```
#importing libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
#load dataset
df=pd.read_csv("/content/train (9).csv")
df
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	E
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	
...	...	...	...	...	...	...	...	...	...	...	...	
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	

891 rows × 12 columns

```
df.isnull().sum()
```

	0
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

```
#filling age column with median
df["Age"].fillna(df["Age"].median(),inplace=True)

#filling Embarked column with mode
df["Embarked"].fillna(df["Embarked"].mode()[0],inplace=True)

#drop cabin column
df.drop("Cabin",axis=1,inplace=True)

df
```

```
/tmp/ipython-input-60-3093640927.py:2: FutureWarning: A value is trying to be set on a copy of
The behavior will change in pandas 3.0. This inplace method will never work because the interm

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: valu


  df["Age"].fillna(df["Age"].median(),inplace=True)
/tmp/ipython-input-60-3093640927.py:5: FutureWarning: A value is trying to be set on a copy of
The behavior will change in pandas 3.0. This inplace method will never work because the interm

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: valu


  df["Embarked"].fillna(df["Embarked"].mode()[0],inplace=True)
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	S
...	...	...	...	...	...	...	...	...	...	...	..
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	28.0	1	2	W./C. 6607	23.4500	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	C

891 rows × 11 columns

```
#Checking for null values
df.isnull().sum()
```

	0
PassengerId	0
Survived	0
Pclass	0
Name	0
Sex	0
Age	0
SibSp	0
Parch	0
Ticket	0
Fare	0
Embarked	0

dtype: int64

```
#Exploratory data analysis

df.describe()
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	891.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.361582	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	13.019697	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	22.000000	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	35.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 11 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          891 non-null    float64
 6   SibSp        891 non-null    int64
```

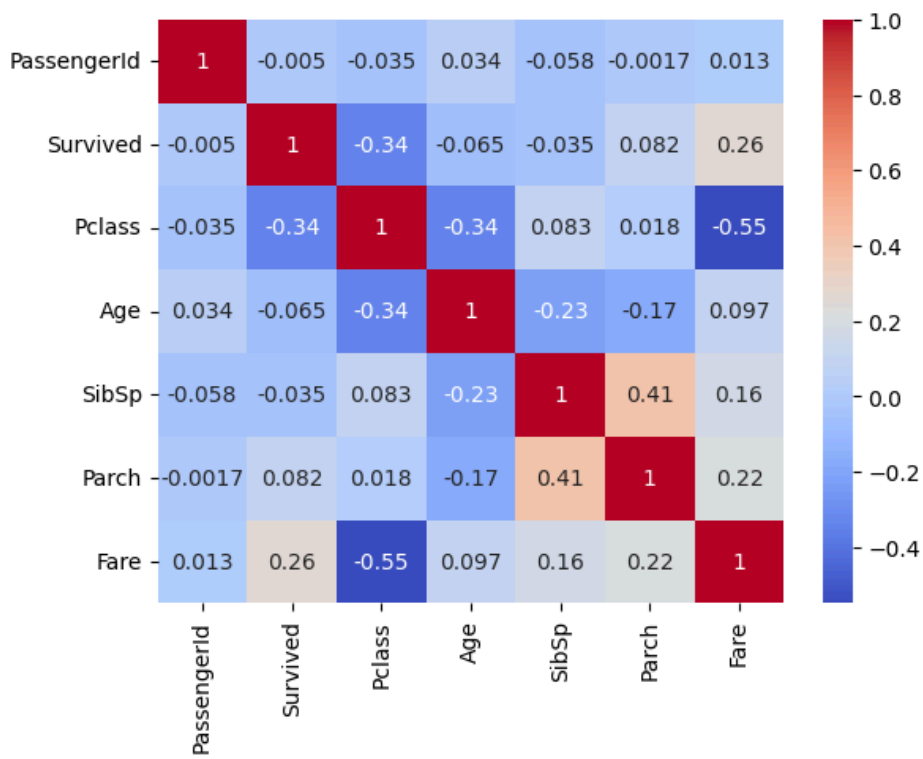
```
 7   Parch        891 non-null    int64
 8   Ticket       891 non-null    object
 9   Fare         891 non-null    float64
 10  Embarked     891 non-null    object
dtypes: float64(2), int64(5), object(4)
memory usage: 76.7+ KB
```

```
df.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	S
				Futrelle, Mrs. Jacques							

```
# Correlation heatmap for numeric features
sns.heatmap(df.corr(numeric_only=True), annot=True, cmap='coolwarm')
```

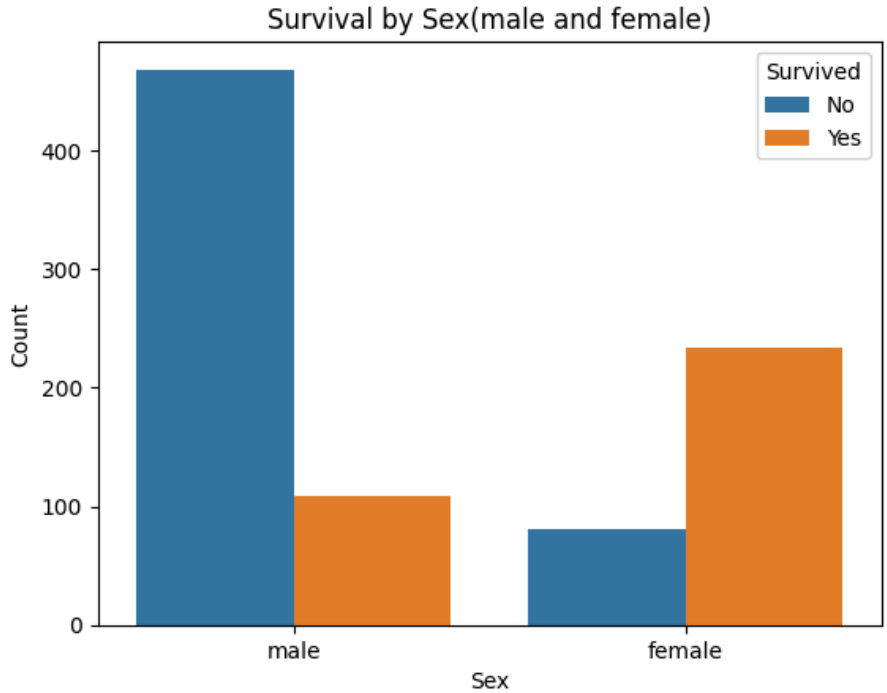
```
<Axes: >
```




```
# Bar plot of survival by sex
sns.countplot(x='Sex', hue='Survived', data=df)
#title for graph
plt.title('Survival by Sex(male and female)')
plt.legend(title='Survived', labels=['No', 'Yes'])
```

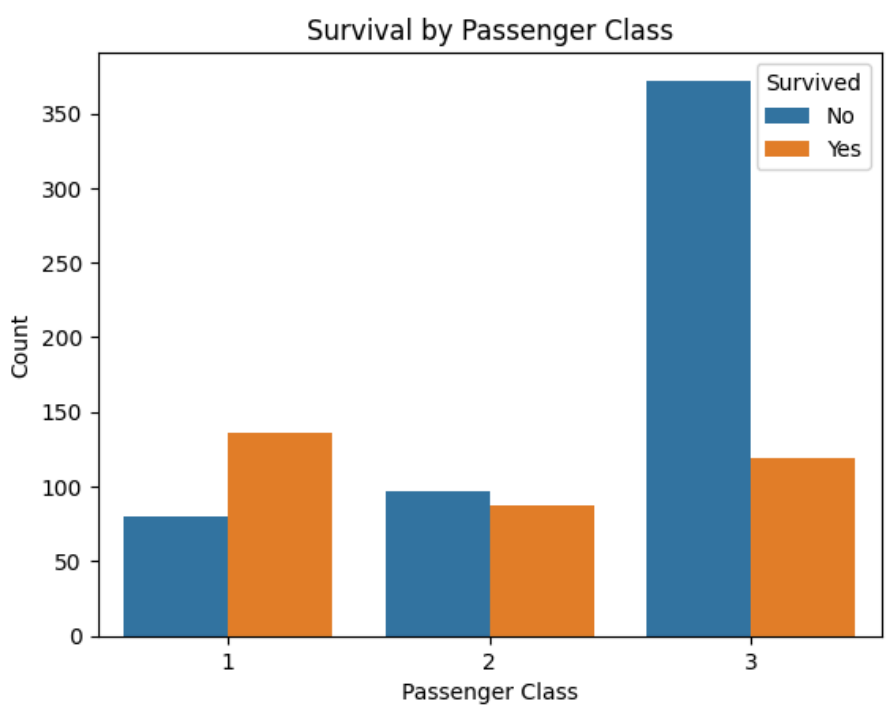
```
plt.xlabel('Sex')
plt.ylabel('Count')
```

```
Text(0, 0.5, 'Count')
```

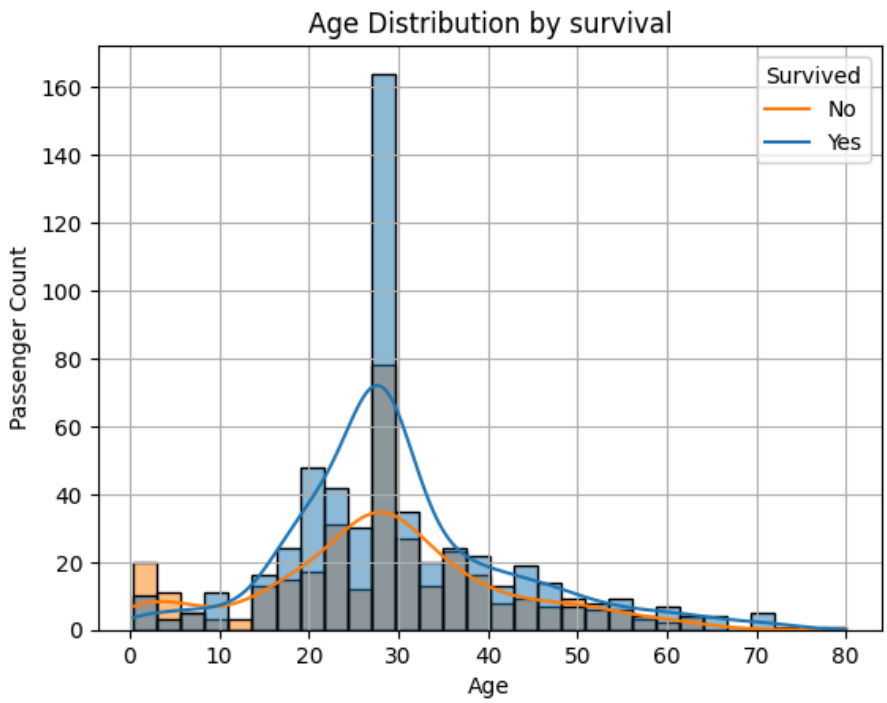



```
# Bar plot of survival by passenger class
sns.countplot(x='Pclass', hue='Survived', data=df)
plt.title('Survival by Passenger Class')
plt.legend(title='Survived', labels=['No', 'Yes'])
plt.xlabel('Passenger Class')
plt.ylabel('Count')
```

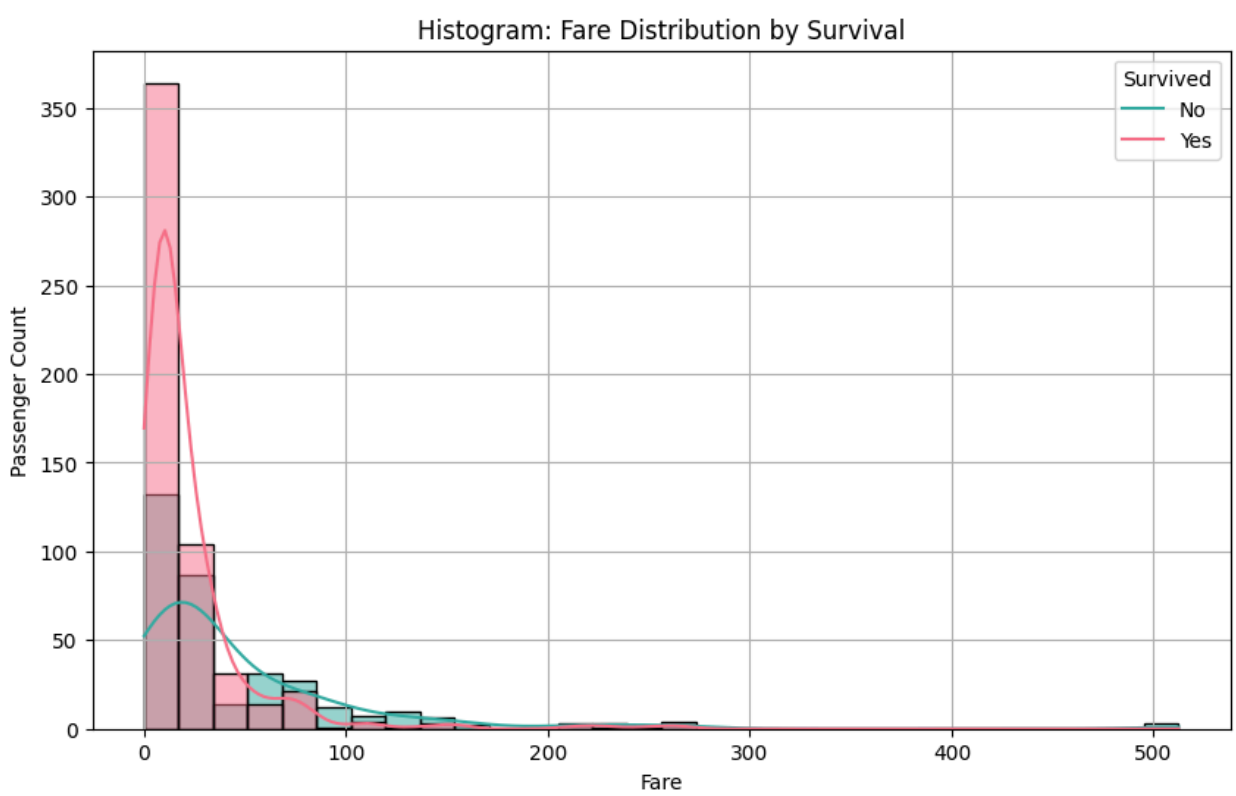
```
Text(0, 0.5, 'Count')
```



```
# Age distribution by survival
sns.histplot(data=df, x='Age', hue='Survived', bins=30, kde=True)
plt.title("Age Distribution by survival")
plt.xlabel("Age")
plt.ylabel("Passenger Count")
plt.legend(title='Survived', labels=['No', 'Yes'])
plt.grid(True)
plt.show()
```




```
# Histogram of Fare by Survival
plt.figure(figsize=(10, 6))
sns.histplot(data=df, x='Fare', hue='Survived', bins=30, kde=True, palette='husl')
plt.title("Histogram: Fare Distribution by Survival")
plt.xlabel("Fare")
plt.ylabel("Passenger Count")
plt.legend(title='Survived', labels=['No', 'Yes'])
plt.grid(True)
plt.show()
```

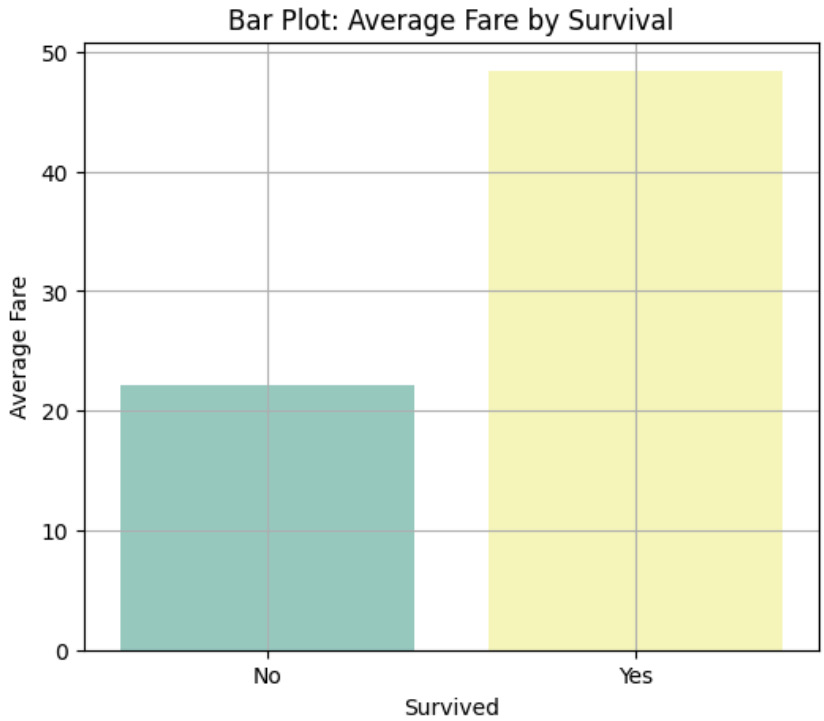



```
# Bar plot of average Fare by Survival
plt.figure(figsize=(6, 5))
avg_fares = df.groupby('Survived')['Fare'].mean().reset_index()
sns.barplot(data=avg_fares, x='Survived', y='Fare', palette='Set3')
plt.title("Bar Plot: Average Fare by Survival")
plt.xlabel("Survived")
plt.ylabel("Average Fare")
plt.xticks([0, 1], ['No', 'Yes'])
plt.grid(True)
plt.show()
```

```
/tmp/ipython-input-96-1180452035.py:4: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign

  sns.barplot(data=avg_fares, x='Survived', y='Fare', palette='Set3')
```




```
# 3. Age Distribution by Survival
sns.kdeplot(data=df[df['Survived'] == 0], x='Age', label='Did not Survive', shade=True)
sns.kdeplot(data=df[df['Survived'] == 1], x='Age', label='Survived', shade=True)
plt.title("Age Distribution by Survival")
plt.xlabel("Age")
plt.ylabel("Density")
plt.legend()
```

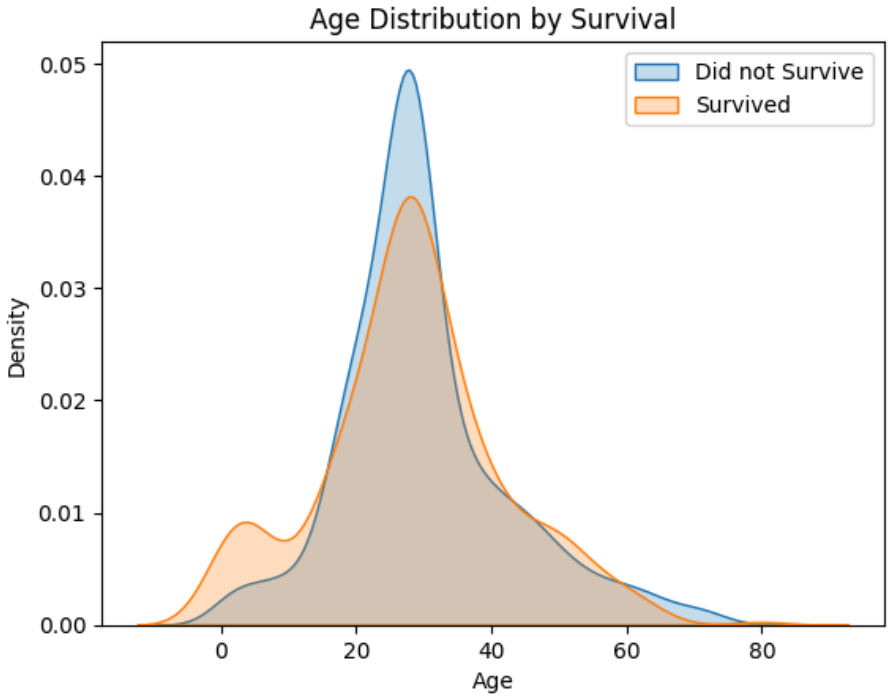
```
/tmp/ipython-input-82-2091817714.py:2: FutureWarning:

`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.

  sns.kdeplot(data=df[df['Survived'] == 0], x='Age', label='Did not Survive', shade=True)
/tmp/ipython-input-82-2091817714.py:3: 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.

  sns.kdeplot(data=df[df['Survived'] == 1], x='Age', label='Survived', shade=True)
<matplotlib.legend.Legend at 0x7a0b12a9edd0>
```

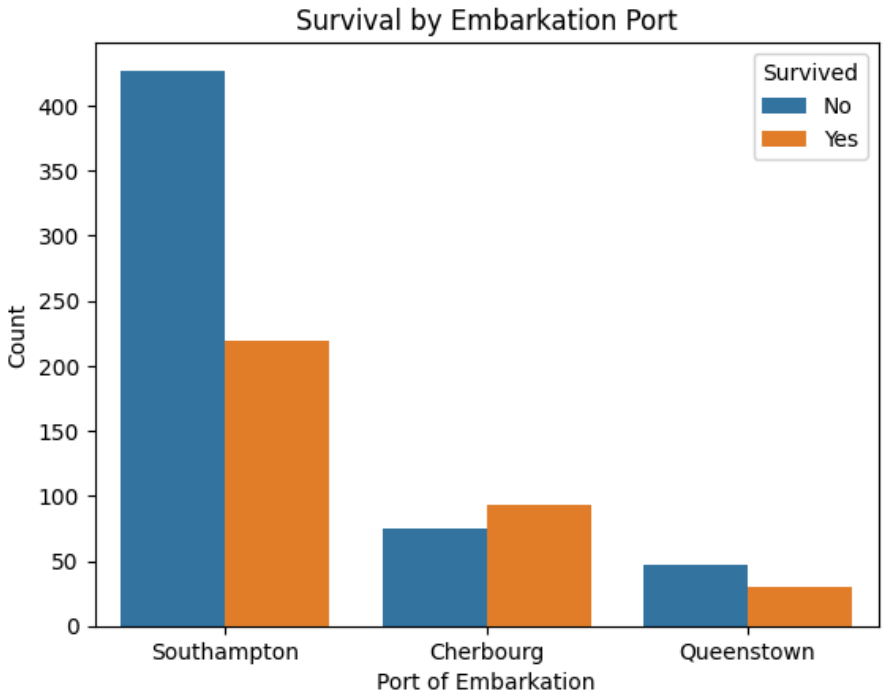



```
# Replace embarkation codes with full names
df['Embarked'] = df['Embarked'].replace({
    'S': 'Southampton',
    'C': 'Cherbourg',
    'Q': 'Queenstown'
```

```
})
```

```
# 5. Embarked Port vs Survival
sns.countplot(data=df, x='Embarked', hue='Survived')
plt.title("Survival by Embarkation Port")
plt.xlabel("Port of Embarkation")
plt.ylabel("Count")
plt.legend(title='Survived', labels=['No', 'Yes'])
```

```
<matplotlib.legend.Legend at 0x7a0b12a4ca90>
```




```
# 1. Survival by Gender
plt.subplot(3, 2, 1)
sns.countplot(data=df, x='Sex', hue='Survived')
plt.title("Survival by Gender")
plt.xlabel("Gender")
plt.ylabel("Count")
plt.legend(title='Survived', labels=['No', 'Yes'])

# 2. Survival by Passenger Class
plt.subplot(3, 2, 2)
sns.countplot(data=df, x='Pclass', hue='Survived')
plt.title("Survival by Passenger Class")
plt.xlabel("Passenger Class")
plt.ylabel("Count")
plt.legend(title='Survived', labels=['No', 'Yes'])

# 3. Age Distribution by Survival
plt.subplot(3, 2, 3)
sns.kdeplot(data=df[df['Survived'] == 0], x='Age', label='Did not Survive', shade=True
sns.kdeplot(data=df[df['Survived'] == 1], x='Age', label='Survived', shade=True)
plt.title("Age Distribution by Survival")
plt.xlabel("Age")
plt.ylabel("Density")
plt.legend()

# 4. Fare Distribution by Survival
```