In [4]:
```
import numpy as np
```

In [2]:
```
import pandas as pd
titanic_df = pd.read_csv('titanic.csv', index_col='PassengerId')
print(titanic_df.describe())
```

```
         Survived      Pclass         Age       SibSp       Parch        Fare
count  891.000000  891.000000  714.000000  891.000000  891.000000  793.000000
mean     0.383838    2.308642   29.699118    0.523008    0.381594   33.123938
std      0.486592    0.836071   14.526497    1.102743    0.806057   51.578312
min      0.000000    1.000000    0.420000    0.000000    0.000000    0.000000
25%      0.000000    2.000000   20.125000    0.000000    0.000000    7.925000
50%      0.000000    3.000000   28.000000    0.000000    0.000000   14.500000
75%      1.000000    3.000000   38.000000    1.000000    0.000000   31.275000
max      1.000000    3.000000   80.000000    8.000000    6.000000  512.329200
```

In [ ]:
```
missing = titanic_df.isnull().sum()
print(missing[missing > 0])
```

```
Age         177
Fare         98
Cabin       687
Embarked      2
dtype: int64
```

In [5]:
```
# Cabin - leaving just deck and in missing spaces i write unknown
titanic_df['Cabin'] = titanic_df['Cabin'].apply(lambda x: x[0] if pd.notnull(x) els
# Embarked - 2 missing values so i just paste there the most common port(S)
most_common_port = titanic_df['Embarked'].mode()[0]
titanic_df['Embarked'].fillna(most_common_port, inplace=True)

# Fare - in place of missing values i fill median fare by class
titanic_df['Fare'] = titanic_df.groupby('Pclass')['Fare'].transform(lambda x: x.fil

# Age - median by age and class
titanic_df['Age'] = titanic_df.groupby(['Sex', 'Pclass'])['Age'].transform(lambda x
titanic_df.isnull().sum()
```

```
C:\Users\sofiy\AppData\Local\Temp\ipykernel_12236\1627639956.py:5: FutureWarning: A
value is trying to be set on a copy of a DataFrame or Series through chained assignm
ent 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.


  titanic_df['Embarked'].fillna(most_common_port, inplace=True)
```

Out[5]:
```
Survived    0
Pclass      0
Name        0
Sex         0
Age         0
SibSp       0
Parch       0
Ticket      0
Fare        0
Cabin       0
Embarked    0
dtype: int64
```

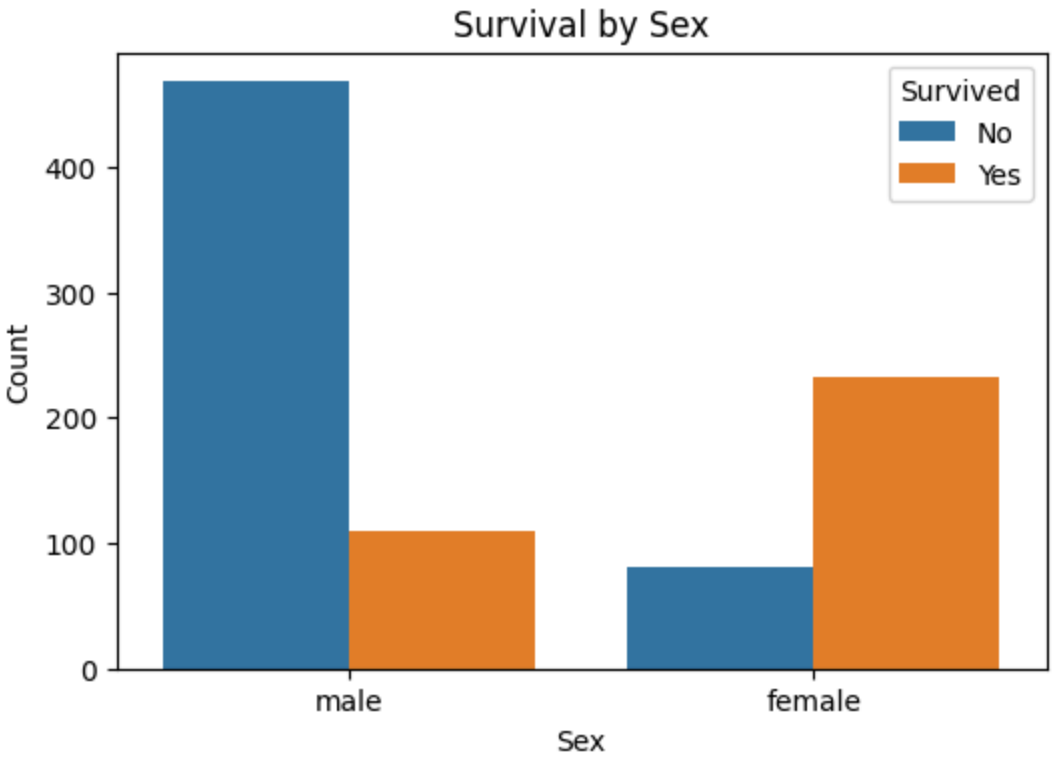
In [ ]:
```
titanic_df['FamilySize'] = titanic_df['SibSp'] + titanic_df['Parch'] + 1
titanic_df['IsAlone'] = (titanic_df['FamilySize'] == 1).astype(int)
```

In [7]:
```
titanic_df = titanic_df.drop(columns=['Name', 'Ticket'])
```

In [8]:
```
titanic_df['Sex'] = titanic_df['Sex'].map({'male': 0, 'female': 1})
# one hot for so model can understand equality that s=q etc.
titanic_df = pd.get_dummies(titanic_df, columns=['Embarked', 'Cabin'], drop_first=T
```

In [14]:
```
import matplotlib.pyplot as plt
import seaborn as sns
```

In [ ]:
```
viz_df = titanic_df.copy()
viz_df['Sex'] = viz_df['Sex'].map({0: 'male', 1: 'female'})
plt.figure(figsize=(6,4))
sns.countplot(x='Sex', hue='Survived', data=viz_df)
plt.title('Survival by Sex')
plt.xlabel('Sex')
plt.ylabel('Count')
plt.legend(title='Survived', labels=['No', 'Yes'])
plt.show()
```

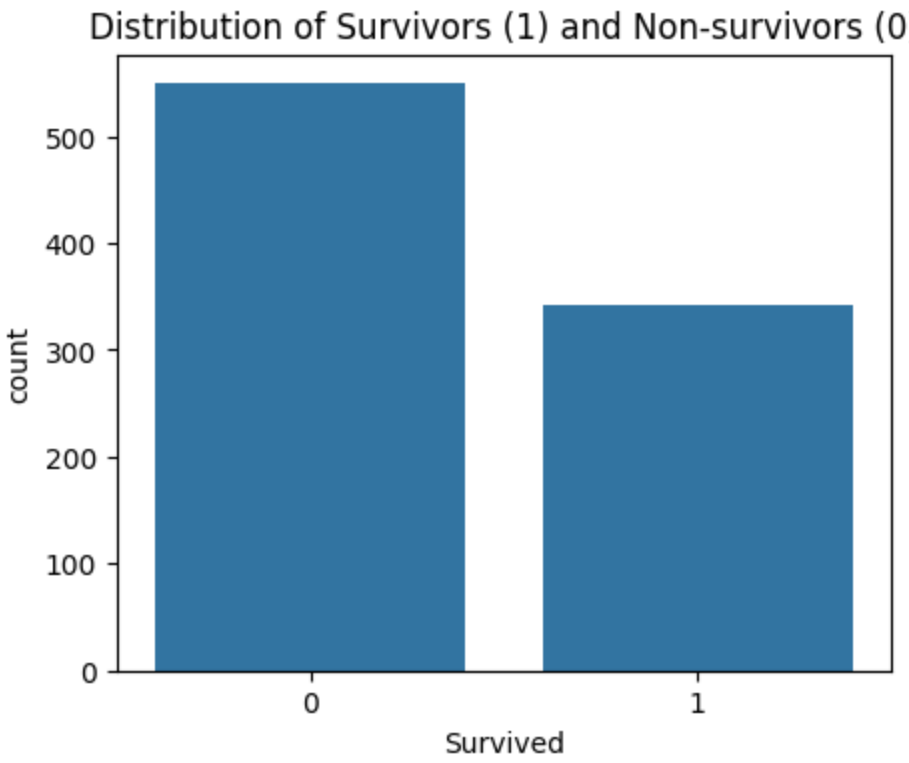


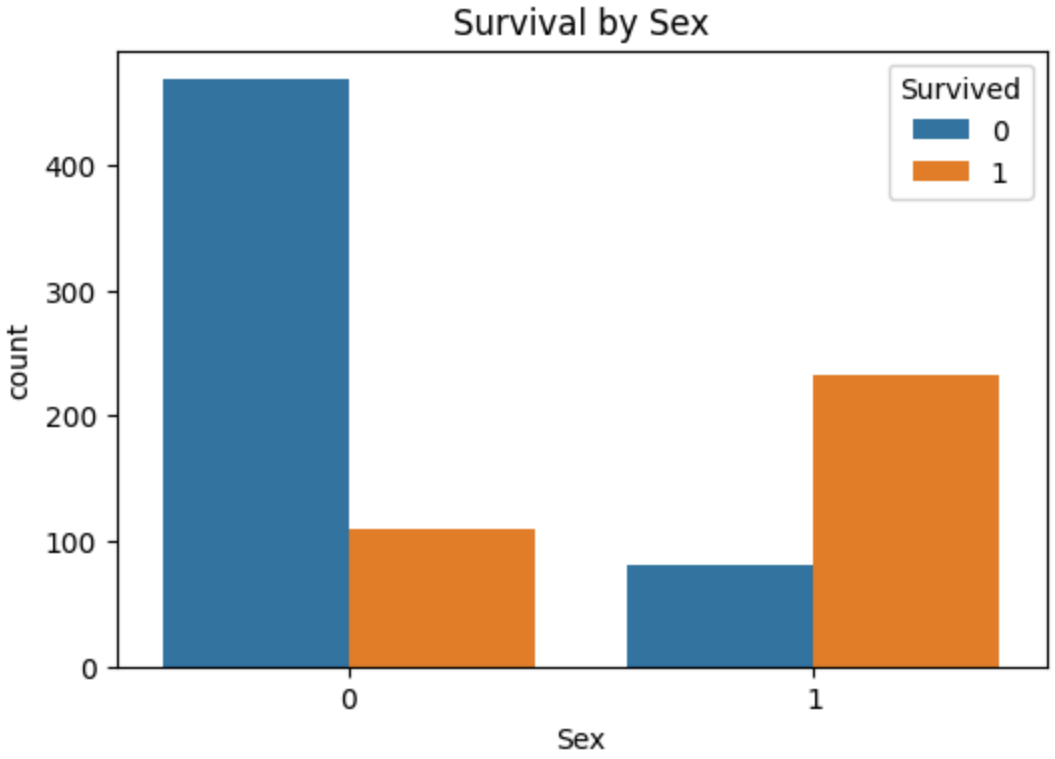
male': 0(blue), 'female': 1(orange)

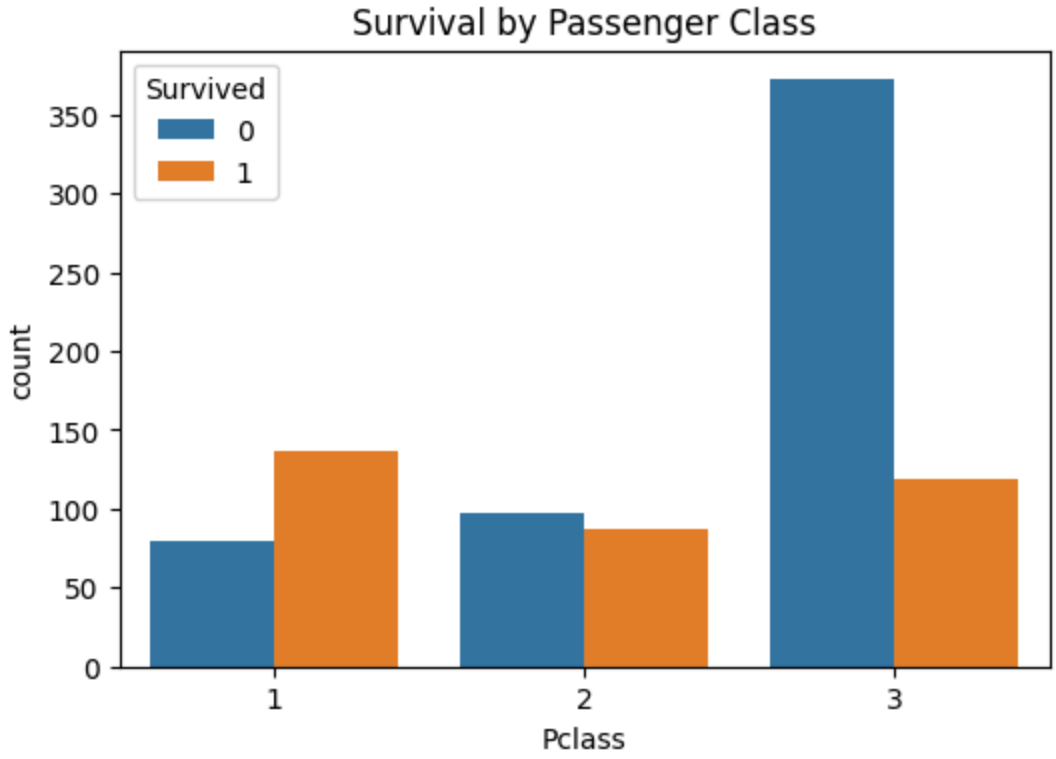
In [11]:

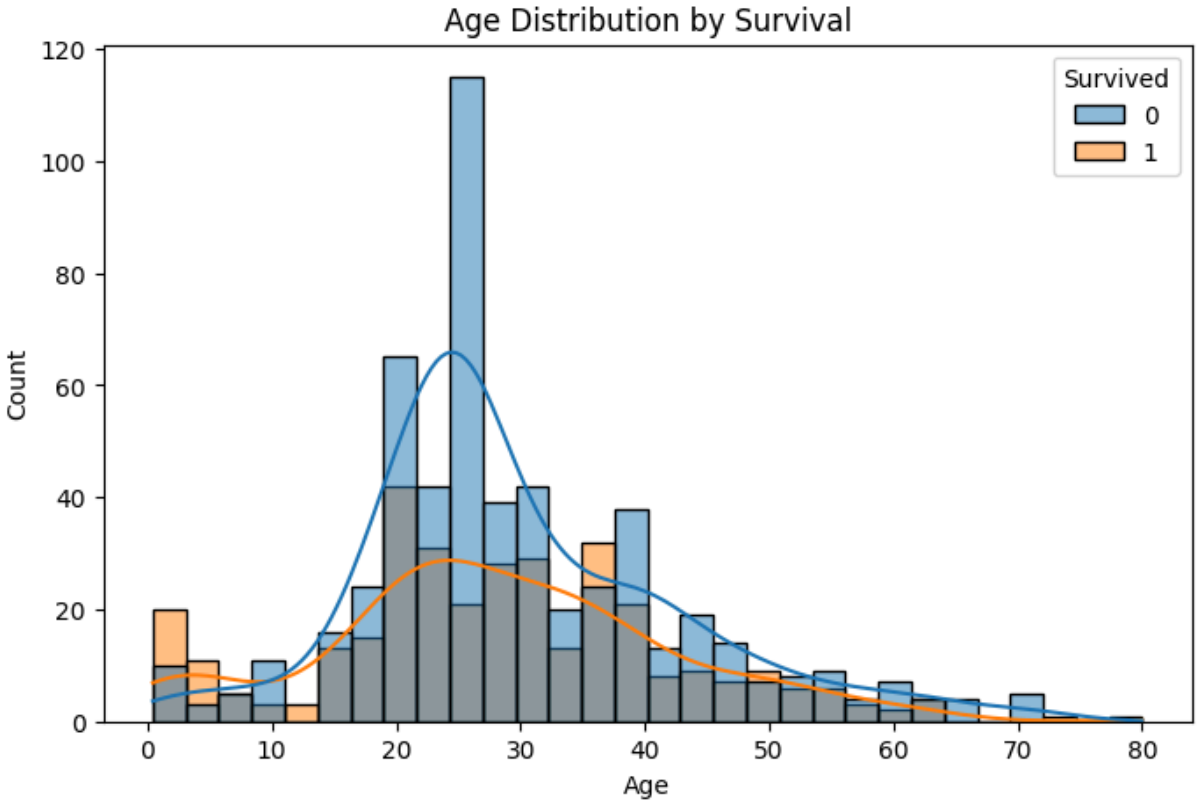
```
plt.figure(figsize=(5,4))
sns.countplot(x='Survived', data=titanic_df)
plt.title('Distribution of Survivors (1) and Non-survivors (0)')
plt.show()
plt.figure(figsize=(6,4))
sns.countplot(x='Sex', hue='Survived', data=titanic_df)
plt.title('Survival by Sex')
plt.show()
plt.figure(figsize=(6,4))
sns.countplot(x='Pclass', hue='Survived', data=titanic_df)
plt.title('Survival by Passenger Class')
plt.show()
plt.figure(figsize=(8,5))
sns.histplot(data=titanic_df, x='Age', hue='Survived', kde=True, bins=30)
plt.title('Age Distribution by Survival')
plt.show()
plt.figure(figsize=(8,5))
sns.boxplot(x='Survived', y='Fare', data=titanic_df)
plt.title('Fare Distribution by Survival')
plt.show()
plt.figure(figsize=(8,5))
sns.countplot(x='FamilySize', hue='Survived', data=titanic_df)
plt.title('Survival by Family Size')
plt.show()
plt.figure(figsize=(5,4))
sns.countplot(x='IsAlone', hue='Survived', data=titanic_df)
plt.title('Survival: Alone or Not')
plt.show()
plt.figure(figsize=(10,5))
```

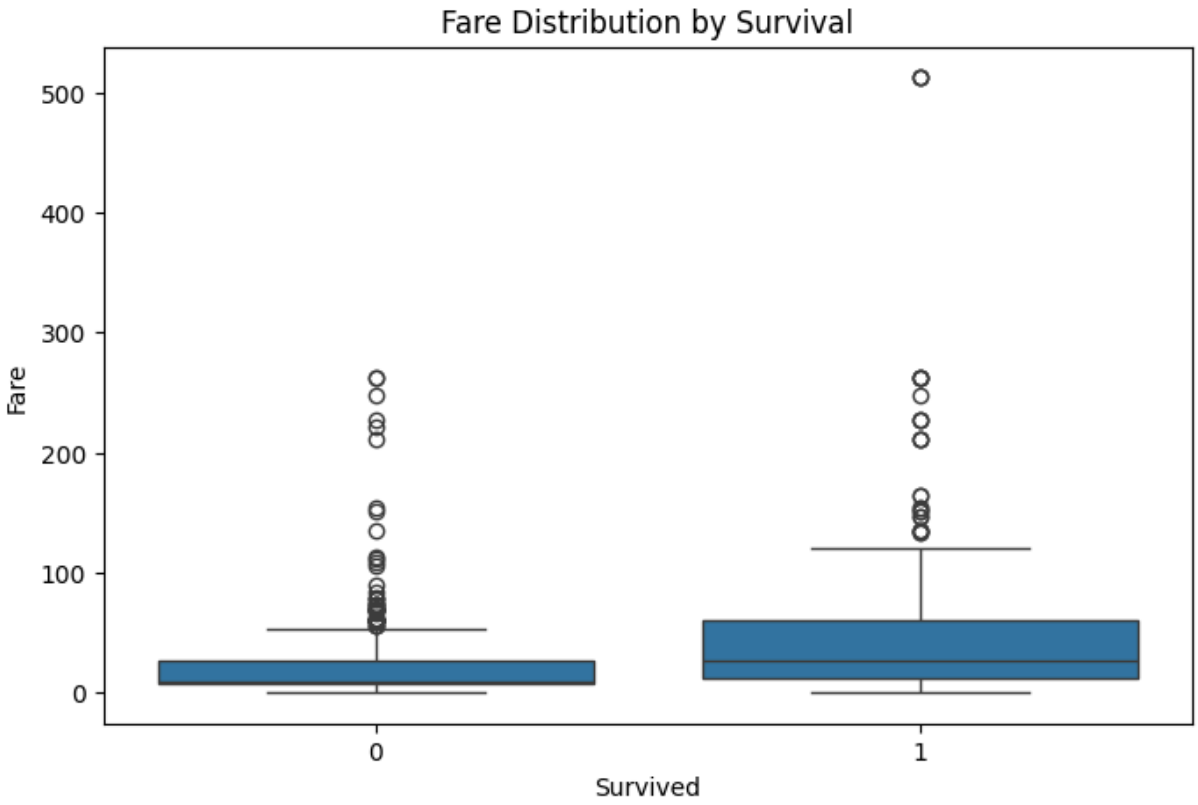
```
sns.countplot(x='Cabin_Unknown', hue='Survived', data=titanic_df)
plt.title('Survival by Cabin Deck (Unknown vs Others)')
plt.show()
if 'Cabin_C' in titanic_df.columns:
    plt.figure(figsize=(5,4))
    sns.countplot(x='Cabin_C', hue='Survived', data=titanic_df)
    plt.title('Survival for Cabin Deck C')
    plt.show()
```

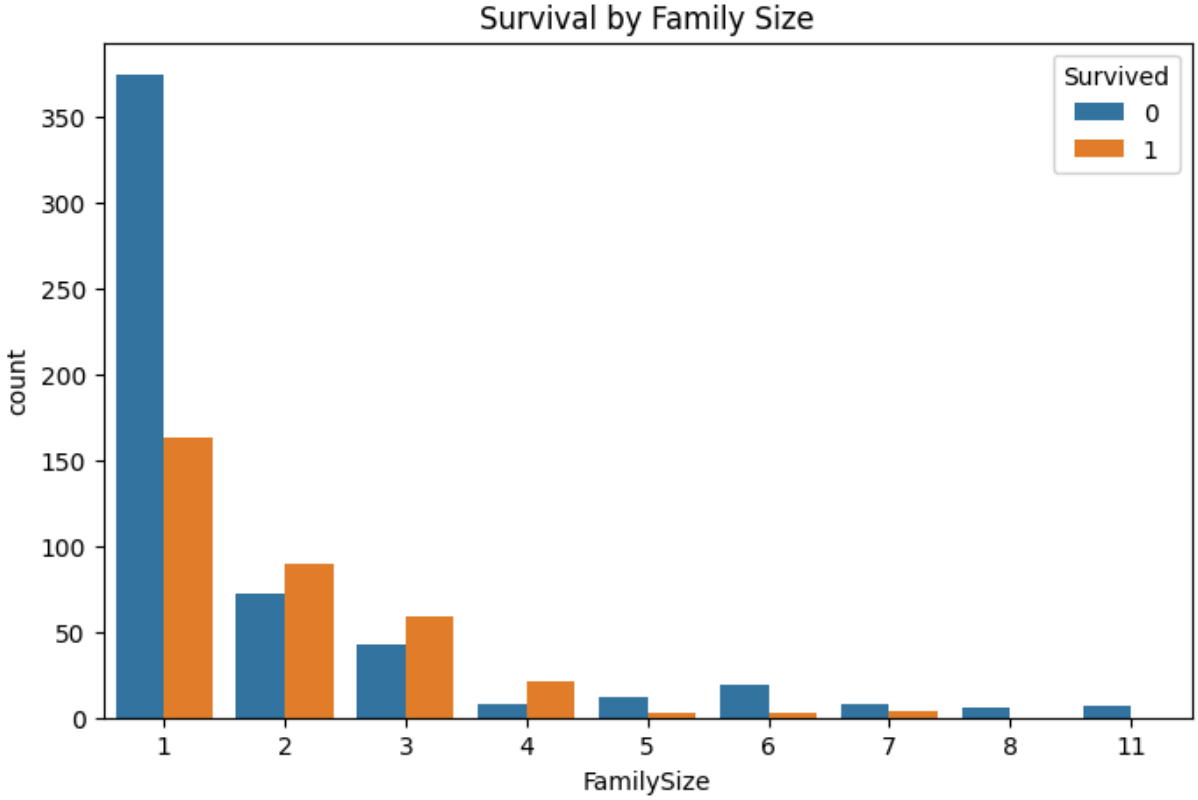


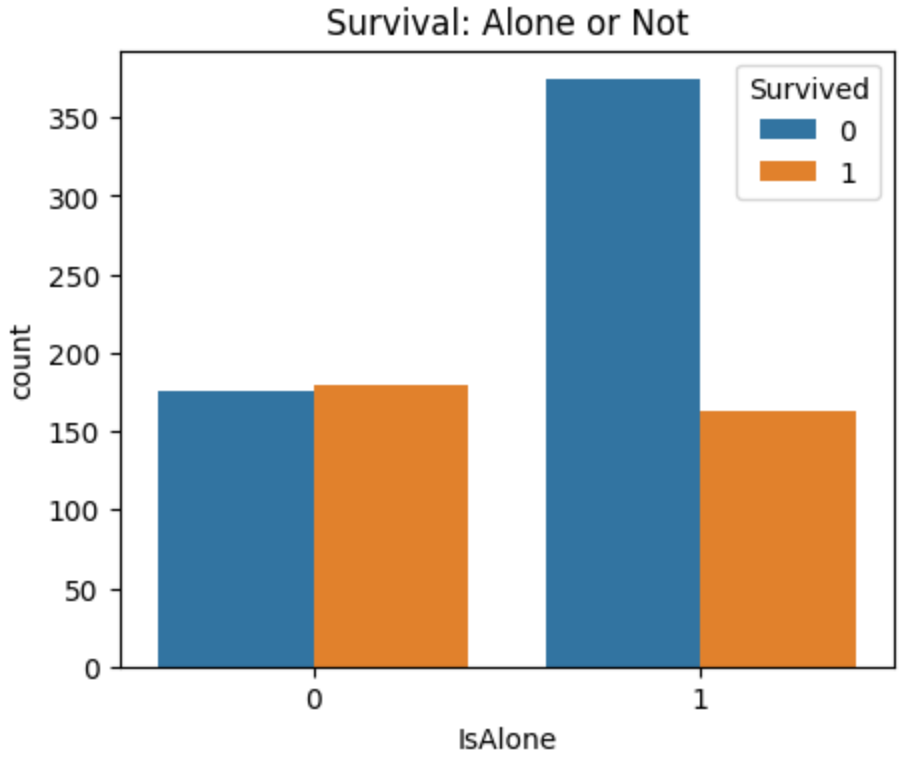


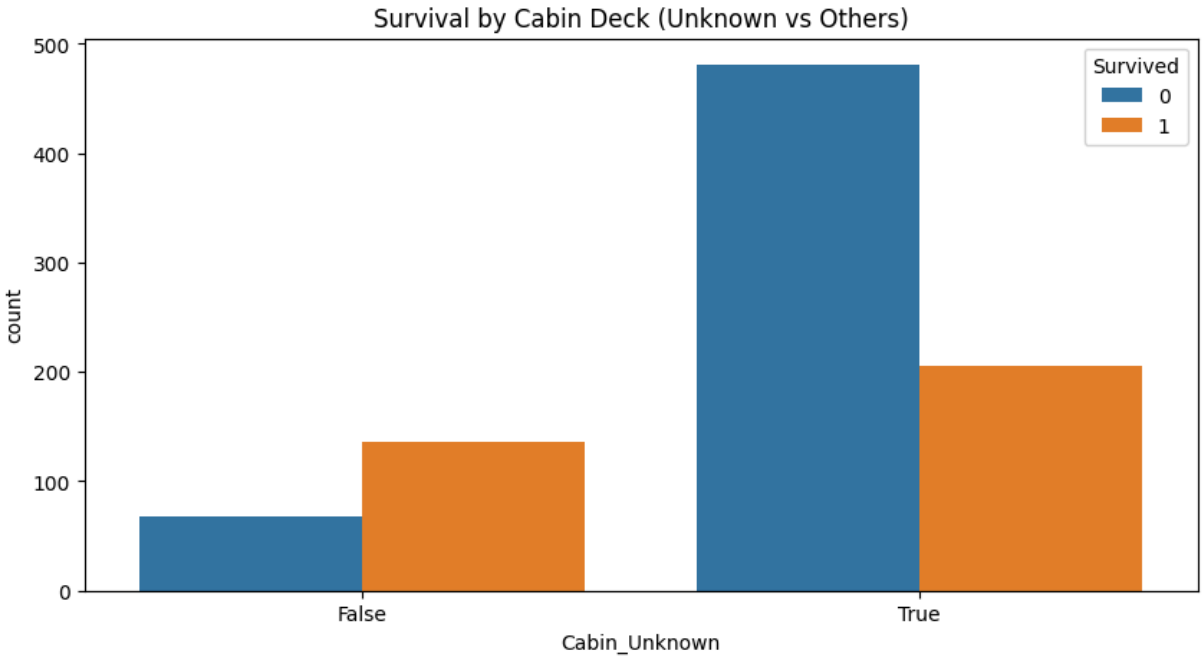

Survival by Passenger Class
Survived
0
1
count
Pclass
0
50
100
150
200
250
300
350
1
2
3

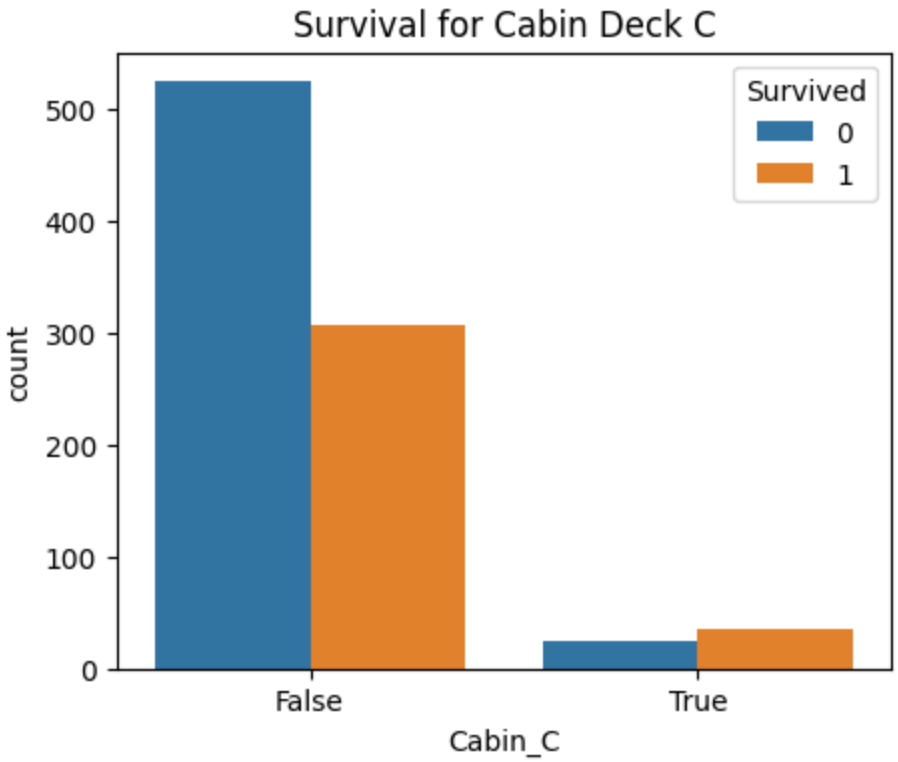

Age Distribution by Survival
Survived
0
1
Count
Age
0
20
40
60
80
100
120
0
10
20
30
40
50
60
70
80


Fare Distribution by Survival
Fare
Survived
0
1
0
100
200
300
400
500


Survival by Family Size
Survived
0
1
count
FamilySize
0
50
100
150
200
250
300
350
1
2
3
4
5
6
7
8
11


Survival: Alone or Not
Survived
0
1
count
350
300
250
200
150
100
50
0
0
1
IsAlone


Survival by Cabin Deck (Unknown vs Others)
Survived
0
1
count
500
400
300
200
100
0
False
True
Cabin_Unknown



In [12]:
```
print(titanic_df.head())
print(titanic_df.info())
```

```
             Survived  Pclass  Sex   Age  SibSp  Parch     Fare  FamilySize  \
PassengerId
1                   0       3    0  22.0      1      0   7.2500           2
2                   1       1    1  38.0      1      0  71.2833           2
3                   1       3    1  26.0      0      0   7.9250           1
4                   1       1    1  35.0      1      0  53.1000           2
5                   0       3    0  35.0      0      0   8.0500           1

             IsAlone  Embarked_Q  Embarked_S  Cabin_B  Cabin_C  Cabin_D  \
PassengerId
1                  0       False        True    False    False    False
2                  0       False       False    False     True    False
3                  1       False        True    False    False    False
4                  0       False        True    False     True    False
5                  1       False        True    False    False    False

             Cabin_E  Cabin_F  Cabin_G  Cabin_T  Cabin_Unknown
PassengerId
1              False    False    False    False           True
2              False    False    False    False          False
3              False    False    False    False           True
4              False    False    False    False          False
5              False    False    False    False           True
<class 'pandas.core.frame.DataFrame'>
Index: 891 entries, 1 to 891
Data columns (total 19 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Survived       891 non-null    int64
 1   Pclass         891 non-null    int64
 2   Sex            891 non-null    int64
 3   Age            891 non-null    float64
 4   SibSp          891 non-null    int64
 5   Parch          891 non-null    int64
 6   Fare           891 non-null    float64
 7   FamilySize     891 non-null    int64
 8   IsAlone        891 non-null    int32
 9   Embarked_Q     891 non-null    bool
 10  Embarked_S     891 non-null    bool
 11  Cabin_B        891 non-null    bool
 12  Cabin_C        891 non-null    bool
 13  Cabin_D        891 non-null    bool
 14  Cabin_E        891 non-null    bool
 15  Cabin_F        891 non-null    bool
 16  Cabin_G        891 non-null    bool
 17  Cabin_T        891 non-null    bool
 18  Cabin_Unknown  891 non-null    bool
dtypes: bool(10), float64(2), int32(1), int64(6)
memory usage: 74.8 KB
None
```

In [ ]: