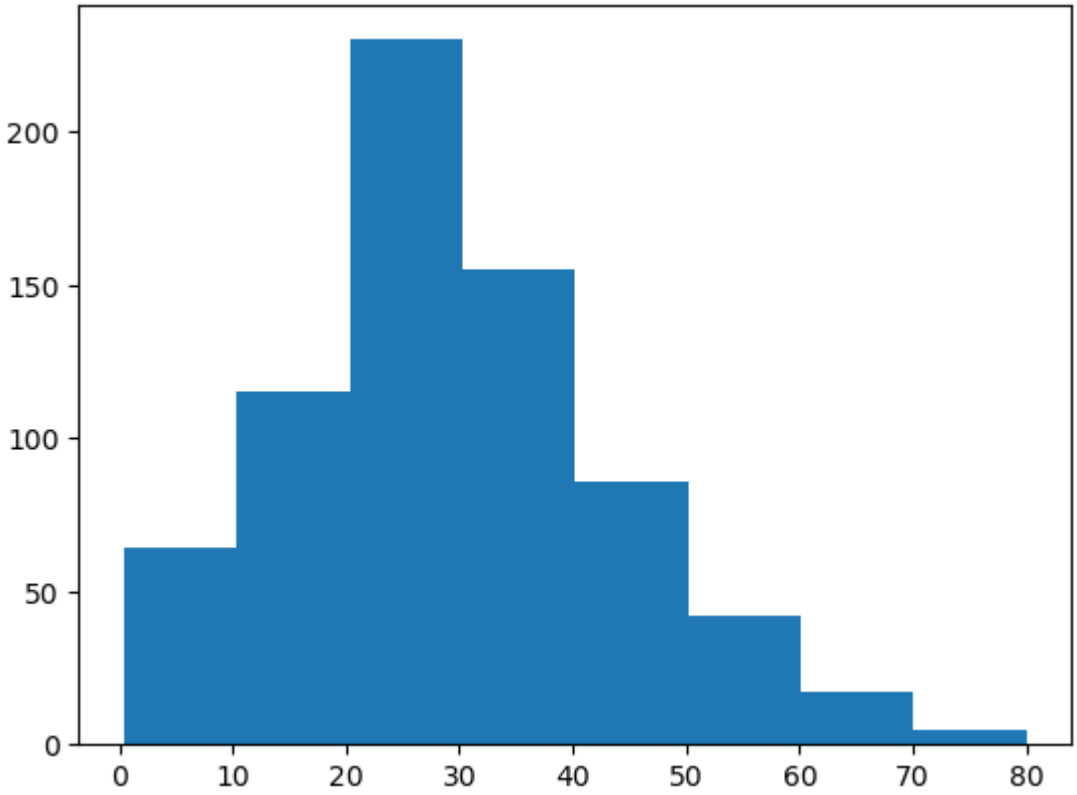
2. NUMERICAL DATA

a. Histogram

In [20]:
```
import matplotlib.pyplot as plt
```

In [29]:
```
plt.hist(df['Age'],bins=8)
```

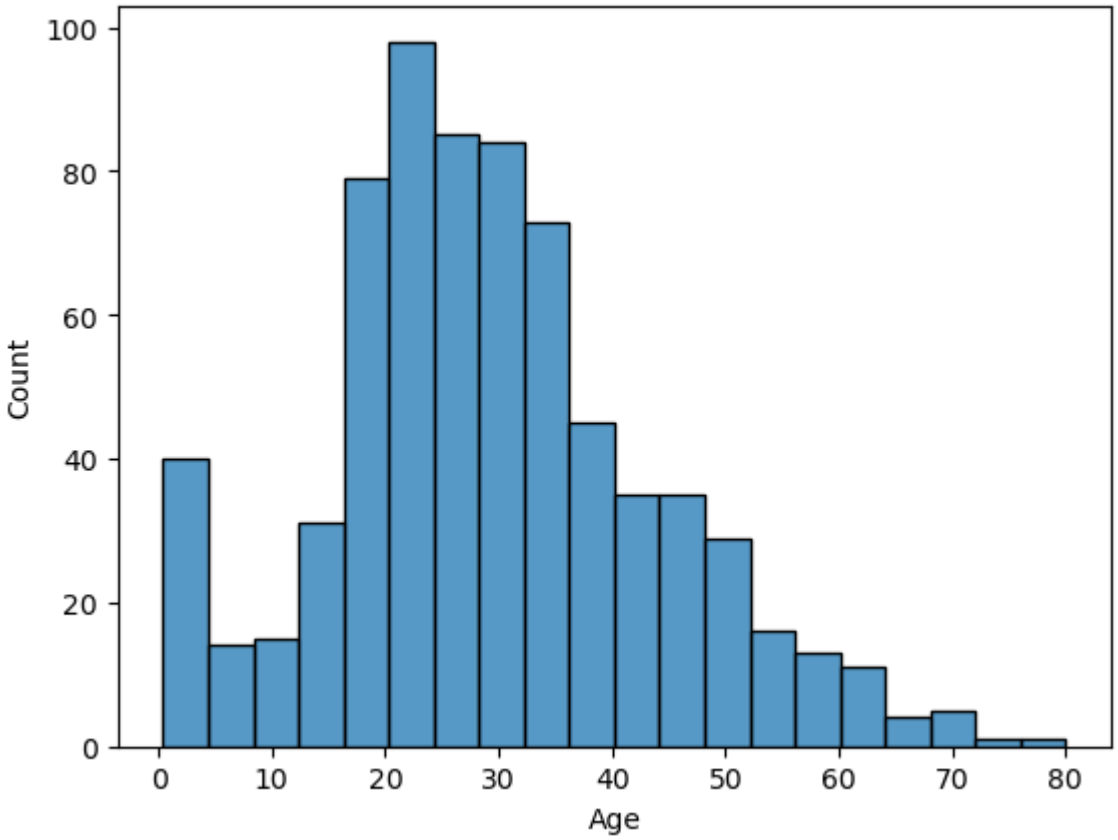
Out[29]:
```
(array([ 64., 115., 230., 155.,  86.,  42.,  17.,   5.]),
 array([ 0.42  , 10.3675, 20.315 , 30.2625, 40.21  , 50.1575, 60.105 ,
        70.0525, 80.    ]),
 <BarContainer object of 8 artists>)
```



b. Distplot / Histplot

In [33]:
```
sns.histplot(df['Age'])
```

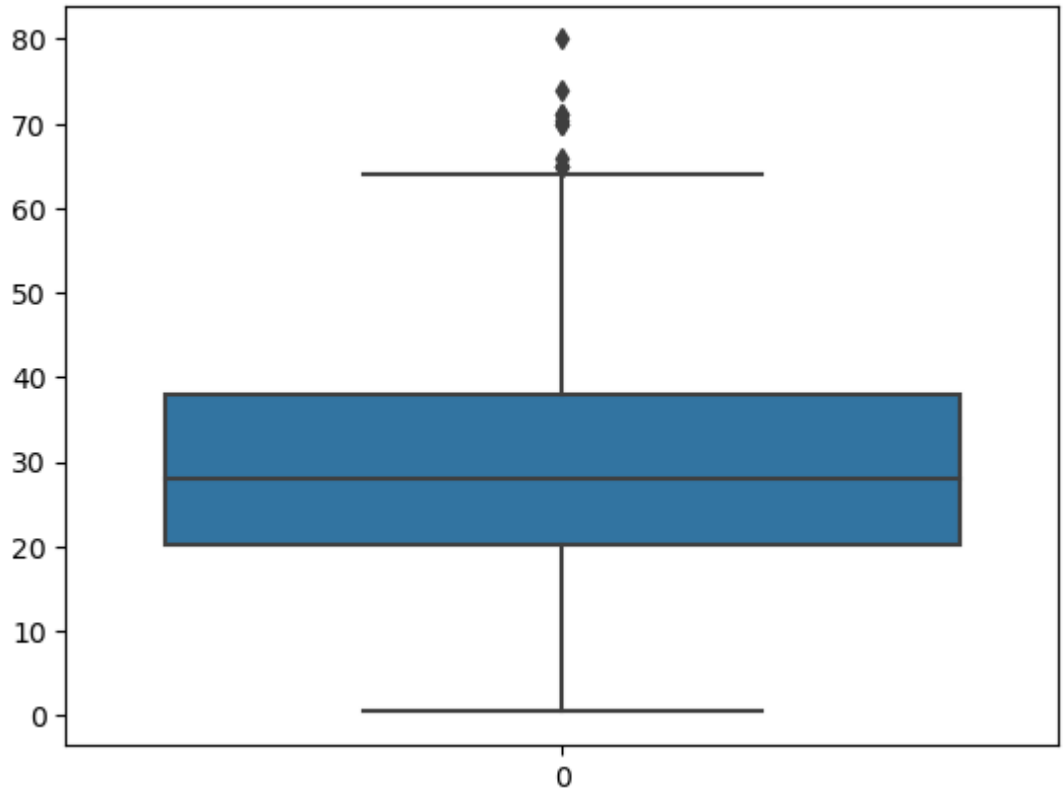
Out[33]:
```
<Axes: xlabel='Age', ylabel='Count'>
```



c. Boxplot

In [34]:
```
sns.boxplot(df['Age'])
```

Out[34]: <Axes: >



In [35]:
```
df['Age'].min()
```

Out[35]: 0.42

In [36]:
```
df['Age'].max()
```

Out[36]: 80.0

In [37]:
```
df['Age'].mean()
```

Out[37]: 29.69911764705882

In [38]:
```
df['Age'].skew()
```

Out[38]: 0.38910778230082704

In [ ]:
```

```