In [4]:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [5]:

```
df = pd.read_csv("tested.csv")
```

Data Cleaning

In [4]:

```
df.head()
```

Out[4]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	892	0	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	Q
1	893	1	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	S
2	894	0	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN	Q
3	895	0	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN	S
4	896	1	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN	S

In [5]:

```
df.tail()
```

Out[5]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
413	1305	0	3	Spector, Mr. Woolf	male	NaN	0	0	A.5. 3236	8.0500	NaN	S
414	1306	1	1	Oliva y Ocana, Dona. Fermina	female	39.0	0	0	PC 17758	108.9000	C105	C
415	1307	0	3	Saether, Mr. Simon Sivertsen	male	38.5	0	0	SOTON/O.Q. 3101262	7.2500	NaN	S
416	1308	0	3	Ware, Mr. Frederick	male	NaN	0	0	359309	8.0500	NaN	S
417	1309	0	3	Peter, Master. Michael J	male	NaN	1	1	2668	22.3583	NaN	C

In [6]:
```
df.shape
```

Out[6]:
```
(418, 12)
```

In [7]:
```
df.describe()
```

Out[7]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	418.000000	418.000000	418.000000	332.000000	418.000000	418.000000	417.000000
mean	1100.500000	0.363636	2.265550	30.272590	0.447368	0.392344	35.627188
std	120.810458	0.481622	0.841838	14.181209	0.896760	0.981429	55.907576
min	892.000000	0.000000	1.000000	0.170000	0.000000	0.000000	0.000000
25%	996.250000	0.000000	1.000000	21.000000	0.000000	0.000000	7.895800
50%	1100.500000	0.000000	3.000000	27.000000	0.000000	0.000000	14.454200
75%	1204.750000	1.000000	3.000000	39.000000	1.000000	0.000000	31.500000
max	1309.000000	1.000000	3.000000	76.000000	8.000000	9.000000	512.329200

In [9]:
```
# Check for missing values
print(df.isnull().sum())
```

```
PassengerId    0
Survived       0
Pclass         0
Name           0
Sex            0
Age            0
SibSp          0
Parch          0
Ticket         0
Fare           0
Cabin          0
Embarked       0
dtype: int64
```

In [10]:

```
# Check for duplicate rows
print(df.duplicated().sum())
```

```
0
```

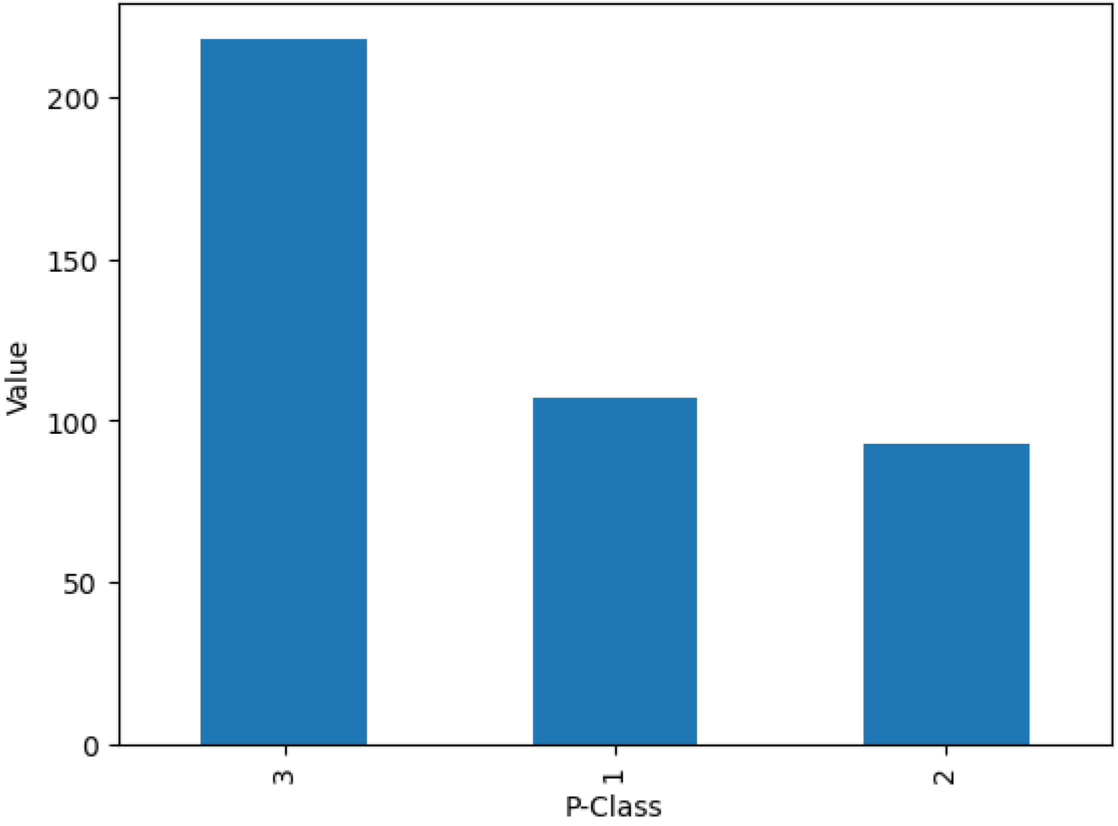
Exploratory Data Analysis (EDA)

In [79]:

```
# Basic statistics
print(df.describe())
```

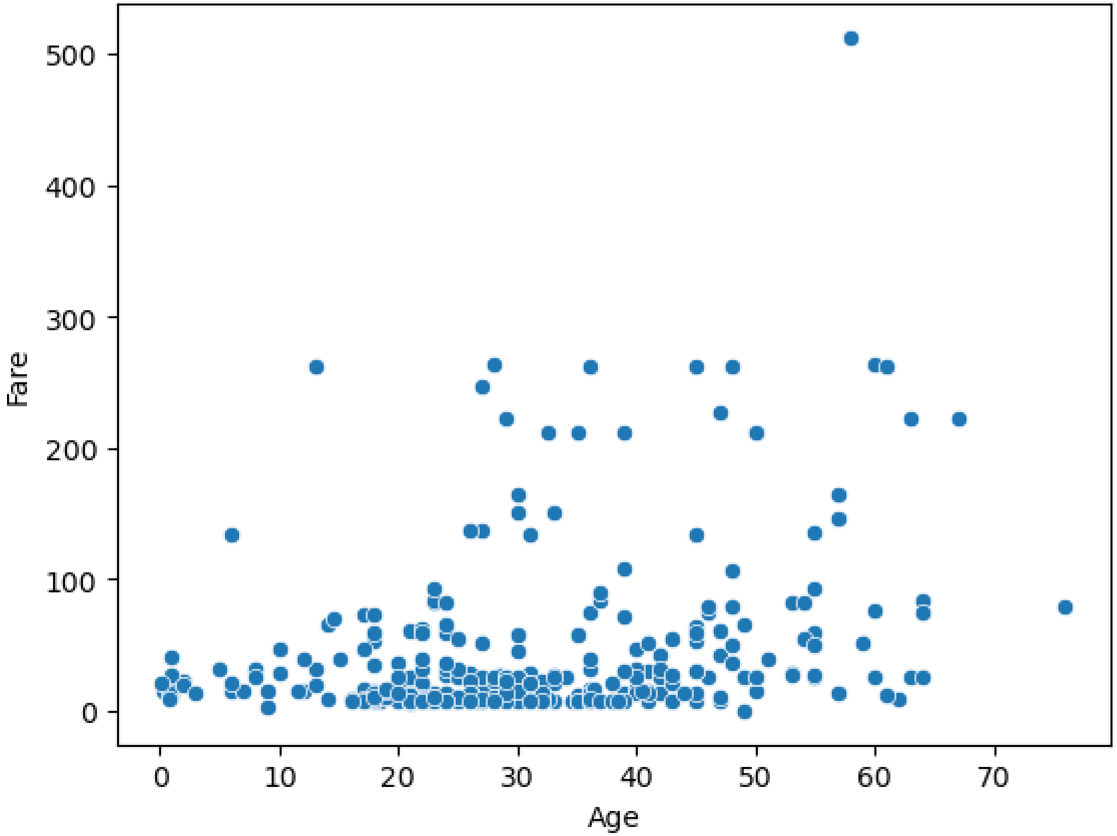
```
       PassengerId   Survived     Pclass        Age      SibSp      Parch  \
count    87.000000  87.000000  87.000000  87.000000  87.000000  87.000000   
mean   1102.712644   0.505747   1.137931  39.247126   0.597701   0.482759   
std     126.751901   0.502865   0.435954  15.218730   0.637214   0.860801   
min     904.000000   0.000000   1.000000   1.000000   0.000000   0.000000   
25%     986.000000   0.000000   1.000000  27.000000   0.000000   0.000000   
50%    1094.000000   1.000000   1.000000  39.000000   1.000000   0.000000   
75%    1216.000000   1.000000   1.000000  50.000000   1.000000   1.000000   
max    1306.000000   1.000000   3.000000  76.000000   3.000000   4.000000   

             Fare  Family_Size  
count   87.000000    87.000000  
mean    98.109198     1.080460  
std     88.177319     1.193182  
min      0.000000     0.000000  
25%     35.339600     0.000000  
50%     71.283300     1.000000  
75%    135.066650     2.000000  
max    512.329200     5.000000  
```

```
In [6]: # Bar plot
        df['Pclass'].value_counts().plot(kind='bar')
        plt.xlabel('P-Class')
        plt.ylabel('Value')
        plt.show()
```

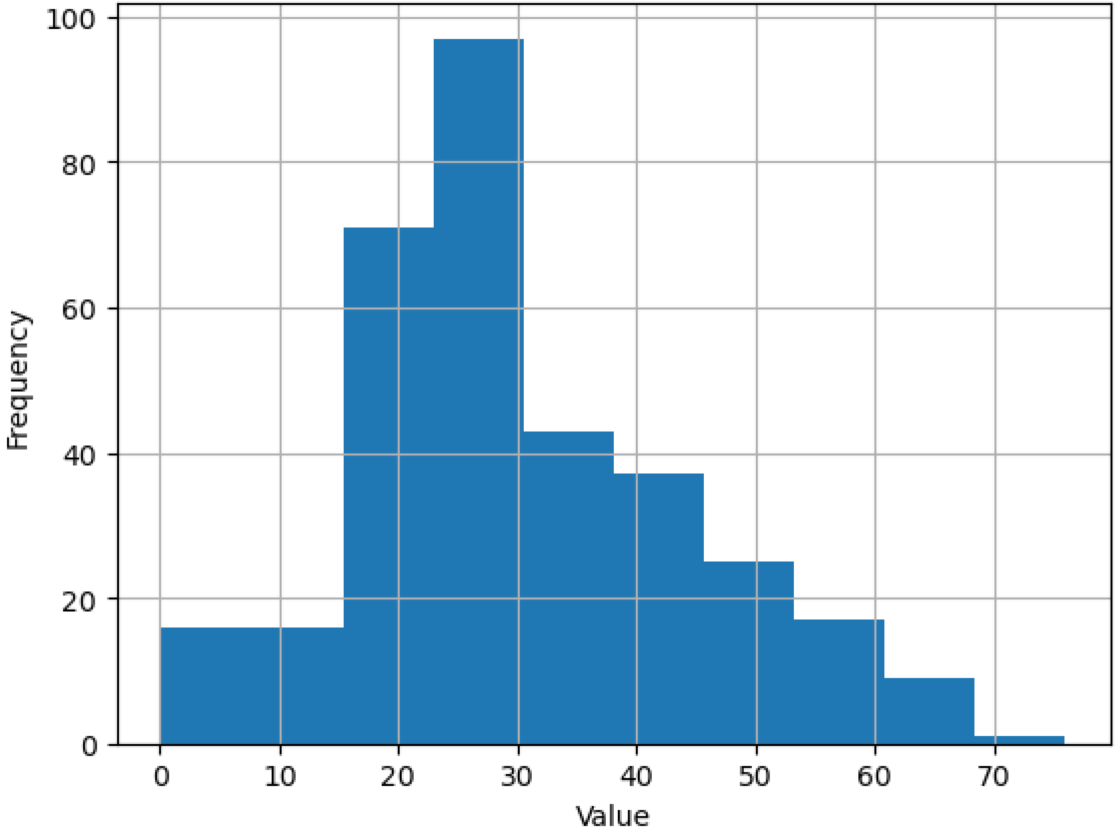


In [7]:

```
# Scatter plot
sns.scatterplot(x='Age', y='Fare', data=df)
plt.show()
```



In [8]:
```
# Histogram
df['Age'].hist()
plt.xlabel('Value')
plt.ylabel('Frequency')
plt.show()
```



Feature Engineering

In [9]:

```python
# Create new column for age group
def age_group(age):
    if age < 18:
        return 'Child'
    else:
        return 'Adult'
df['Age_Group'] = df['Age'].apply(age_group)
```

In [10]:

```python
df
```

Out[10]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	Age_Group
0	892	0	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	Q	Adult
1	893	1	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	S	Adult
2	894	0	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN	Q	Adult
3	895	0	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN	S	Adult
4	896	1	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN	S	Adult
...	...	...	...	...	...	...	...	...	...	...	...	...	...
413	1305	0	3	Spector, Mr. Woolf	male	NaN	0	0	A.5. 3236	8.0500	NaN	S	Adult
414	1306	1	1	Oliva y Ocana, Dona. Fermina	female	39.0	0	0	PC 17758	108.9000	C105	C	Adult
415	1307	0	3	Saether, Mr. Simon Sivertsen	male	38.5	0	0	SOTON/O.Q. 3101262	7.2500	NaN	S	Adult
416	1308	0	3	Ware, Mr. Frederick	male	NaN	0	0	359309	8.0500	NaN	S	Adult
417	1309	0	3	Peter, Master. Michael J	male	NaN	1	1	2668	22.3583	NaN	C	Adult

418 rows × 13 columns

In [38]:

```
# Create new column for family size
df['Family_Size'] = df['SibSp'] + df['Parch']
```

In [40]:

```
df
```

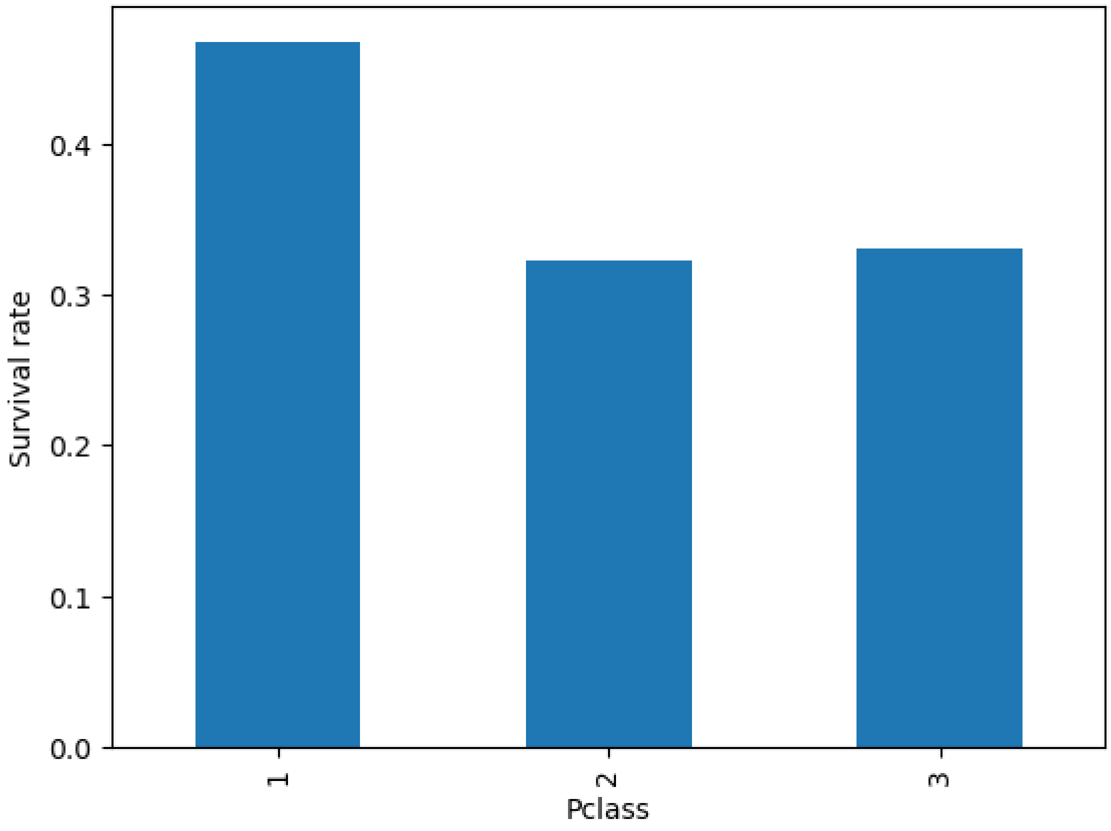
Out[40]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	Age_Group	Family_Size
12	904	1	1	Snyder, Mrs. John Pillsbury (Nelle Stevenson)	female	23.0	1	0	21228	82.2667	B45	S	Adult	1
14	906	1	1	Chaffee, Mrs. Herbert Fuller (Carrie Constance...	female	47.0	1	0	W.E.P. 5734	61.1750	E31	S	Adult	1
24	916	1	1	Ryerson, Mrs. Arthur Larned (Emily Maria Borie)	female	48.0	1	3	PC 17608	262.3750	B57 B59 B63 B66	C	Adult	4
26	918	1	1	Ostby, Miss. Helene Ragnhild	female	22.0	0	1	113509	61.9792	B36	C	Adult	1
28	920	0	1	Brady, Mr. John Bertram	male	41.0	0	0	113054	30.5000	A21	S	Adult	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
404	1296	0	1	Frauenthal, Mr. Isaac Gerald	male	43.0	1	0	17765	27.7208	D40	C	Adult	1
405	1297	0	2	Nourney, Mr. Alfred (Baron von Drachstedt")"	male	20.0	0	0	SC/PARIS 2166	13.8625	D38	C	Adult	0
407	1299	0	1	Widener, Mr. George Dunton	male	50.0	1	1	113503	211.5000	C80	C	Adult	2
411	1303	1	1	Minahan, Mrs. William Edward (Lillian E Thorpe)	female	37.0	1	0	19928	90.0000	C78	Q	Adult	1
414	1306	1	1	Oliva y Ocana, Dona. Fermina	female	39.0	0	0	PC 17758	108.9000	C105	C	Adult	0

87 rows × 14 columns

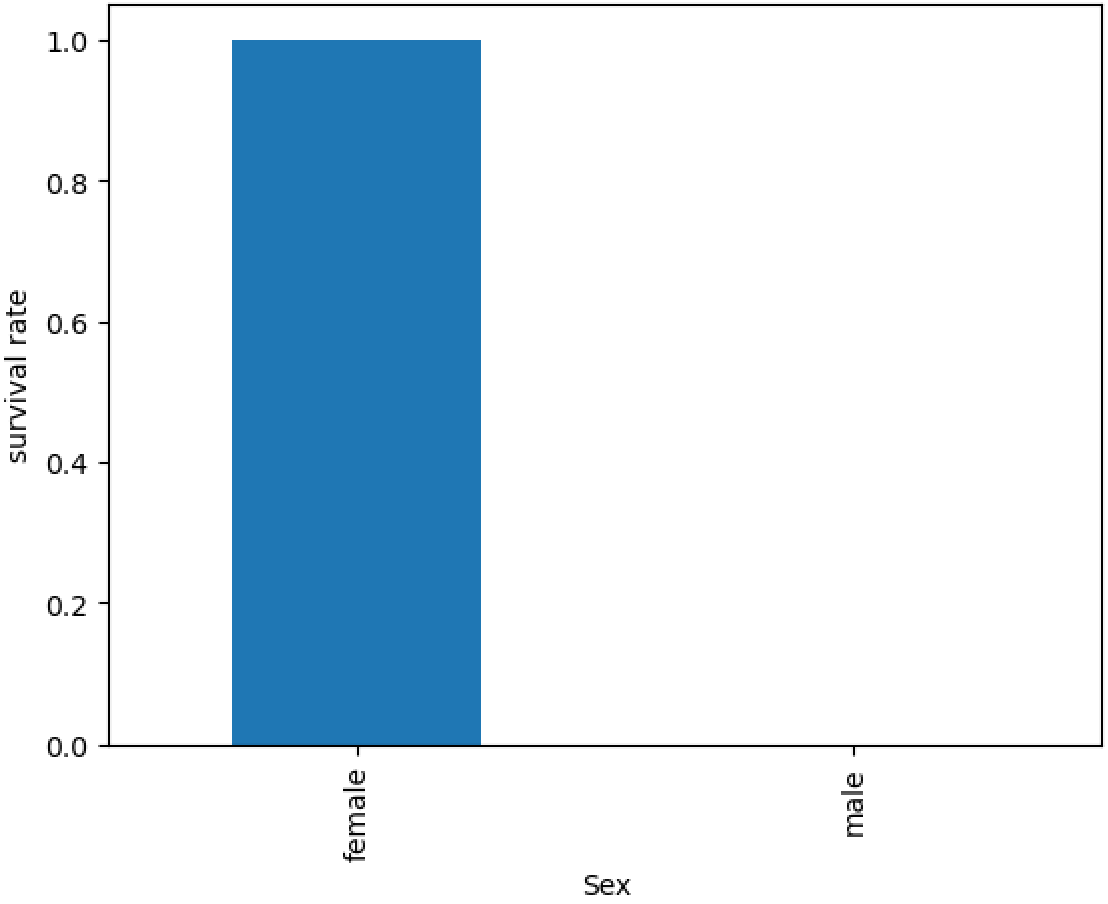
Survival Analysis

```
In [11]: # Bar plot of survival rate by class
         class_survival = df.groupby(['Pclass'])['Survived'].mean()
         class_survival.plot(kind='bar')
         plt.ylabel('Survival rate')
         plt.show()
```



In [12]:

```
# Bar plot of survival rate by sex
sex_survival = df.groupby(['Sex'])['Survived'].mean()
sex_survival.plot(kind='bar')
plt.ylabel('survival rate')
plt.show()
```



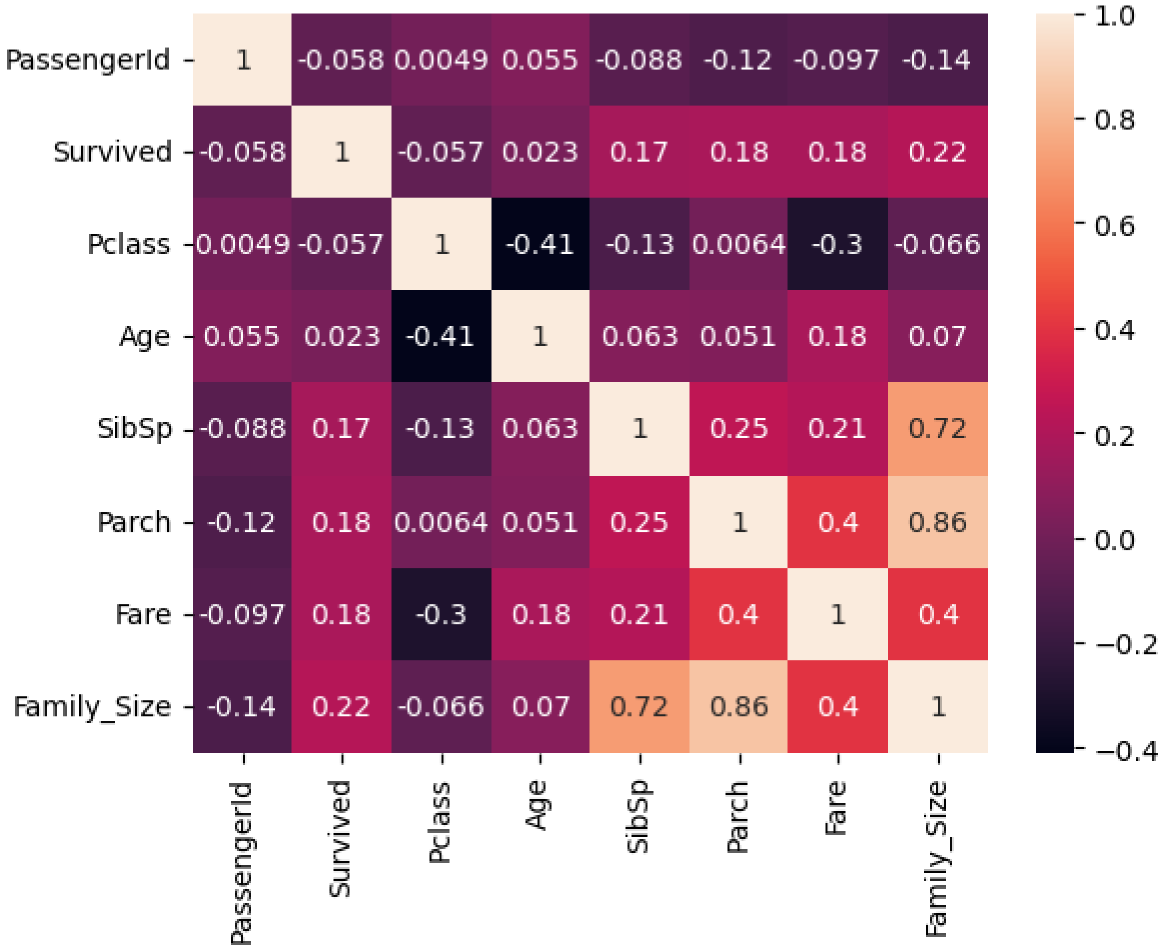
Correlation Analysis

In [13]:
```
# Correlation matrix
corr = df.corr()
print(corr)
```

```
             PassengerId  Survived    Pclass       Age     SibSp     Parch  \
PassengerId     1.000000 -0.023245 -0.026751 -0.034102  0.003818  0.043080   
Survived       -0.023245  1.000000 -0.108615 -0.000013  0.099943  0.159120   
Pclass         -0.026751 -0.108615  1.000000 -0.492143  0.001087  0.018721   
Age            -0.034102 -0.000013 -0.492143  1.000000 -0.091587 -0.061249   
SibSp           0.003818  0.099943  0.001087 -0.091587  1.000000  0.306895   
Parch           0.043080  0.159120  0.018721 -0.061249  0.306895  1.000000   
Fare            0.008211  0.191514 -0.577147  0.337932  0.171539  0.230046   

                 Fare  
PassengerId  0.008211  
Survived     0.191514  
Pclass      -0.577147  
Age          0.337932  
SibSp        0.171539  
Parch        0.230046  
Fare         1.000000  
```

In [74]:

```
# Heatmap
sns.heatmap(corr, annot=True)
plt.show()
```



Grouping and Aggregating

In [75]:

```
# Group by class and calculate mean fare
class_fare = df.groupby(['Pclass'])['Fare'].mean()
print(class_fare)
```

```
Pclass
1    107.378955
2     21.393750
3     10.526400
Name: Fare, dtype: float64
```

In [76]:

```
# Group by class and sex and calculate mean fare
class_sex_fare = df.groupby(['Pclass', 'Sex'])['Fare'].mean()
print(class_sex_fare)
```

```
Pclass  Sex
1       female    122.359380
        male       91.610087
2       female     29.500000
        male       13.287500
3       female     16.700000
        male        7.439600
Name: Fare, dtype: float64
```

Pivot Tables

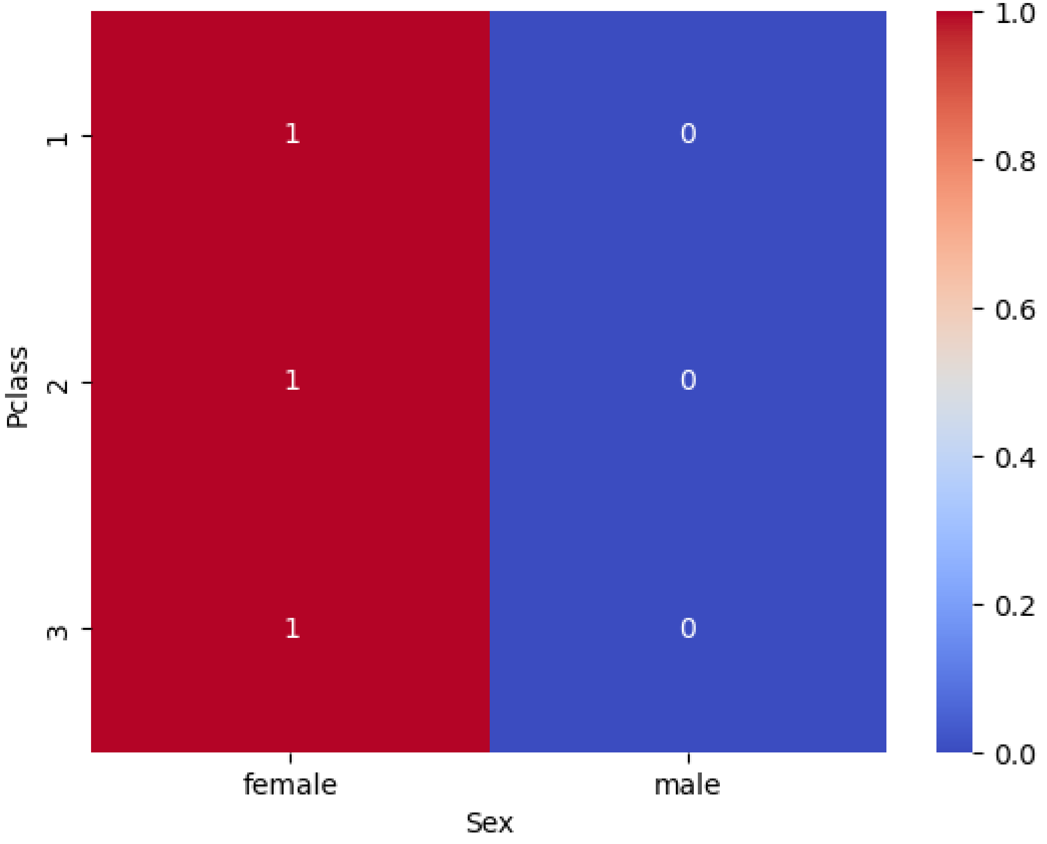
In [77]:

```
# Pivot table of survival rate by class and sex
pivot = df.pivot_table(values='Survived', index='Pclass', columns='Sex')
print(pivot)
```

```
Sex     female  male
Pclass
1            1     0
2            1     0
3            1     0
```

In [78]:

```
# Plot pivot table as a heatmap
sns.heatmap(pivot, annot=True, cmap='coolwarm')
plt.show()
```



In [ ]: