In [ ]:

```
import numpy as np # Data Handling
import matplotlib.pyplot as plt # Data Visualization
import pandas as pd # # Data Handling
import os # Working Directory
from sklearn.preprocessing import LabelEncoder, OneHotEncoder # Transformation of Categor
from sklearn.compose import ColumnTransformer # Transformation same as level encoding and
from sklearn.model_selection import train_test_split # Splitting Data into Train & Test
from sklearn.preprocessing import StandardScaler # Neural Networks --> generally standari
from sklearn.metrics import confusion_matrix # Model Evaluation
from sklearn.metrics import classification_report # Model Evaluation
import keras # Deep Learning Framework
from keras.models import Sequential # Adding layers in the Neural Network
from keras.layers import Dense # Adding layers in the Neural Network
```

In [8]:

```
train = pd.read_csv("train.csv")
test = pd.read_csv("test.csv")
ss = pd.read_csv("gender_submission.csv")
```

In [9]:

```
train.head()
```

Out[9]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500
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 [10]:

```
test.head()
```

Out[10]:

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

In [11]:

```
print("Training set shape: ", train.shape)
print("Test set shape: ", test.shape)
```

```
Training set shape:  (891, 12)
Test set shape:  (418, 11)
```

In [12]:

```
ss.head()
```

Out[12]:

	PassengerId	Survived
0	892	0
1	893	1
2	894	0
3	895	0
4	896	1

In [13]:

```
ss.shape
```

Out[13]:

```
(418, 2)
```

In [14]:

```
train.info()
print('-'*40)
test.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
----------------------------------------
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 11 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  418 non-null    int64  
 1   Pclass       418 non-null    int64  
 2   Name         418 non-null    object 
 3   Sex          418 non-null    object 
 4   Age          332 non-null    float64
 5   SibSp        418 non-null    int64  
 6   Parch        418 non-null    int64  
 7   Ticket       418 non-null    object 
 8   Fare         417 non-null    float64
 9   Cabin        91 non-null     object 
 10  Embarked     418 non-null    object 
dtypes: float64(2), int64(4), object(5)
memory usage: 36.0+ KB
```

In [15]:

```
train.isnull().sum().sort_values(ascending = False)
```

Out[15]:

```
Cabin          687
Age            177
Embarked         2
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
SibSp            0
Parch            0
Ticket           0
Fare             0
dtype: int64
```

In [16]:

```
test.isnull().sum().sort_values(ascending = False)
```

Out[16]:

```
Cabin          327
Age             86
Fare             1
PassengerId      0
Pclass           0
Name             0
Sex              0
SibSp            0
Parch            0
Ticket           0
Embarked         0
dtype: int64
```

In [17]:

```
train.describe()
```

Out[17]:

	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 [18]:

```
# Summary statistics for test set

test.describe()
```

Out[18]:

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

In [19]:

```
# Value counts of the sex column

train['Sex'].value_counts(dropna = False)

# Comment: There are more male passengers than female passengers on titanic
```

Out[19]:

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

In [20]:

```
train[['Sex', 'Survived']].groupby('Sex', as_index = False).mean().sort_values(by = 'Surv
```

Out[20]:

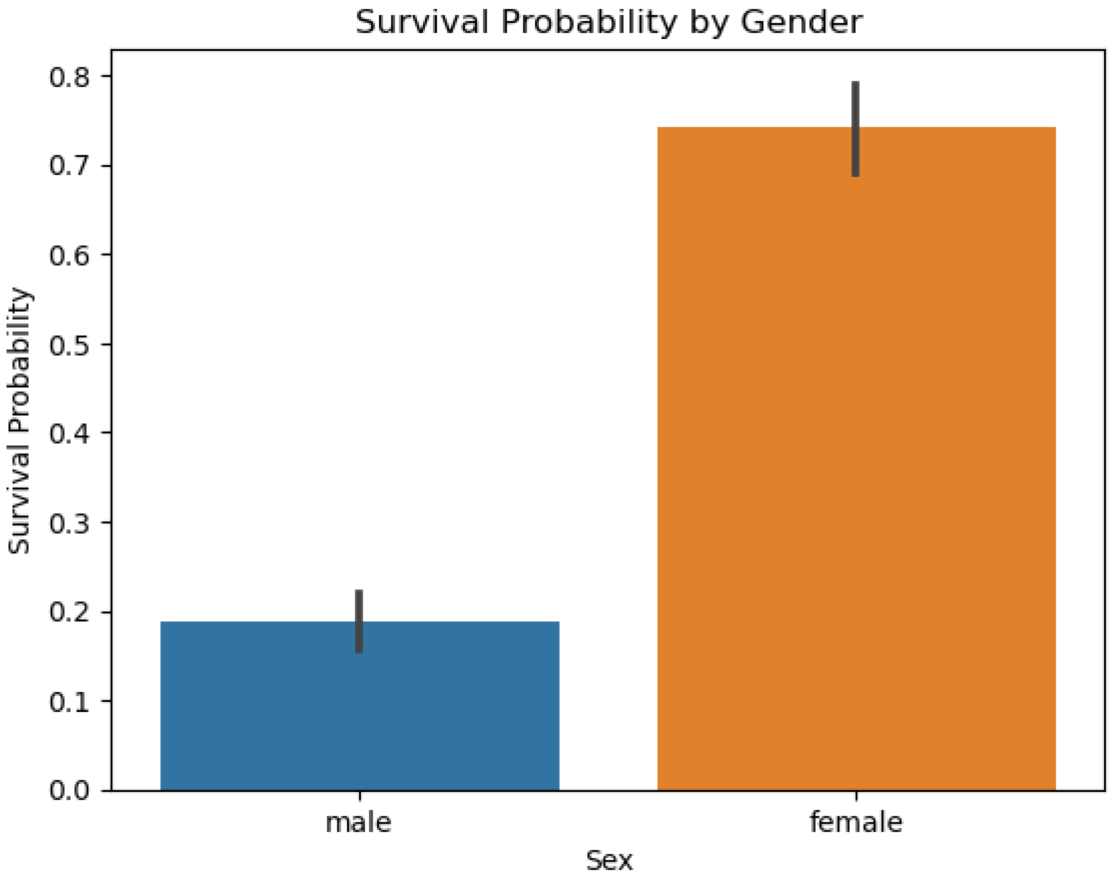
	Sex	Survived
0	female	0.742038
1	male	0.188908

In [21]:

```
sns.barplot(x = 'Sex', y ='Survived', data = train)
plt.ylabel('Survival Probability')
plt.title('Survival Probability by Gender')
```

Out[21]:

```
Text(0.5, 1.0, 'Survival Probability by Gender')
```



In [22]:

```
# Value counts of the Pclass column

train['Pclass'].value_counts(dropna = False)
```

Out[22]:

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

In [23]:

```
# Mean of survival by passenger class

train[['Pclass', 'Survived']].groupby(['Pclass'], as_index = False).mean().sort_values(by
```

Out[23]:

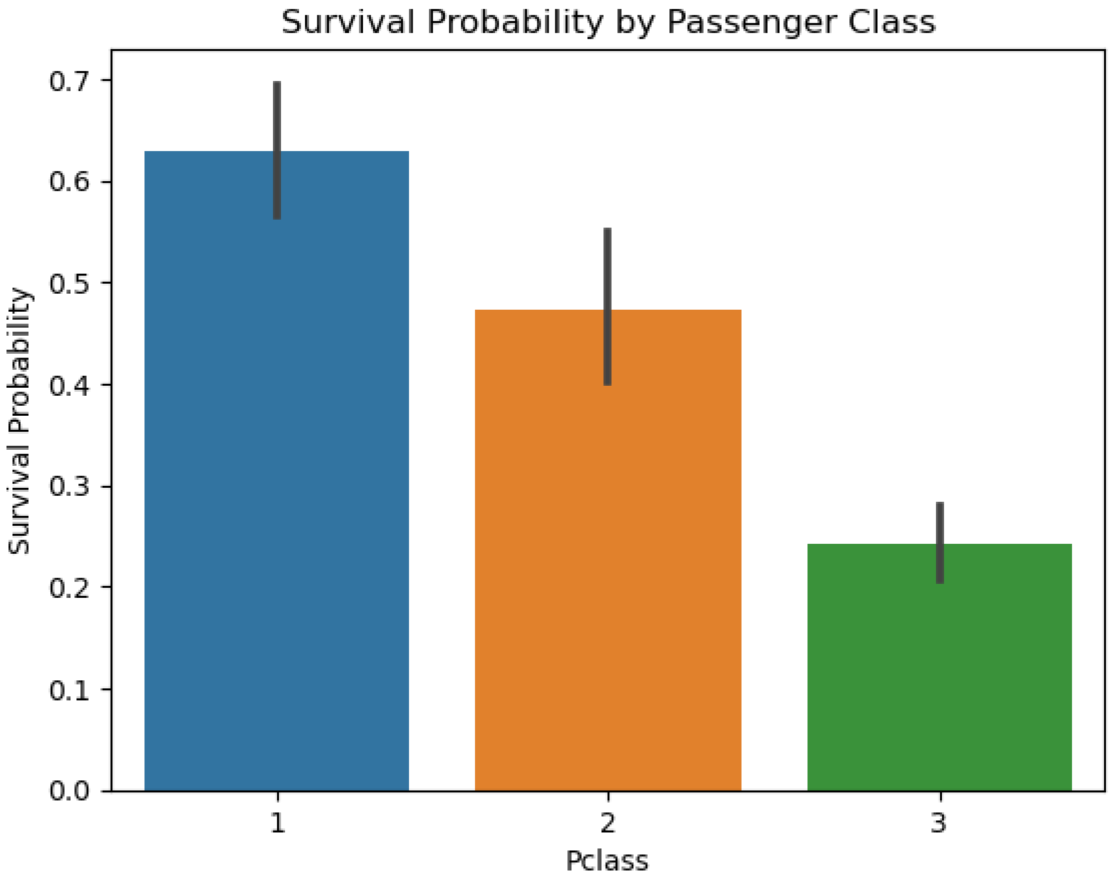
	Pclass	Survived
0	1	0.629630
1	2	0.472826
2	3	0.242363

In [24]:

```
sns.barplot(x = 'Pclass', y ='Survived', data = train)
plt.ylabel('Survival Probability')
plt.title('Survival Probability by Passenger Class')
```

Out[24]:

```
Text(0.5, 1.0, 'Survival Probability by Passenger Class')
```

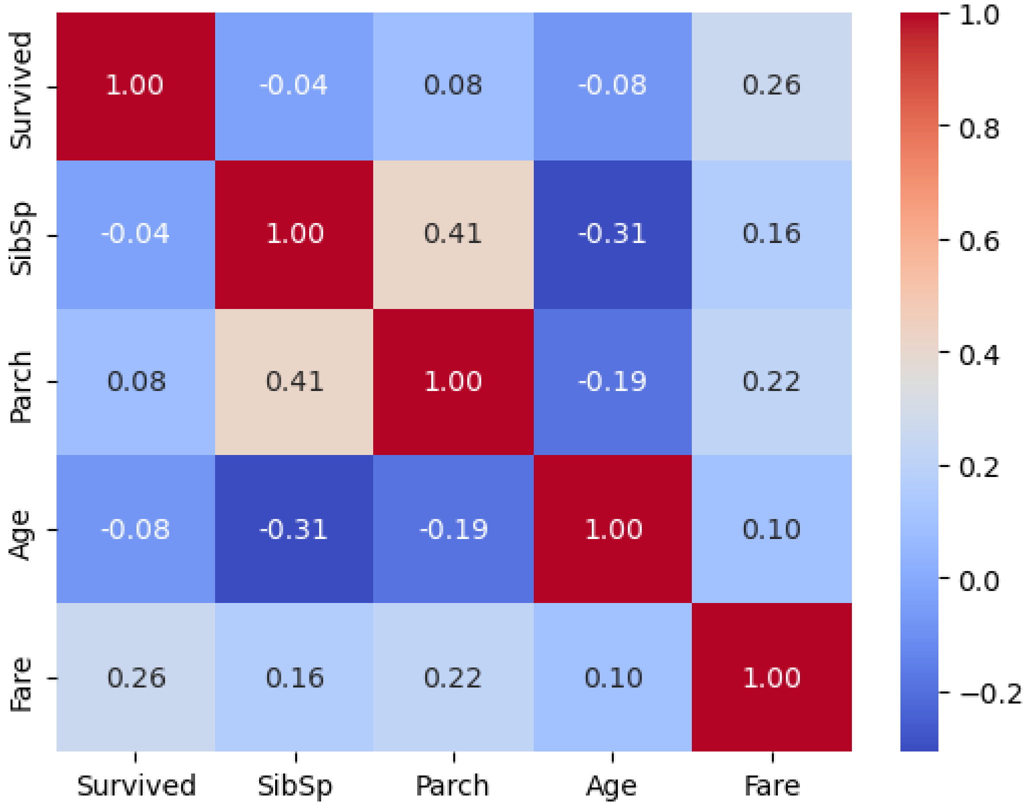


In [26]:

```
sns.heatmap(train[['Survived', 'SibSp', 'Parch', 'Age', 'Fare']].corr(), annot = True, fm

# Comment: Fare seems to be the only feature that has a substantial correlation with surv
```

Out[26]:

```
<Axes: >
```



In [27]:

```
train['SibSp'].value_counts(dropna = False)
```

Out[27]:

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

In [28]:

```
# Mean of survival by SibSp

train[['SibSp', 'Survived']].groupby('SibSp', as_index = False).mean().sort_values(by = '
```

Out[28]:

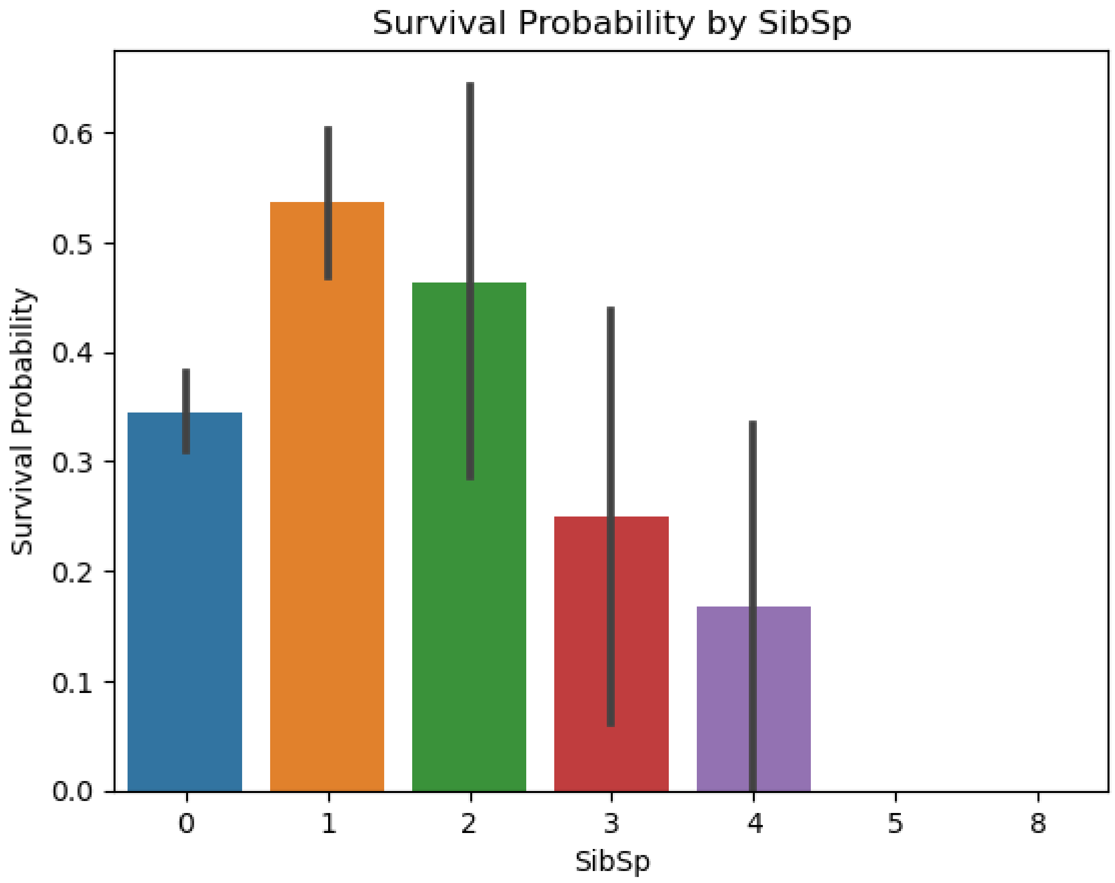
	SibSp	Survived
1	1	0.535885
2	2	0.464286
0	0	0.345395
3	3	0.250000
4	4	0.166667
5	5	0.000000
6	8	0.000000

In [29]:

```
sns.barplot(x = 'SibSp', y ='Survived', data = train)
plt.ylabel('Survival Probability')
plt.title('Survival Probability by SibSp')
```

Out[29]:

```
Text(0.5, 1.0, 'Survival Probability by SibSp')
```



In [30]:

```
# Value counts of the Parch column

train['Parch'].value_counts(dropna = False)
```

Out[30]:

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

In [31]:

```
# Mean of survival by Parch

train[['Parch', 'Survived']].groupby('Parch', as_index = False).mean().sort_values(by = '
```

Out[31]:

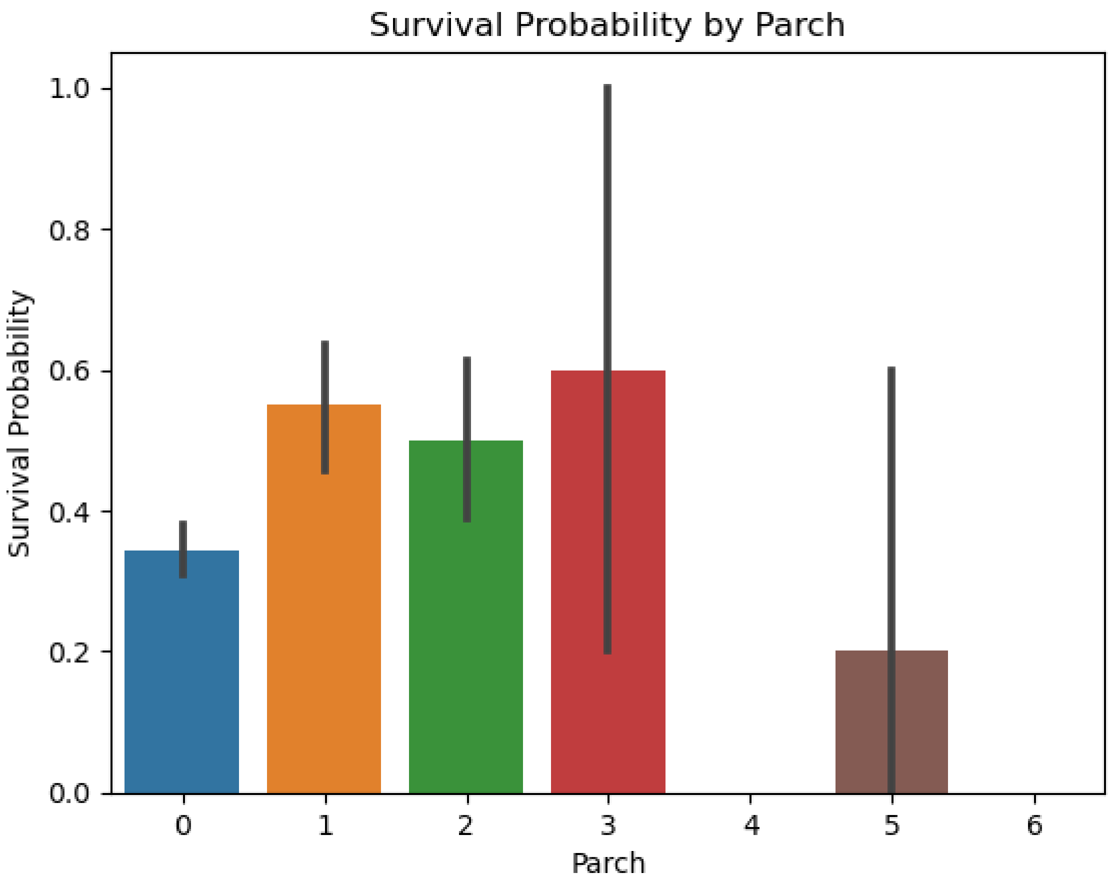
	Parch	Survived
3	3	0.600000
1	1	0.550847
2	2	0.500000
0	0	0.343658
5	5	0.200000
4	4	0.000000
6	6	0.000000

In [32]:

```
sns.barplot(x = 'Parch', y ='Survived', data = train)
plt.ylabel('Survival Probability')
plt.title('Survival Probability by Parch')
```

Out[32]:

```
Text(0.5, 1.0, 'Survival Probability by Parch')
```



In [33]:

```
# Null values in Age column

train['Age'].isnull().sum()
```

Out[33]:

```
177
```

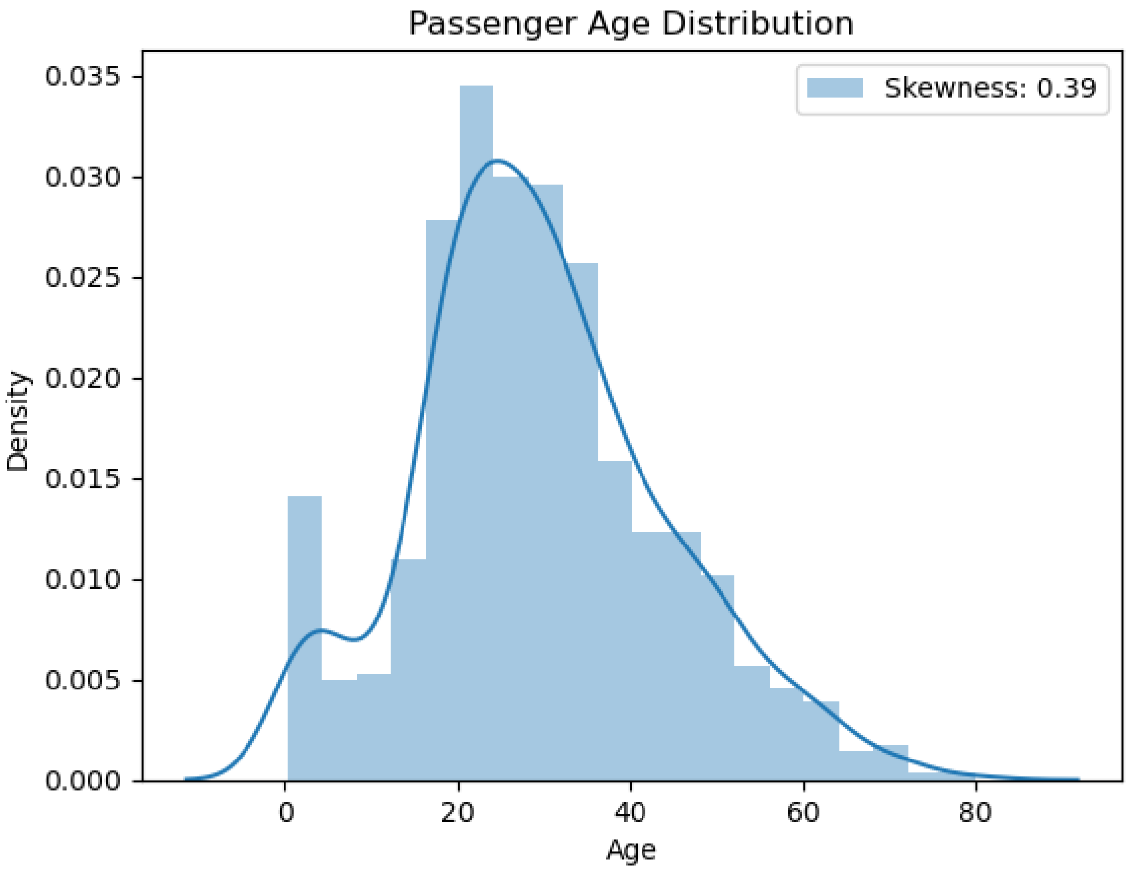
In [34]:

```
# Passenger age distribution

sns.distplot(train['Age'], label = 'Skewness: %.2f'%(train['Age'].skew()))
plt.legend(loc = 'best')
plt.title('Passenger Age Distribution')
```

```
C:\Users\rahul\AppData\Local\Temp\ipykernel_7916\385672501.py:3: UserWarni
ng:

`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(train['Age'], label = 'Skewness: %.2f'%(train['Age'].skew
()))
```

Out[34]:

```
Text(0.5, 1.0, 'Passenger Age Distribution')
```



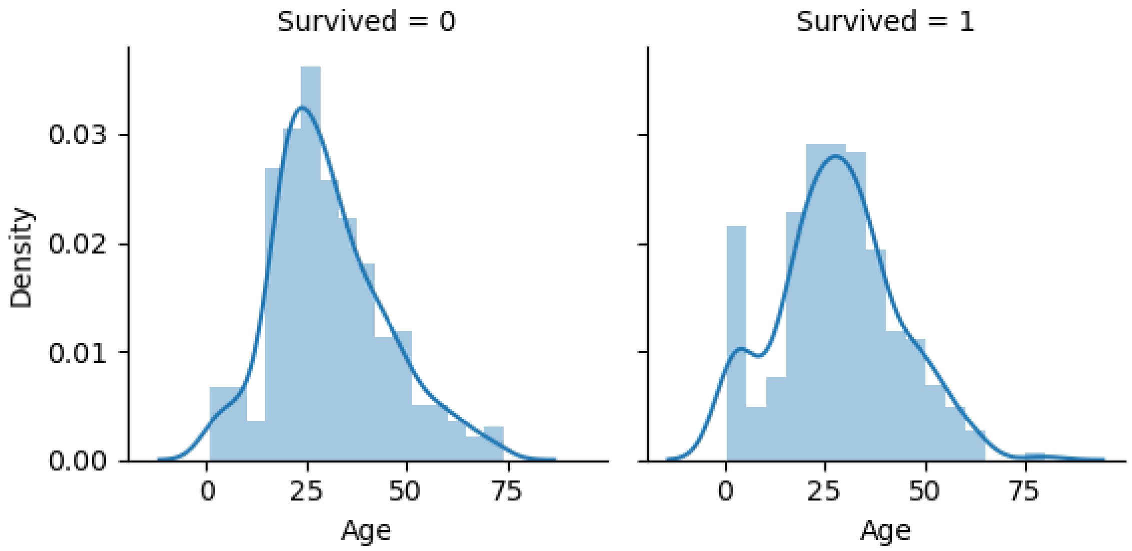
In [35]:

```
# Age distribution by survival

g = sns.FacetGrid(train, col = 'Survived')
g.map(sns.distplot, 'Age')
```

```
E:\anaconda\lib\site-packages\seaborn\axisgrid.py:848: 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)

  func(*plot_args, **plot_kwargs)
E:\anaconda\lib\site-packages\seaborn\axisgrid.py:848: 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)

  func(*plot_args, **plot_kwargs)
```

Out[35]:

```
<seaborn.axisgrid.FacetGrid at 0x20cf7f6f0d0>
```

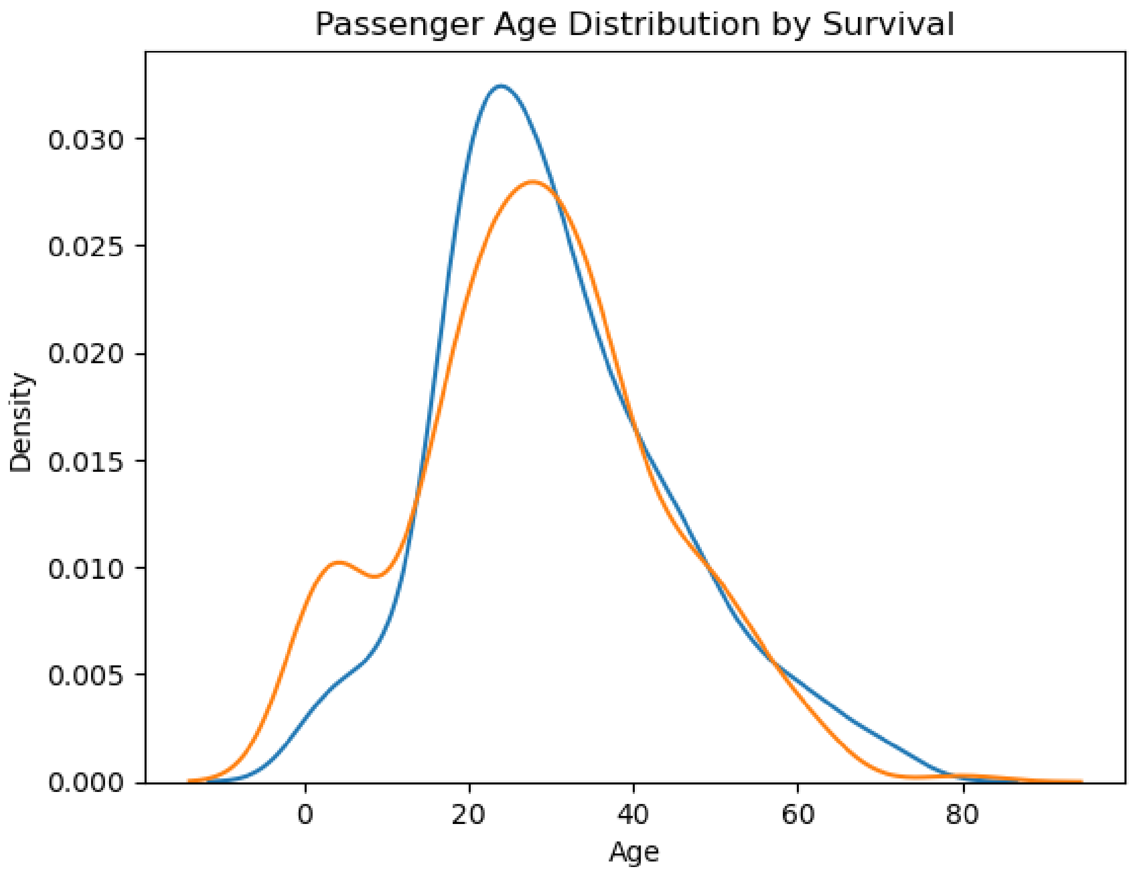


In [36]:

```
sns.kdeplot(train['Age'][train['Survived'] == 0], label = 'Did not survive')
sns.kdeplot(train['Age'][train['Survived'] == 1], label = 'Survived')
plt.xlabel('Age')
plt.title('Passenger Age Distribution by Survival')
```

Out[36]:

```
Text(0.5, 1.0, 'Passenger Age Distribution by Survival')
```



In [37]:

```
train['Fare'].isnull().sum()
```

Out[37]:

```
0
```

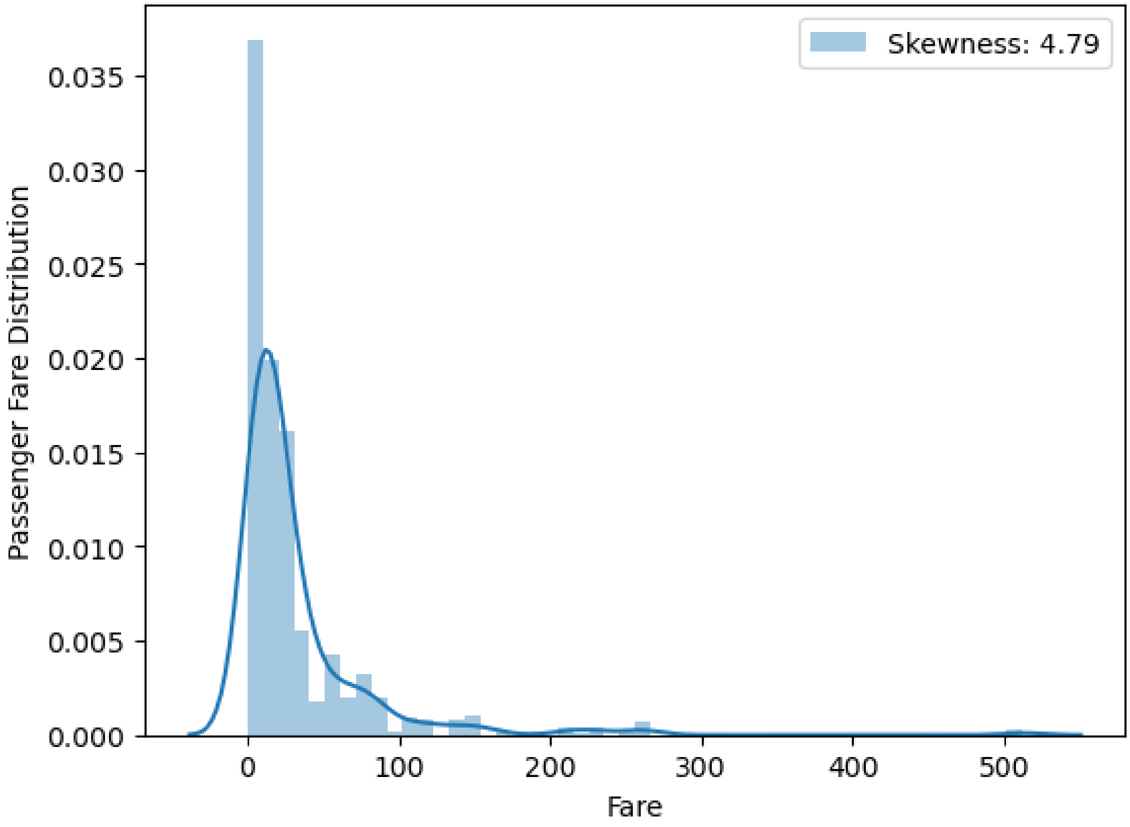
In [38]:

```
# Passenger fare distribution

sns.distplot(train['Fare'], label = 'Skewness: %.2f'%(train['Fare'].skew()))
plt.legend(loc = 'best')
plt.ylabel('Passenger Fare Distribution')
```

```
C:\Users\rahul\AppData\Local\Temp\ipykernel_7916\1143978767.py:3: UserWarn
ing:

`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(train['Fare'], label = 'Skewness: %.2f'%(train['Fare'].skew
()))
```

Out[38]:

```
Text(0, 0.5, 'Passenger Fare Distribution')
```



In [39]:

```
X_train = train.drop('Survived', axis = 1)
Y_train = train['Survived']
X_test = test.drop('PassengerId', axis = 1).copy()
print("X_train shape: ", X_train.shape)
print("Y_train shape: ", Y_train.shape)
print("X_test shape: ", X_test.shape)
```

```
X_train shape:  (891, 11)
Y_train shape:  (891,)
X_test shape:  (418, 10)
```

In [41]:

```
train = train.drop(['Ticket', 'Cabin'], axis = 1)
test = test.drop(['Ticket', 'Cabin'], axis = 1)
```

In [42]:

```
# Missing values in training set

train.isnull().sum().sort_values(ascending = False)
```

Out[42]:

```
Age            177
Embarked         2
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
SibSp            0
Parch            0
Fare             0
dtype: int64
```

In [43]:

```
# Compute the most frequent value of Embarked in training set

mode = train['Embarked'].dropna().mode()[0]
mode
```

Out[43]:

```
'S'
```

In [44]:

```
# Fill missing value in Embarked with mode

train['Embarked'].fillna(mode, inplace = True)
```

In [45]:

```
test.isnull().sum().sort_values(ascending = False)
```

Out[45]:

```
Age            86
Fare            1
PassengerId     0
Pclass          0
Name            0
Sex             0
SibSp           0
Parch           0
Embarked        0
dtype: int64
```

In [46]:

```
# Compute median of Fare in test set

median = test['Fare'].dropna().median()
median
```

Out[46]:

```
14.4542
```

In [47]:

```
# Fill missing value in Fare with median

test['Fare'].fillna(median, inplace = True)
```

In [48]:

```
# Combine training set and test set

combine = pd.concat([train, test], axis = 0).reset_index(drop = True)
combine.head()
```

Out[48]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Embarked
0	1	0.0	3	Braund, Mr. Owen Harris	male	22.0	1	0	7.2500	S
1	2	1.0	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	71.2833	C
2	3	1.0	3	Heikkinen, Miss. Laina	female	26.0	0	0	7.9250	S
3	4	1.0	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	53.1000	S
4	5	0.0	3	Allen, Mr. William Henry	male	35.0	0	0	8.0500	S

In [49]:

```
# Missing values in the combined dataset

combine.isnull().sum().sort_values(ascending = False)
```

Out[49]:

```
Survived       418
Age            263
PassengerId      0
Pclass           0
Name             0
Sex              0
SibSp            0
Parch            0
Fare             0
Embarked         0
dtype: int64
```

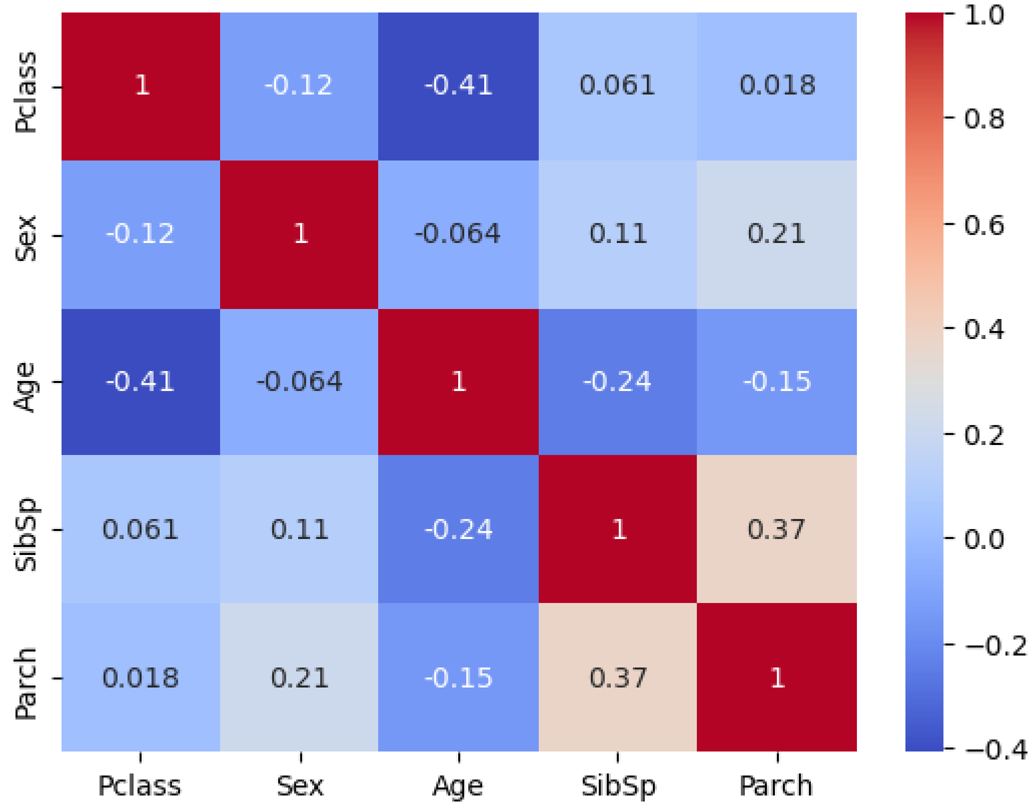
In [52]:

```
sns.heatmap(combine.drop(['Survived', 'Name', 'PassengerId', 'Fare'], axis = 1).corr(), a
```

```
C:\Users\rahul\AppData\Local\Temp\ipykernel_7916\2145782946.py:1: FutureWa
rning: The default value of numeric_only in DataFrame.corr is deprecated.
In a future version, it will default to False. Select only valid columns o
r specify the value of numeric_only to silence this warning.
  sns.heatmap(combine.drop(['Survived', 'Name', 'PassengerId', 'Fare'], ax
is = 1).corr(), annot = True, cmap = 'coolwarm')
```

Out[52]:

```
<Axes: >
```



In [ ]: