Assignment 8

```
import numpy as np
import pandas as pd
import seaborn as sns

df=sns.load_dataset('titanic')
df
```

```
     survived  pclass     sex   age  sibsp  parch     fare embarked
class  \
0           0       3    male  22.0      1      0   7.2500        S
Third
1           1       1  female  38.0      1      0  71.2833        C
First
2           1       3  female  26.0      0      0   7.9250        S
Third
3           1       1  female  35.0      1      0  53.1000        S
First
4           0       3    male  35.0      0      0   8.0500        S
Third
..        ...     ...     ...   ...    ...    ...      ...       ...
...
886         0       2    male  27.0      0      0  13.0000        S
Second
887         1       1  female  19.0      0      0  30.0000        S
First
888         0       3  female   NaN      1      2  23.4500        S
Third
889         1       1    male  26.0      0      0  30.0000        C
First
890         0       3    male  32.0      0      0   7.7500        Q
Third

       who  adult_male deck  embark_town alive  alone
0      man        True  NaN  Southampton    no  False
1    woman       False    C    Cherbourg   yes  False
2    woman       False  NaN  Southampton   yes   True
3    woman       False    C  Southampton   yes  False
4      man        True  NaN  Southampton    no   True
..     ...         ...  ...          ...   ...    ...
886    man        True  NaN  Southampton    no   True
887  woman       False    B  Southampton   yes   True
888  woman       False  NaN  Southampton    no  False
889    man        True    C    Cherbourg   yes   True
890    man        True  NaN   Queenstown    no   True

[891 rows x 15 columns]
```

```
df.head()
```

```
   survived  pclass     sex   age  sibsp  parch     fare embarked
class  \
0         0       3    male  22.0      1      0   7.2500        S
Third
1         1       1  female  38.0      1      0  71.2833        C
First
2         1       3  female  26.0      0      0   7.9250        S
Third
3         1       1  female  35.0      1      0  53.1000        S
First
4         0       3    male  35.0      0      0   8.0500        S
Third

     who  adult_male deck  embark_town alive  alone
0    man        True  NaN  Southampton    no  False
1  woman       False    C    Cherbourg   yes  False
2  woman       False  NaN  Southampton   yes   True
3  woman       False    C  Southampton   yes  False
4    man        True  NaN  Southampton    no   True
```

```
df.shape
```

```
(891, 15)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 15 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   survived     891 non-null    int64
 1   pclass       891 non-null    int64
 2   sex          891 non-null    object
 3   age          714 non-null    float64
 4   sibsp        891 non-null    int64
 5   parch        891 non-null    int64
 6   fare         891 non-null    float64
 7   embarked     889 non-null    object
 8   class        891 non-null    category
 9   who          891 non-null    object
 10  adult_male   891 non-null    bool
 11  deck         203 non-null    category
 12  embark_town  889 non-null    object
 13  alive        891 non-null    object
 14  alone        891 non-null    bool
dtypes: bool(2), category(2), float64(2), int64(4), object(5)
memory usage: 80.7+ KB
```

```
df.describe()
```

```
         survived      pclass         age       sibsp       parch
fare
count  891.000000  891.000000  714.000000  891.000000  891.000000
891.000000
mean     0.383838    2.308642   29.699118    0.523008    0.381594
32.204208
std      0.486592    0.836071   14.526497    1.102743    0.806057
49.693429
min      0.000000    1.000000    0.420000    0.000000    0.000000
0.000000
25%      0.000000    2.000000   20.125000    0.000000    0.000000
7.910400
50%      0.000000    3.000000   28.000000    0.000000    0.000000
14.454200
75%      1.000000    3.000000   38.000000    1.000000    0.000000
31.000000
max      1.000000    3.000000   80.000000    8.000000    6.000000
512.329200
```

```
df.isnull().sum()
```

```
survived         0
pclass           0
sex              0
age            177
sibsp            0
parch            0
fare             0
embarked         2
class            0
who              0
adult_male       0
deck           688
embark_town      2
alive            0
alone            0
dtype: int64
```

```
df.drop(columns=['deck','embark_town'],axis=1,inplace=True)
```

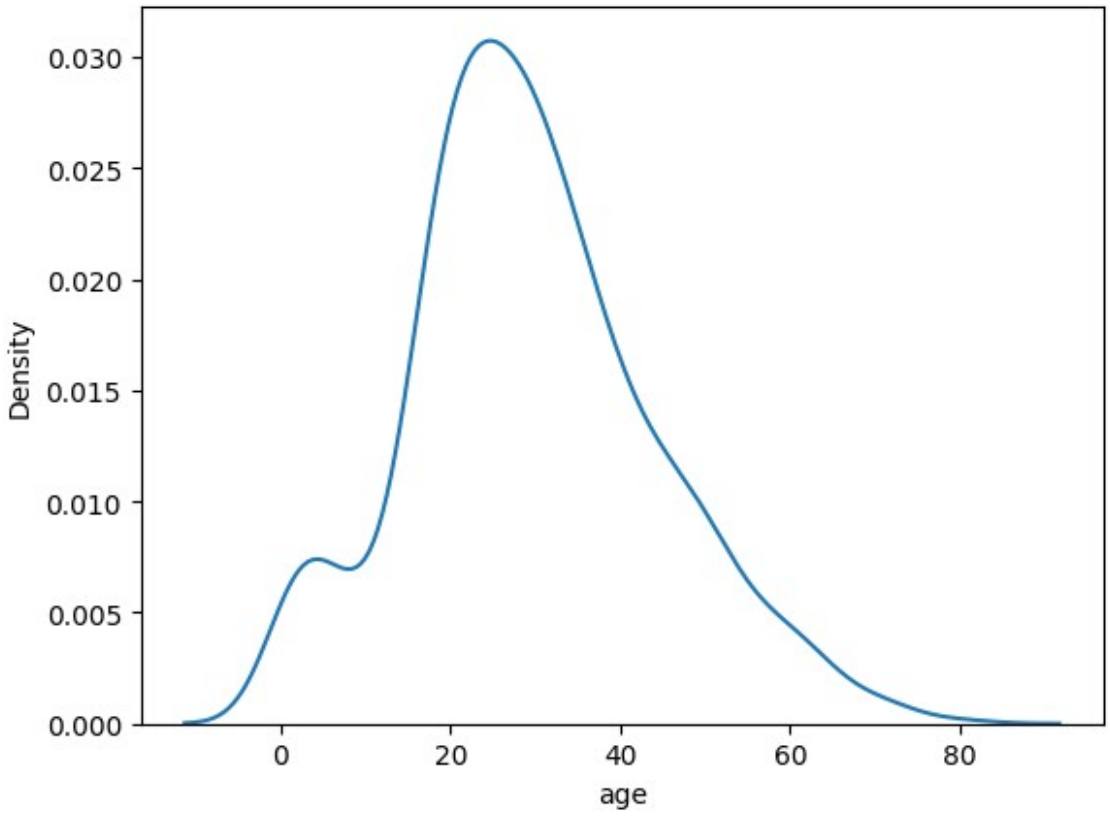
```
df.isnull().sum()
```

```
survived        0
pclass          0
sex             0
age           177
sibsp           0
parch           0
fare            0
```

```
embarked         2
class            0
who              0
adult_male       0
alive            0
alone            0
dtype: int64
```

```
sns.kdeplot(df['age'])
```

```
<Axes: xlabel='age', ylabel='Density'>
```




```
df['age'].skew()
```

```
0.38910778230082704
```

```
df['age'].fillna(df['age'].mean(),inplace=True)
```

```
C:\Users\ANKIT\AppData\Local\Temp\ipykernel_6448\1492264711.py:1:
FutureWarning: A value is trying to be set on a copy of a DataFrame or
Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never
work because the intermediate object on which we are setting values
always behaves as a copy.
```

```
For example, when doing 'df[col].method(value, inplace=True)', try
using 'df.method({col: value}, inplace=True)' or df[col] =
df[col].method(value) instead, to perform the operation inplace on the
original object.
```

```
  df['age'].fillna(df['age'].mean(),inplace=True)
```

```
df['embarked'].fillna(df['embarked'].mode()[0],inplace=True)
```

```
C:\Users\ANKIT\AppData\Local\Temp\ipykernel_6448\1024298632.py:1:
FutureWarning: A value is trying to be set on a copy of a DataFrame or
Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never
work because the intermediate object on which we are setting values
always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try
using 'df.method({col: value}, inplace=True)' or df[col] =
df[col].method(value) instead, to perform the operation inplace on the
original object.
```

```
  df['embarked'].fillna(df['embarked'].mode()[0],inplace=True)
```

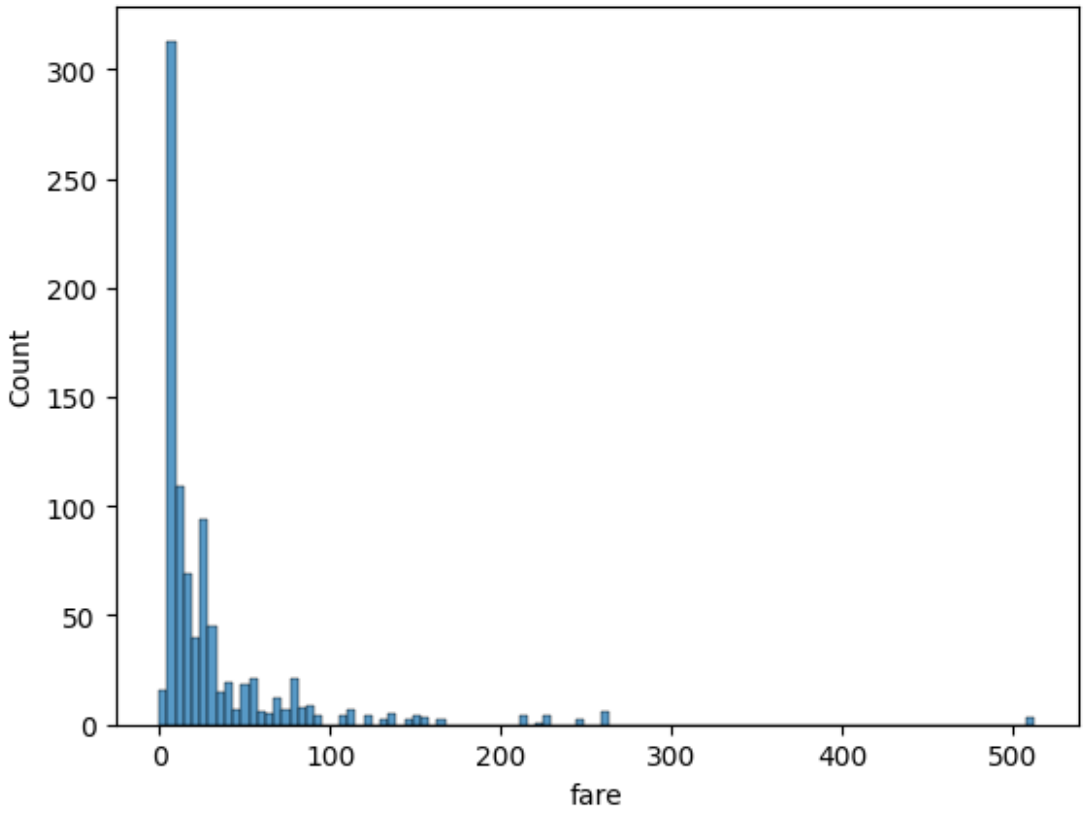
```
df.isnull().sum()
```

```
survived      0
pclass        0
sex           0
age           0
sibsp         0
parch         0
fare          0
embarked      0
class         0
who           0
adult_male    0
alive         0
alone         0
dtype: int64
```

EDA Exploratory data analysis

```
sns.histplot(df['fare']) #DATA IN FARE COLUMN IS HIGHLY DISTRIBUTED
```

```
<Axes: xlabel='fare', ylabel='Count'>
```

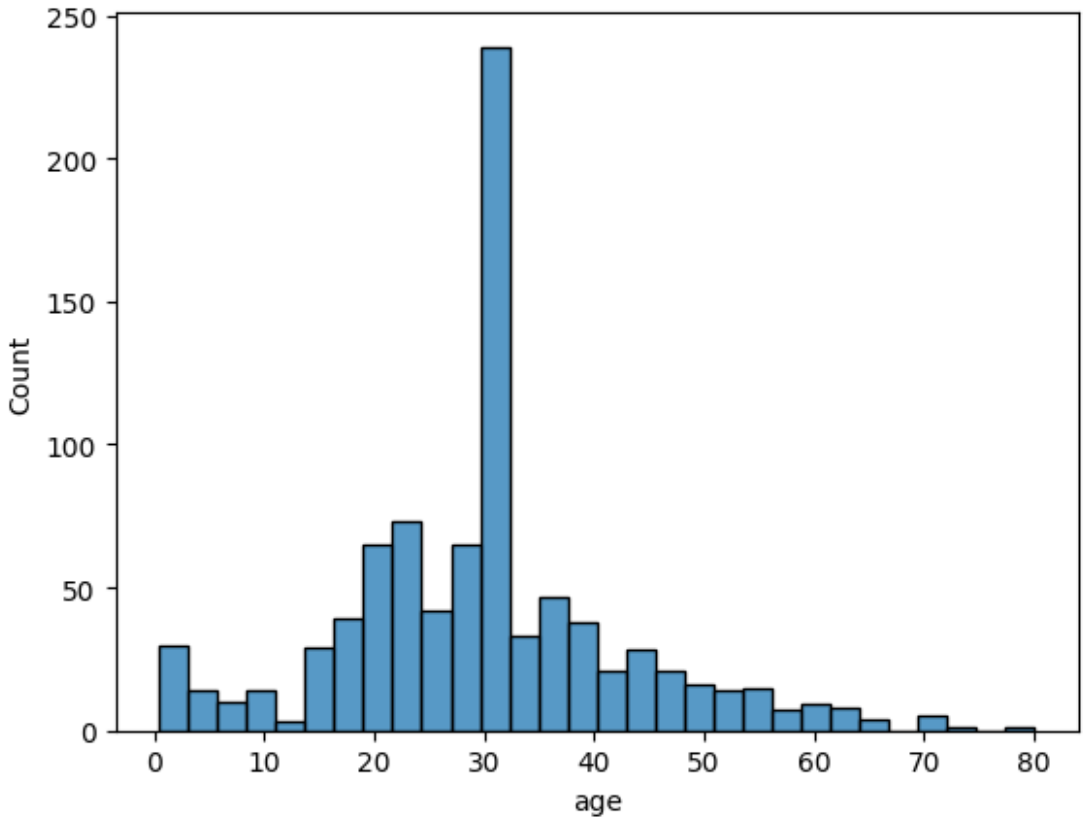



```
df['fare'].skew()
```

```
4.787316519674893
```

```
#Univariate analysis
sns.histplot(df['age'])
```

```
<Axes: xlabel='age', ylabel='Count'>
```



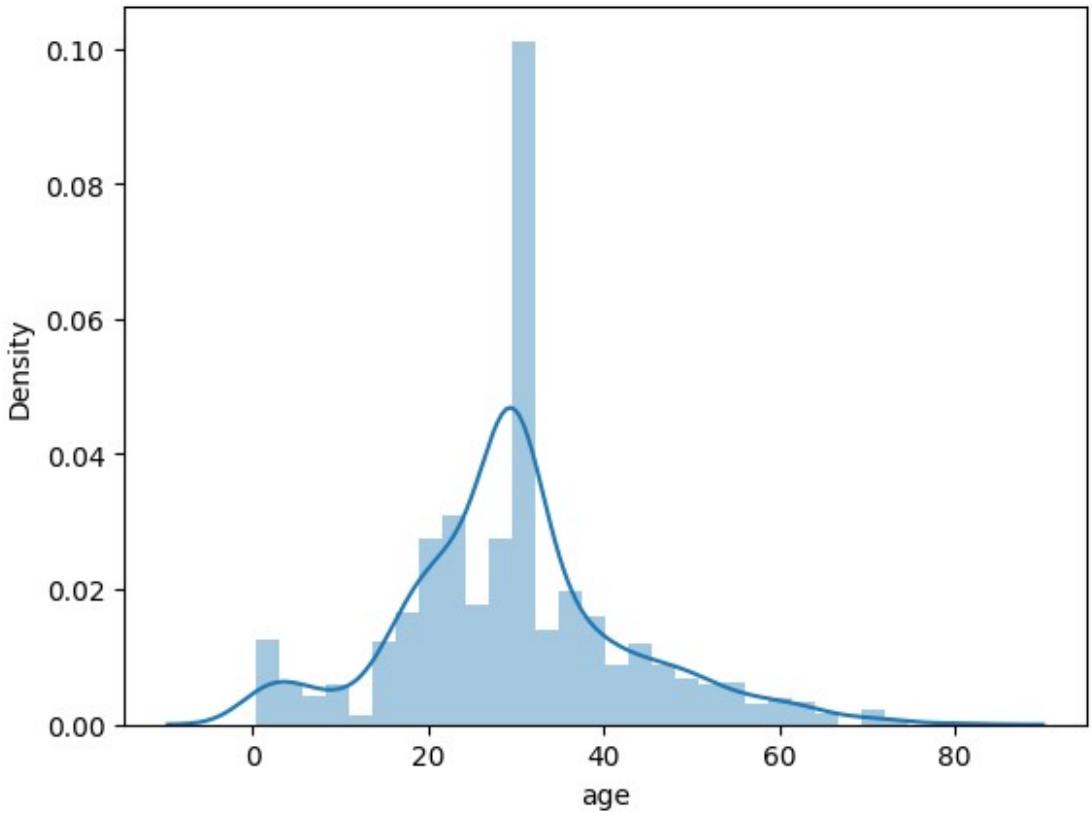

```
df['age'].skew()
```

```
0.4344880940129925
```

```
sns.distplot(df['age']) #Age column is normally distributed
```

```
C:\Users\ANKIT\AppData\Local\Temp\ipykernel_6448\738922914.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']) #Age column is normally distributed
```

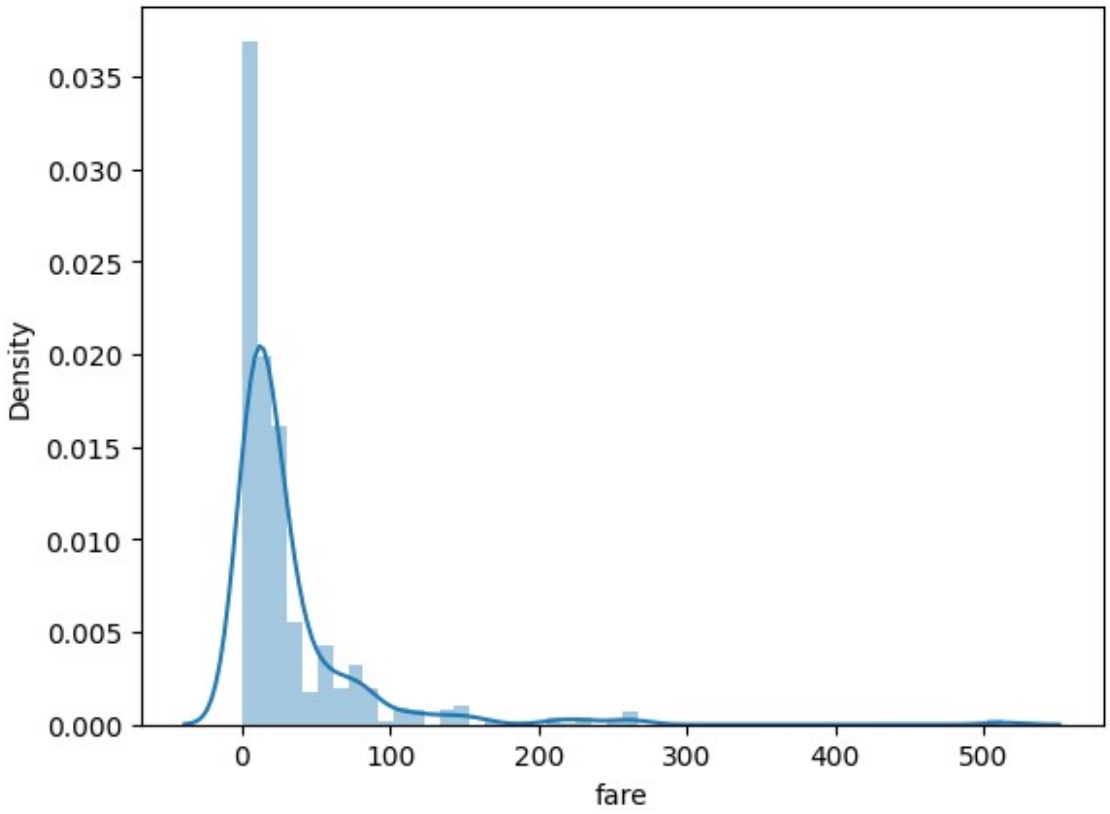
```
<Axes: xlabel='age', ylabel='Density'>
```




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

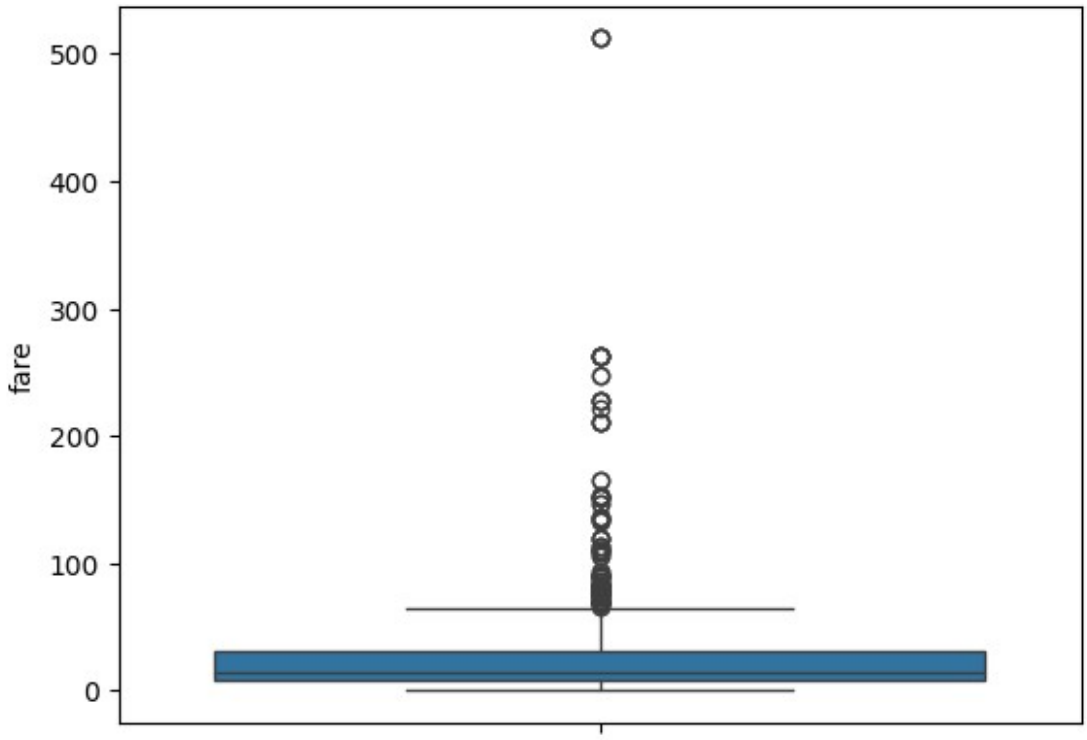
```
C:\Users\ANKIT\AppData\Local\Temp\ipykernel_6448\1195996103.py:1:
UserWarning:

`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['fare'])
```

```
<Axes: xlabel='fare', ylabel='Density'>
```



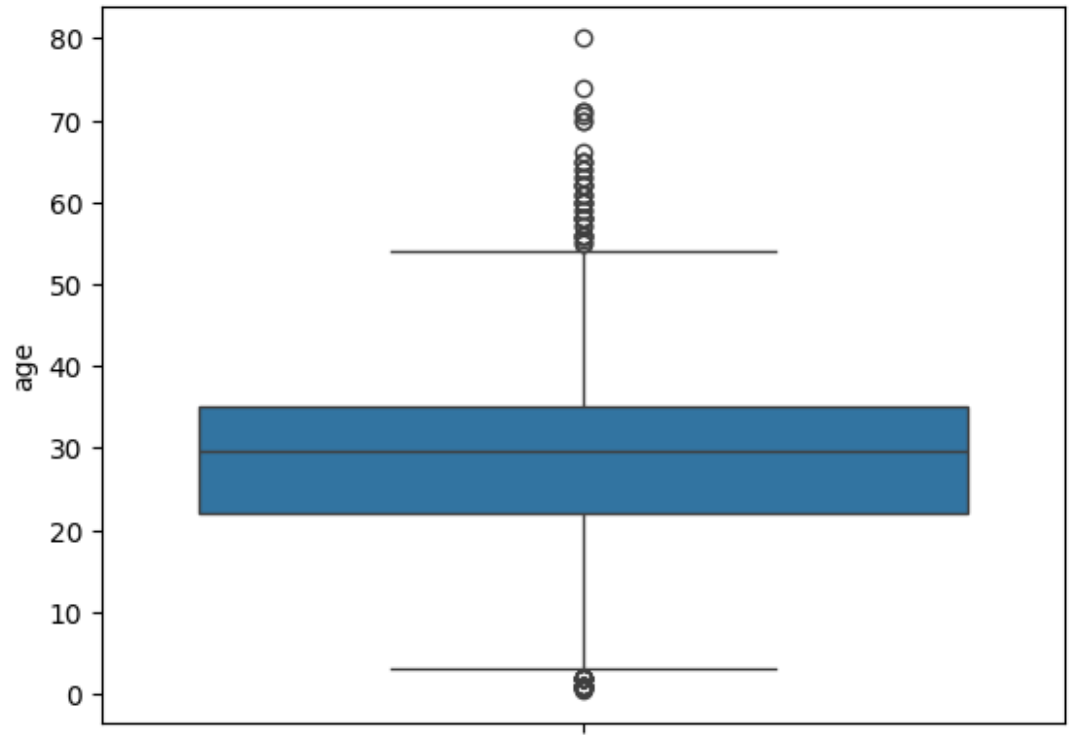

```
sns.boxplot(df['fare']) #WE SAW OUTLIERS IN FARE ON UPPER FENCE
```

```
<Axes: ylabel='fare'>
```



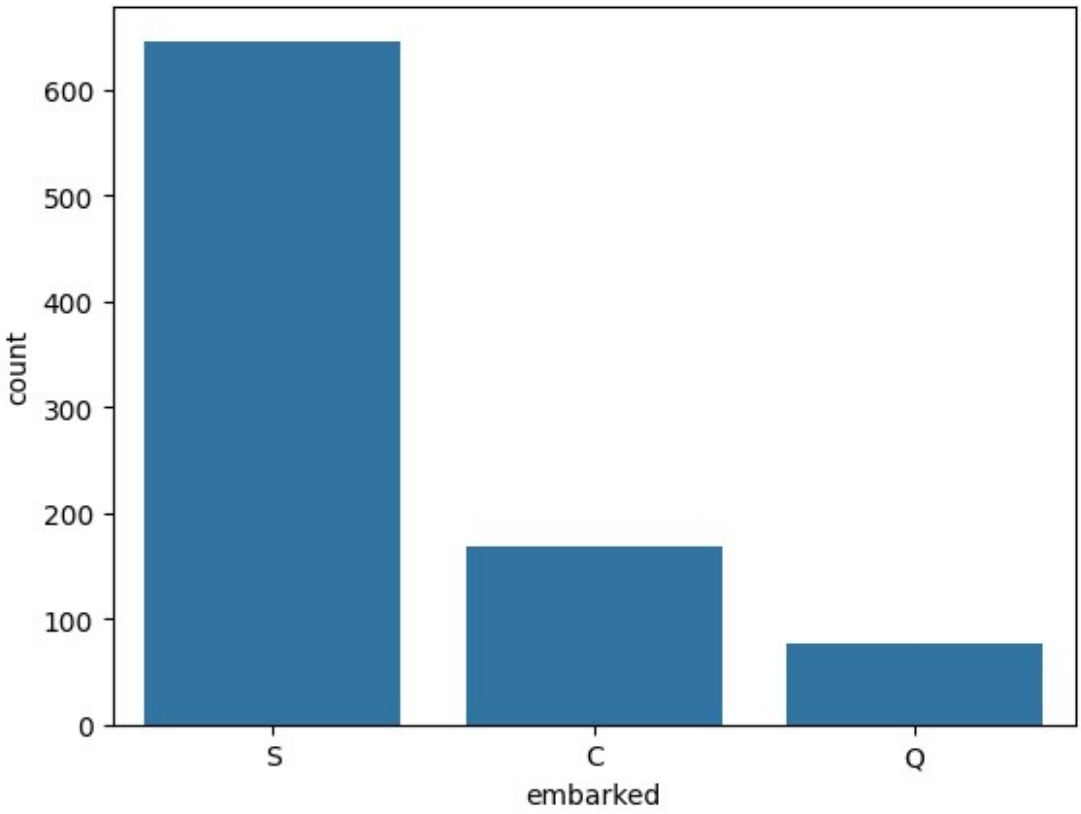

```
sns.boxplot(df['age']) #WE SAW OUTLIERS IN AGE
```

```
<Axes: ylabel='age'>
```



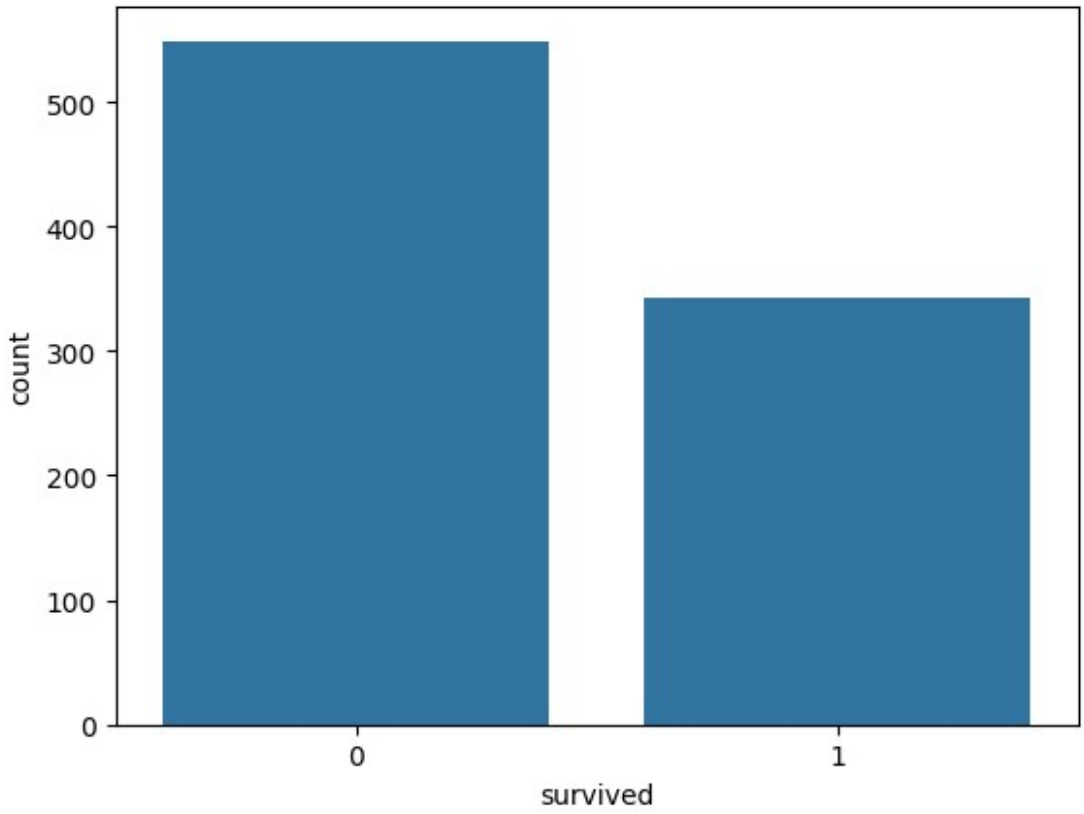
```
sns.countplot(x=df['embarked']) #As we see that there are mainly passengers are from Southampton
```

```
<Axes: xlabel='embarked', ylabel='count'>
```



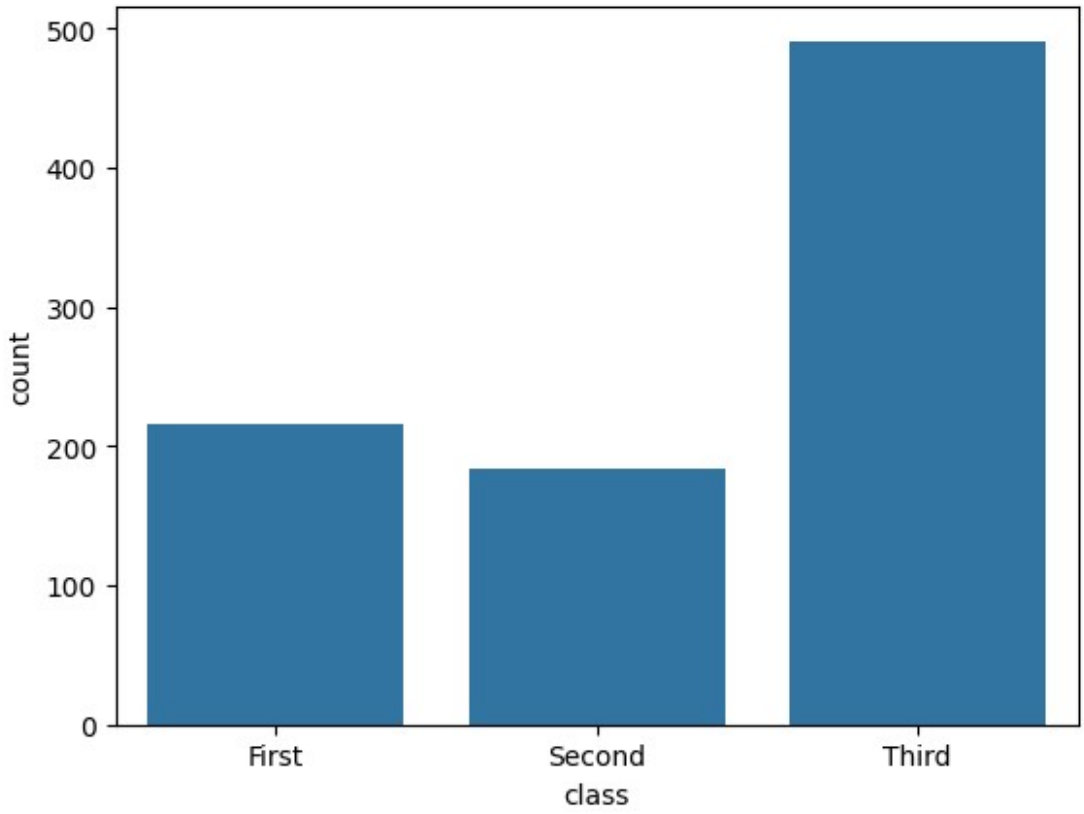

```
sns.countplot(x=df['survived']) #As we see that most of the people not survived
```

```
<Axes: xlabel='survived', ylabel='count'>
```



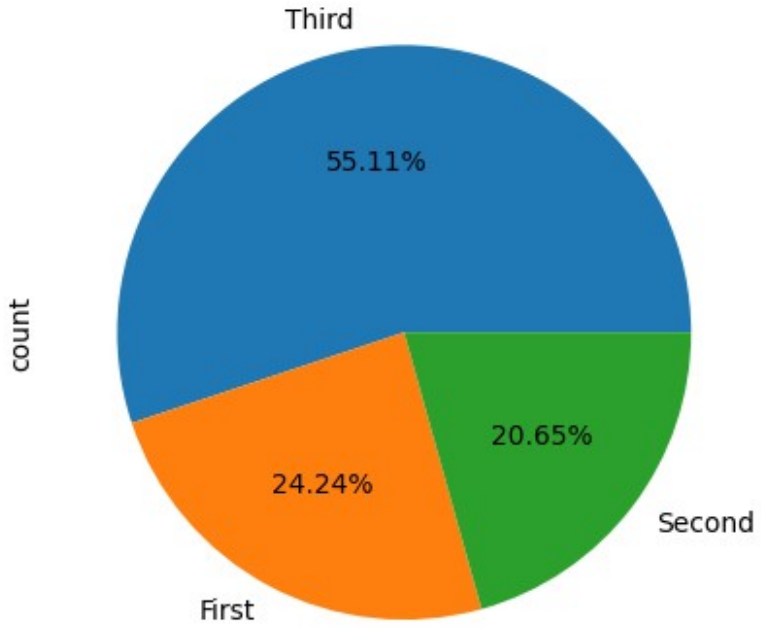

```
sns.countplot(x=df['class']) #Most of the people are from third Class
```

```
<Axes: xlabel='class', ylabel='count'>
```



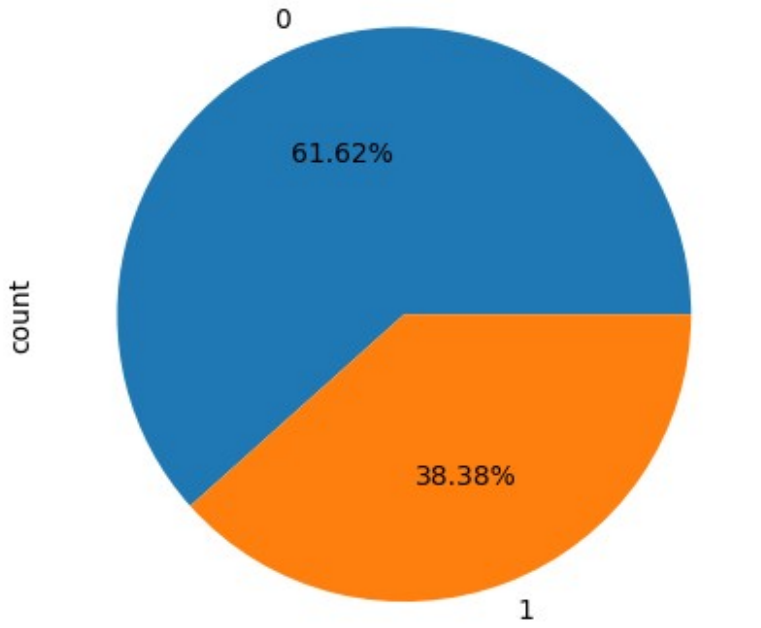

```
df['class'].value_counts().plot(kind='pie',autopct='')
```

```
<Axes: ylabel='count'>
```




```
df['survived'].value_counts().plot(kind='pie',autopct='%.2f%%')
```

```
<Axes: ylabel='count'>
```



```
df['embarked'].value_counts().plot(kind='pie',autopct='%.2f%%')
```

```
<Axes: ylabel='count'>
```