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

In [3]:
```python
df = pd.read_csv(r"C:\Users\Owner\Downloads\titanic.csv")
df.head()
```

Out[3]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town	alive	alone	Unnamed: 15
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	NaN	Southampton	no	False	NaN
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	C	Cherbourg	yes	False	NaN
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	NaN	Southampton	yes	True	NaN
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	C	Southampton	yes	False	NaN
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	NaN	Southampton	no	True	NaN

In [4]:
```python
df.shape
```

Out[4]:
```
(891, 16)
```

In [5]:
```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 16 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
 15  Unnamed: 15  0 non-null      float64
dtypes: bool(2), float64(3), int64(4), object(7)
memory usage: 99.3+ KB
```

In [6]:
```python
df.describe()
```

Out[6]:

	survived	pclass	age	sibsp	parch	fare	Unnamed: 15
count	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000	0.0
mean	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208	NaN
std	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429	NaN
min	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000	NaN
25%	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400	NaN
50%	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200	NaN
75%	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000	NaN
max	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200	NaN

In [7]:
```python
df.isnull().sum()
```

Out[7]:
```
survived         0
pclass           0
sex              0
age            177
sibsp            0
parch            0
fare             0
embarked         2
class            0
who              0
adult_male       0
deck           688
embark_town      2
alive            0
alone            0
Unnamed: 15    891
dtype: int64
```

In [8]:
```python
df.duplicated().sum()
```

Out[8]:
```
107
```

In [9]:
```python
titantic = df.isna().sum().sort_values(ascending=False)
titantic = titantic[titantic>107]
titantic
```

Out[9]:
```
Unnamed: 15    891
deck           688
age            177
dtype: int64
```

In [11]:
```python
# checking what % colunm contain missing values
df.isnull().sum().sort_values(ascending=False)*100/len(df)
```

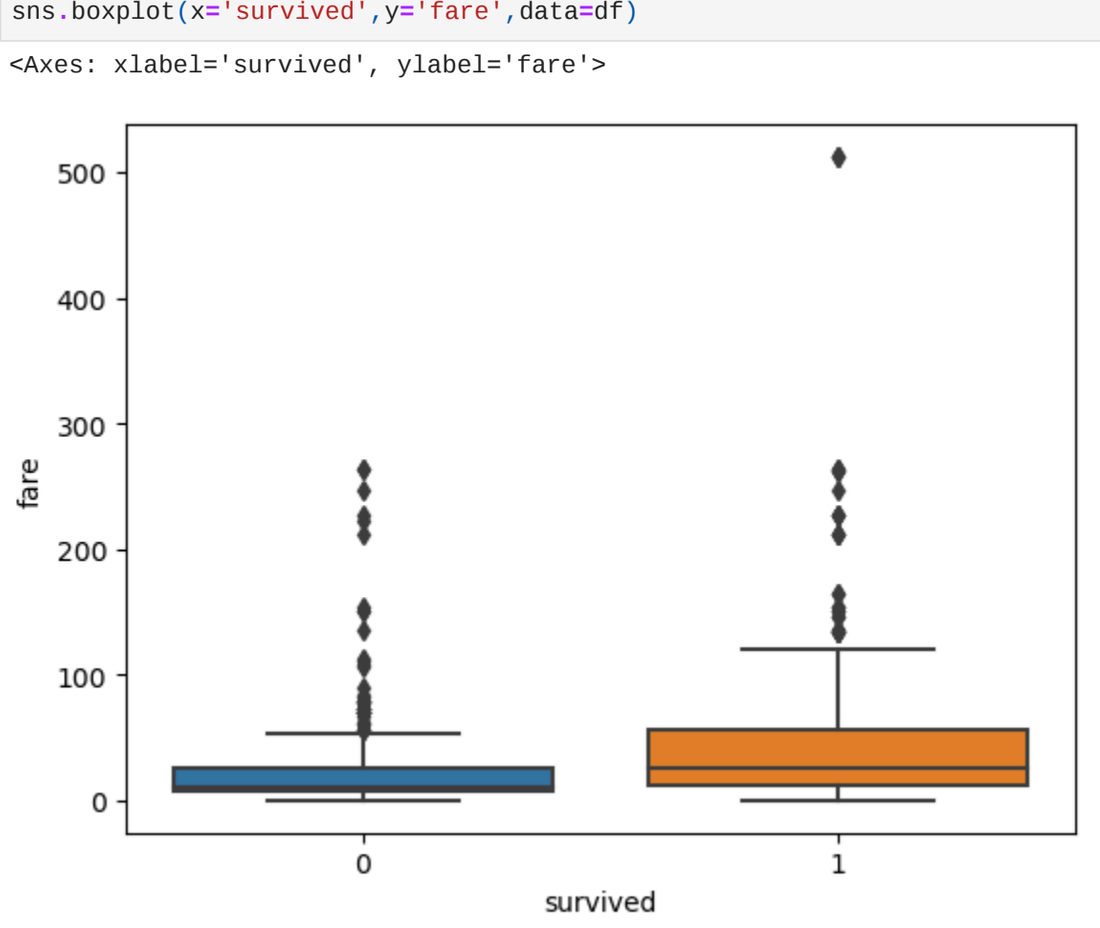
Out[11]:
```
Unnamed: 15    100.000000
deck            77.216611
age             19.865320
embarked         0.224467
embark_town      0.224467
survived         0.000000
pclass           0.000000
sex              0.000000
sibsp            0.000000
parch            0.000000
fare             0.000000
class            0.000000
who              0.000000
adult_male       0.000000
alive            0.000000
alone            0.000000
dtype: float64
```

In [14]:
```python
df.columns.values
```

Out[14]:
```
array(['survived', 'pclass', 'sex', 'age', 'sibsp', 'parch', 'fare',
       'embarked', 'class', 'who', 'adult_male', 'deck', 'embark_town',
       'alive', 'alone', 'Unnamed: 15'], dtype=object)
```

In [17]:
```python
sns.boxplot(x='survived',y='fare',data=df)
```

Out[17]:
```
<Axes: xlabel='survived', ylabel='fare'>
```



Passengers who paid over 300

In [18]:
```python
df[df['fare']>300]
```

Out[18]:

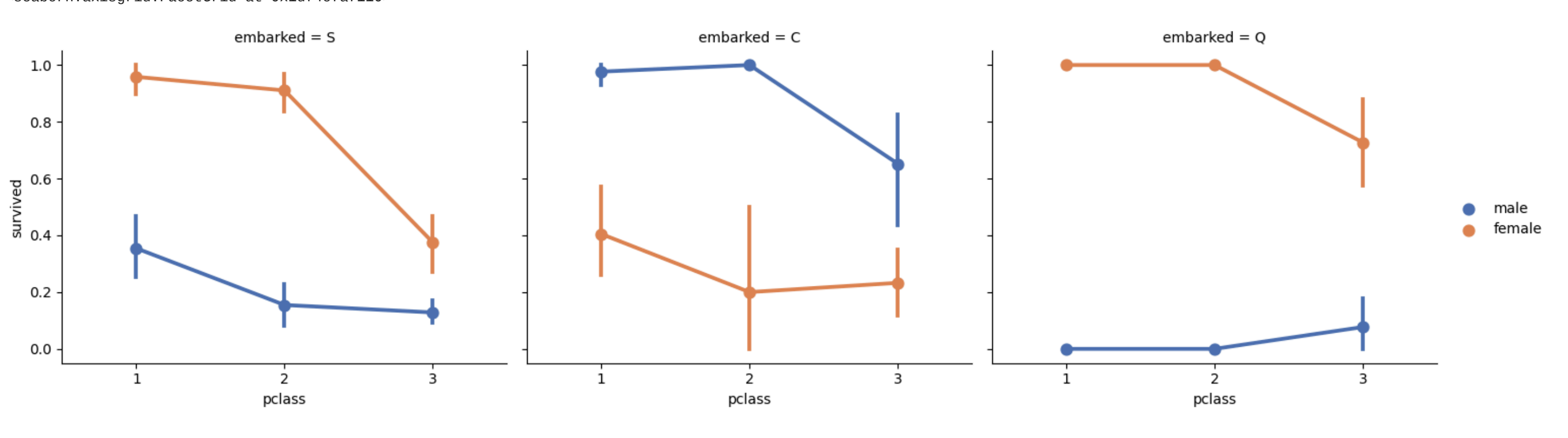
	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town	alive	alone	Unnamed: 15
258	1	1	female	35.0	0	0	512.3292	C	First	woman	False	NaN	Cherbourg	yes	True	NaN
679	1	1	male	36.0	0	1	512.3292	C	First	man	True	B	Cherbourg	yes	False	NaN
737	1	1	male	35.0	0	0	512.3292	C	First	man	True	B	Cherbourg	yes	True	NaN

Embarked, Pclass and Sex

In [21]:
```python
FacetGrid = sns.FacetGrid(df, col='embarked', height=4, aspect=1.2)
FacetGrid.map(sns.pointplot, 'pclass', 'survived', 'sex', ci=95.0, palette='deep', order=None, hue_order=None)
FacetGrid.add_legend()
```

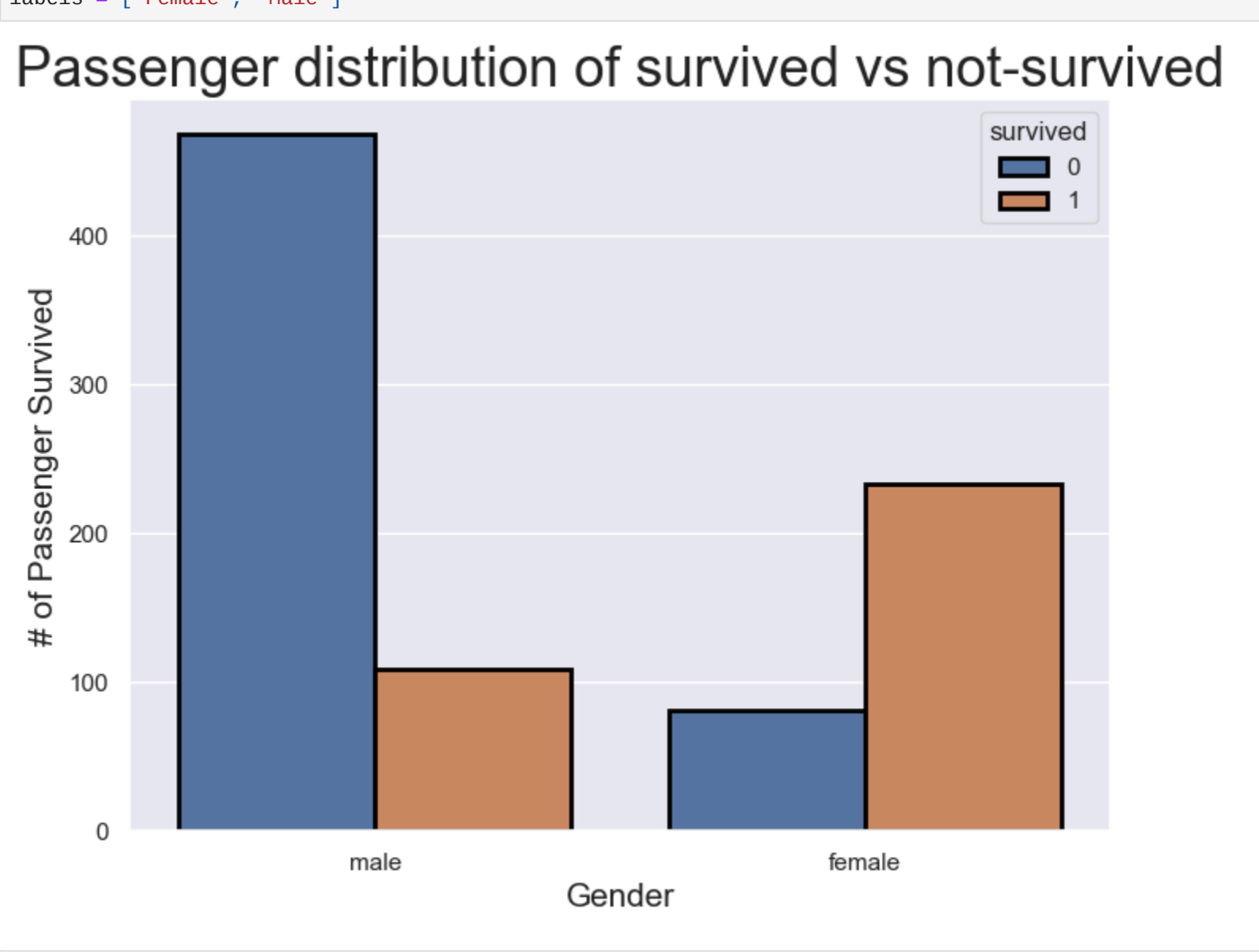
```
C:\Users\Owner\Desktop\a\Lib\site-packages\seaborn\axisgrid.py:848: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=('ci', 95.0)` for the same effect.

  func(*plot_args, **plot_kwargs)
C:\Users\Owner\Desktop\a\Lib\site-packages\seaborn\axisgrid.py:848: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=('ci', 95.0)` for the same effect.

  func(*plot_args, **plot_kwargs)
C:\Users\Owner\Desktop\a\Lib\site-packages\seaborn\axisgrid.py:848: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=('ci', 95.0)` for the same effect.

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

Out[21]:
```
<seaborn.axisgrid.FacetGrid at 0x1df4c7a7110>
```



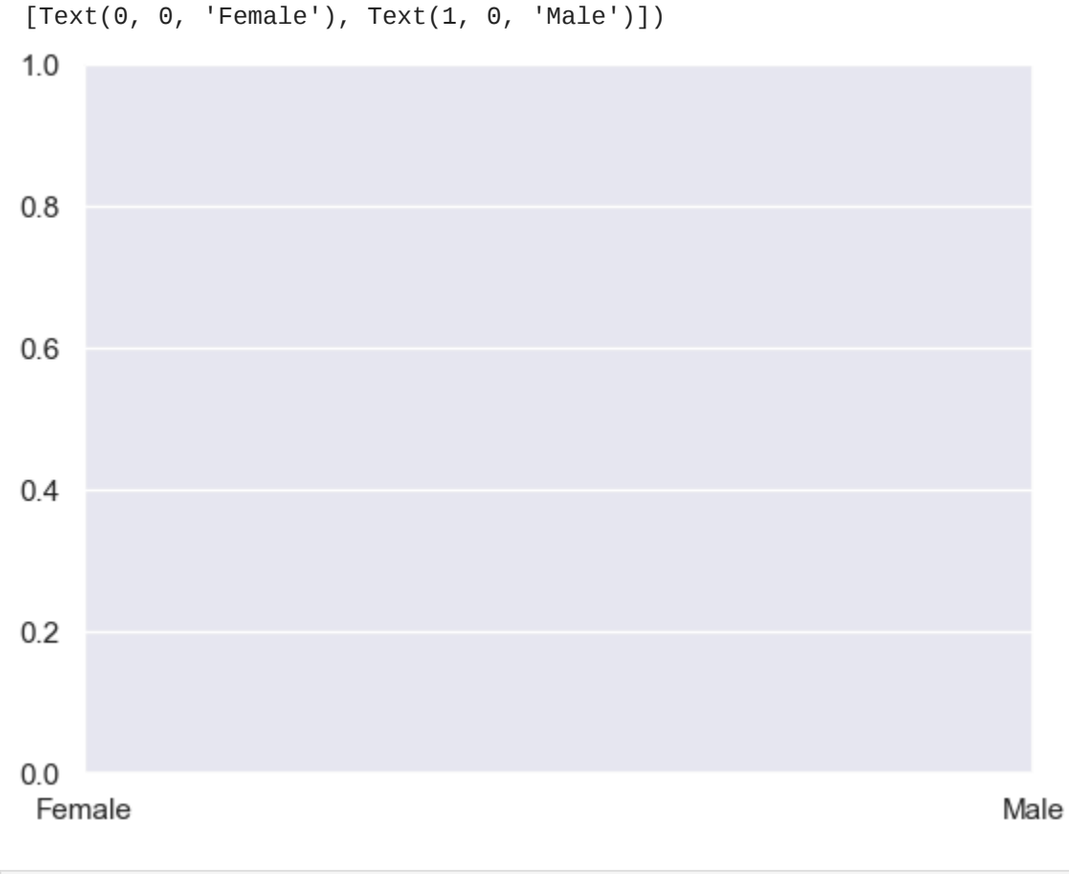
Distribution of Pclass and Survived

In [23]:
```python
sns.set(style='darkgrid')
plt.subplots(figsize = (8,6))
ax=sns.countplot(x='sex', data = df, hue='survived', edgecolor=(0,0,0), linewidth=2)

# Fixing title, xlabel and ylabel
plt.title('Passenger distribution of survived vs not-survived', fontsize=25)
plt.xlabel('Gender', fontsize=15)
plt.ylabel("# of Passenger Survived", fontsize = 15)
labels = ['Female', 'Male']
```



In [26]:
```python
# Fixing xticks.
plt.xticks(sorted(df.survived.unique()),labels)
```

Out[26]:
```
([<matplotlib.axis.XTick at 0x1df4cf14090>,
  <matplotlib.axis.XTick at 0x1df4cf7fcd0>],
 [Text(0, 0, 'Female'), Text(1, 0, 'Male')])
```



In [27]:
```python
df.groupby(['sex']).mean()
```

```
C:\Users\Owner\AppData\Local\Temp\ipykernel_14652\1056145912.py:1: FutureWarning: The default value of numeric_only in DataFrameGroupBy.mean is deprecated. In a future version, numeric_only will default to False. Either specify numeric_only or select only columns which should be valid for the function.
  df.groupby(['sex']).mean()
```

Out[27]:

sex	survived	pclass	age	sibsp	parch	fare	adult_male	alone	Unnamed: 15
female	0.742038	2.159236	27.915709	0.694268	0.649682	44.479818	0.000000	0.401274	NaN
male	0.188908	2.389948	30.726645	0.429809	0.235702	25.523893	0.930676	0.712305	NaN

Looking deeper into differences between females and males statistics

In [28]:
```python
df.groupby(['sex','pclass']).mean()
```

```
C:\Users\Owner\AppData\Local\Temp\ipykernel_14652\3662533120.py:1: FutureWarning: The default value of numeric_only in DataFrameGroupBy.mean is deprecated. In a future version, numeric_only will default to False. Either specify numeric_only or select only columns which should be valid for the function.
  df.groupby(['sex','pclass']).mean()
```

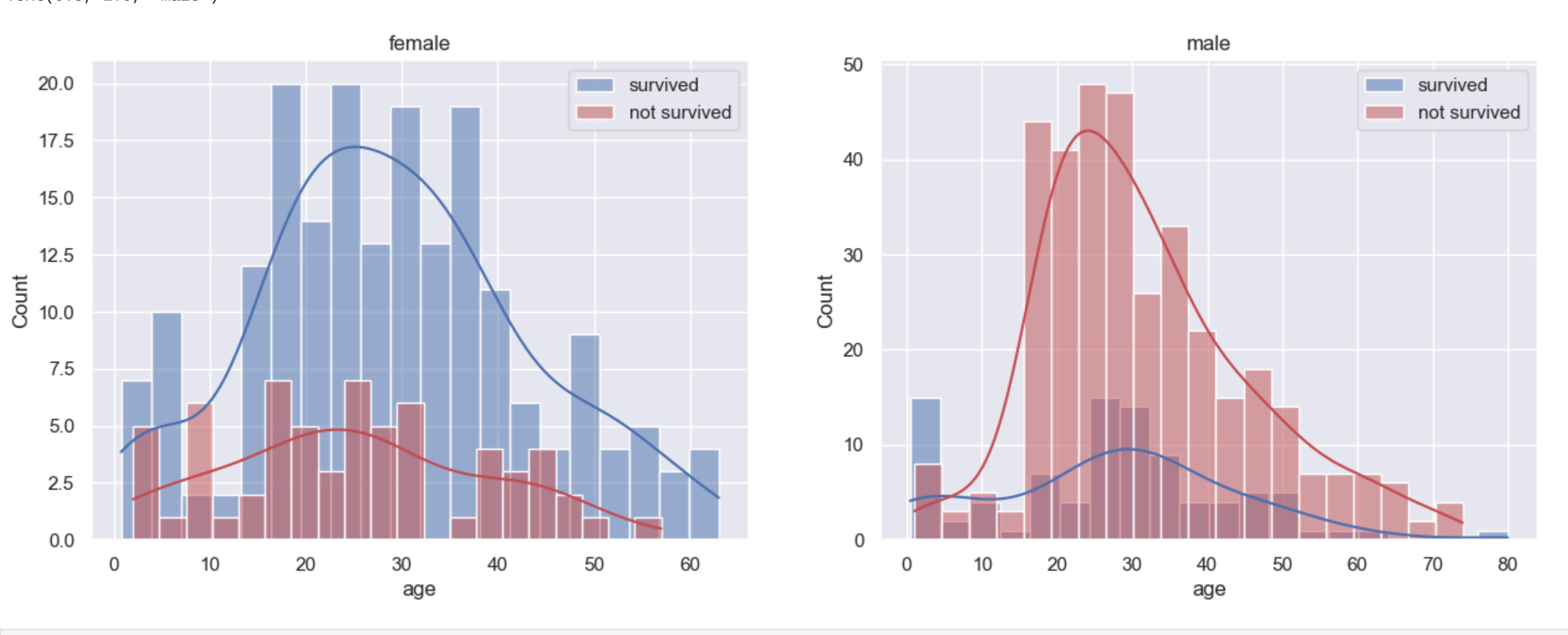
Out[28]:

sex	pclass	survived	age	sibsp	parch	fare	adult_male	alone	Unnamed: 15
female	1	0.968085	34.611765	0.553191	0.457447	106.125798	0.000000	0.361702	NaN
	2	0.921053	28.722973	0.486842	0.605263	21.970121	0.000000	0.421053	NaN
	3	0.500000	21.750000	0.895833	0.798611	16.118810	0.000000	0.416667	NaN
male	1	0.368852	41.281386	0.311475	0.278689	67.226127	0.975410	0.614754	NaN
	2	0.157407	30.740707	0.342593	0.222222	19.741782	0.916667	0.666667	NaN
	3	0.135447	26.507589	0.498559	0.224784	12.661633	0.919308	0.760807	NaN

Age and Sex distributions

In [29]:
```python
survived = 'survived'
not_survived = 'not survived'

fig, axes = plt.subplots(nrows=1, ncols=2, figsize=(15, 5))

women = df[df['sex']=='female']
men = df[df['sex']=='male']

# Plot Female Survived vs Not-Survived distribution
ax = sns.histplot(women[women['survived']==1].age.dropna(), bins=20, label = survived, ax = axes[0],color='b', kde=True)
ax = sns.histplot(women[women['survived']==0].age.dropna(), bins=20, label = not_survived, ax = axes[0],color='r', kde=True)
ax.legend()
ax.set_title('female')

# Plot Male Survived vs Not-Survived distribution
ax = sns.histplot(men[men['survived']==1].age.dropna(), bins=20, label = survived, ax = axes[1],color='b', kde=True)
ax = sns.histplot(men[men['survived']==0].age.dropna(), bins=20, label = not_survived, ax = axes[1],color='r', kde=True)
ax.legend()
ax.set_title('male')
```

Out[29]:
```
Text(0.5, 1.0, 'male')
```



In [30]:
```python
df[df['age']<18].groupby(['sex','pclass']).mean()
```

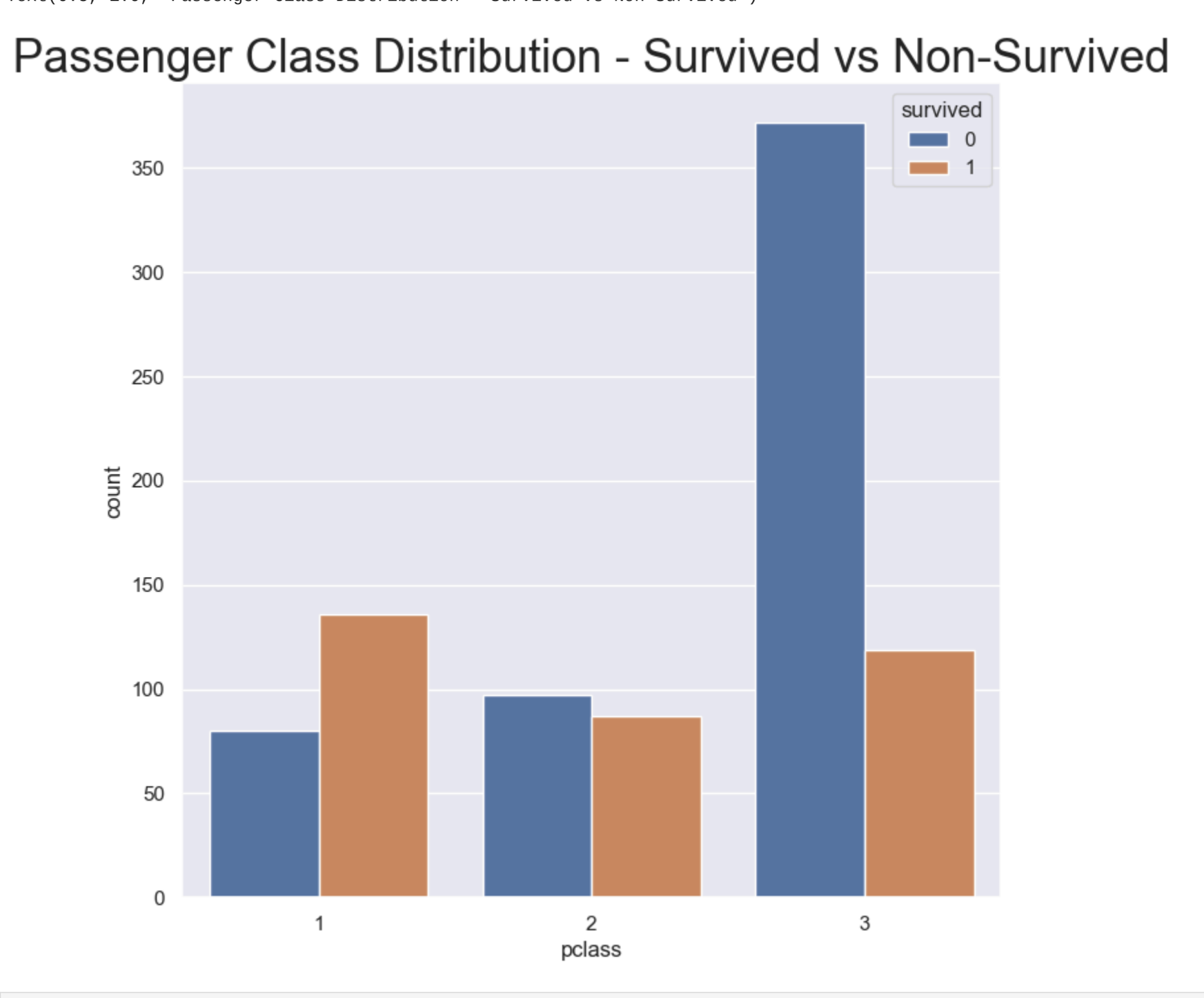
```
C:\Users\Owner\AppData\Local\Temp\ipykernel_14652\3275984015.py:1: FutureWarning: The default value of numeric_only in DataFrameGroupBy.mean is deprecated. In a future version, numeric_only will default to False. Either specify numeric_only or select only columns which should be valid for the function.
  df[df['age']<18].groupby(['sex','pclass']).mean()
```

Out[30]:

sex	pclass	survived	age	sibsp	parch	fare	adult_male	alone	Unnamed: 15
female	1	0.875000	14.125000	0.500000	0.875000	104.083337	0.000000	0.125000	NaN
	2	1.000000	8.333333	0.583333	1.083333	26.241667	0.000000	0.166667	NaN
	3	0.542857	8.428571	1.571429	1.057143	18.727977	0.000000	0.228571	NaN
male	1	1.000000	8.230000	0.500000	2.000000	116.072900	0.250000	0.000000	NaN
	2	0.818182	4.757273	0.727273	1.000000	25.659473	0.181818	0.181818	NaN
	3	0.232558	9.963256	2.069767	1.000000	22.752523	0.348837	0.232558	NaN

In [31]:
```python
plt.subplots(figsize = (8,8))
ax=sns.countplot(x='pclass',hue='survived',data=df)
plt.title("Passenger Class Distribution - Survived vs Non-Survived", fontsize = 25)
```

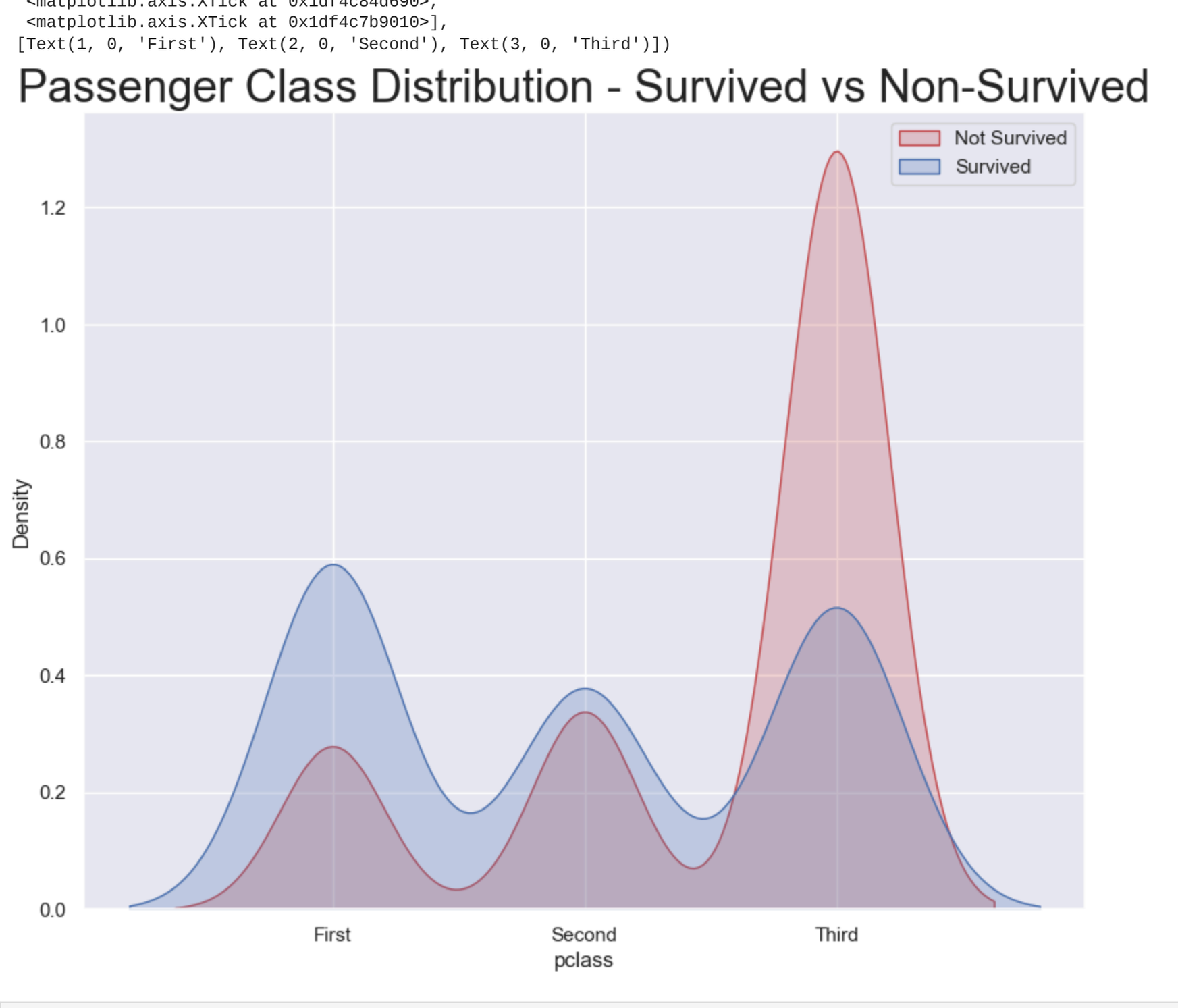
Out[31]:
```
Text(0.5, 1.0, 'Passenger Class Distribution - Survived vs Non-Survived')
```



In [33]:
```python
plt.subplots(figsize=(10,8))
ax=sns.kdeplot(df.loc[(df['survived'] == 0),'pclass'],shade=True,color='r',label='Not Survived')
ax.legend()
ax=sns.kdeplot(df.loc[(df['survived'] == 1),'pclass'],shade=True,color='b',label='Survived')
ax.legend()

plt.title("Passenger Class Distribution - Survived vs Non-Survived", fontsize = 25)
labels = ['First', 'Second', 'Third']
plt.xticks(sorted(df.pclass.unique()),labels)
```

```
C:\Users\Owner\AppData\Local\Temp\ipykernel_14652\3303278358.py:2: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  ax=sns.kdeplot(df.loc[(df['survived'] == 0),'pclass'],shade=True,color='r',label='Not Survived')
C:\Users\Owner\AppData\Local\Temp\ipykernel_14652\3303278358.py:4: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  ax=sns.kdeplot(df.loc[(df['survived'] == 1),'pclass'],shade=True,color='b',label='Survived')
```

Out[33]:
```
([<matplotlib.axis.XTick at 0x1df4c71b990>,
  <matplotlib.axis.XTick at 0x1df4c84d690>,
  <matplotlib.axis.XTick at 0x1df4c7b9010>],
 [Text(1, 0, 'First'), Text(2, 0, 'Second'), Text(3, 0, 'Third')])
```



In [34]:
```python
plt.subplots(figsize = (8,6))
sns.barplot(x='pclass', y='survived', data=df)
plt.title("Passenger Class Distribution - Survived Passengers", fontsize = 25)
```

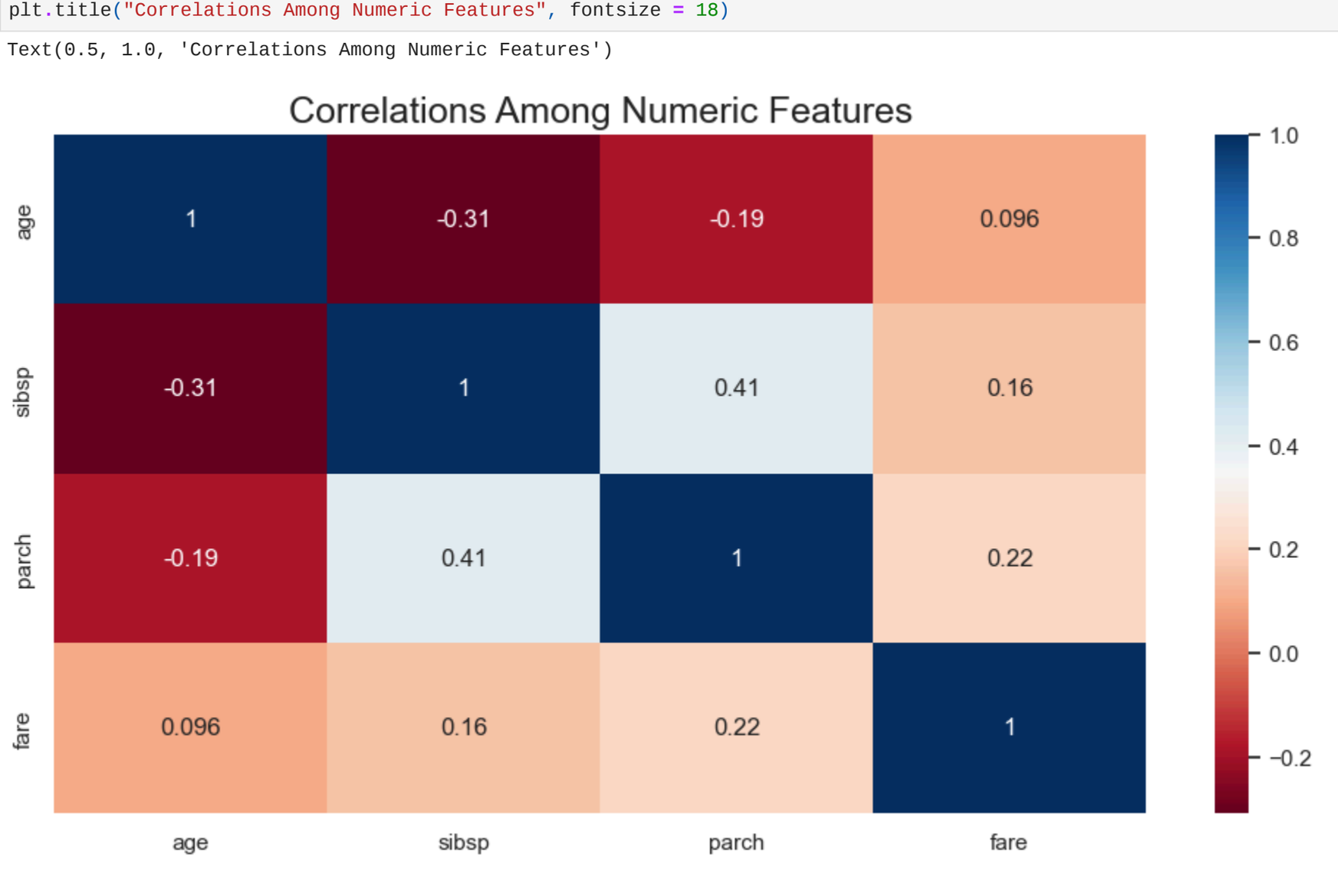
Out[34]:
```
Text(0.5, 1.0, 'Passenger Class Distribution - Survived Passengers')
```

Correlation Matrix and Heatmap

In [37]:
```python
# Look at numeric and categorical values separately
df_num = df[['age','sibsp','parch','fare']]
df_cat = df[['survived','pclass','sex','embarked']]
```

In [38]:
```python
plt.subplots(figsize = (12,6))
sns.heatmap(df_num.corr(), annot=True,cmap="RdBu")
plt.title("Correlations Among Numeric Features", fontsize = 18)
```

Out[38]:
```
Text(0.5, 1.0, 'Correlations Among Numeric Features')
```

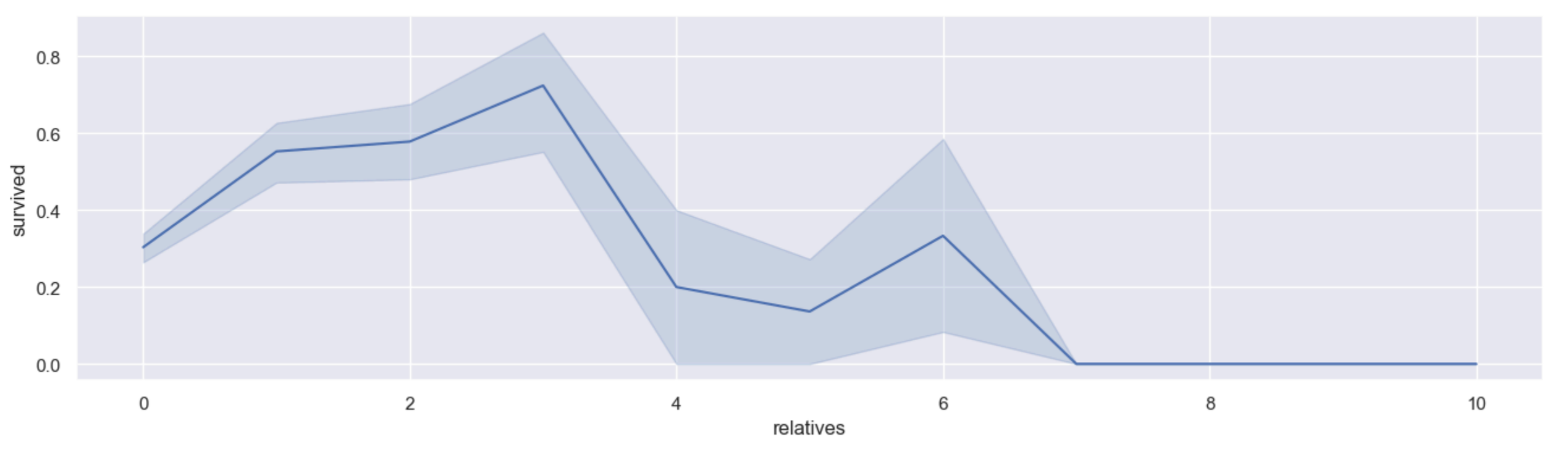


Combining SibSp and Parch

In [39]:
```python
data = [df]
for dataset in data:
    dataset['relatives'] = dataset['sibsp'] + dataset['parch']
    dataset.loc[dataset['relatives'] > 0, 'not_alone'] = 0
    dataset.loc[dataset['relatives'] == 0, 'not_alone'] = 1
    dataset['not_alone'] = dataset['not_alone'].astype(int)
df['not_alone'].value_counts()
```

Out[39]:
```
1    537
0    354
Name: not_alone, dtype: int64
```

In [40]:
```python
plt.subplots(figsize = (16,4))
ax = sns.lineplot(x='relatives',y='survived', data=df)
```



Missing Data

In [46]:
```python
data = [df]

for dataset in data:
    mean = df["age"].mean()
    std = df["age"].std()
    is_null = dataset["age"].isnull().sum()

    # Compute random numbers between the mean, std and is_null
    rand_age = np.random.randint(mean - std, mean + std, size = is_null)

    # Fill NaN values in Age column with random values generated
    age_slice = dataset["age"].copy()
    age_slice[np.isnan(age_slice)] = rand_age
    dataset["age"] = age_slice
    dataset["age"] = df["age"].astype(int)
```

In [47]:
```python
df["age"].isnull().sum()
```

Out[47]:
```
0
```

In [48]:
```python
df['embarked'].describe()
```

Out[48]:
```
count     889
unique      3
top         S
freq      644
Name: embarked, dtype: object
```

In [49]:
```python
common_value = 'S'
data = [df]

for dataset in data:
    dataset['embarked'] = dataset['embarked'].fillna(common_value)
```

In [50]:
```python
df['embarked'].isnull().sum()
```

Out[50]:
```
0
```

In [51]:
```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 18 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          891 non-null    int32
 4   sibsp        891 non-null    int64
 5   parch        891 non-null    int64
 6   fare         891 non-null    float64
 7   embarked     891 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
 15  Unnamed: 15  0 non-null      float64
 16  relatives    891 non-null    int64
 17  not_alone    891 non-null    int32
dtypes: bool(2), float64(2), int32(2), int64(5), object(7)
memory usage: 106.3+ KB
```

Sex

Convert feature 'Sex' into numeric values

male = 0 female = 1

In [52]:
```python
genders = {"male": 0, "female": 1}
data = [df]

for dataset in data:
    dataset['sex'] = dataset['sex'].map(genders)
```

In [53]:
```python
df.head()
```

Out[53]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town	alive	alone	Unnamed: 15	relatives	not_alone
0	0	3	0	22	1	0	7.2500	S	Third	man	True	NaN	Southampton	no	False	NaN	1	0
1	1	1	1	38	1	0	71.2833	C	First	woman	False	C	Cherbourg	yes	False	NaN	1	0
2	1	3	1	26	0	0	7.9250	S	Third	woman	False	NaN	Southampton	yes	True	NaN	0	1
3	1	1	1	35	1	0	53.1000	S	First	woman	False	C	Southampton	yes	False	NaN	1	0
4	0	3	0	35	0	0	8.0500	S	Third	man	True	NaN	Southampton	no	True	NaN	0	1

Convert 'Embarked' feature into numeric values

In [54]:
```python
ports = {"S": 0, "C": 1, "Q": 2}
data = [df]

for dataset in data:
    dataset['embarked'] = dataset['embarked'].map(ports)
```

In [55]:
```python
df.head()
```

Out[55]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town	alive	alone	Unnamed: 15	relatives	not_alone
0	0	3	0	22	1	0	7.2500	0	Third	man	True	NaN	Southampton	no	False	NaN	1	0
1	1	1	1	38	1	0	71.2833	1	First	woman	False	C	Cherbourg	yes	False	NaN	1	0
2	1	3	1	26	0	0	7.9250	0	Third	woman	False	NaN	Southampton	yes	True	NaN	0	1
3	1	1	1	35	1	0	53.1000	0	First	woman	False	C	Southampton	yes	False	NaN	1	0
4	0	3	0	35	0	0	8.0500	0	Third	man	True	NaN	Southampton	no	True	NaN	0	1

In [57]:
```python
data = [df]
for dataset in data:
    dataset['age'] = dataset['age'].astype(int)
    dataset.loc[ dataset['age'] <= 11, 'age'] = 0
    dataset.loc[(dataset['age'] > 11) & (dataset['age'] <= 18), 'age'] = 1
    dataset.loc[(dataset['age'] > 18) & (dataset['age'] <= 22), 'age'] = 2
    dataset.loc[(dataset['age'] > 22) & (dataset['age'] <= 27), 'age'] = 3
    dataset.loc[(dataset['age'] > 27) & (dataset['age'] <= 33), 'age'] = 4
    dataset.loc[(dataset['age'] > 33) & (dataset['age'] <= 40), 'age'] = 5
    dataset.loc[(dataset['age'] > 40) & (dataset['age'] <= 66), 'age'] = 6
    dataset.loc[ dataset['age'] > 66, 'age'] = 6
```

In [58]:
```python
# Checking the distribution
df['age'].value_counts()
```

Out[58]:
```
6    173
0    165
4    150
5    148
3    131
2    124
Name: age, dtype: int64
```

In [59]:
```python
df.head()
```

Out[59]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town	alive	alone	Unnamed: 15	relatives	not_alone
0	0	3	0	2	1	0	7.2500	0	Third	man	True	NaN	Southampton	no	False	NaN	1	0
1	1	1	1	5	1	0	71.2833	1	First	woman	False	C	Cherbourg	yes	False	NaN	1	0
2	1	3	1	3	0	0	7.9250	0	Third	woman	False	NaN	Southampton	yes	True	NaN	0	1
3	1	1	1	5	1	0	53.1000	0	First	woman	False	C	Southampton	yes	False	NaN	1	0
4	0	3	0	5	0	0	8.0500	0	Third	man	True	NaN	Southampton	no	True	NaN	0	1

In [60]:
```python
pd.qcut(df['fare'], q=6)
```

Out[60]:
```
0       (-0.001, 7.775]
1      (52.369, 512.329]
2        (7.775, 8.662]
3      (52.369, 512.329]
4        (7.775, 8.662]
             ...
886     (8.662, 14.454]
887       (26.0, 52.369]
888      (14.454, 26.0]
889       (26.0, 52.369]
890     (-0.001, 7.775]
Name: fare, Length: 891, dtype: category
Categories (6, interval[float64, right]): [(-0.001, 7.775] < (7.775, 8.662] < (8.662, 14.454] < (14.454, 26.0] < (26.0, 52.369] < (52.369, 512.329]]
```

In [61]:
```python
data = [df]

for dataset in data:
    dataset.loc[ dataset['fare'] <= 7, 'fare'] = 0
    dataset.loc[(dataset['fare'] > 7) & (dataset['fare'] <= 8), 'fare'] = 1
    dataset.loc[(dataset['fare'] > 8) & (dataset['fare'] <= 14), 'fare']   = 2
    dataset.loc[(dataset['fare'] > 14) & (dataset['fare'] <= 26), 'fare']   = 3
    dataset.loc[(dataset['fare'] > 26) & (dataset['fare'] <= 52), 'fare']   = 4
    dataset.loc[dataset['fare'] > 52, 'fare'] = 5
    dataset['fare'] = dataset['fare'].astype(int)
```

In [62]:
```python
df.head()
```

Out[62]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town	alive	alone	Unnamed: 15	relatives	not_alone
0	0	3	0	2	1	0	1	0	Third	man	True	NaN	Southampton	no	False	NaN	1	0
1	1	1	1	5	1	0	5	1	First	woman	False	C	Cherbourg	yes	False	NaN	1	0
2	1	3	1	3	0	0	1	0	Third	woman	False	NaN	Southampton	yes	True	NaN	0	1
3	1	1	1	5	1	0	5	0	First	woman	False	C	Southampton	yes	False	NaN	1	0
4	0	3	0	5	0	0	2	0	Third	man	True	NaN	Southampton	no	True	NaN	0	1