Prodigy Infotech Internship Task 2

Name : Pavan yadav

Task : EDA of Titanic dataset

In [48]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [49]:
```
df = pd.read_csv('train.csv')
df
```

Out[49]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Far
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.250
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.283
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.925
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.100
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.050
...	...	...	...	...	...	...	...	...	...	.
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.000
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.000
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.450
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.000
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.750

891 rows × 12 columns

In [50]:
```
df.head()
```

Out[50]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
0	1	0 7.2500	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	

In

1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	

In [51]:
```
df.tail()
```

Out[51]:

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

In [52]:
```
df.shape
```

Out[52]: (891, 12)

In [53]:
```
df.columns.values
```

Out[53]:
```
array(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp',
       'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'], dtype=object)
```

In [54]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
  0PassengerId  891 non-null    int64
  1Survived     891 non-null    int64
  2Pclass       891 non-null    int64
  3Name         891 non-null    object
  4Sex          891 non-null    object
  5Age          714 non-null    float64
  6SibSp        891 non-null    int64
  7Parch        891 non-null    int64
  8Ticket       891 non-null    object
  9Fare         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 [55]:
```
df.isnull().sum()
```

Out[55]:
```
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 [56]:
```
# dropping the cabin column
df.drop(columns=['Cabin'],inplace=True)
```

In [57]:
```
#imputing the missing value with the mean
df['Age'].fillna(df['Age'].mean() , inplace = True)
```

In [58]:
```
#imputing missing value of embarked
#counting the value appereard most number of times
df['Embarked'].value_counts()
df['Embarked'].fillna('S', inplace = True)
```

In [59]:
```
df['Survived'] = df['Survived'].astype('category')
df['Pclass'] = df['Pclass'].astype('category')
df['Sex'] = df['Sex'].astype('category')
df['Age'] = df['Age'].astype('int')
df['Embarked'] = df['Embarked'].astype('category')
```

In [60]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 11 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
  0PassengerId  891 non-null    int64
  1Survived     891 non-null    category
  2Pclass       891 non-null    category
  3Name         891 non-null    object
  4Sex          891 non-null    category
  5Age          891 non-null    int32
  6SibSp        891 non-null    int64
  7Parch        891 non-null    int64
  8Ticket       891 non-null    object
  9Fare         891 non-null    float64
  10     Embarked     891 non-null    categorydtypes: category(4),
 float64(1), int32(1), int64(3), object(2) memory usage: 49.4+ KB
```

In [61]:
```
df.describe()
```

Out[61]:

	PassengerId	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	891.000000	891.000000
mean	446.000000	29.544332	0.523008	0.381594	32.204208
std	257.353842	13.013778	1.102743	0.806057	49.693429
min	1.000000	0.000000	0.000000	0.000000	0.000000
25%	223.500000	22.000000	0.000000	0.000000	7.910400
50%	446.000000	29.000000	0.000000	0.000000	14.454200
75%	668.500000	35.000000	1.000000	0.000000	31.000000
max	891.000000	80.000000	8.000000	6.000000	512.329200

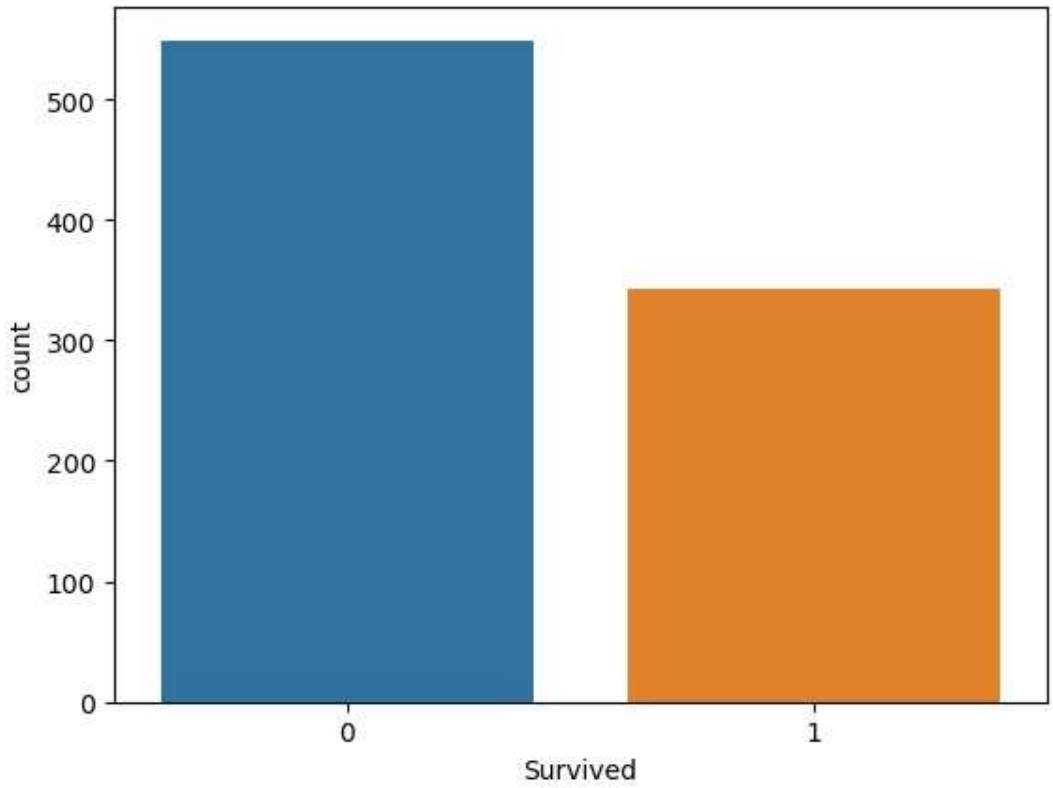
In [62]:
```
df.isnull().sum()
```

Out[62]:
```
PassengerId    0
Survived       0
Pclass         0
Name           0
Sex            0
Age            0
SibSp          0
Parch          0
Ticket         0
```

In

```
Fare          0
Embarked      0
dtype: int64
```

[63]:
```
# Univariate Summary
sns.countplot(x=df['Survived'])
plt.show()
death=round(df['Survived'].value_counts().values[0])
print("Out of 891 , {} people died in the accident".format(death))
```



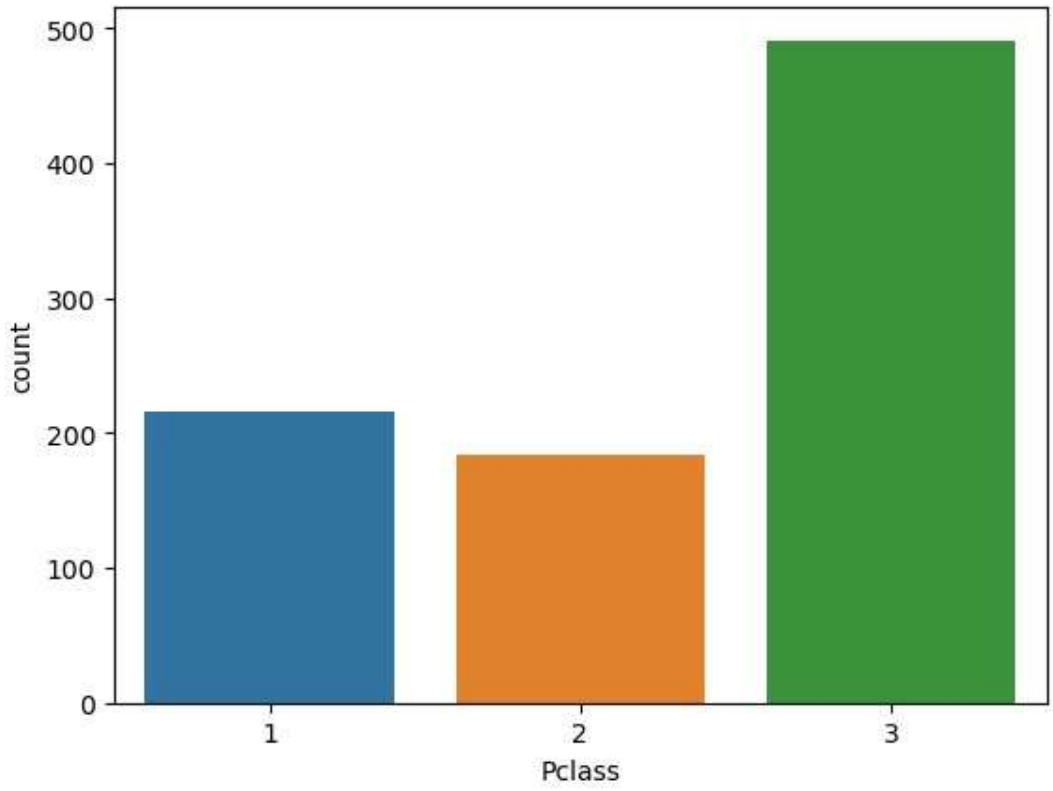

```
Out of 891 , 549 people died in the accident
```

[64]:
```
#print((df['Pclass'].value_counts()/891)*100)
print((df['Pclass'].value_counts())) sns.countplot(x=df['Pclass'])
```

```
3    491
1    216
2    184
Name: Pclass, dtype: int64
```

Out[64]: <Axes: xlabel='Pclass', ylabel='count'>

In

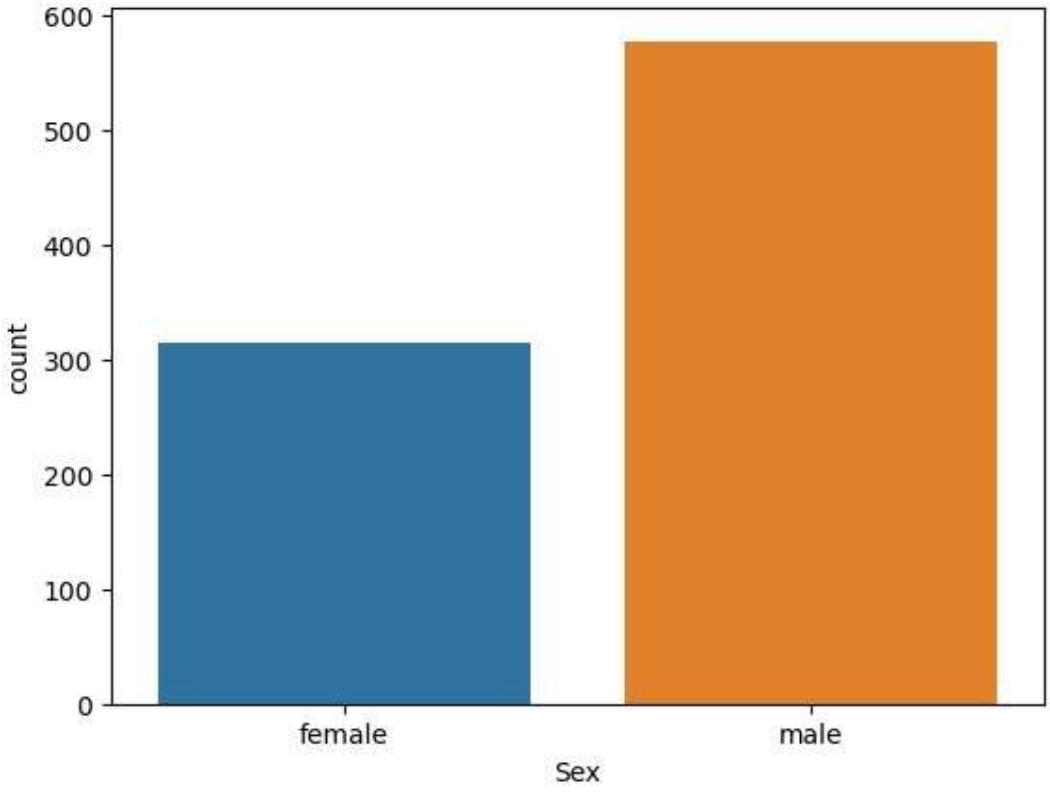


[65]:
```
#print((df['Sex'].value_counts()/891)*100)
print((df['Sex'].value_counts()))
sns.countplot(x=df['Sex'])
```

```
male      577 female
314 Name: Sex, dtype:
int64
```

Out[65]: <Axes: xlabel='Sex', ylabel='count'>

In

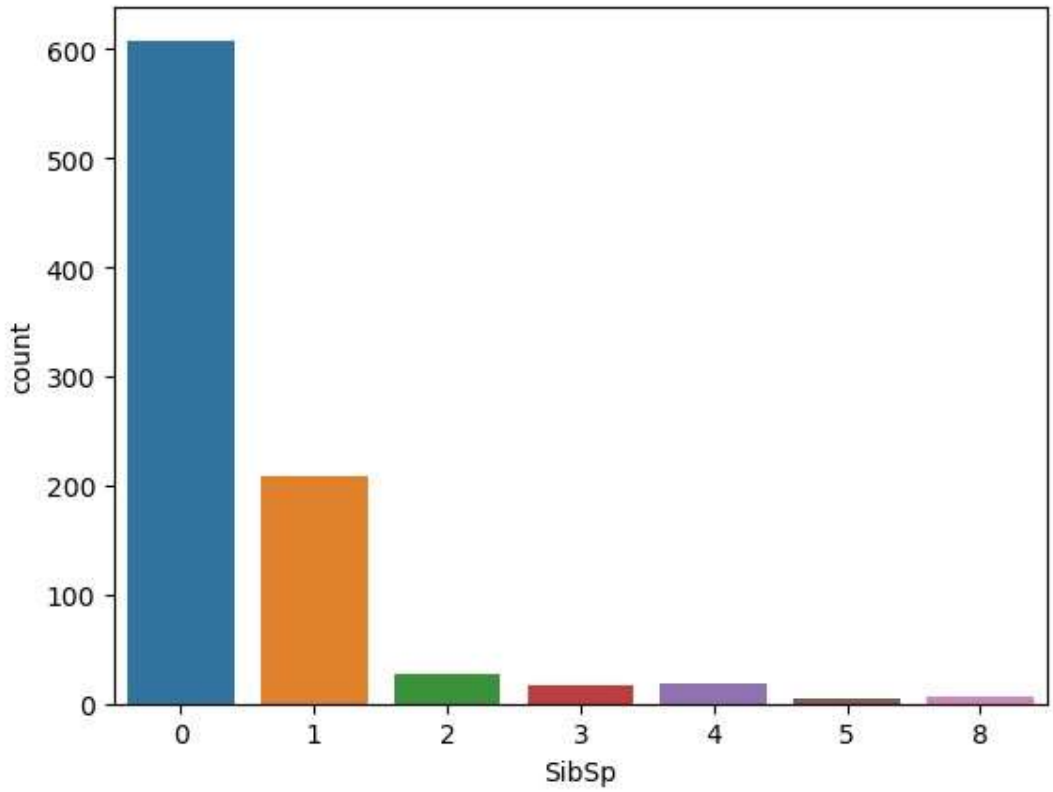


In

[66]:
```
print(df['SibSp'].value_counts())
sns.countplot(x=df['SibSp'])
```

```
0     608
1     209
2      28
4      18
3      16
8       7
5       5
Name: SibSp, dtype: int64
```

Out[66]: <Axes: xlabel='SibSp', ylabel='count'>



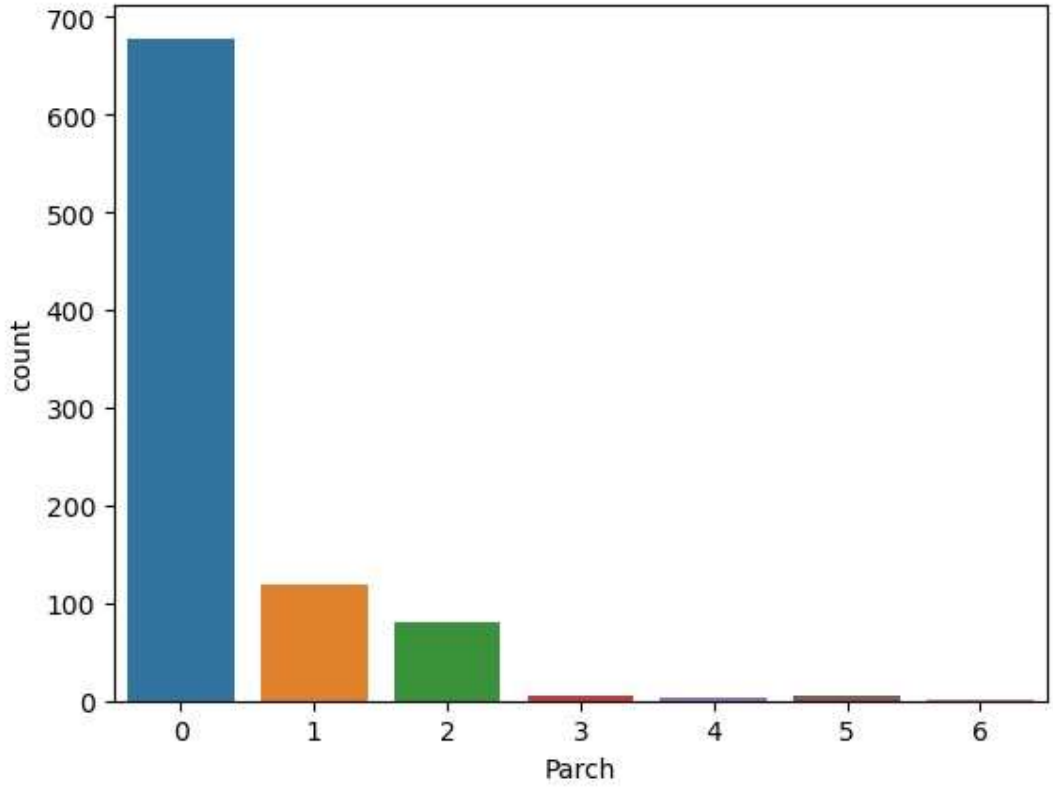
[67]:
```
#print(df['Parch'].value_counts()/891)*100)
print(df['Parch'].value_counts())
sns.countplot(x=df['Parch'])
```

```
0     678
```

In

```
1     118
2     80
5      5
3      5
4      4
6      1
Name: Parch, dtype: int64
```

Out[67]: <Axes: xlabel='Parch', ylabel='count'>

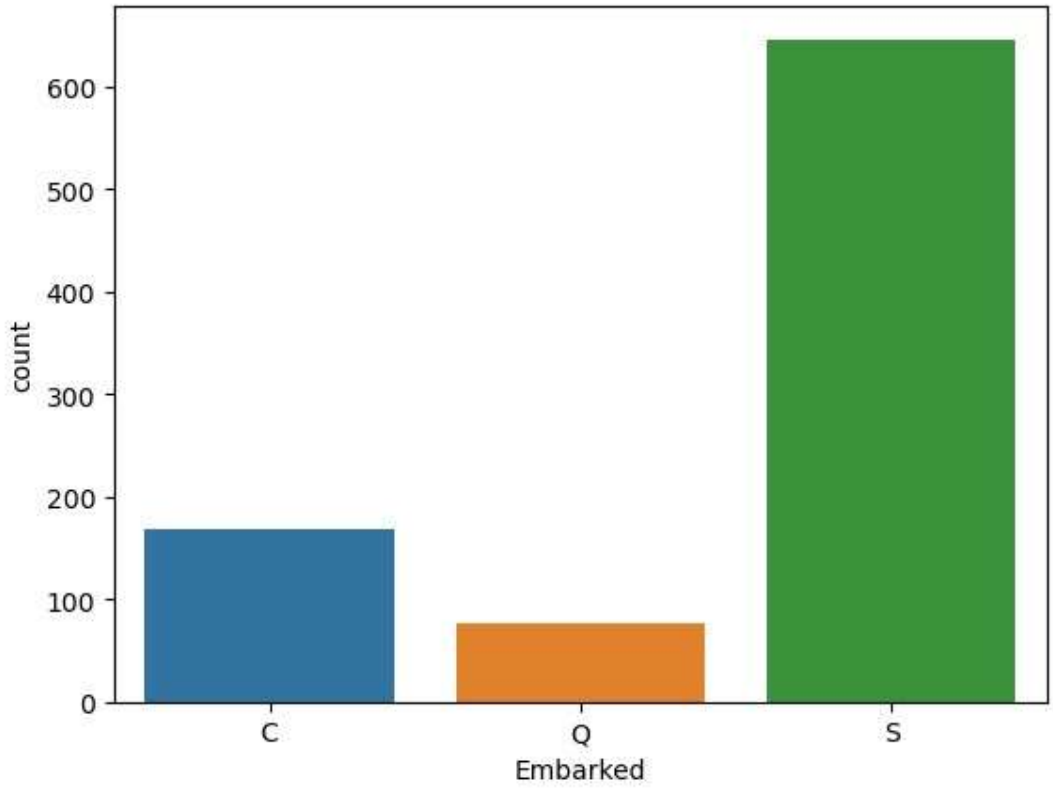


[68]:
```
print(df['Embarked'].value_counts())
sns.countplot(x=df['Embarked'])
```

```
S    646
C    168
Q     77
Name: Embarked, dtype: int64
```

Out[68]: <Axes: xlabel='Embarked', ylabel='count'>

In

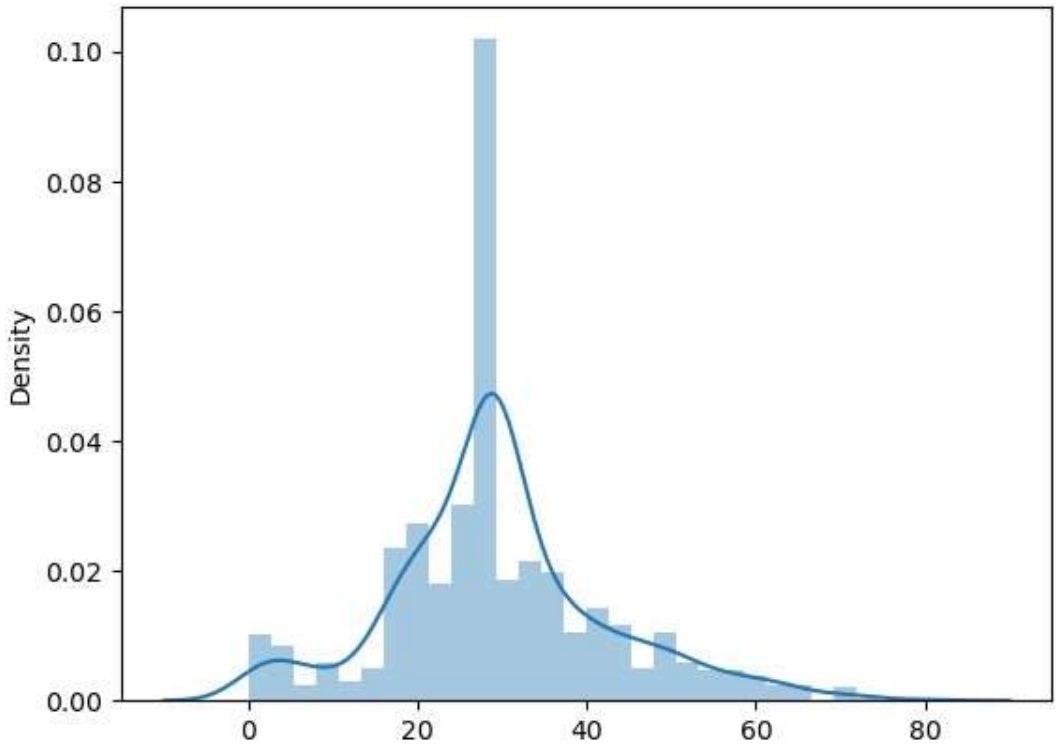


[69]:
```
 # Age
sns.distplot(x=df['Age'])
print(df['Age'].skew())
print(df['Age'].kurt())
```

```
C:\Users\pranavkumar landage\AppData\Local\Temp\ipykernel_12828\282126825
1.py:2: 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 wi
th
```

In

```
similar flexibility) or `histplot` (an axes-level function for histogram
s).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751 (https://
gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751)
sns.distplot(x=df['Age'])

0.45956263424701577
0.9865867453652877
```

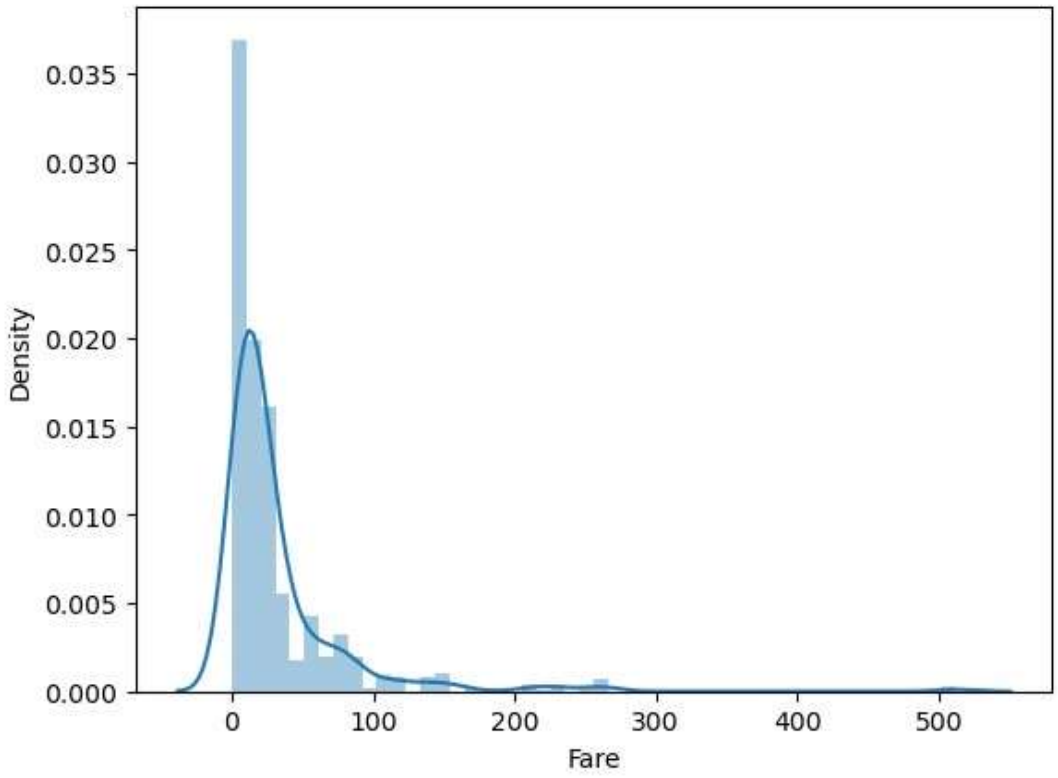


In [70]:
```
#Fare Column
sns.distplot(df['Fare'])
```

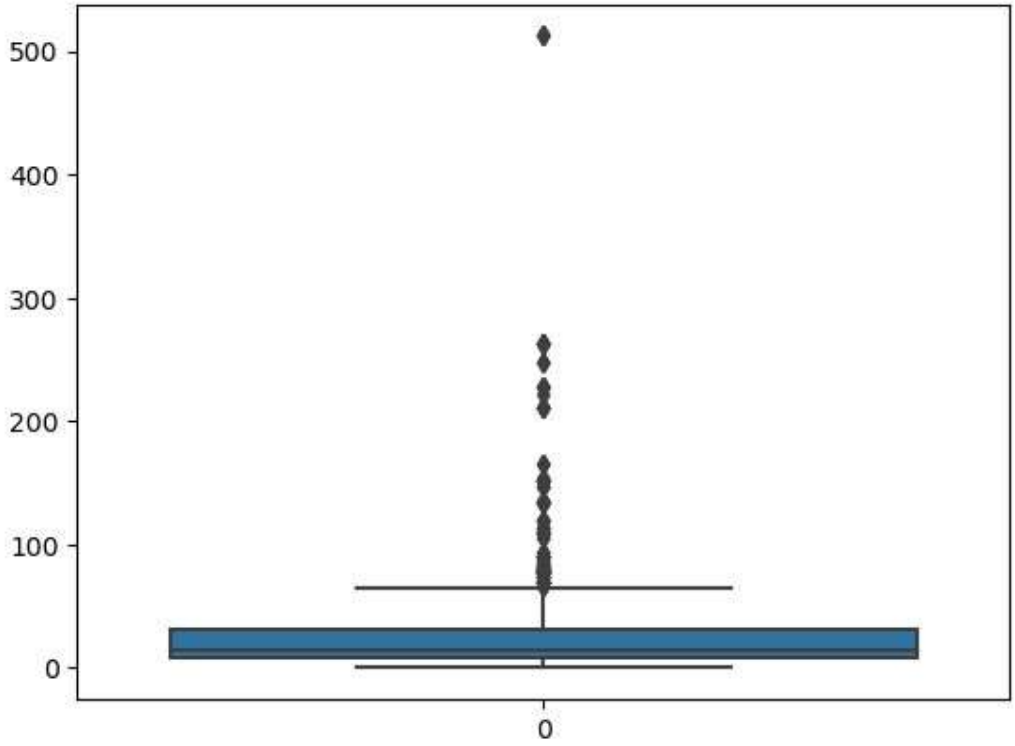
```
C:\Users\pranavkumar landage\AppData\Local\Temp\ipykernel_12828\666492110.
py:2: UserWarning:

`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 wi
th
similar flexibility) or `histplot` (an axes-level function for histogram
s).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751 (https://
gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751)
sns.distplot(df['Fare'])
```

Out[70]: <Axes: xlabel='Fare', ylabel='Density'>

In

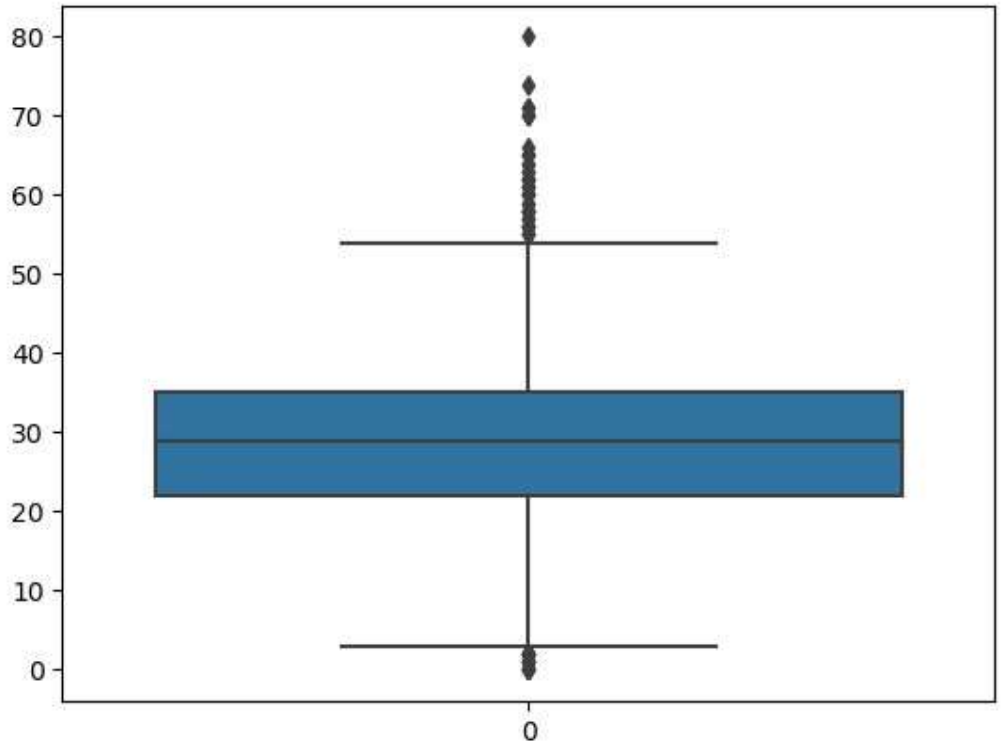


[71]:
```
sns.boxplot(df['Age'])
```



In

Out[71]: <Axes: >



In [72]:
```
sns.boxplot(df['Fare'])
```

Out[72]: <Axes: >

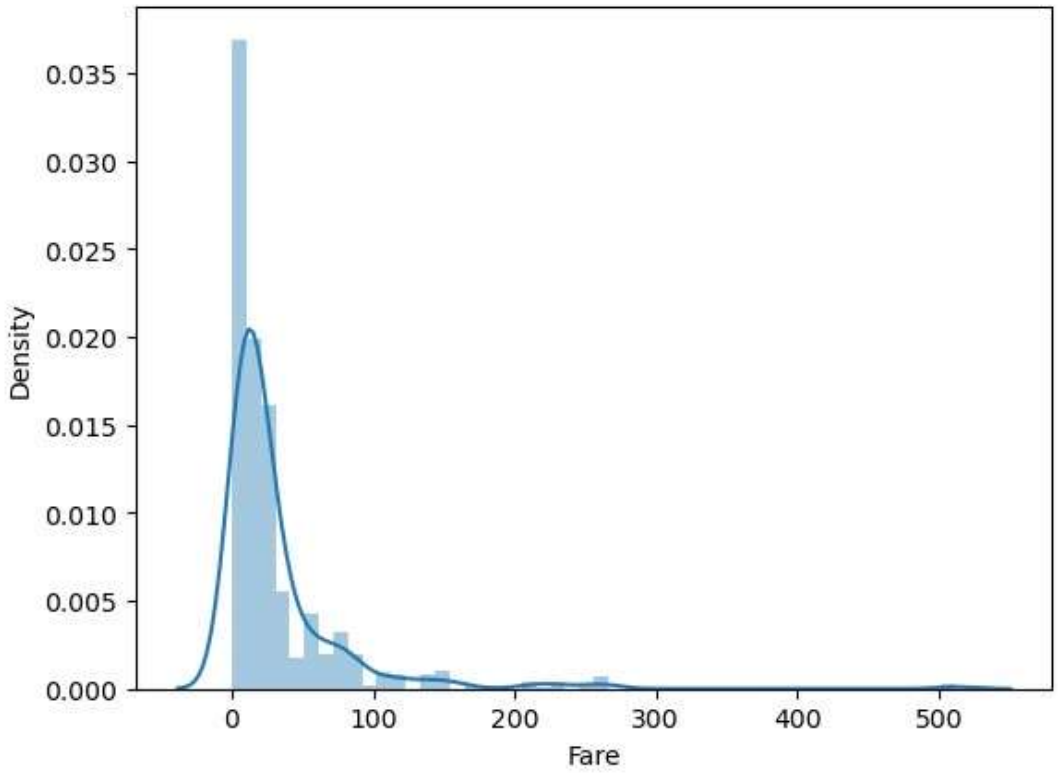
[73]:
```
print("People with age in between 60 and 70 are", df[(df['Age']>60) & (df['A
print("People with age greater than 70 and 75 are", df[ (df['Age']>=70) & (d
print("People with age greater than 75 are", df[df['Age']>75].shape[0])
print('-'*50)
print("People with age between 0 and 1", df[df['Age']<1].shape[0])
```

```
People with age in between 60 and 70 are 15
People with age greater than 70 and 75 are 6
People with age greater than 75 are 1
--------------------------------------------------
People with age between 0 and 1 7
```

In [74]:
```
#Fare Column
sns.distplot(df['Fare'])
```

```
C:\Users\pranavkumar landage\AppData\Local\Temp\ipykernel_12828\666492110.
py:2: 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 wi
th
similar flexibility) or `histplot` (an axes-level function for histogram
s).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751 (https://
gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751)
sns.distplot(df['Fare'])
```

Out[74]: <Axes: xlabel='Fare', ylabel='Density'>

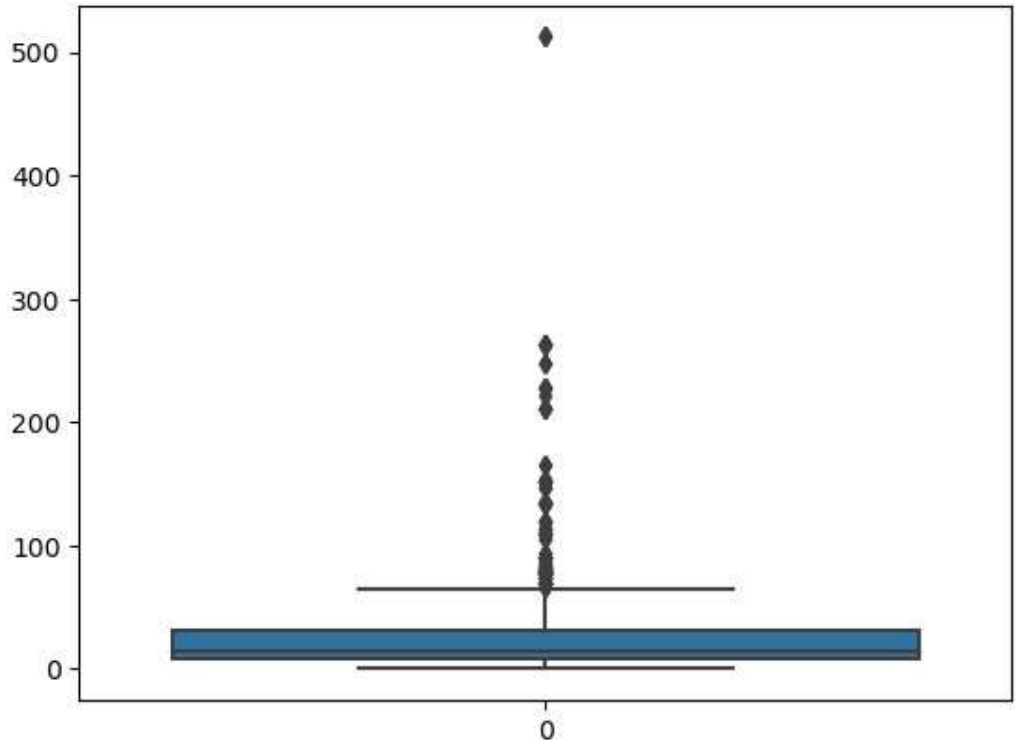


In [75]:
```
print(df['Fare'].skew())
print(df['Fare'].kurt())
```

```
4.787316519674893
33.39814088089868
```

In [76]:
```
sns.boxplot(df['Fare'])
```

Out[76]: <Axes: >

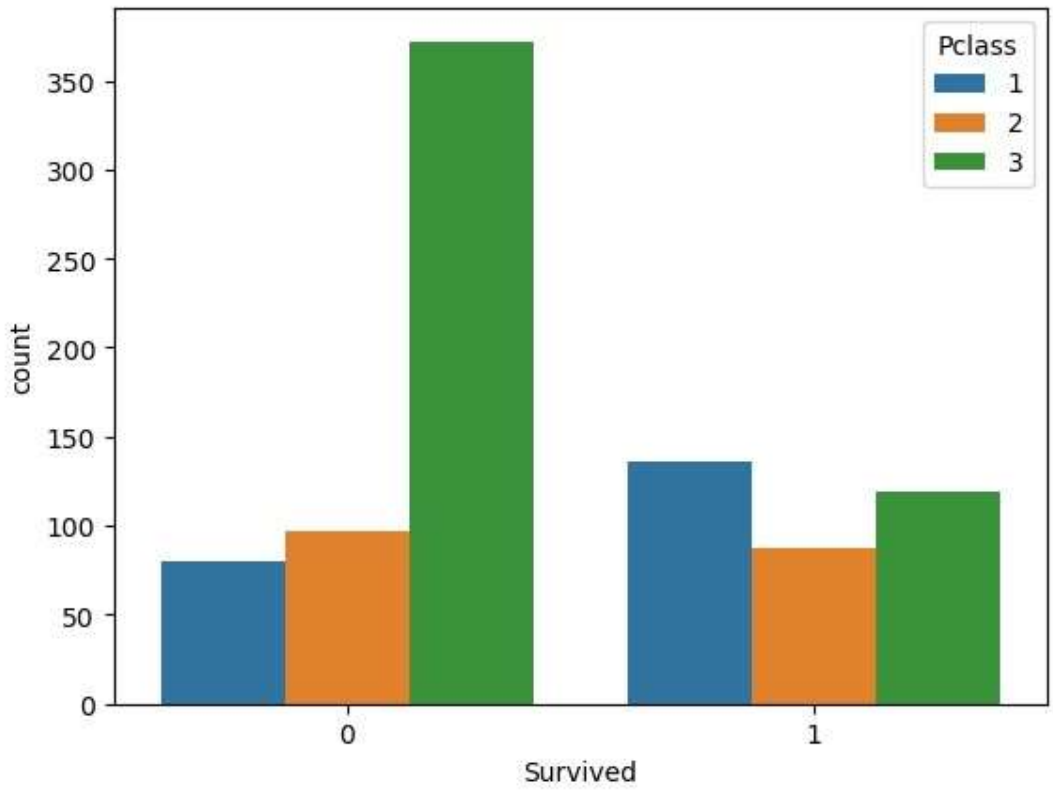


In [77]:
```
print("People with fare in between $200 and $300", df[(df['Fare']>200) & (df
print("People with fare in greater than $308", df[df['Fare']>300].shape[0])
```

```
People with fare in between $200 and $300 17
People with fare in greater than $308 3  [78]: # Multivariate Analysis
#Survival with Pclass
sns.countplot(x=df['Survived'],hue=df['Pclass'])
pd.crosstab(index=df['Pclass'],columns=df['Survived'])
```

Out[78]:

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

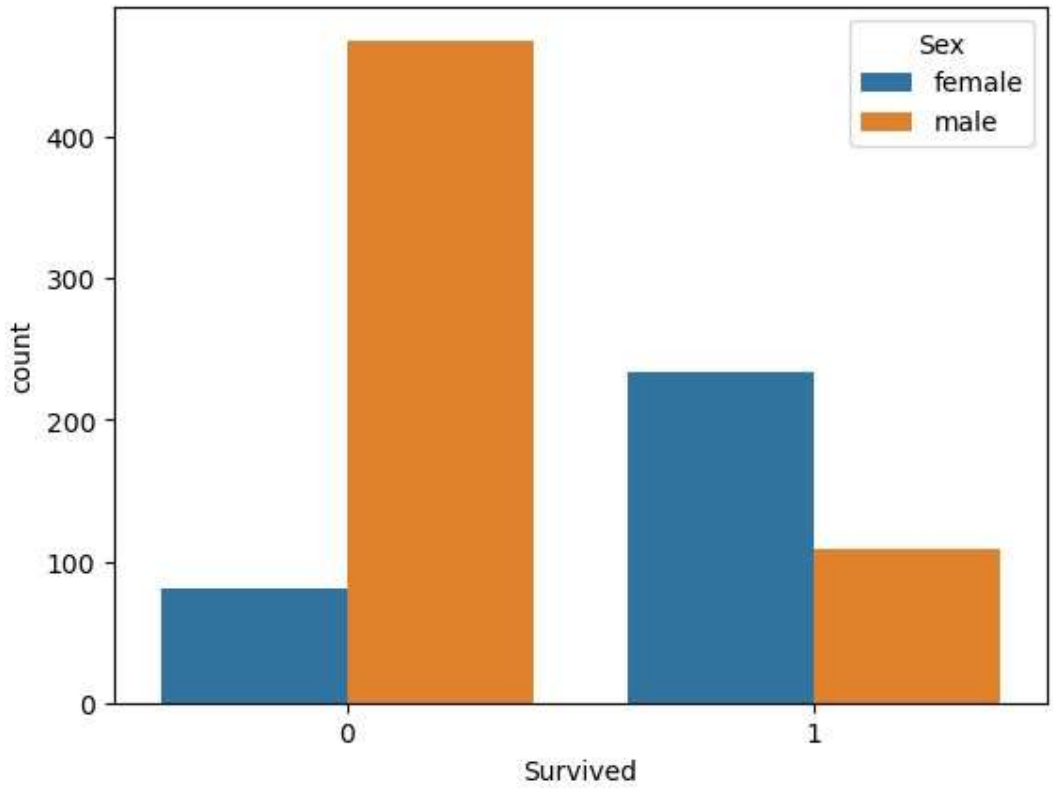


[79]:
```
sns.countplot(x=df['Survived'],hue=df['Sex'])
pd.crosstab(df['Sex'],df['Survived'])
```

Out[79]:

Survived	0	1
Sex		
female	81	233
male	468	109

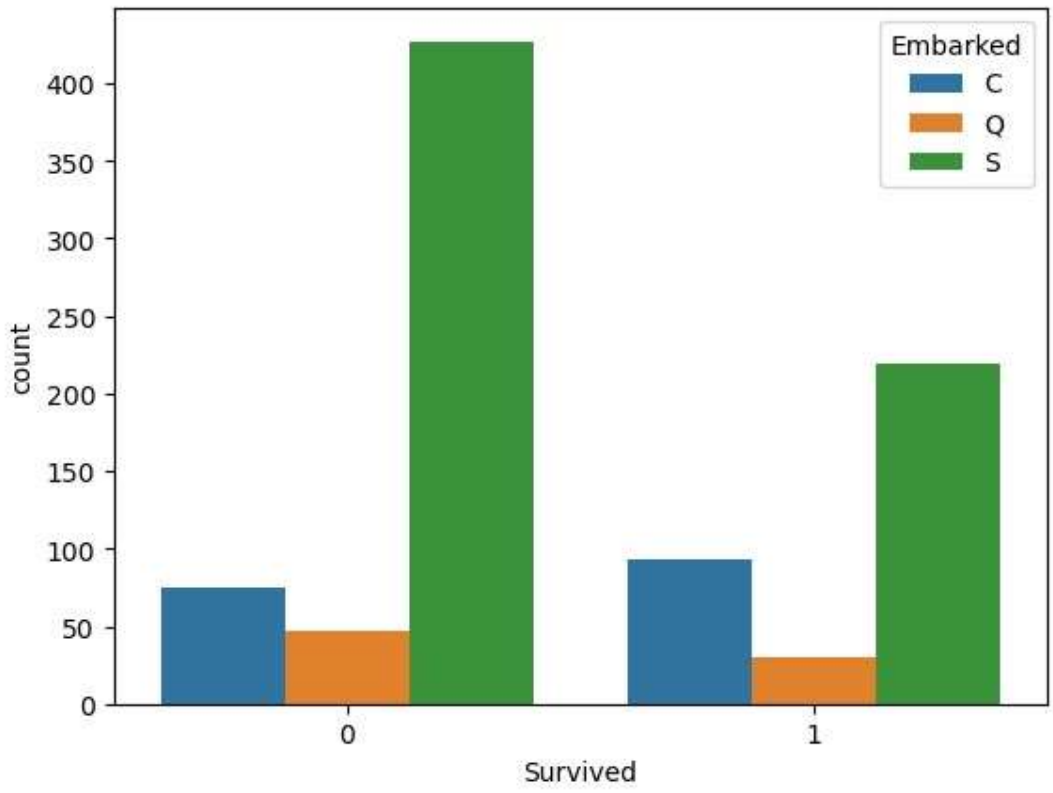
In



[80]:
```
sns.countplot(x=df['Survived'],hue=df['Embarked'])
pd.crosstab(df['Embarked'],df['Survived'])
```

Out[80]:

Survived	0	1
Embarked		
C	75	93
Q	47	30
S	427	219



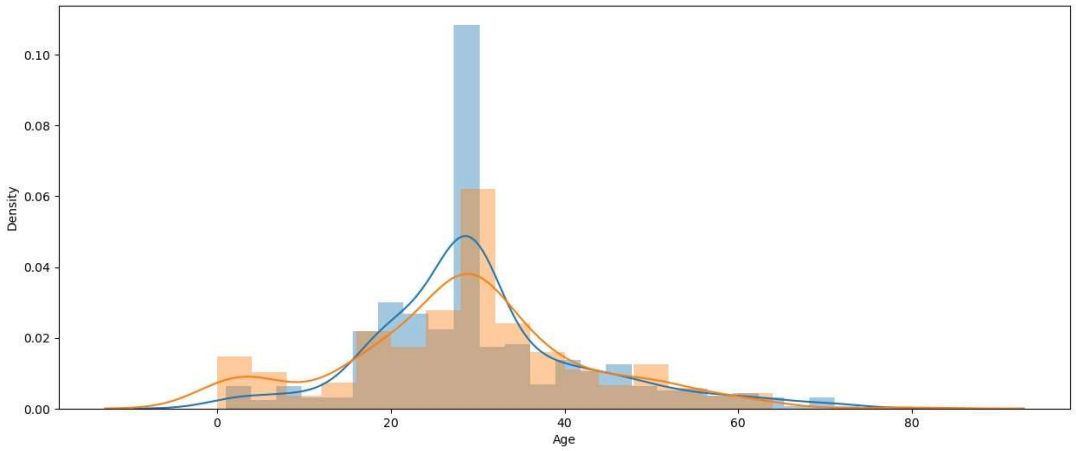
[81]:
```
# survived with age
plt.figure(figsize=(15,6))
sns.distplot(df[df['Survived']==0]['Age'])
sns.distplot(df[df['Survived']==1]['Age'])
```

```
C:\Users\pranavkumar landage\AppData\Local\Temp\ipykernel_12828\229325668
7.py:3: 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 wi
th
similar flexibility) or `histplot` (an axes-level function for histogram
s).

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

  sns.distplot(df[df['Survived']==0]['Age'])
C:\Users\pranavkumar landage\AppData\Local\Temp\ipykernel_12828\229325668
7.py:4: 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 wi
th
similar flexibility) or `histplot` (an axes-level function for histogram
s).
```

In

```
For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751 (https://
gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751)
sns.distplot(df[df['Survived']==1]['Age'])
```

Out[81]: <Axes: xlabel='Age', ylabel='Density'>

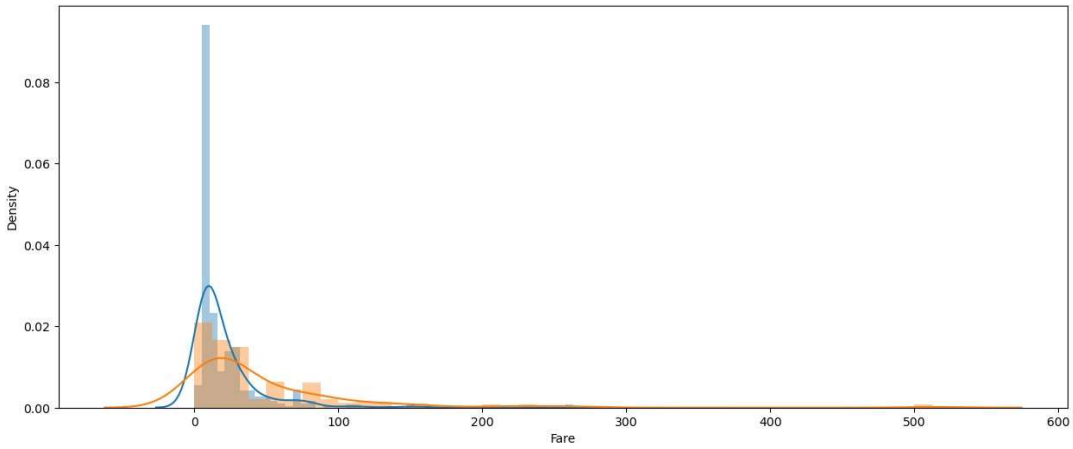


[82]:
```
# survived with Fare
plt.figure(figsize=(15,6))
sns.distplot(df[df['Survived']==0]['Fare'])
sns.distplot(df[df['Survived']==1]['Fare'])
```

```
C:\Users\pranavkumar landage\AppData\Local\Temp\ipykernel_12828\157101336
3.py:3: 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 wi
th
similar flexibility) or `histplot` (an axes-level function for histogram
s).

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

  sns.distplot(df[df['Survived']==0]['Fare'])
C:\Users\pranavkumar landage\AppData\Local\Temp\ipykernel_12828\157101336
3.py:4: 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 wi
th
similar flexibility) or `histplot` (an axes-level function for histogram
s).

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

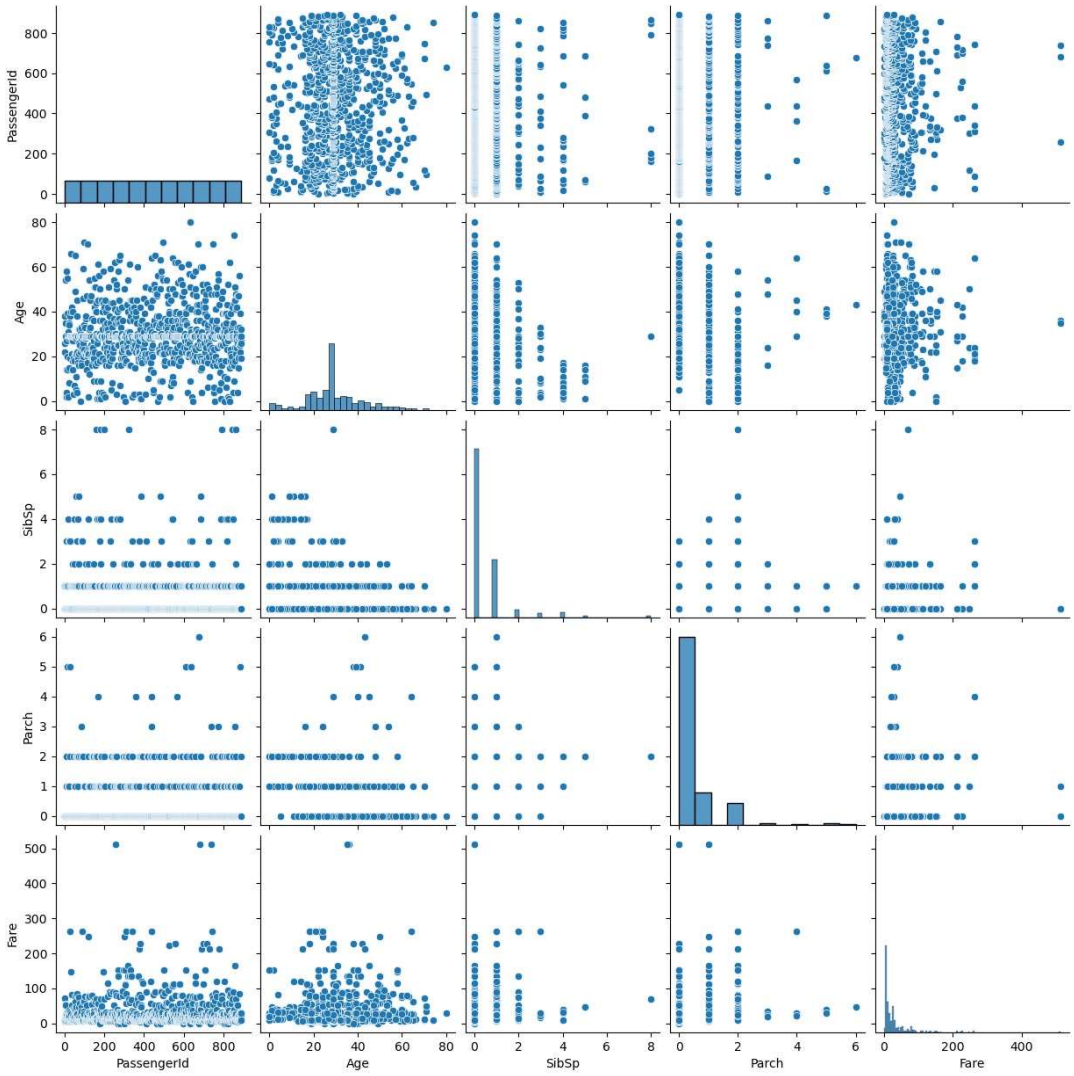
```
gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751)
sns.distplot(df[df['Survived']==1]['Fare'])
```

Out[82]: <Axes: xlabel='Fare', ylabel='Density'>



[83]:
```
sns.pairplot(df)
```

Out[83]: <seaborn.axisgrid.PairGrid at 0x2115a4479d0>

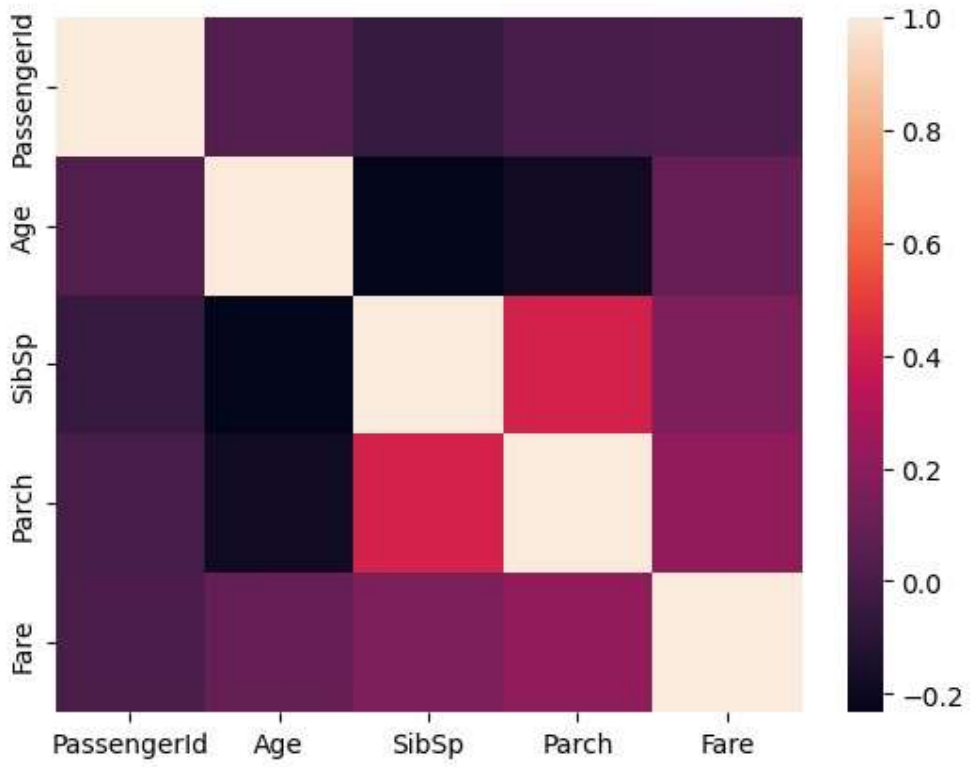


[84]:
```
sns.heatmap(df.corr())
```

In

```
C:\Users\pranavkumar landage\AppData\Local\Temp\ipykernel_12828\58359773.py:1: FutureWarning: The default value of numeric_only 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.
sns.heatmap(df.corr())
```

Out[84]: <Axes: >



In [85]:
```
#handling outlier from age
df = df[df['Age']<df['Age'].mean() + 3 * df['Age'].std()]
df.shape
```

Out[85]: (884, 11)

[86]:
```
#We will create a new column by the name of family which will be the sum of
df['family_size']=df['Parch']+df['SibSp'] df.sample(5)
```

```
C:\Users\pranavkumar landage\AppData\Local\Temp\ipykernel_12828\3619013328.py:2: SettingWithCopyWarning: A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-aview-versus-a-copy)  df['family_size']=df['Parch']+df['SibSp']
```

Out[86]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare

845	846	0	male	42	0	0	C.A. 5547	7.550
492	493	0	male	55	0	0	113787	30.500
757	758	0	male	18	0	0	29108	11.500
694	695	0	male	60	0	0	113800	26.550
392	393	0	male	28	2	0	3101277	7.925

In [87]:
```
#Now we will enginner a new feature by the name of family type
def family_type(number):
 if number==0:
     return "Alone"
 elif number >0 and number <= 4:
     return "Medium"
 else:
     return"Large"
```

[88]:
```
df.head()
```

Out[88]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
0	1	0	3	Braund, Mr. Owen Harris	male	22	1	0	A/5 21171	7.2500
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38	1	0	PC 17599	71.2833
2	3	1	3	Heikkinen, Miss. Laina	female	26	0	0	STON/O2. 3101282	7.9250
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35	1	0	113803	53.1000
4	5	0	3	Allen, Mr. William Henry	male	35	0	0	373450	8.0500

In

Conclusion

Chance of female survival is higher than male survival

Travelling in Pclass 3 was deadliest

Somehow, people going to C survived more

People in the age range of 20 to 40 had a higher chance of not surviving

People travelling with smaller familes had a higher chance of surviving the accident in comparison to people with large families

Thank You

In [ ]: