In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [4]:
```
# Load the data
train = pd.read_csv('train.csv')
test = pd.read_csv('test.csv')

train.head(10)
```

Out[4]:

	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
5	6	0	3	Moran, Mr. James	male	NaN	0	0	330877	8.4583	NaN	Q
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.8625	E46	S
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	349909	21.0750	NaN	S
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	347742	11.1333	NaN	S
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	237736	30.0708	NaN	C

In [5]:
```
test.head(10)
```

Out[5]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	Q
1	893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	S
2	894	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN	Q
3	895	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN	S
4	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN	S
5	897	3	Svensson, Mr. Johan Cervin	male	14.0	0	0	7538	9.2250	NaN	S
6	898	3	Connolly, Miss. Kate	female	30.0	0	0	330972	7.6292	NaN	Q
7	899	2	Caldwell, Mr. Albert Francis	male	26.0	1	1	248738	29.0000	NaN	S
8	900	3	Abrahim, Mrs. Joseph (Sophie Halaut Easu)	female	18.0	0	0	2657	7.2292	NaN	C
9	901	3	Davies, Mr. John Samuel	male	21.0	2	0	A/4 48871	24.1500	NaN	S

In [6]:
```
train.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 [7]:
```
train.describe()
```

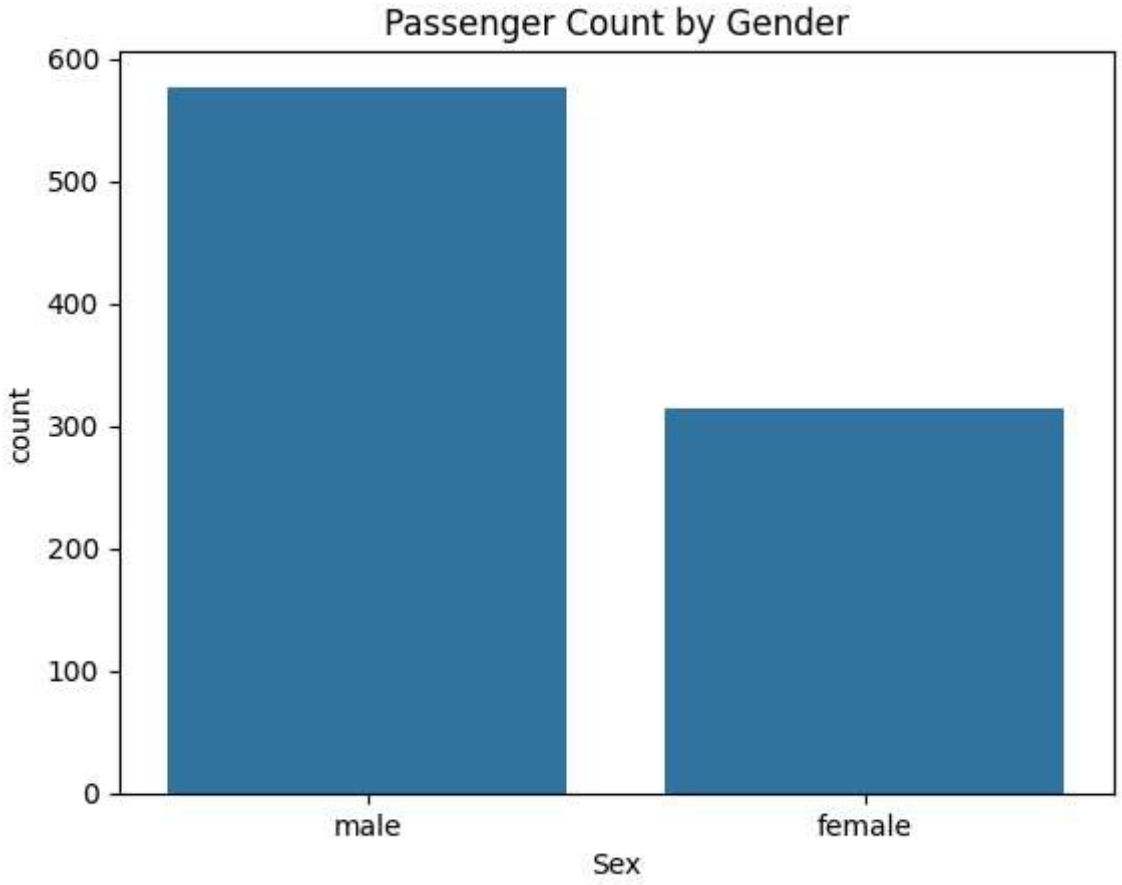
Out[7]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

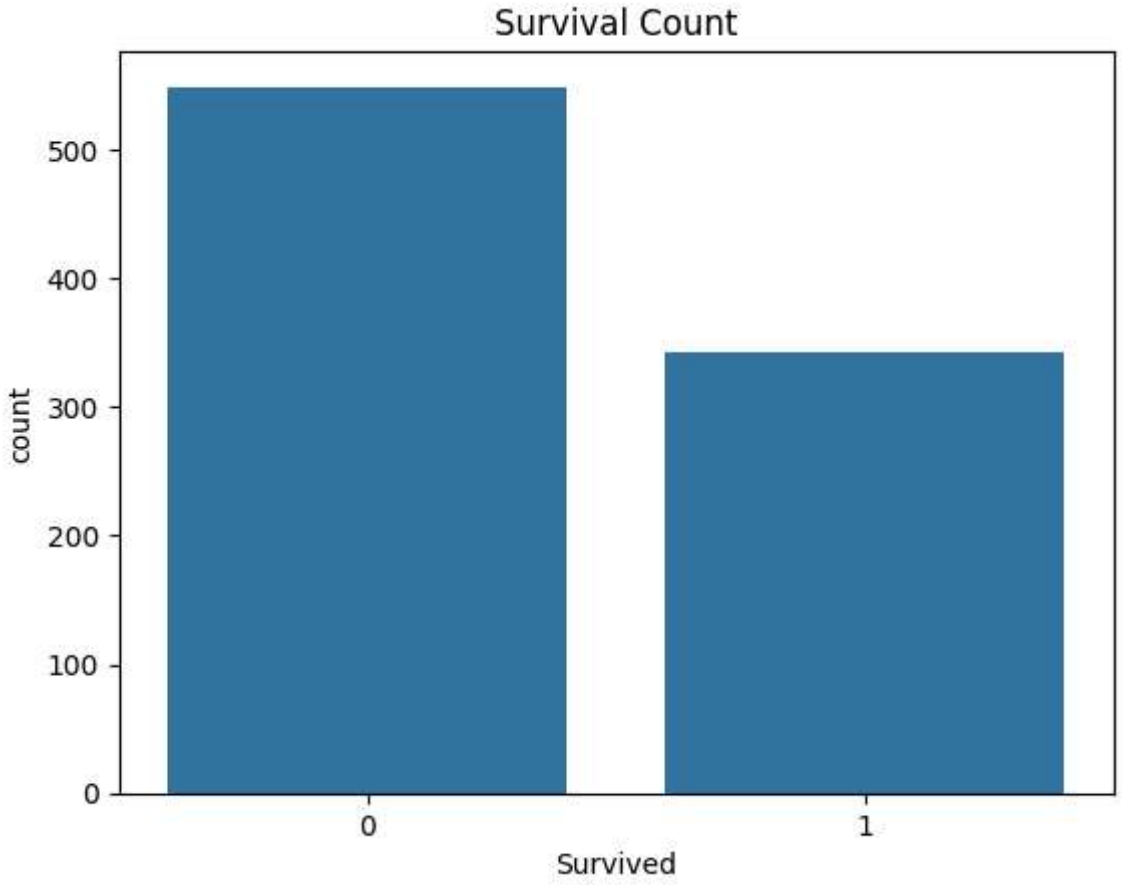
In [8]:
```
# Check for missing values
train.isnull().sum()
```

Out[8]:
```
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
```

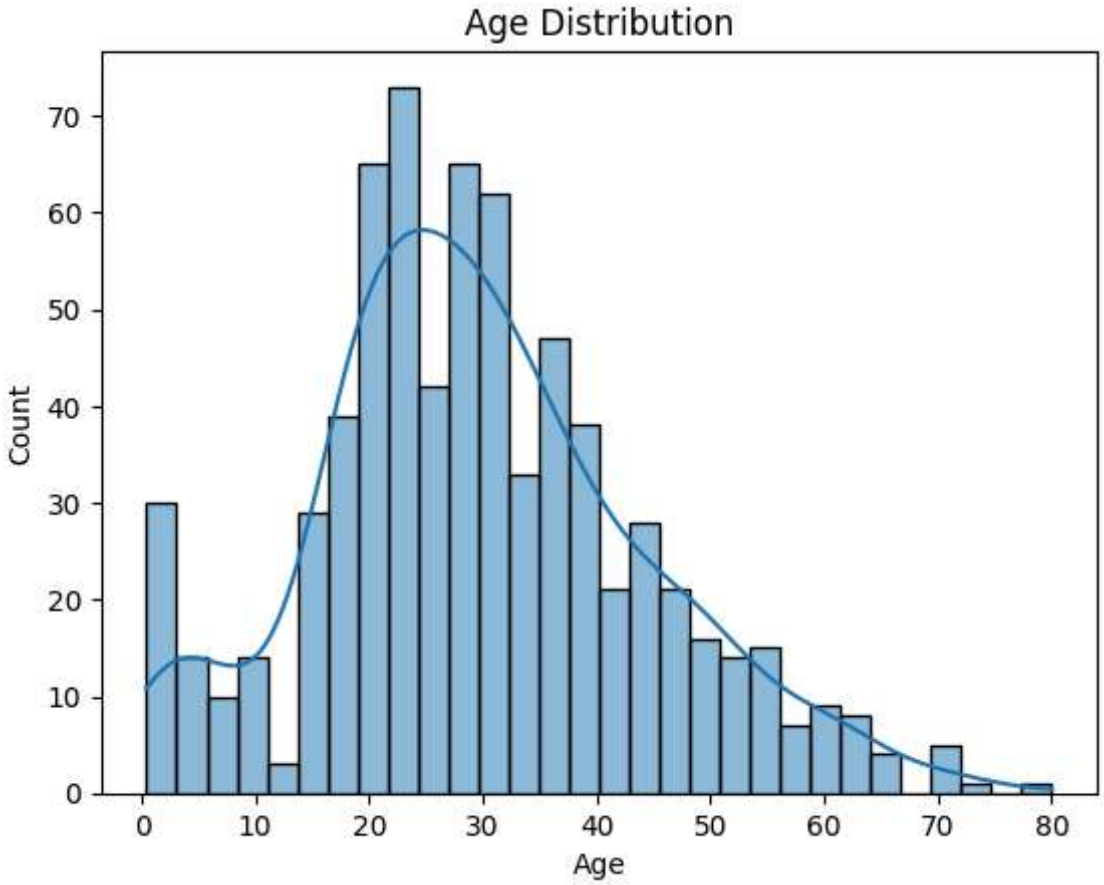
In [10]:
```
#gender count

sns.countplot(data=train, x='Sex')
plt.title("Passenger Count by Gender")
plt.show()
```



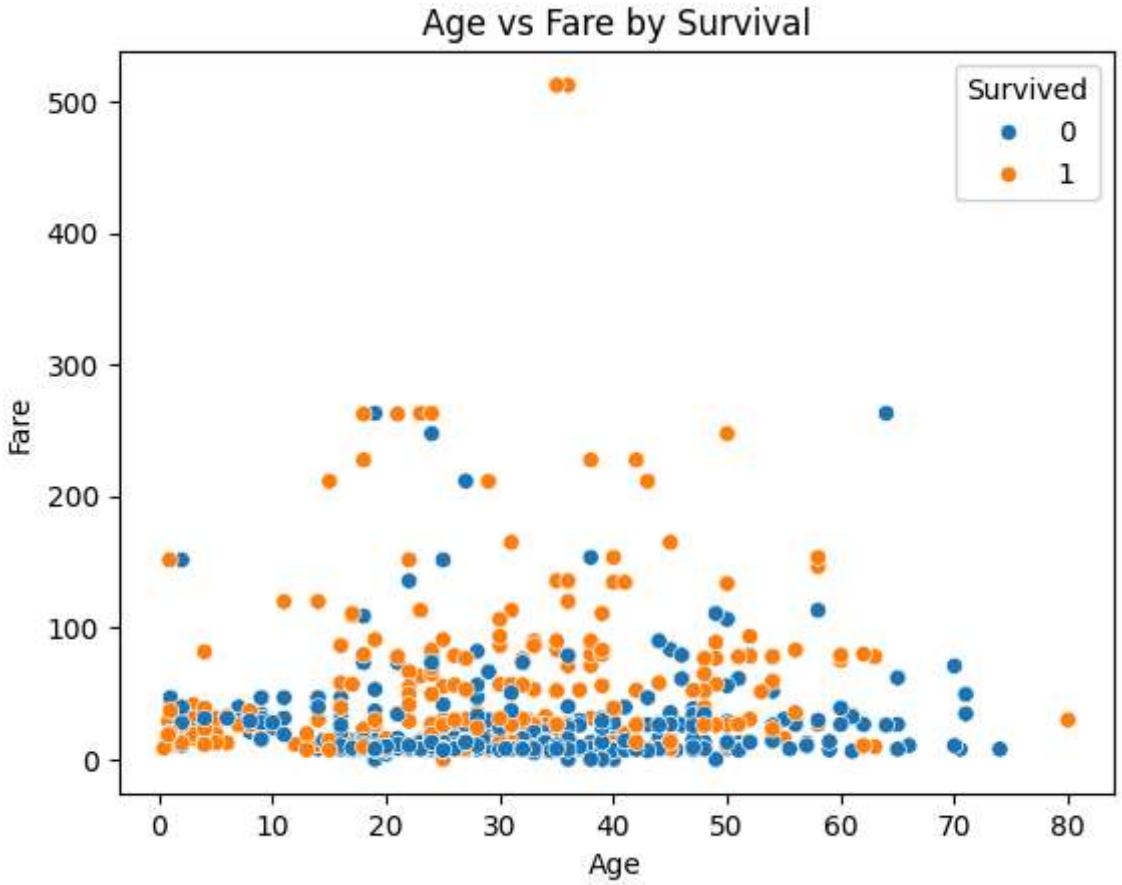
In [11]:
```
#survival distribution

sns.countplot(data=train, x='Survived')
plt.title("Survival Count")
plt.show()
```



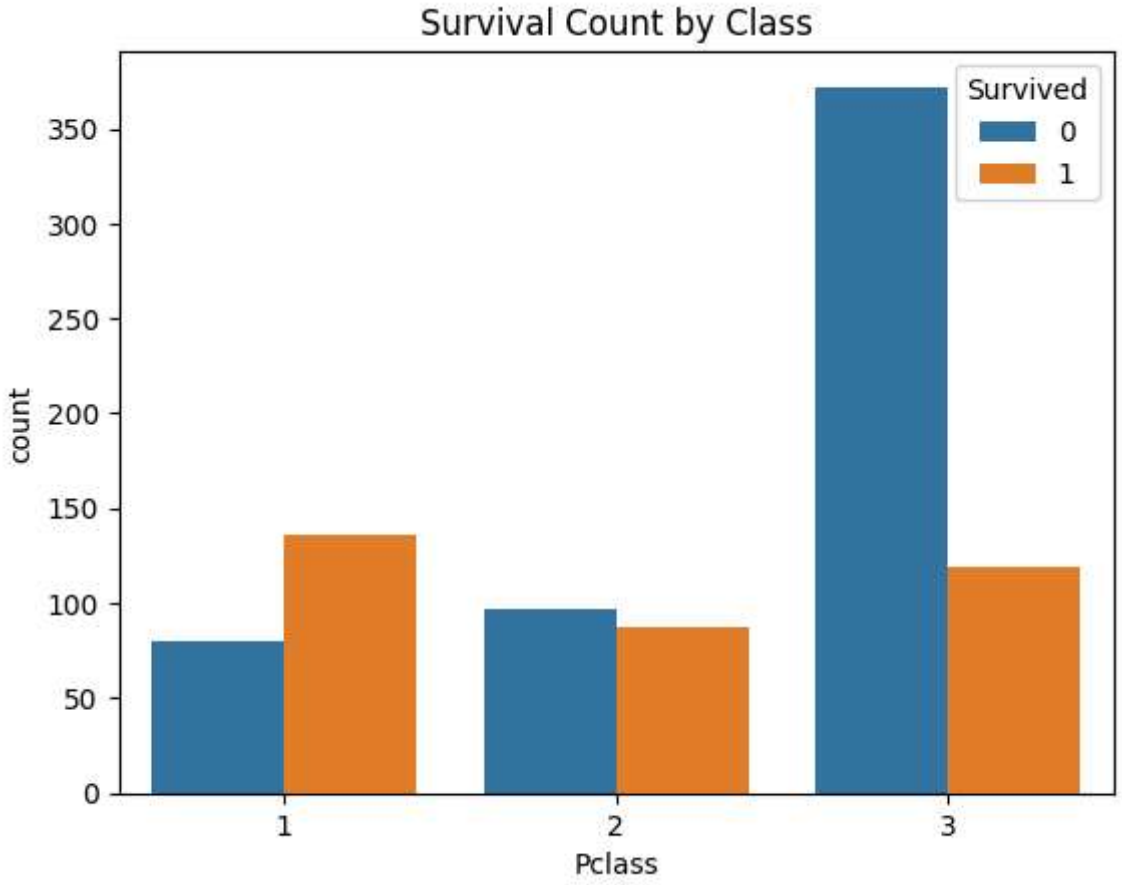
In [12]:
```
#Age distribution

sns.histplot(data=train, x='Age', bins=30, kde=True)
plt.title("Age Distribution")
plt.show()
```



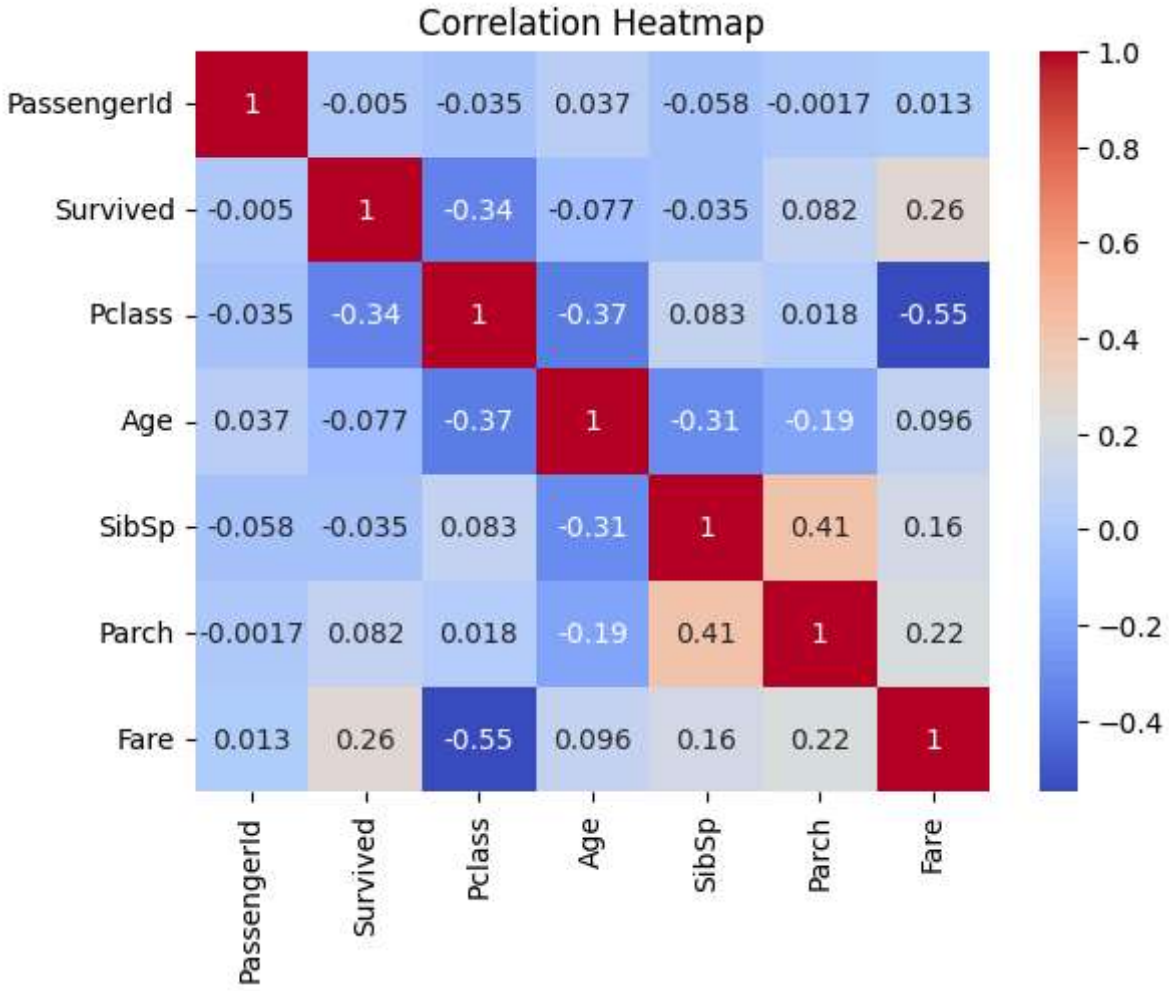
In [13]:
```
# Age VS Fare

sns.scatterplot(data=train, x='Age', y='Fare', hue='Survived')
plt.title("Age vs Fare by Survival")
plt.show()
```



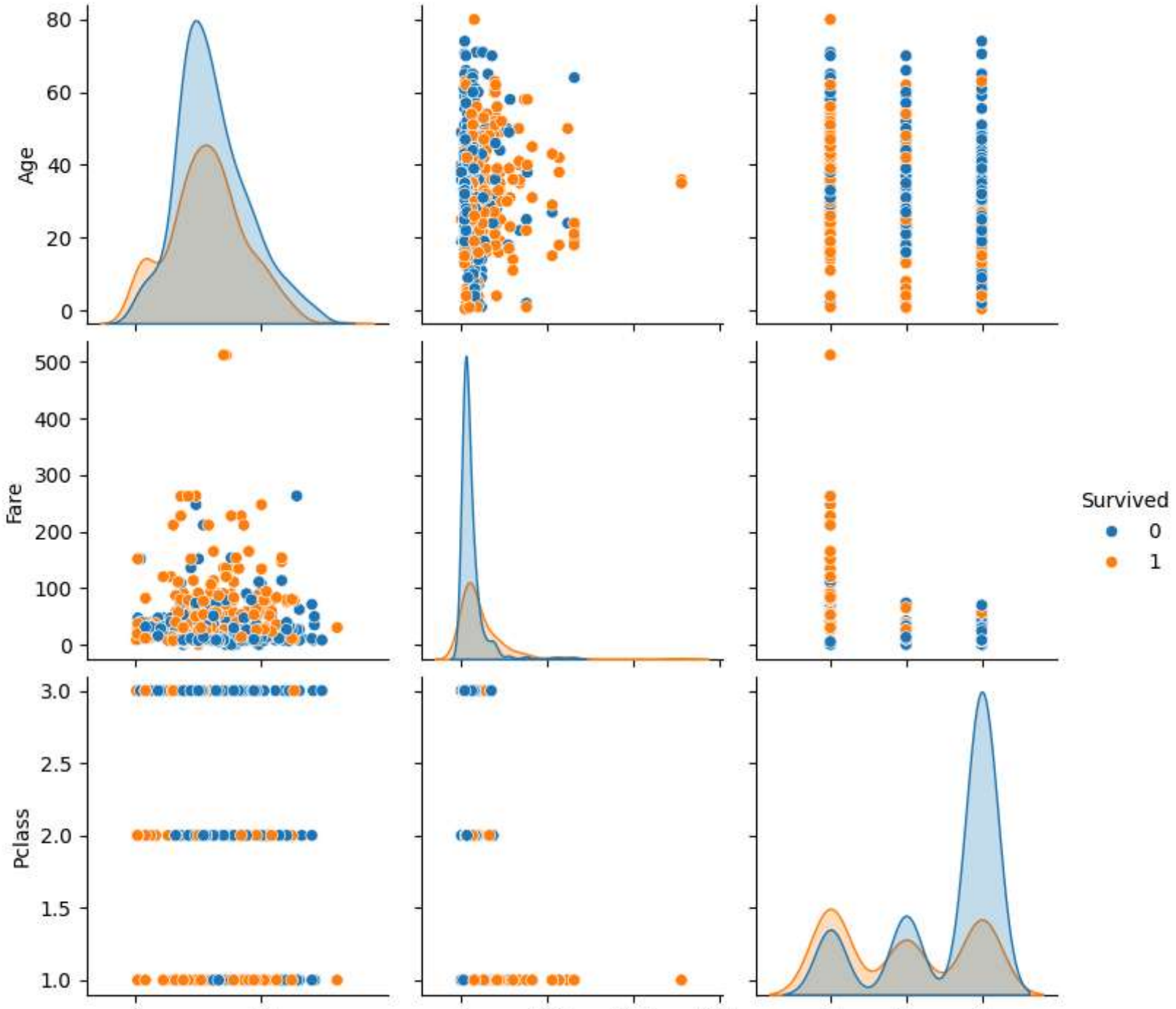
In [14]:
```
# Class vs Survival

sns.countplot(data=train, x='Pclass', hue='Survived')
plt.title("Survival Count by Class")
plt.show()
```



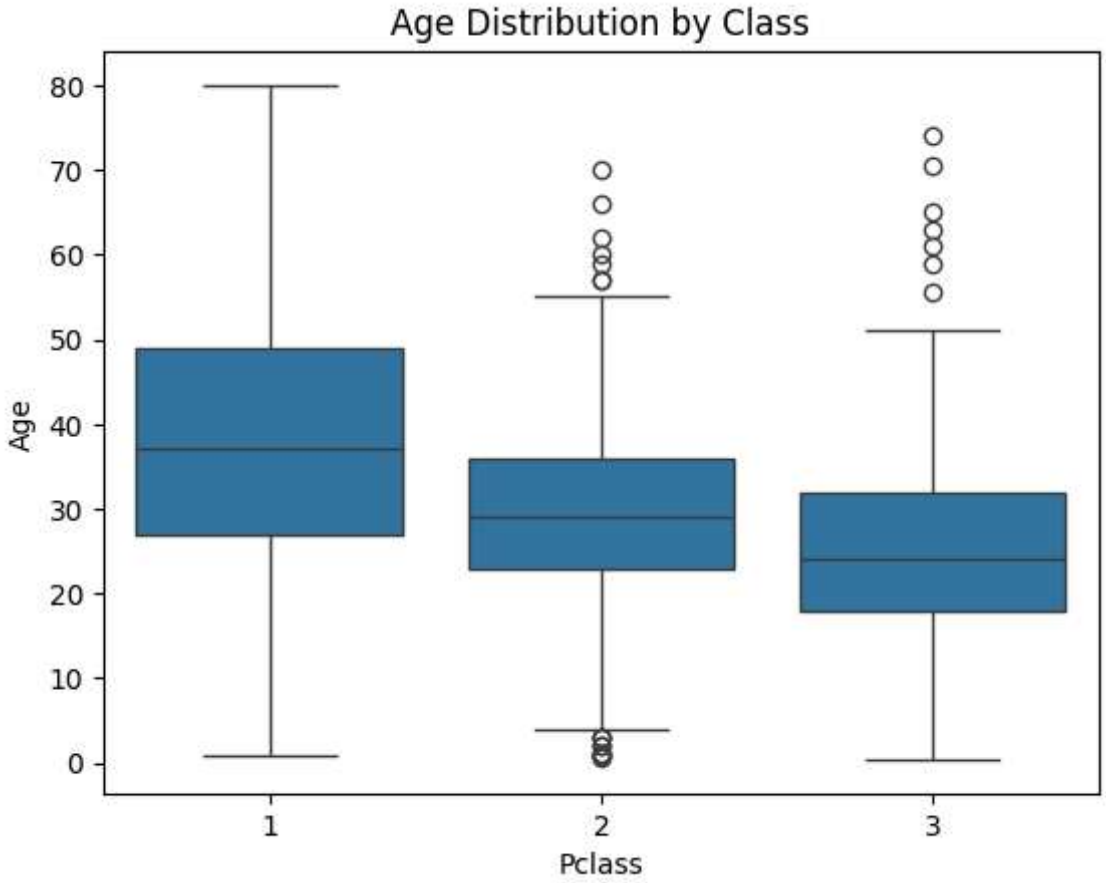
In [15]:
```
corr = train.corr(numeric_only=True)
sns.heatmap(corr, annot=True, cmap='coolwarm')
plt.title("Correlation Heatmap")
plt.show()
```



In [16]:
```
# suset for readability
sns.pairplot(train[['Survived', 'Age', 'Fare', 'Pclass']], hue='Survived')
plt.show()
```

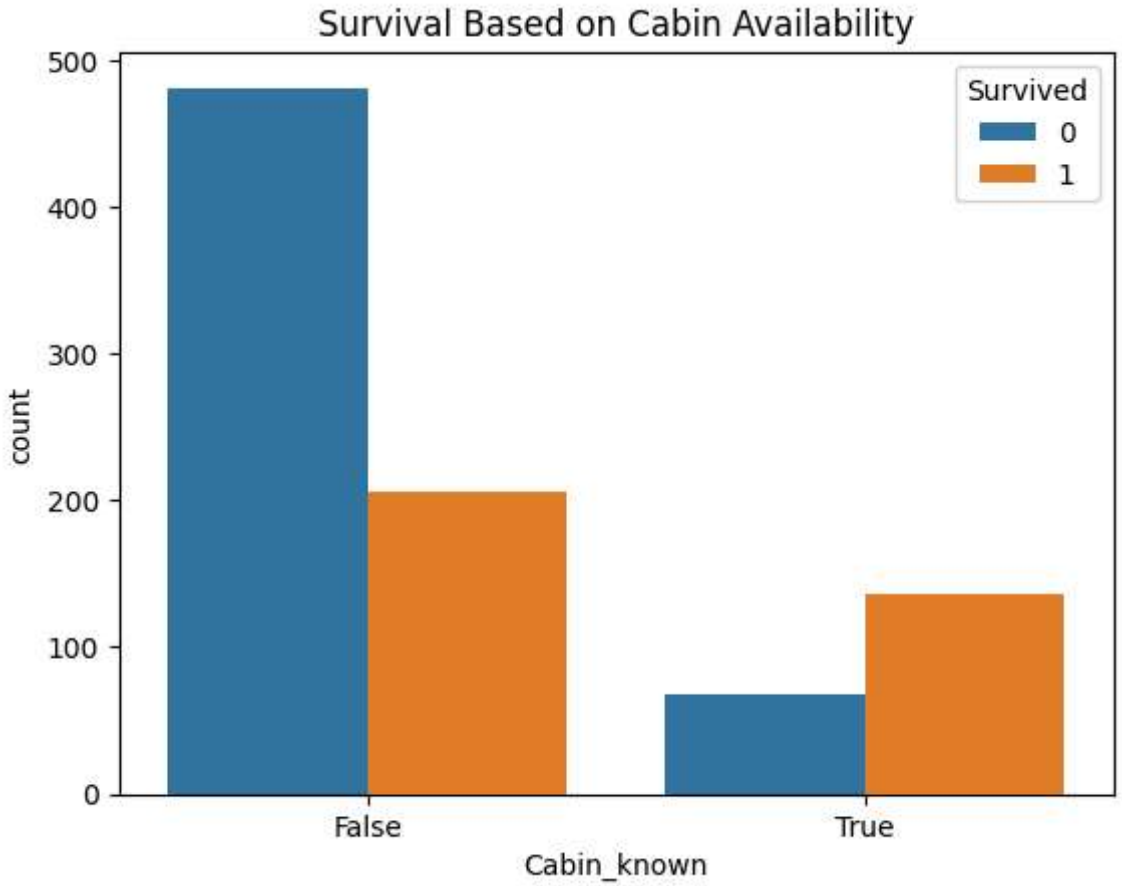

Age
Fare
Pclass
80
60
40
20
0
500
400
300
200
100
0
3.0
2.5
2.0
1.5
1.0
Survived
0
1

In [18]:
```
# Box plot

sns.boxplot(data=train, x='Pclass', y='Age')
plt.title("Age Distribution by Class")
plt.show()
```

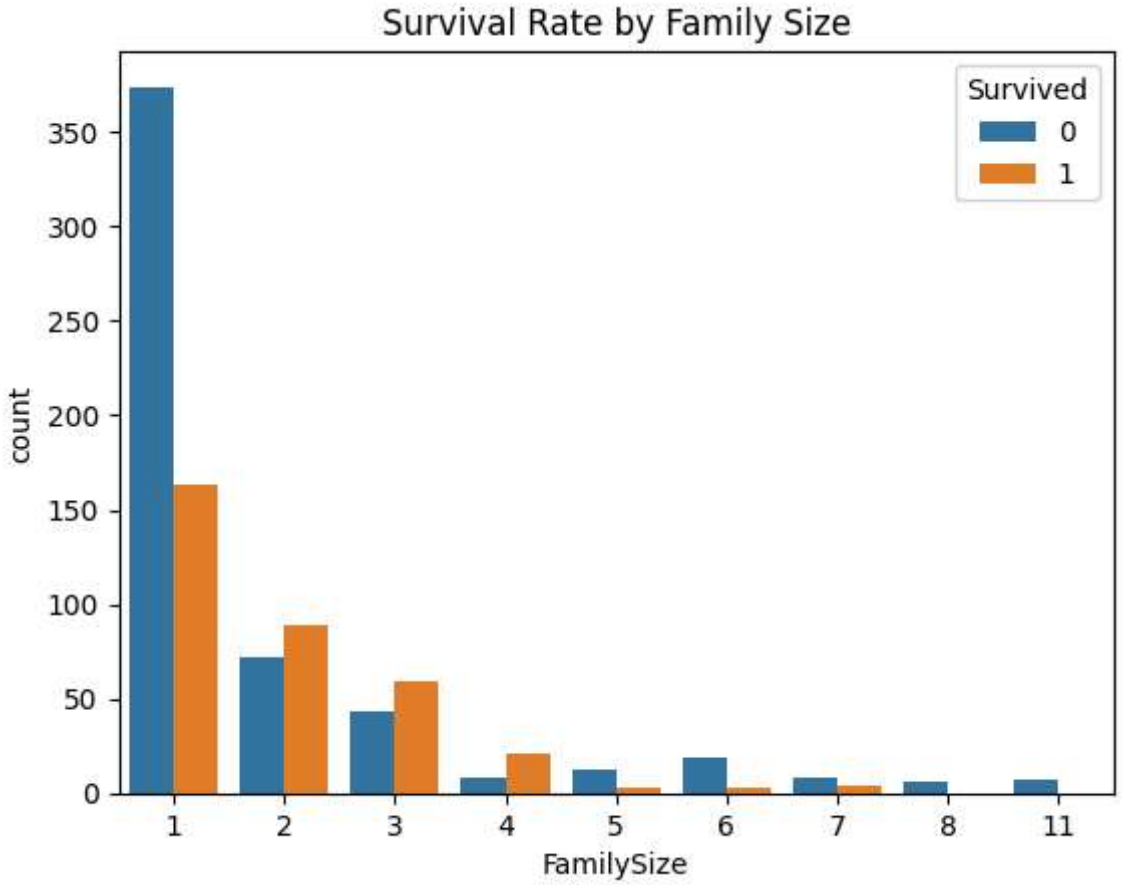


In [19]:
```
# Cabin data missing pattern

train['Cabin_known'] = train['Cabin'].notnull()
```

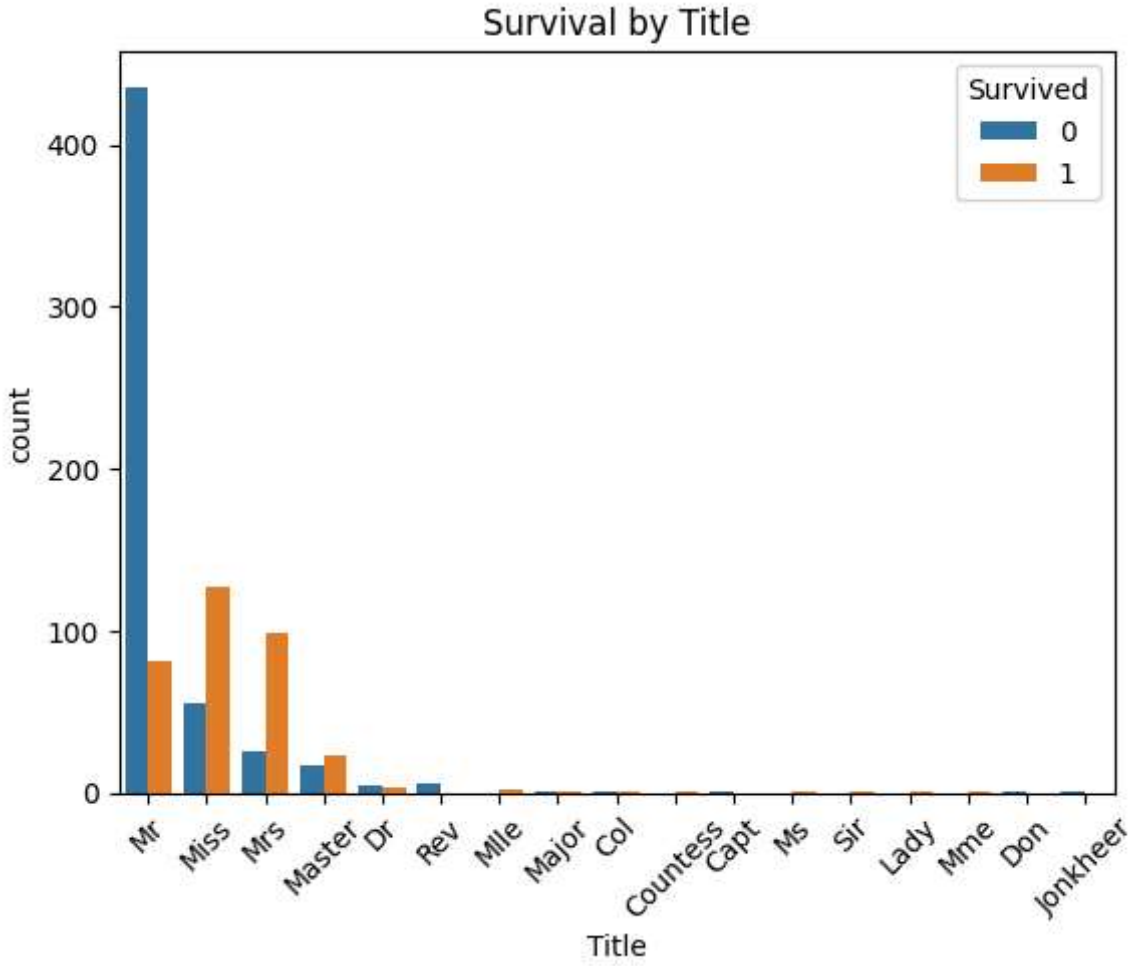
```
sns.countplot(data=train, x='Cabin_known', hue='Survived')
plt.title("Survival Based on Cabin Availability")
plt.show()

# Note : Passengers with known Cabin info had higher survival rates — likely first-class passengers.
```



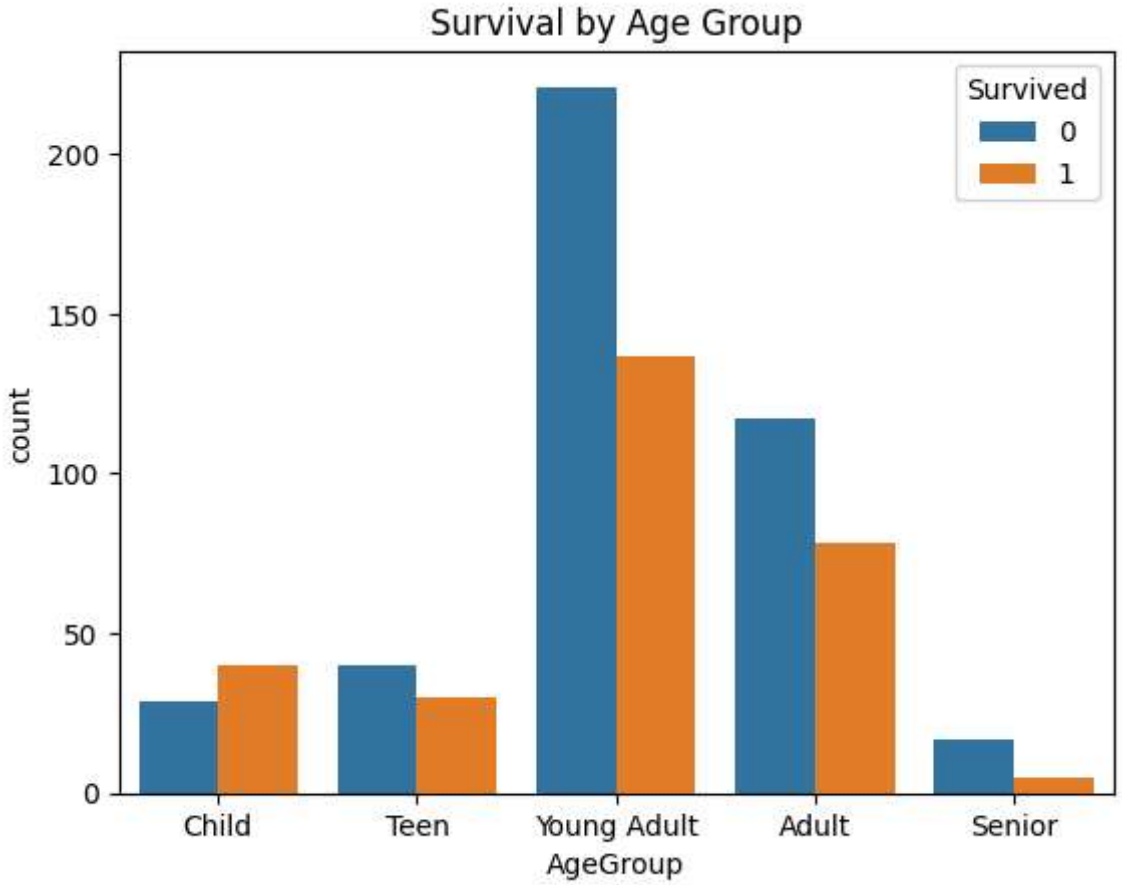
In [20]:
```
# family size feature

train['FamilySize'] = train['SibSp'] + train['Parch'] + 1
sns.countplot(data=train, x='FamilySize', hue='Survived')
plt.title("Survival Rate by Family Size")
plt.show()
```



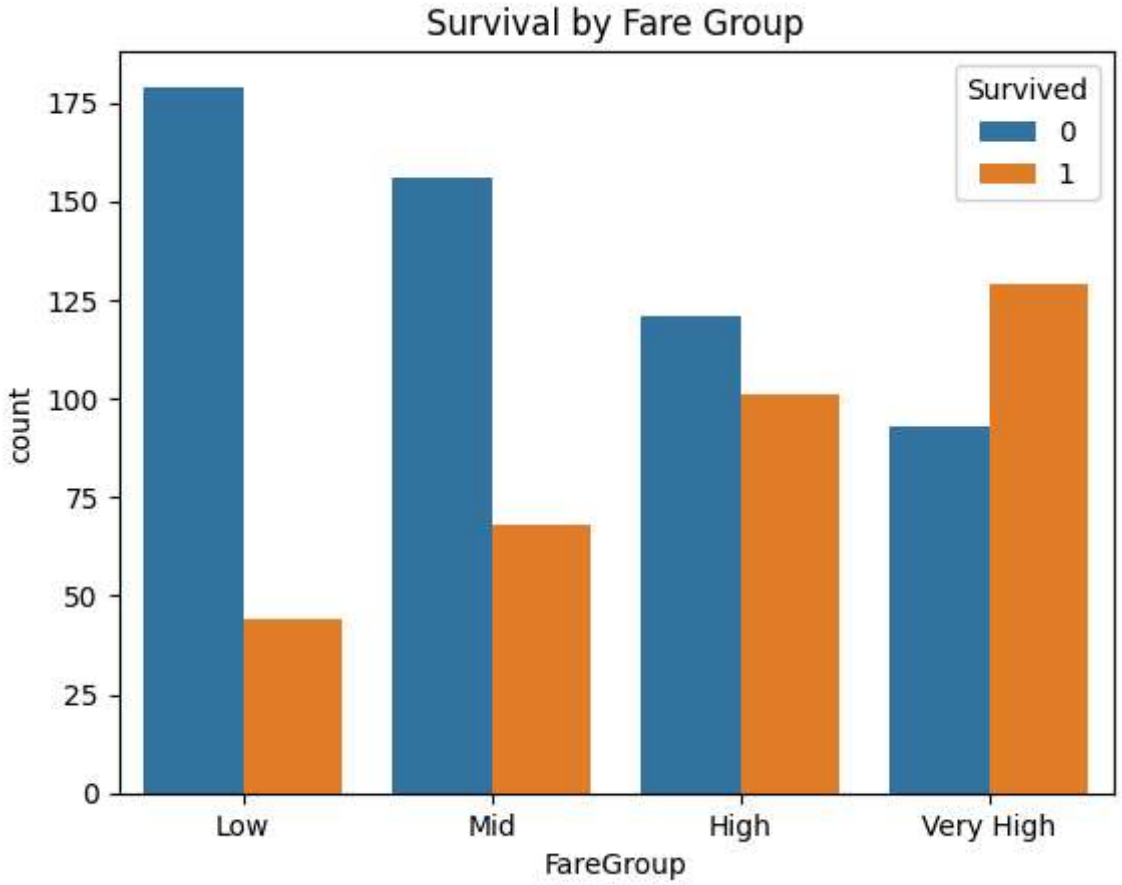
In [22]:
```
# Title extraction from name

train['Title'] = train['Name'].str.extract(' ([A-Za-z]+)\.', expand=False)
sns.countplot(data=train, x='Title', order=train['Title'].value_counts().index, hue='Survived')
plt.xticks(rotation=45)
plt.title("Survival by Title")
plt.show()
```



In [23]:
```
# categorizing the age group

train['AgeGroup'] = pd.cut(train['Age'], bins=[0, 12, 18, 35, 60, 80],
                           labels=['Child', 'Teen', 'Young Adult', 'Adult', 'Senior'])
sns.countplot(data=train, x='AgeGroup', hue='Survived')
plt.title("Survival by Age Group")
plt.show()
```



In [24]:
```
# categorizing fare

train['FareGroup'] = pd.qcut(train['Fare'], 4, labels=['Low', 'Mid', 'High', 'Very High'])
sns.countplot(data=train, x='FareGroup', hue='Survived')
plt.title("Survival by Fare Group")
plt.show()
```


Survival by Fare Group
Survived
0
1
count
175
150
125
100
75
50
25
0
Low
Mid
High
Very High
FareGroup

In [ ]: