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

In [2]:
```
data= pd.read_csv(r"H:\Data\titanic.csv")
```

In [3]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 15 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   survived     891 non-null    int64  
 1   pclass       891 non-null    int64  
 2   sex          891 non-null    object 
 3   age          714 non-null    float64
 4   sibsp        891 non-null    int64  
 5   parch        891 non-null    int64  
 6   fare         891 non-null    float64
 7   embarked     889 non-null    object 
 8   class        891 non-null    object 
 9   who          891 non-null    object 
 10  adult_male   891 non-null    bool   
 11  deck         203 non-null    object 
 12  embark_town  889 non-null    object 
 13  alive        891 non-null    object 
 14  alone        891 non-null    bool   
dtypes: bool(2), float64(2), int64(4), object(7)
memory usage: 92.4+ KB
```

In [4]:
```
data.head()
```

Out[4]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	NaN	Southa
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	C	Che
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	NaN	Southa
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	C	Southa
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	NaN	Southa

Removing irrelevant and null data from the train and test datasets.

In [5]:
```
data= data.drop('deck', axis=1)
data= data.drop('embarked', axis=1)
```

In [6]:
```
data.isna().sum()
```

Out[6]:
```
survived         0
pclass           0
sex              0
age            177
sibsp            0
parch            0
fare             0
class            0
who              0
adult_male       0
embark_town      2
alive            0
alone            0
dtype: int64
```

In [7]:
```
data['age']= data['age'].fillna(data['age'].mean())
```

In [8]:
```
data= data.dropna()
```

In [9]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 889 entries, 0 to 890
Data columns (total 13 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   survived     889 non-null    int64  
 1   pclass       889 non-null    int64  
 2   sex          889 non-null    object 
 3   age          889 non-null    float64
 4   sibsp        889 non-null    int64  
 5   parch        889 non-null    int64  
 6   fare         889 non-null    float64
 7   class        889 non-null    object 
 8   who          889 non-null    object 
 9   adult_male   889 non-null    bool   
 10  embark_town  889 non-null    object 
 11  alive        889 non-null    object 
 12  alone        889 non-null    bool   
dtypes: bool(2), float64(2), int64(4), object(5)
memory usage: 85.1+ KB
```

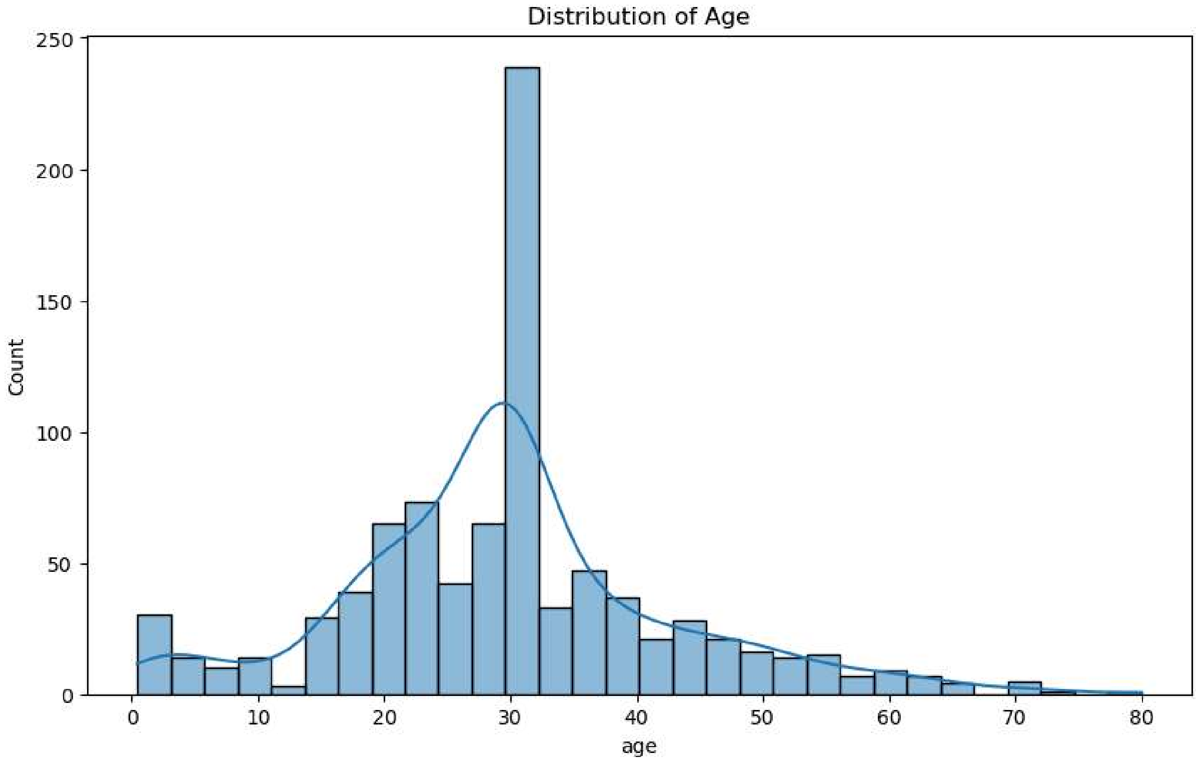
In [10]:
```
data['embark_town'].value_counts()
```

Out[10]:
```
Southampton    644
Cherbourg      168
Queenstown      77
Name: embark_town, dtype: int64
```

In [11]:
```
data['embark_town']= data['embark_town'].replace({'S': 'Southampton', 'C': 'Cherbourg',
```

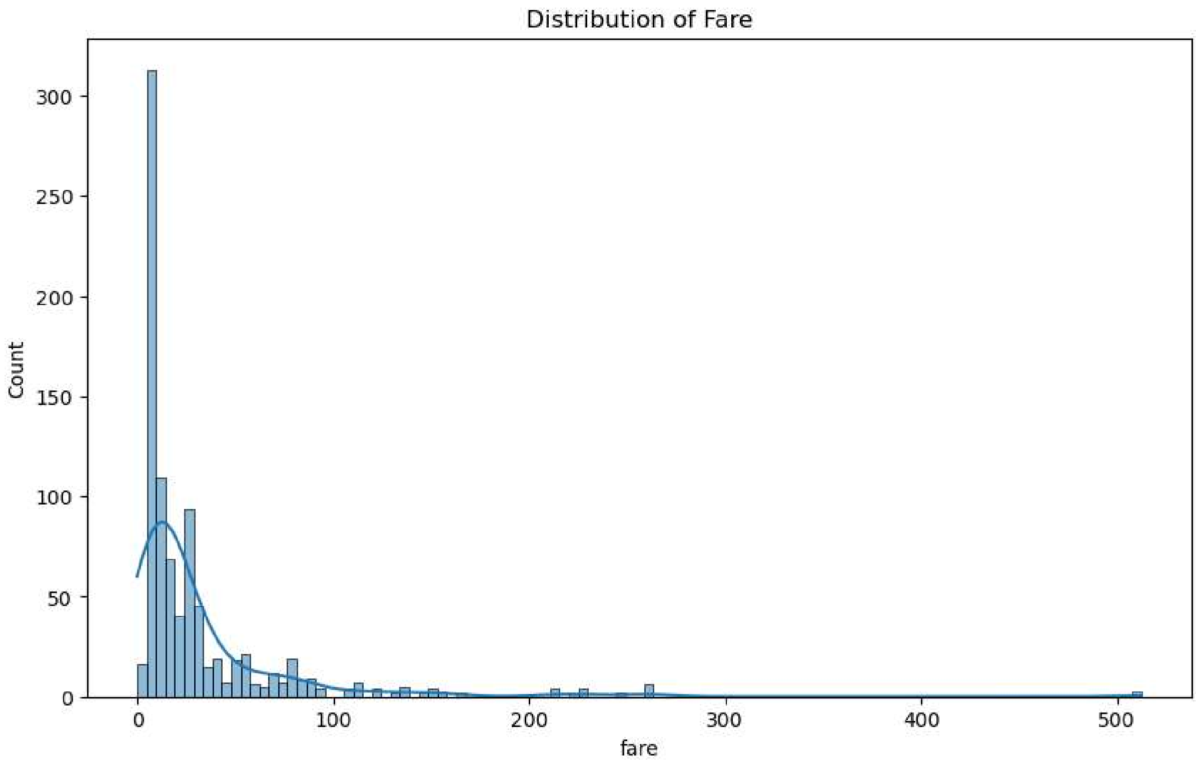
By observing Histogram for the distribution of Age I observed that the majority of the passengers falls in the range of age 18 to 35.

In [12]:
```
plt.figure(figsize=(10,6))
sns.histplot(data['age'], kde= True)
plt.title('Distribution of Age')
plt.show()
```

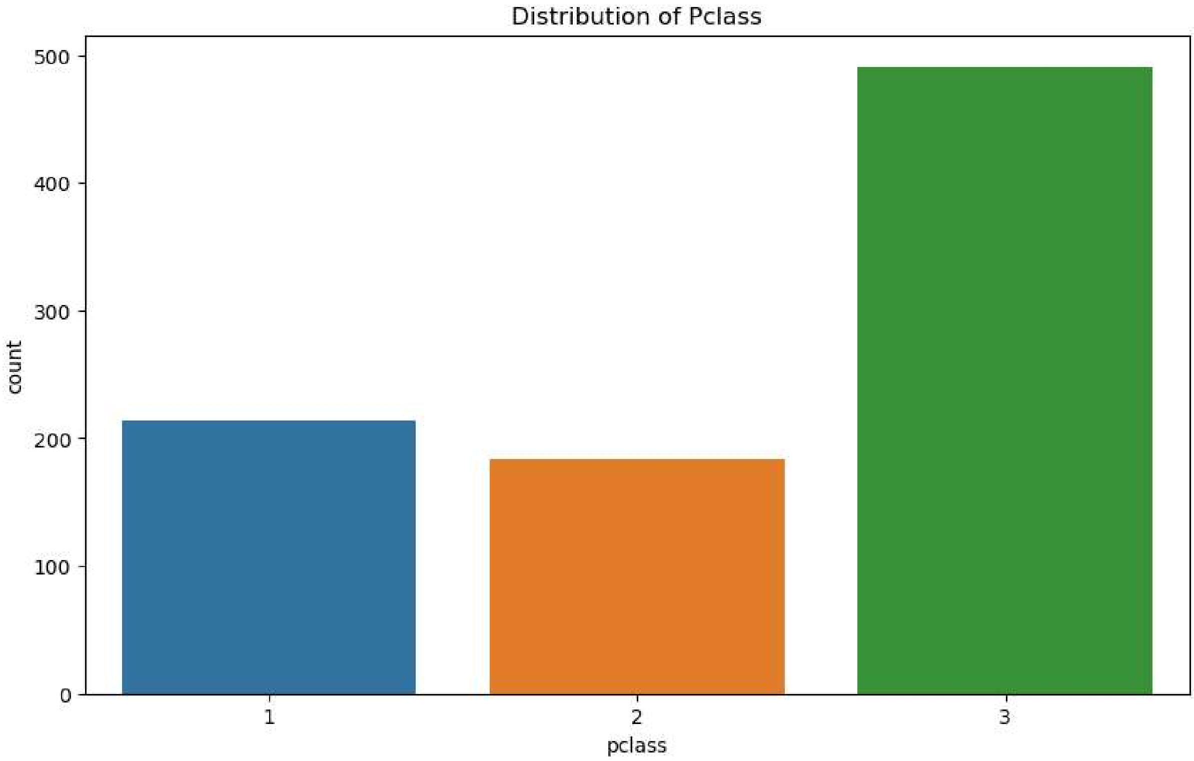


As expected a lot of passengers had bought the ticket which was low in fare.

In [13]:
```
plt.figure(figsize=(10, 6))
sns.histplot(data['fare'], kde=True)
plt.title('Distribution of Fare')
plt.show()
```

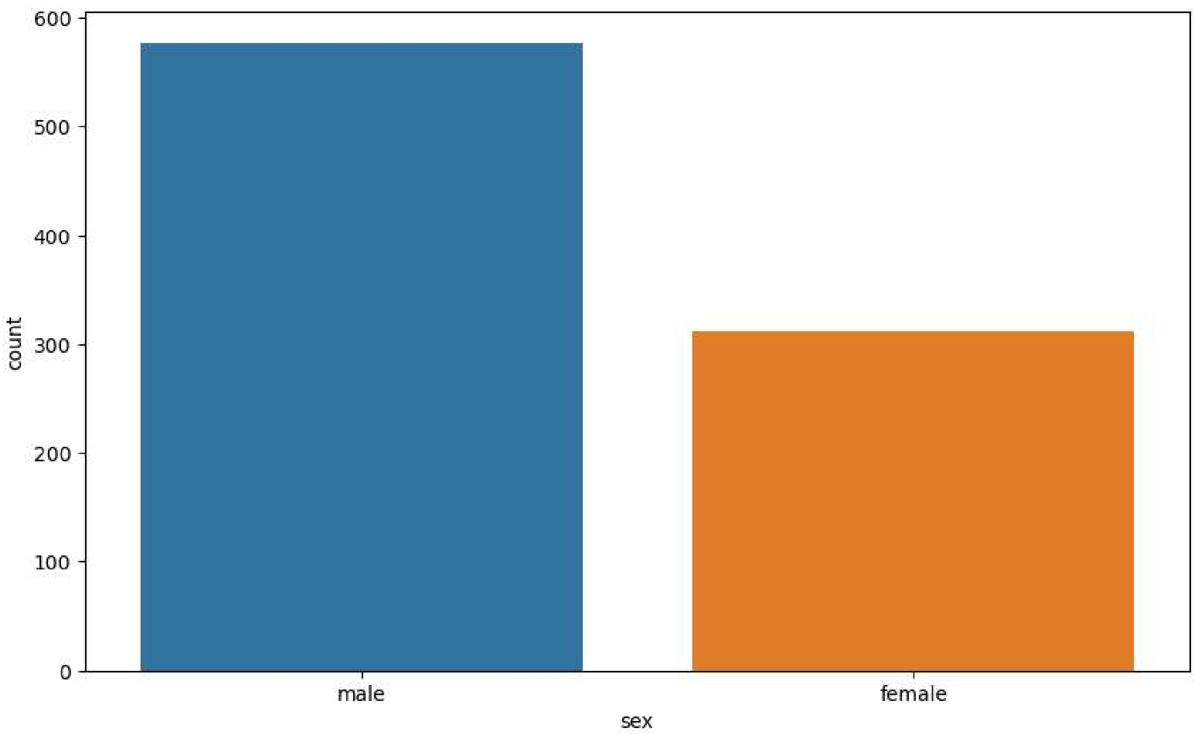


In [14]:
```
plt.figure(figsize=(10,6))
sns.countplot(data= data, x= 'pclass')
plt.title('Distribution of Pclass')
plt.show()
```



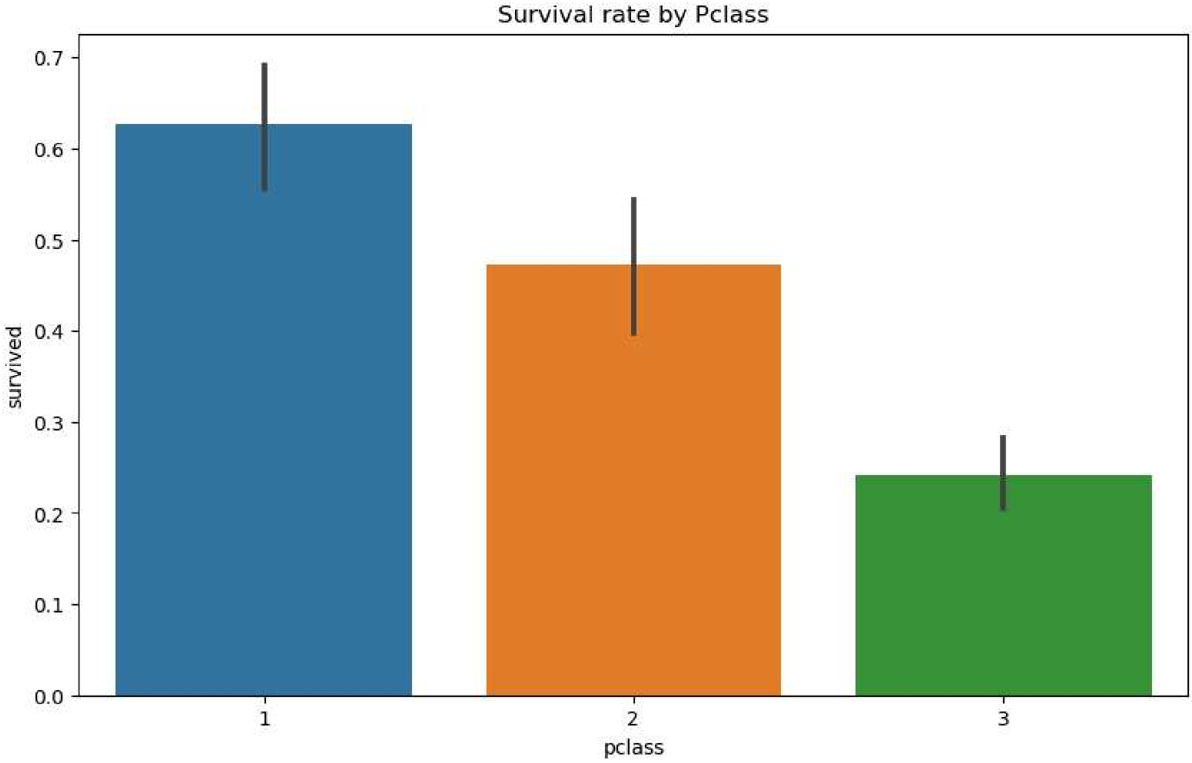
There were more number of males as compared to the females present on the ship. Male passengers were almost 1.5 times of the females.

In [15]:
```
plt.figure(figsize=(10,6))
sns.countplot(data= data, x='sex')
plt.show()
```



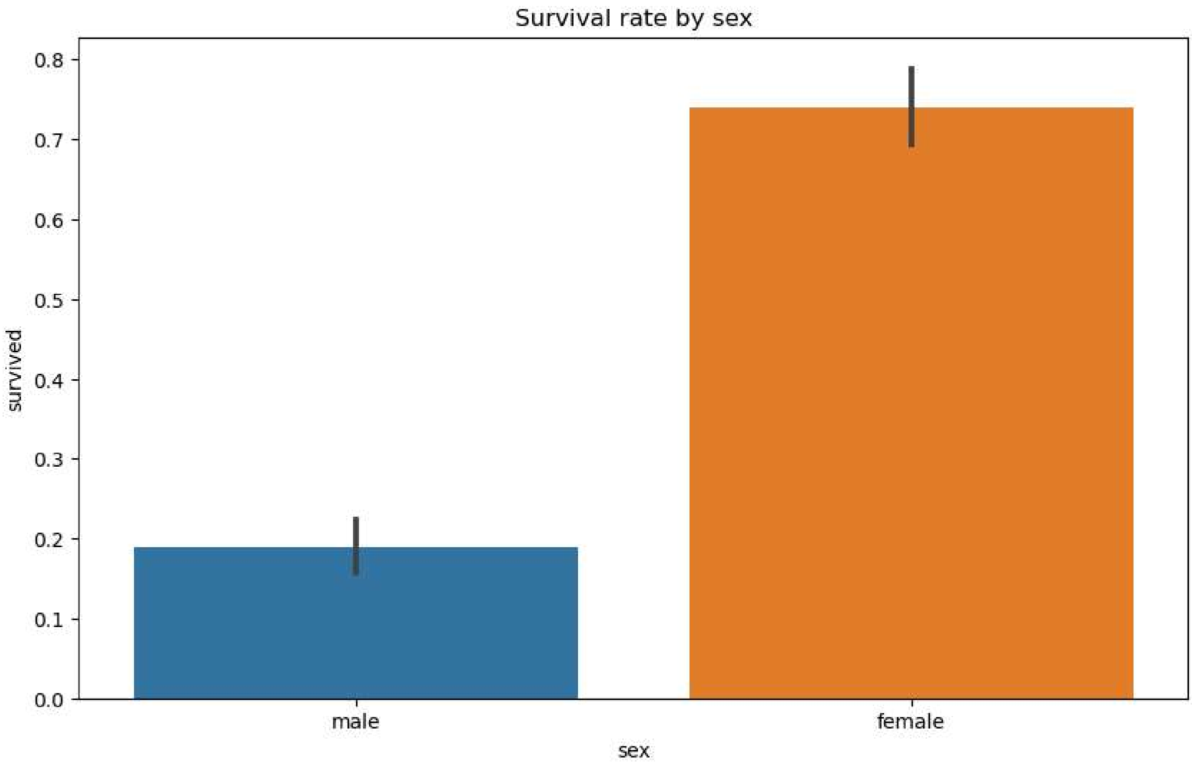
Most of the passengers that had survived were of the first class and second class.

In [16]:
```
plt.figure(figsize=(10,6))
sns.barplot(data= data, x='pclass', y='survived')
plt.title('Survival rate by Pclass')
plt.show()
```



Most of the females had survived while most of the males had to loose their lives.

In [17]:
```
plt.figure(figsize=(10,6))
sns.barplot(data= data, x='sex', y='survived')
plt.title('Survival rate by sex')
plt.show()
```



In [18]:
```
age_bins= [0, 12, 18, 35, 60, 80]
age_labels= ['Child', 'Teen', 'Adult', 'Mid_Age', 'Senior']
data['Age_Group']= pd.cut(data['age'], bins= age_bins, labels= age_labels, right= False)
```

In [19]:
```
from sklearn.preprocessing import LabelEncoder
le= LabelEncoder()
```

In [35]:
```
for col in data.columns:
    if data[col].dtype== 'object' or data[col].dtype== 'category':
        data[col]= le.fit_transform(data[col])
        print(col, le.classes_)
```

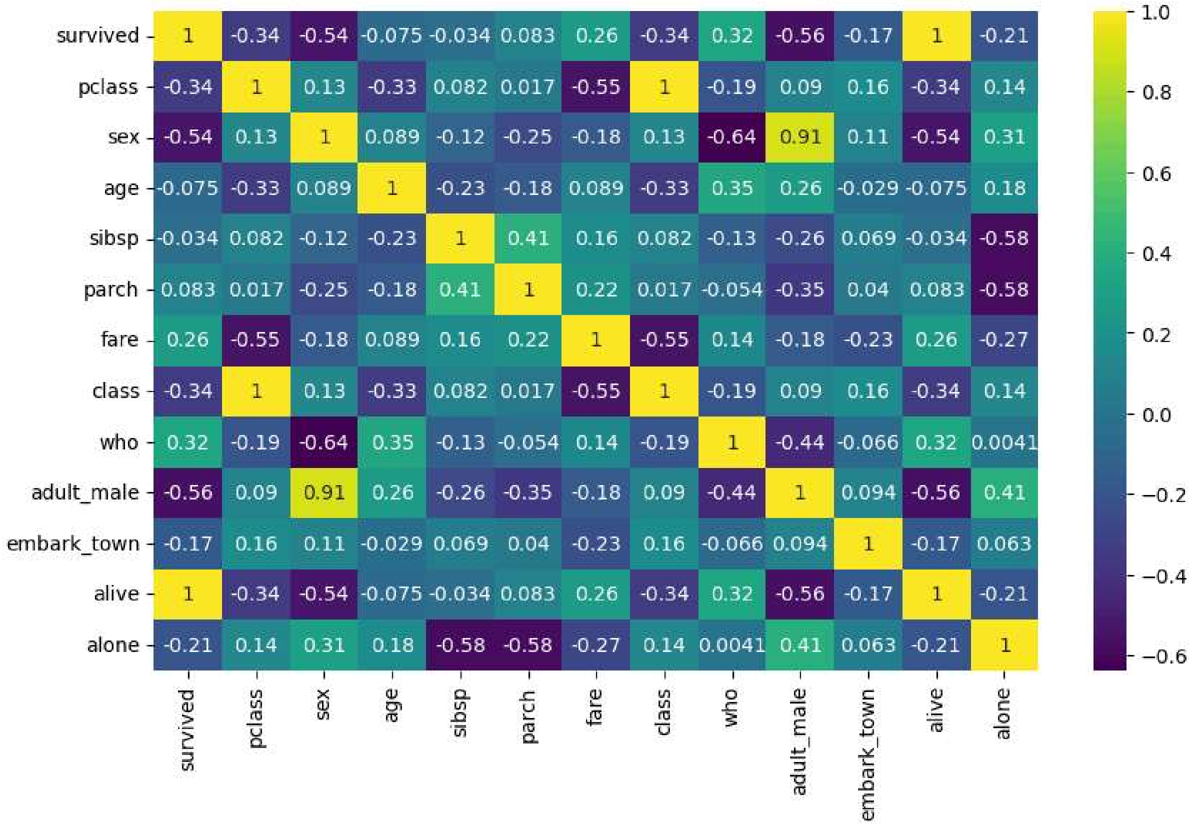
```
Age_Group ['Adult' 'Child' 'Mid_Age' 'Senior' 'Teen' nan]
```

In [36]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 889 entries, 0 to 890
Data columns (total 14 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   survived     889 non-null    int64  
 1   pclass       889 non-null    int64  
 2   sex          889 non-null    int32  
 3   age          889 non-null    float64
 4   sibsp        889 non-null    int64  
 5   parch        889 non-null    int64  
 6   fare         889 non-null    float64
 7   class        889 non-null    int32  
 8   who          889 non-null    int32  
 9   adult_male   889 non-null    bool   
 10  embark_town  889 non-null    int32  
 11  alive        889 non-null    int32  
 12  alone        889 non-null    bool   
 13  Age_Group    889 non-null    int32  
dtypes: bool(2), float64(2), int32(6), int64(4)
memory usage: 71.2 KB
```

In [21]:
```python
plt.figure(figsize=(10,6))
sns.heatmap(data=data.corr(), annot= True, cmap='viridis')
plt.show
```

Out[21]: <function matplotlib.pyplot.show(close=None, block=None)>



We can observe from the above plot the survival of an adult male is very low as compare to other passengers. As there is a signigicant negative correlation of 'adult_male' column with 'survived' column.

In [24]:
```python
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
```

In [38]:
```python
x= data.drop('survived', axis=1)
y= data['survived']
```

In [39]:
```python
print(x.shape)
print(y.shape)
```

```
(889, 13)
(889,)
```

In [40]:
```python
x_train, x_test, y_train, y_test= train_test_split(x,y, train_size= 0.75, random_state=4
```

In [41]:
```python
lr= LogisticRegression()
```

In [42]:
```
data.head()
```

Out[42]:

	survived	pclass	sex	age	sibsp	parch	fare	class	who	adult_male	embark_town	alive	alone	A
0	0	3	1	22.0	1	0	7.2500	2	1	True	2	0	False	
1	1	1	0	38.0	1	0	71.2833	0	2	False	0	1	False	
2	1	3	0	26.0	0	0	7.9250	2	2	False	2	1	True	
3	1	1	0	35.0	1	0	53.1000	0	2	False	2	1	False	
4	0	3	1	35.0	0	0	8.0500	2	1	True	2	0	True	

In [43]:
```
lr.fit(x_train, y_train)
```

```
C:\Users\hp\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:814: Converge
nceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html (https://scikit-learn.or
g/stable/modules/preprocessing.html)
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression (http
s://scikit-learn.org/stable/modules/linear_model.html#logistic-regression)
  n_iter_i = _check_optimize_result(
```

Out[43]:
```
LogisticRegression()
```

In [44]:
```
preds= lr.predict(x_test)
```

In [45]:
```
from sklearn.metrics import *
```

In [46]:
```
acc= accuracy_score(y_test, preds)
acc
```

Out[46]:
```
1.0
```