Prodigy Infotech Internship Task 2

Name : Uditanshu

Task : EDA of Titanic dataset

In [377]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [378]:
```
df = pd.read_csv('train.csv')
df
```

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
...	...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	Q

In [379]:
```
df.head()
```

Out[379]:

	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 [380]:
```
df.tail()
```

Out[380]:

	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 [381]:
```
df.shape
```

Out[381]:
```
(891, 12)
```

In [382]:
```
df.columns.values
```

Out[382]:
```
array(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp',
       'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'], dtype=object)
```

In [383]:
```
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 [384]:
```
df.isnull().sum()
```

Out[384]:
```
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 [385]:
```
# dropping the cabin column
df.drop(columns=['Cabin'],inplace=True)
```

In [386]:
```
#imputing the missing value with the mean
df['Age'].fillna(df['Age'].mean() , inplace = True)
```

In [387]:
```
#imputing missing value of embarked
#counting the value appereard most number of times
df['Embarked'].value_counts()

df['Embarked'].fillna('S', inplace = True)
```

In [388]:
```
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 [389]:
```
df.info()
```

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

In [390]:
```
df.describe()
```

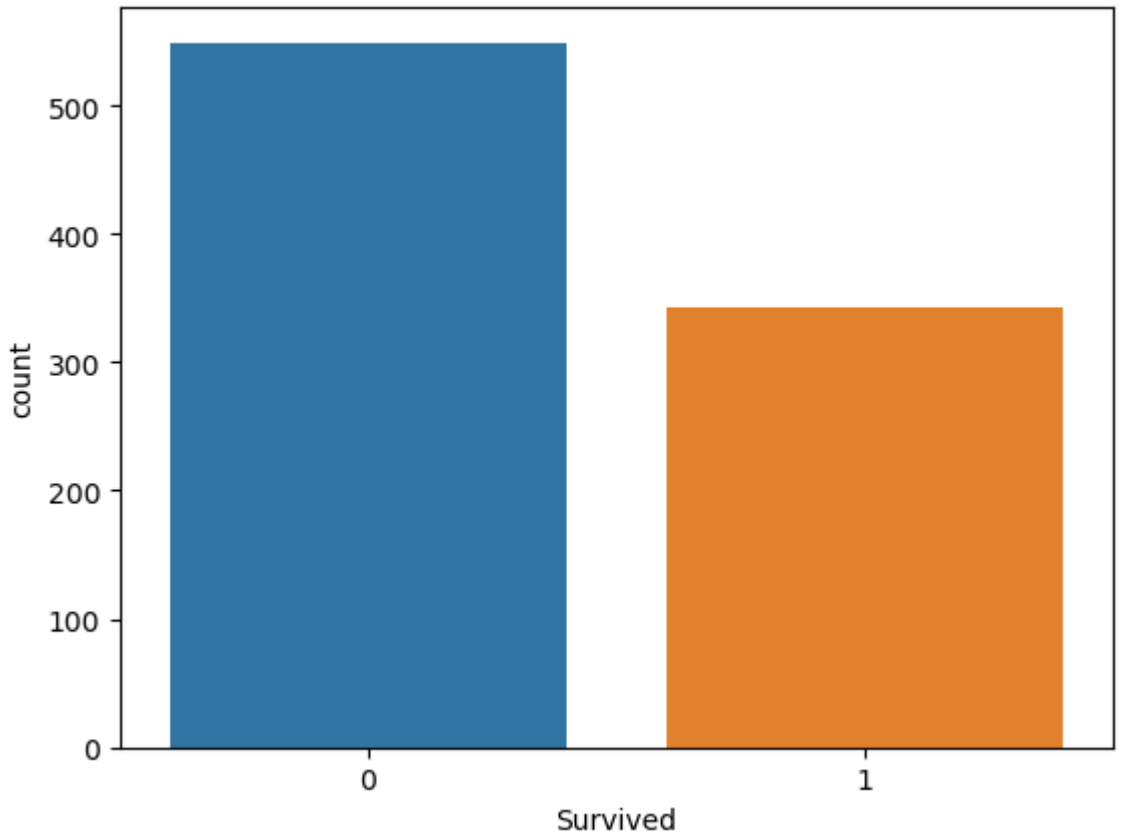
Out[390]:

	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 [391]:
```
df.isnull().sum()
```

Out[391]:
```
PassengerId    0
Survived       0
Pclass         0
Name           0
Sex            0
Age            0
SibSp          0
Parch          0
Ticket         0
Fare           0
Embarked       0
dtype: int64
```

In [392]:
```
# 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))
```

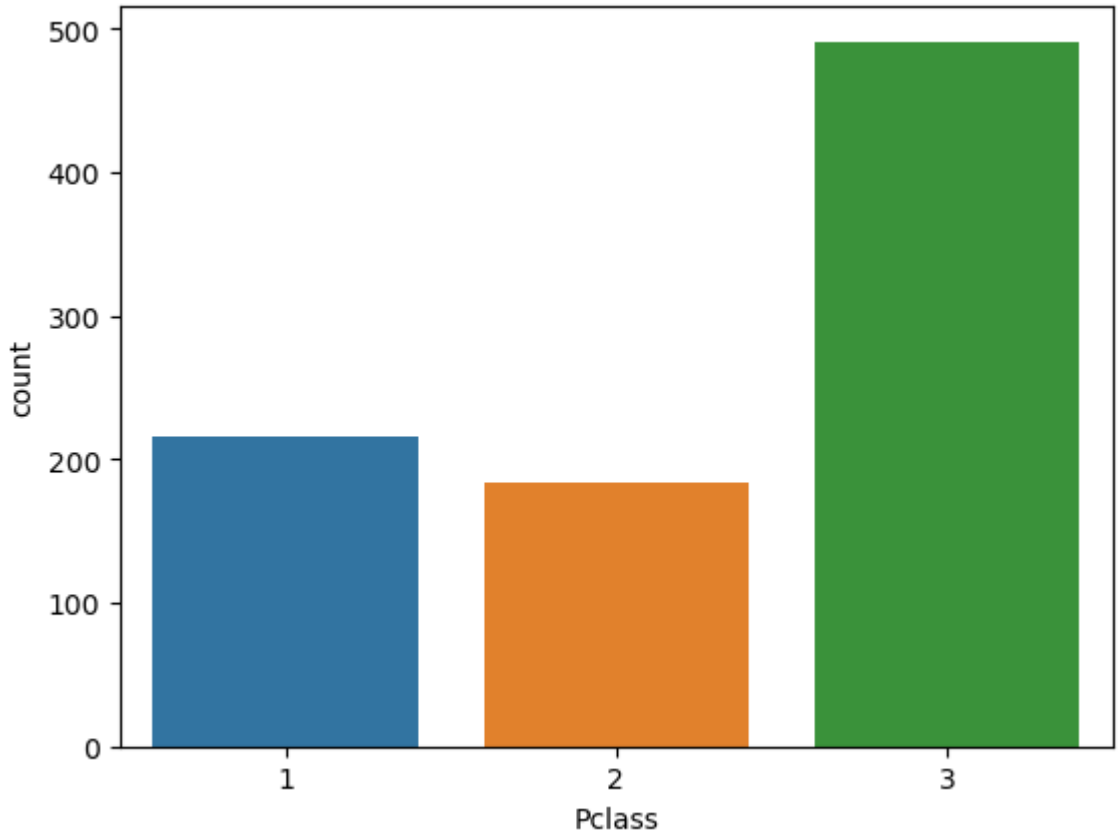



```
Out of 891 , 549  people died in the accident
```

In [393]:
```
#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
```

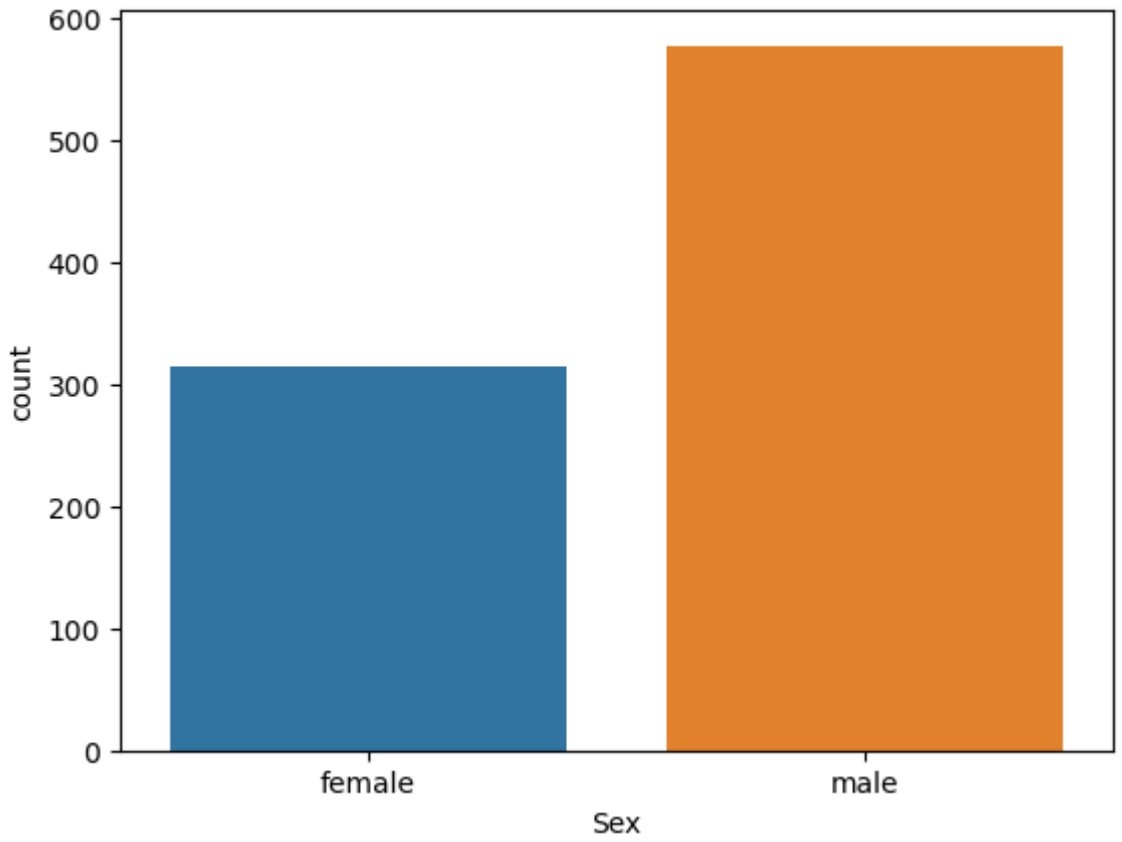
Out[393]: <Axes: xlabel='Pclass', ylabel='count'>



In [394]:
```
#rint((df['Sex'].value_counts()/891)*100)
print((df['Sex'].value_counts()))
sns.countplot(x=df['Sex'])
```

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

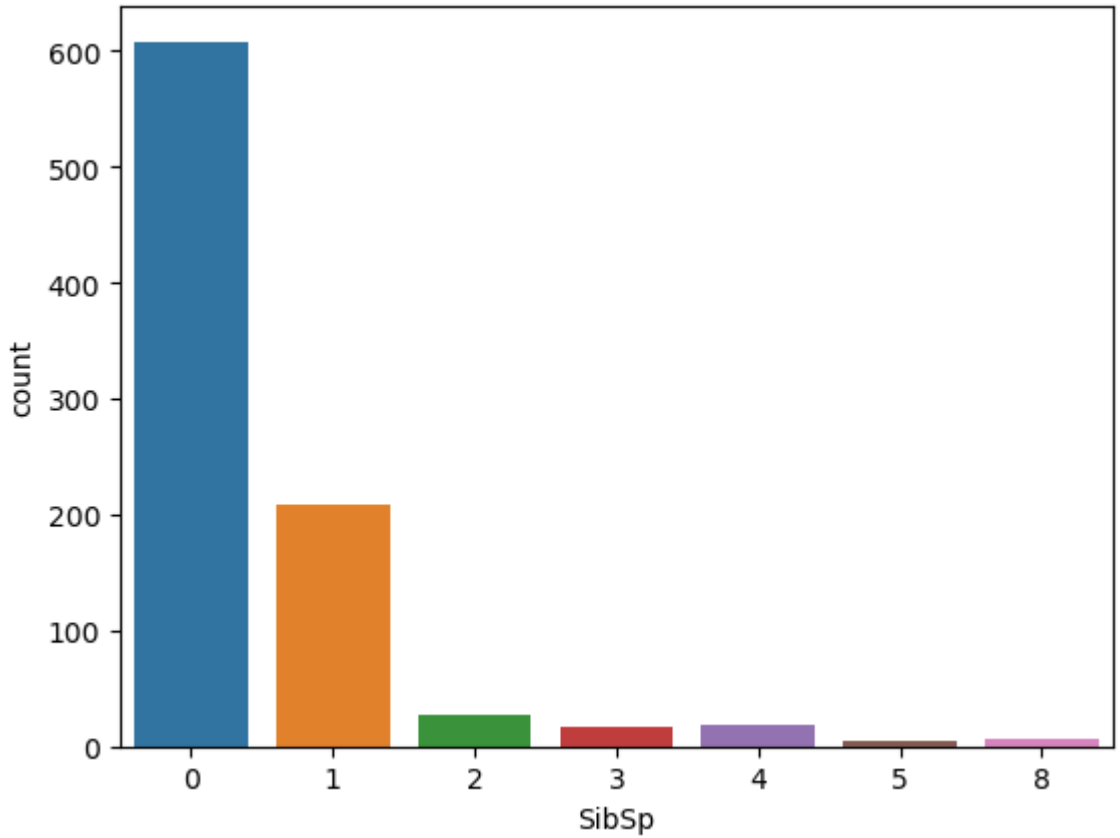
Out[394]: <Axes: xlabel='Sex', ylabel='count'>



In [395]:
```
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
```

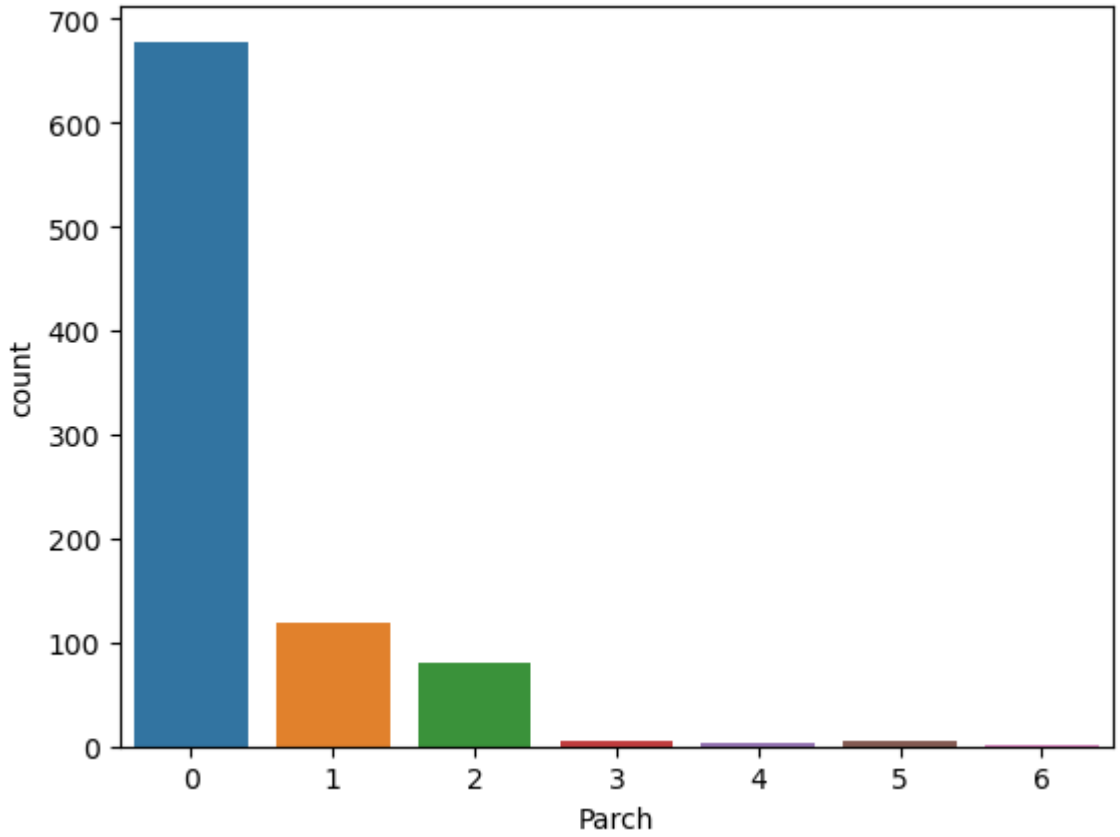
Out[395]: <Axes: xlabel='SibSp', ylabel='count'>



In [396]:
```
#print(df['Parch'].value_counts()/891)*100)
print(df['Parch'].value_counts())
sns.countplot(x=df['Parch'])
```

```
0    678
1    118
2     80
5      5
3      5
4      4
6      1
Name: Parch, dtype: int64
```

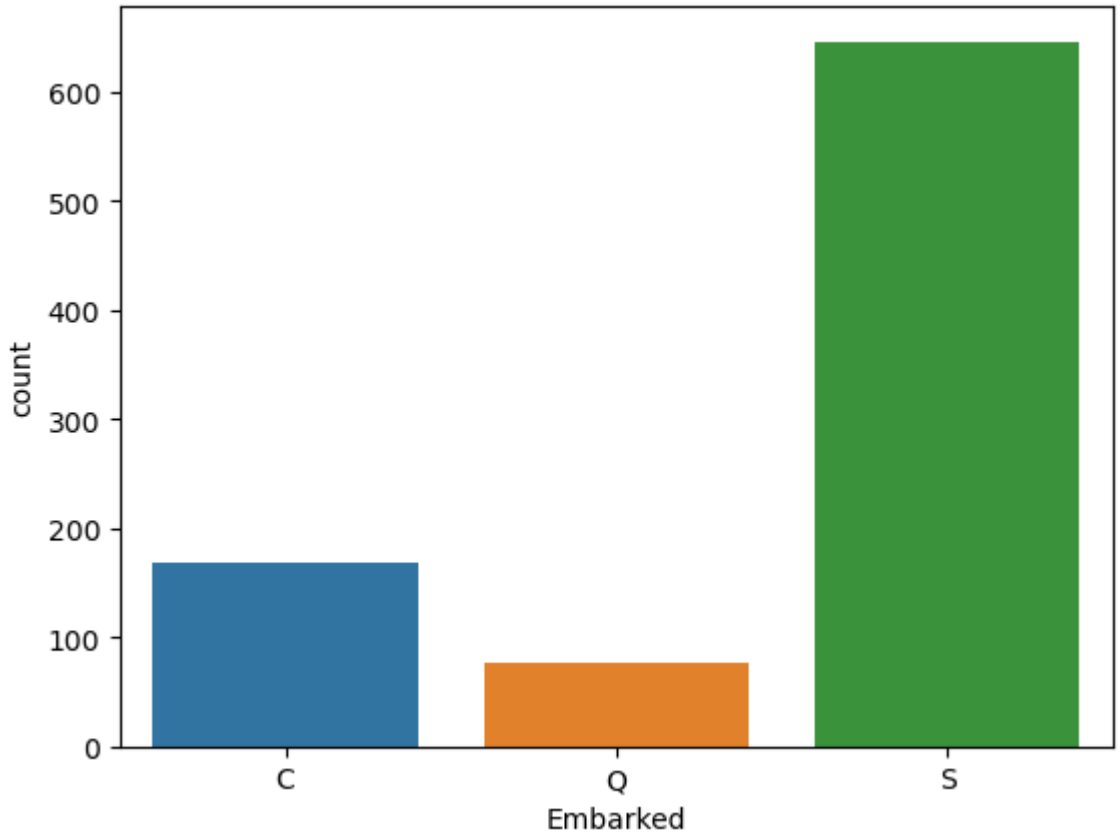
Out[396]: <Axes: xlabel='Parch', ylabel='count'>



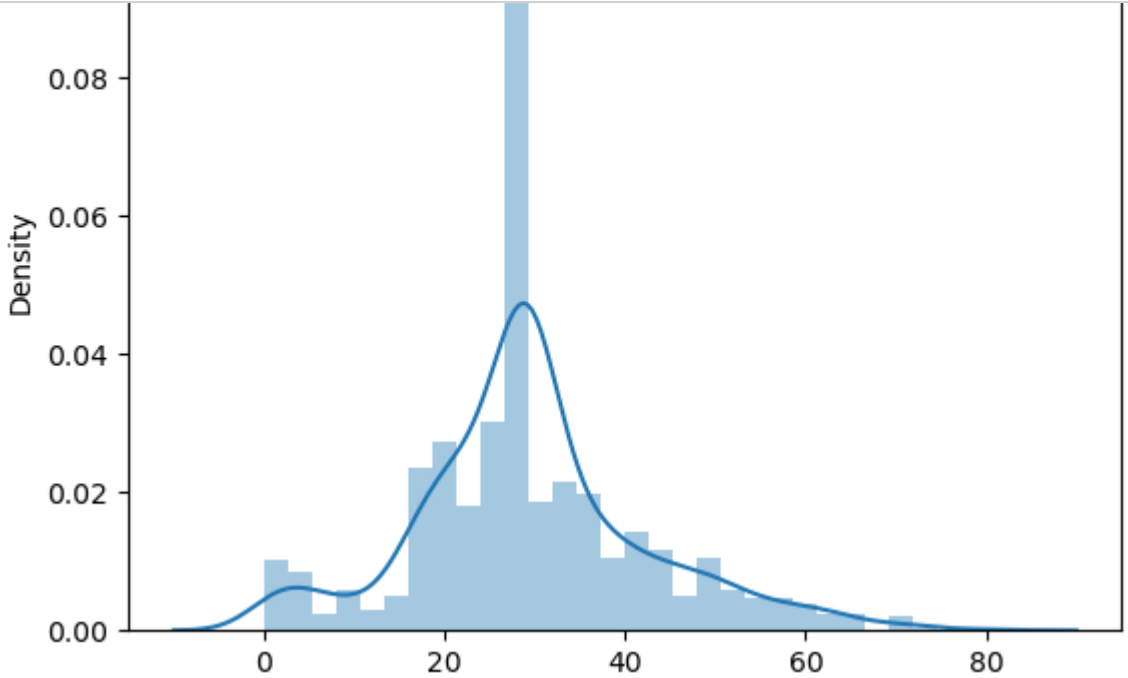
In [397]:
```
print(df['Embarked'].value_counts())
sns.countplot(x=df['Embarked'])
```

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

Out[397]: <Axes: xlabel='Embarked', ylabel='count'>



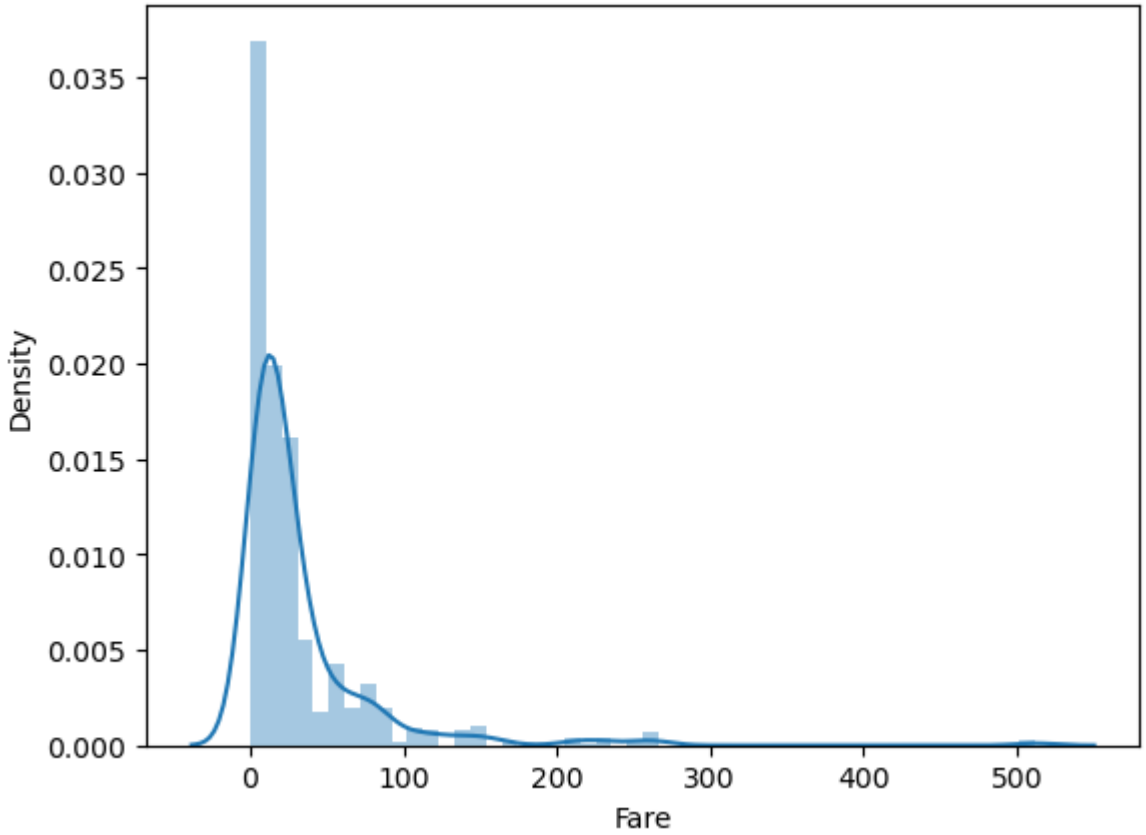
In [398]:
```
# Age

sns.distplot(x=df['Age'])

print(df['Age'].skew())

print(df['Age'].kurt())
```



In [399]:
```
#Fare Column
sns.distplot(df['Fare'])
```

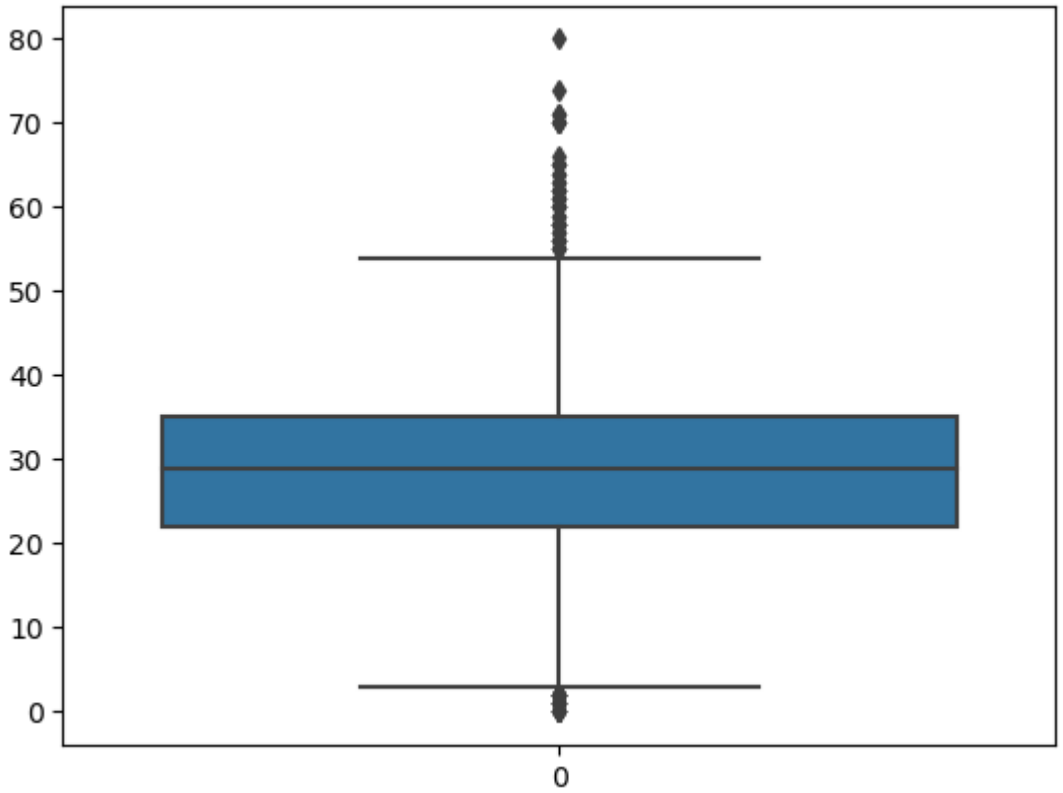
```
C:\Users\pranavkumar landage\AppData\Local\Temp\ipykernel_6768\666492110.py:2: UserWarning:

`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 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 (https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751)

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

Out[399]:
```
<Axes: xlabel='Fare', ylabel='Density'>
```


0.035
0.030
0.025
0.020
0.015
0.010
0.005
0.000
Density
0
100
200
300
400
500
Fare

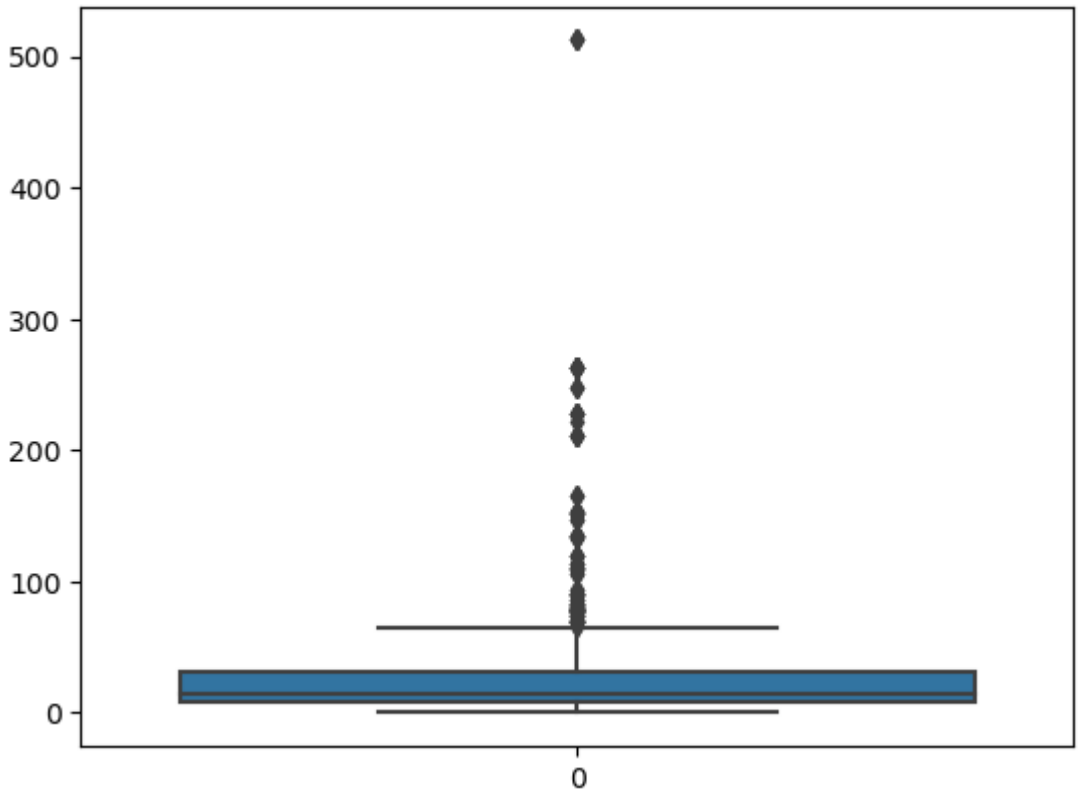
In [400]:
```
sns.boxplot(df['Age'])
```

Out[400]: <Axes: >



In [401]:
```
sns.boxplot(df['Fare'])
```

Out[401]: <Axes: >



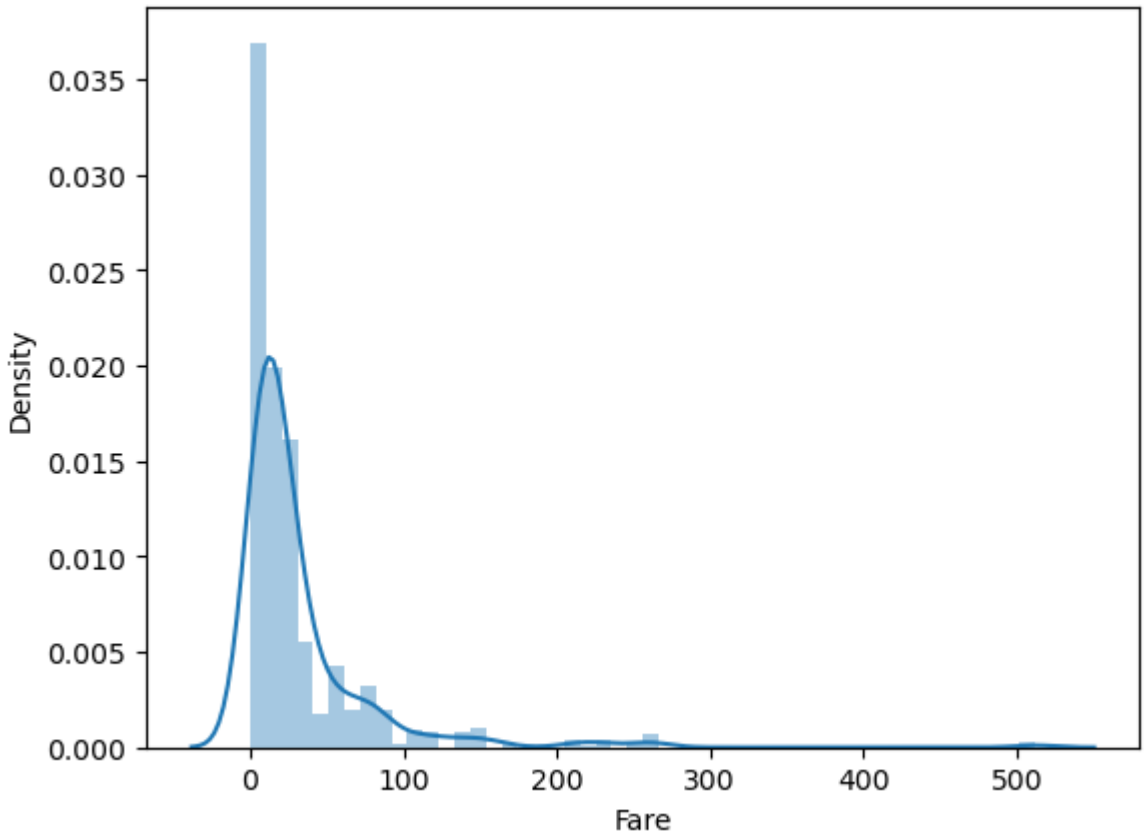
In [402]:
```
print("People with age in between 60 and 70 are", df[(df['Age']>60) & (df['Age']<70)].shape[0])
print("People with age greater than 70 and 75 are", df[ (df['Age']>=70) & (df['Age']<=75)].shape[0])
print("People with age greater than 75 are", df[df['Age']>75].shape[0])

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 [403]:
```
#Fare Column
sns.distplot(df['Fare'])
```

```
C:\Users\pranavkumar landage\AppData\Local\Temp\ipykernel_6768\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 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 (https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751)

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

Out[403]:
```
<Axes: xlabel='Fare', ylabel='Density'>
```

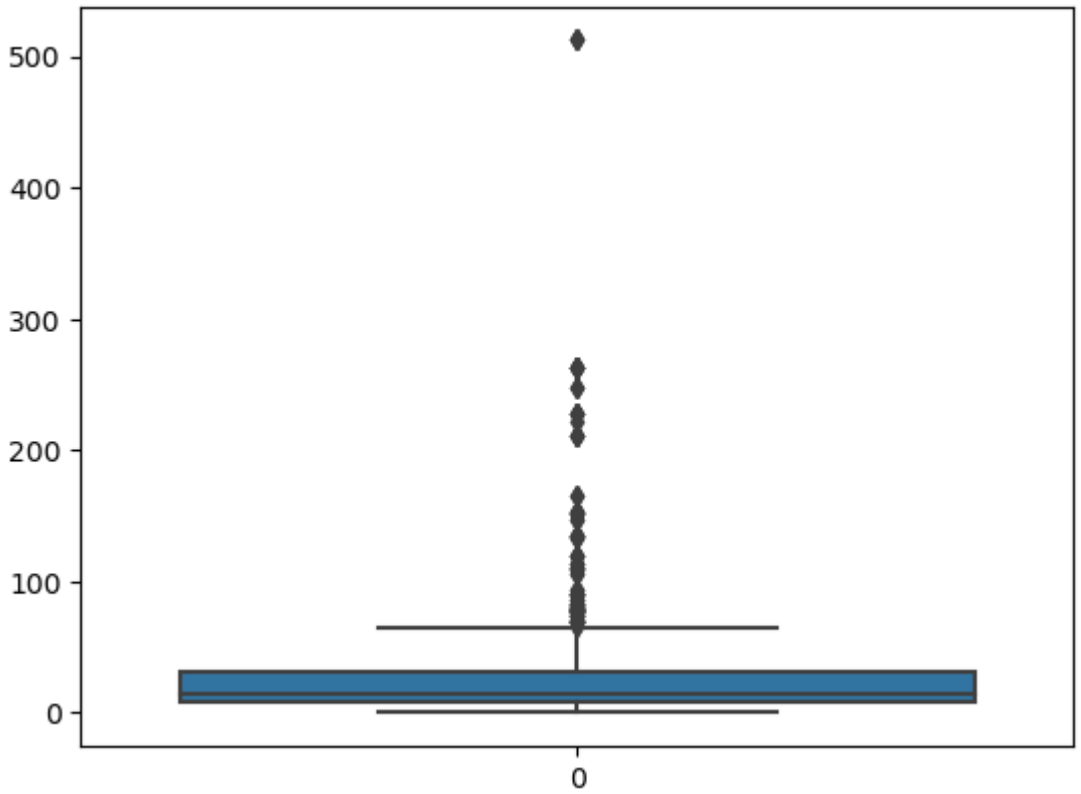


In [404]:
```
print(df['Fare'].skew())
print(df['Fare'].kurt())
```

```
4.787316519674893
33.39814088089868
```

In [405]:
```
sns.boxplot(df['Fare'])
```

Out[405]: <Axes: >



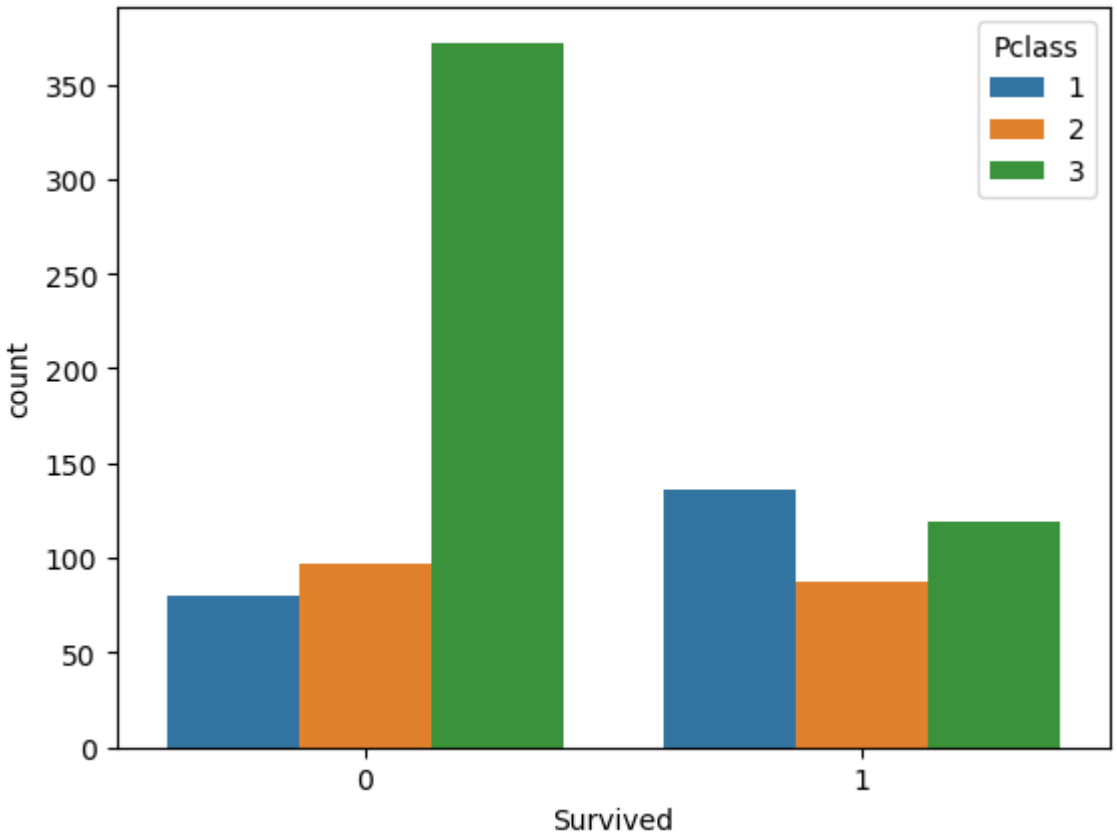
In [406]:
```
print("People with fare in between $200 and $300", df[(df['Fare']>200) & (df['Fare']<300)].shape[0])
print("People with fare in greater than $308", df[df['Fare']>300].shape[0])
```

```
People with fare in between $200 and $300 17
People with fare in greater than $308 3
```

In [407]:
```
# Multivariate Analysis

#Survival with Pclass

sns.countplot(x=df['Survived'],hue=df['Pclass'])

pd.crosstab(index=df['Pclass'],columns=df['Survived'])
```

Out[407]:

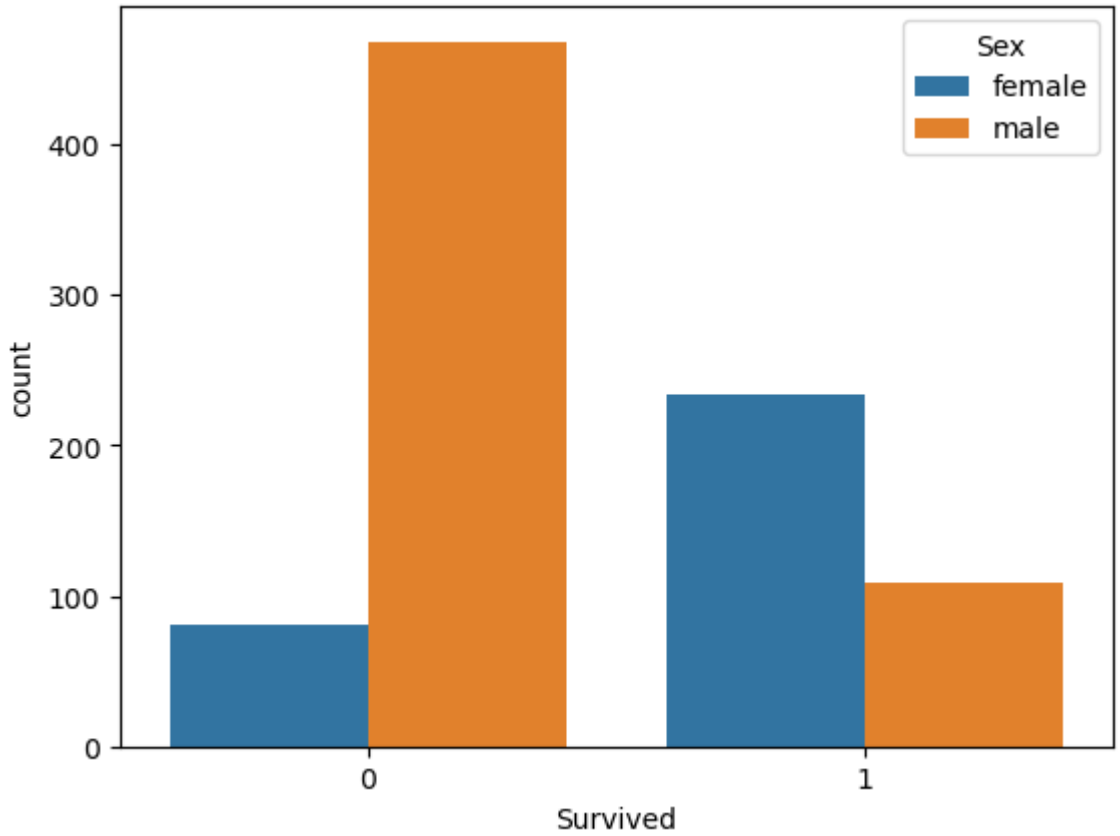
Survived	0	1
Pclass		
1	80	136
2	97	87
3	372	119


Pclass
1
2
3
350
300
250
200
150
100
50
0
count
0
1
Survived

In [408]:
```
sns.countplot(x=df['Survived'],hue=df['Sex'])
pd.crosstab(df['Sex'],df['Survived'])
```

Out[408]:

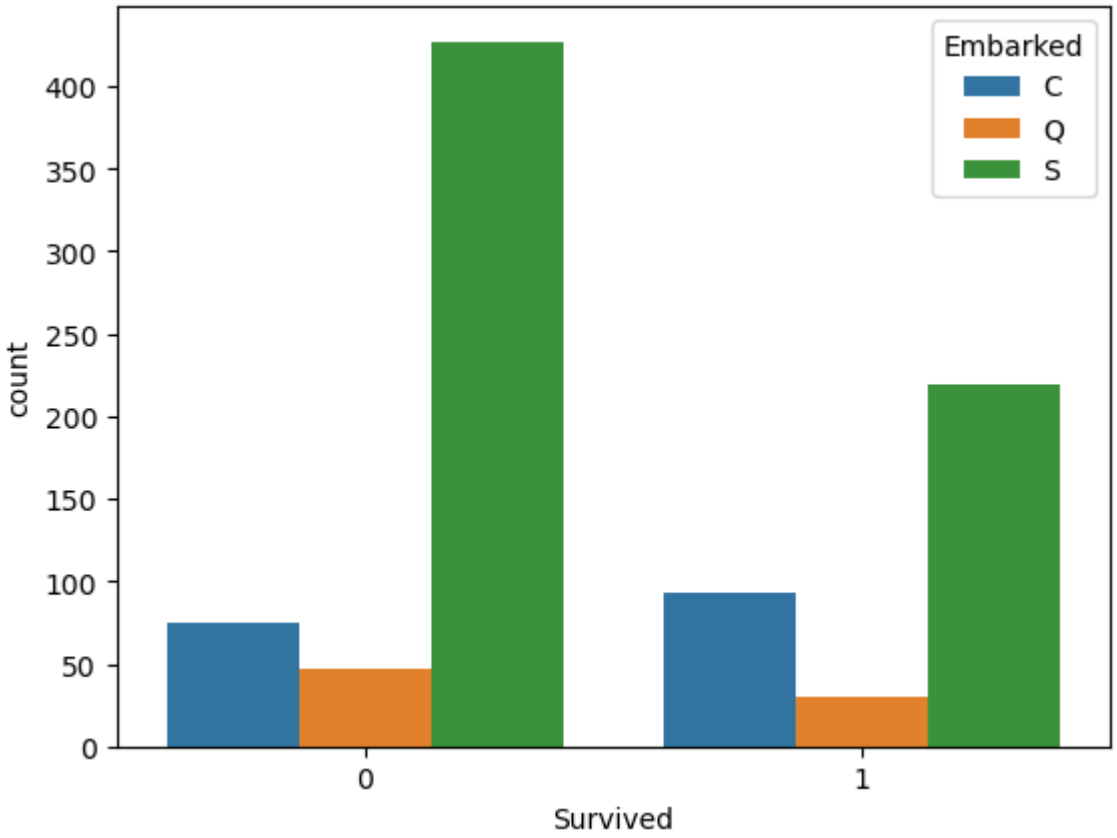
Survived	0	1
Sex		
female	81	233
male	468	109



In [409]:
```
sns.countplot(x=df['Survived'],hue=df['Embarked'])
pd.crosstab(df['Embarked'],df['Survived'])
```

Out[409]:

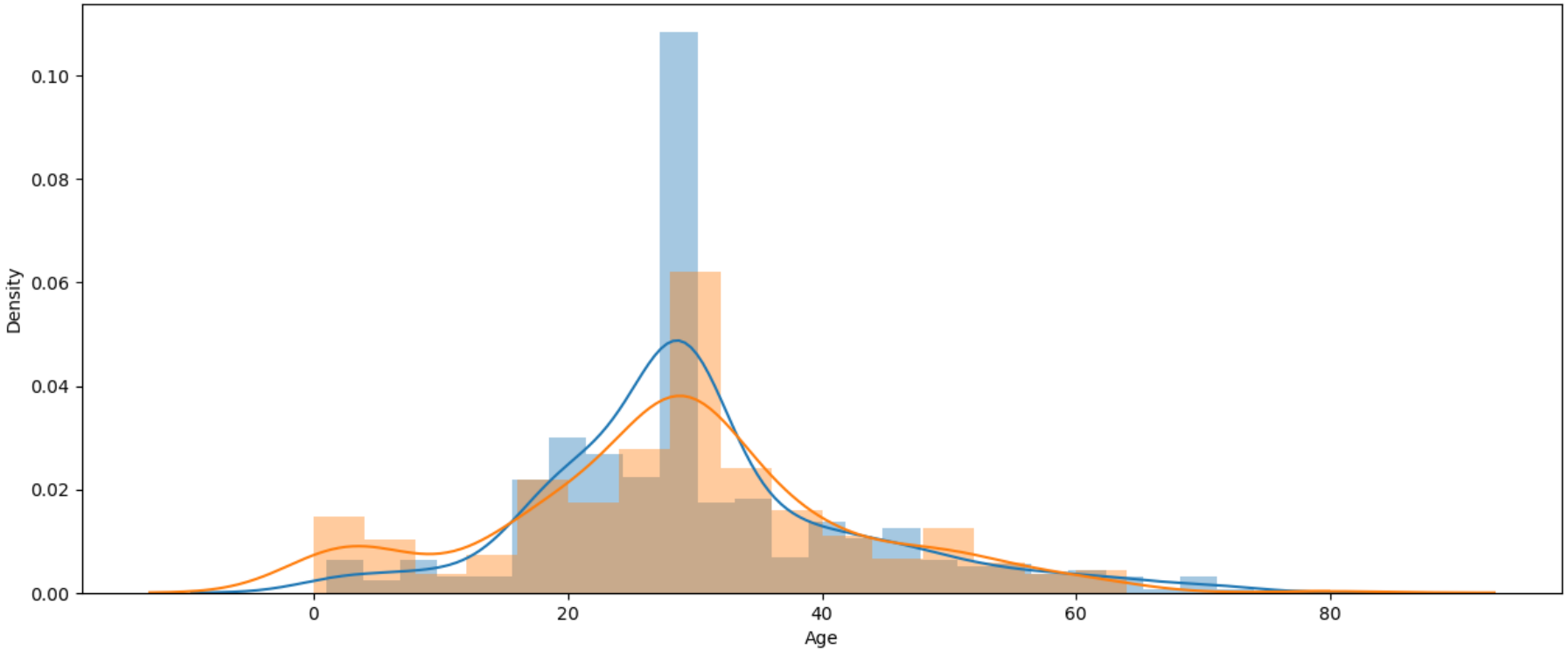
Survived	0	1
Embarked		
C	75	93
Q	47	30
S	427	219



In [410]:
```
# 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_6768\1300477796.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 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 (https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751)

  sns.distplot(df[df['Survived']==0]['Age'])
C:\Users\pranavkumar landage\AppData\Local\Temp\ipykernel_6768\1300477796.py:5: 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 (https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751)

  sns.distplot(df[df['Survived']==1]['Age'])
```

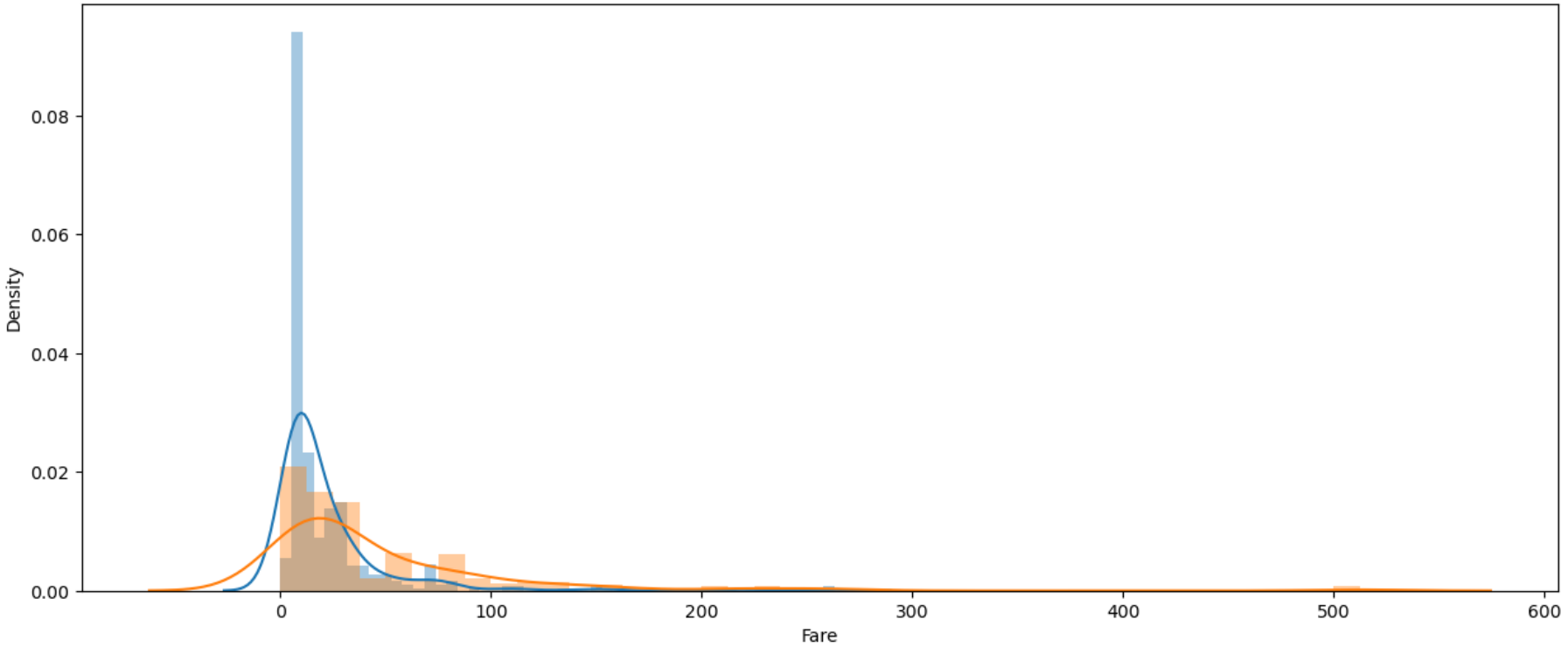
Out[410]:
```
<Axes: xlabel='Age', ylabel='Density'>
```


0.10
0.08
0.06
0.04
0.02
0.00
Density
0
20
40
60
80
Age

In [411]:
```
# survived with Fare

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

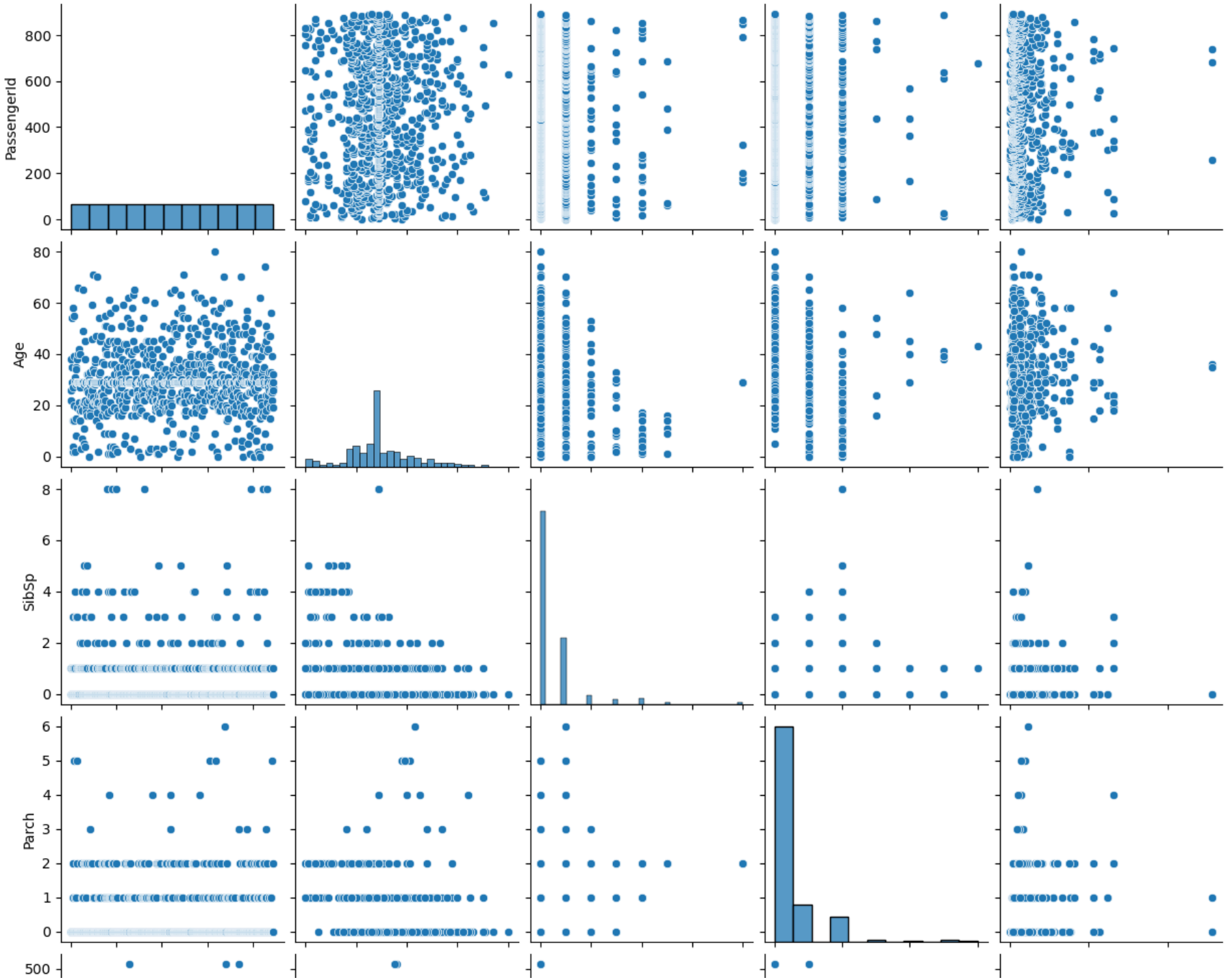
```
C:\Users\pranavkumar landage\AppData\Local\Temp\ipykernel_6768\2721700718.py:4: UserWarning:

`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 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 (https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751)

  sns.distplot(df[df['Survived']==0]['Fare'])
C:\Users\pranavkumar landage\AppData\Local\Temp\ipykernel_6768\2721700718.py:5: 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 (https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751)

  sns.distplot(df[df['Survived']==1]['Fare'])
```

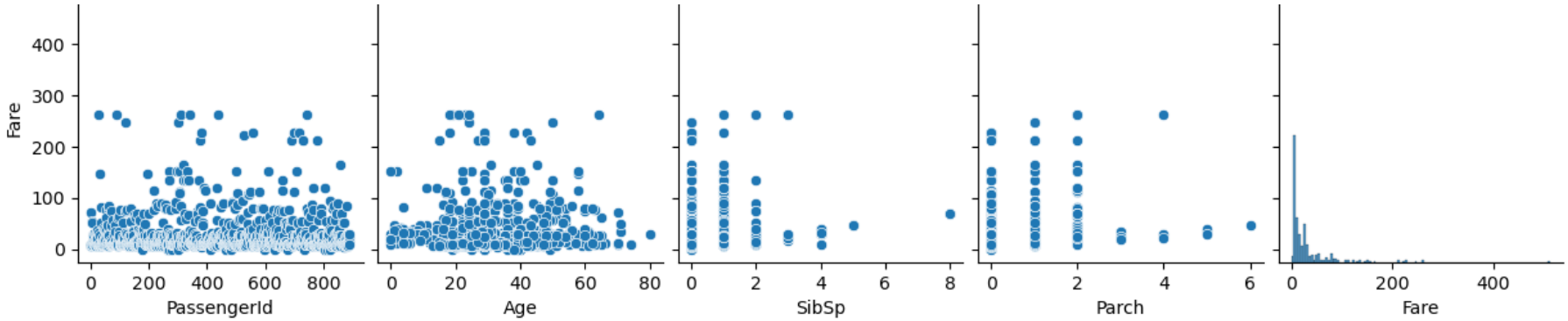
Out[411]:
```
<Axes: xlabel='Fare', ylabel='Density'>
```


0.08
0.06
0.04
0.02
0.00
Density
0
100
200
300
400
500
600
Fare

In [412]:
```
sns.pairplot(df)
```

Out[412]:
```
<seaborn.axisgrid.PairGrid at 0x21fa3cb4150>
```

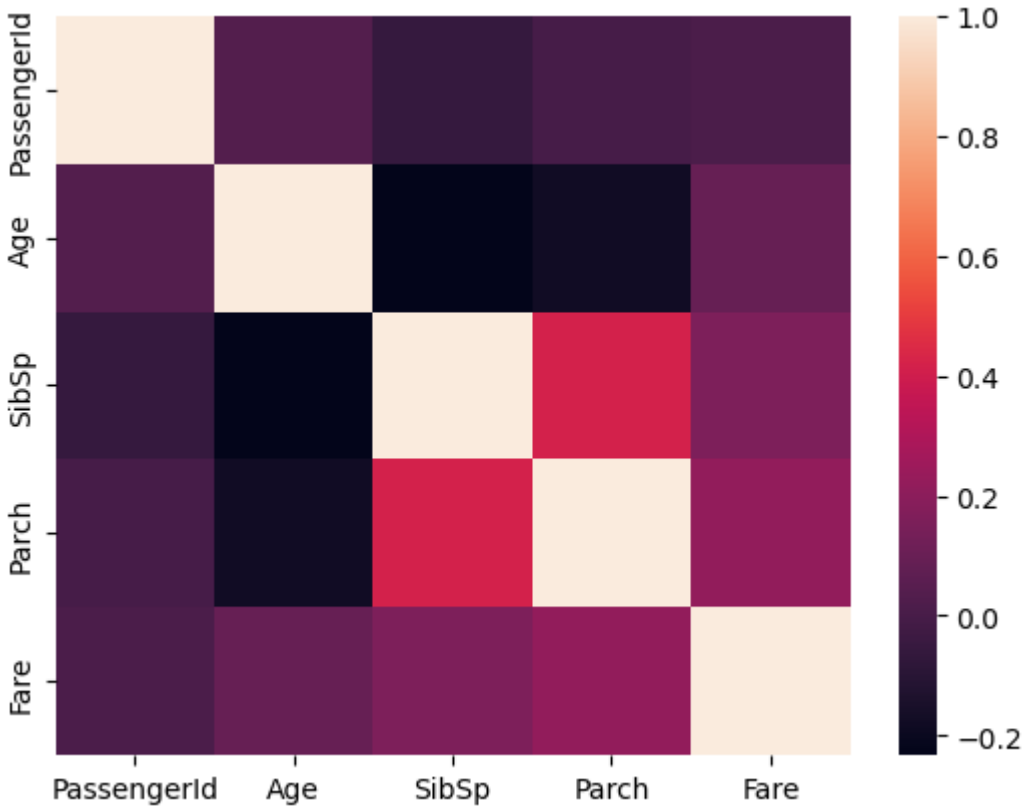

PassengerId
Age
SibSp
Parch


Fare
400
300
200
100
0
0
200
400
600
800
PassengerId
0
20
40
60
80
Age
0
2
4
6
8
SibSp
0
2
4
6
Parch
0
200
400
Fare

In [413]:
```
sns.heatmap(df.corr())
```

```
C:\Users\pranavkumar landage\AppData\Local\Temp\ipykernel_6768\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[413]: <Axes: >



In [414]:
```
#Detecting Outlier
```

In [415]:
```
#handling outlier from age
df = df[df['Age']<df['Age'].mean() + 3 * df['Age'].std()]
df.shape
```

Out[415]:
```
(884, 11)
```

In [416]:
```
#We will create a new column by the name of family which will be the sum of SibSp and Parch cols

df['family_size']=df['Parch']+df['SibSp']
df.sample(5)
```

```
C:\Users\pranavkumar landage\AppData\Local\Temp\ipykernel_6768\180938510.py:3: 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-a-view-versus-a-copy)
  df['family_size']=df['Parch']+df['SibSp']
```

Out[416]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Embarked	family_size
412	413	1	1	Minahan, Miss. Daisy E	female	33	1	0	19928	90.0000	Q	1
340	341	1	2	Navratil, Master. Edmond Roger	male	2	1	1	230080	26.0000	S	2
145	146	0	2	Nicholls, Mr. Joseph Charles	male	19	1	1	C.A. 33112	36.7500	S	2
451	452	0	3	Hagland, Mr. Ingvald Olai Olsen	male	29	1	0	65303	19.9667	S	1
95	96	0	3	Shorney, Mr. Charles Joseph	male	29	0	0	374910	8.0500	S	0

In [417]:
```
#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"
```

In [418]:
```
df.head()
```

Out[418]:

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

Type *Markdown* and LaTeX: α^2

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 and

Thank You

In [ ]:

In [ ]: