Data Visualization I:

1. Use the inbuilt dataset 'titanic'. The dataset contains 891 rows and contains information about the passengers who boarded the unfortunate Titanic ship. Use the Seaborn library to see if we can find any patterns in the data.
2. Write a code to check how the price of the ticket (column name: 'fare') for each passenger is distributed by plotting a histogram.

```
pip install seaborn
```

```
Requirement already satisfied: seaborn in c:\users\rutuja habib\appdata\local\programs\python\python313\lib\site-packages (0.13.2)
Requirement already satisfied: numpy!=1.24.0,>=1.20 in c:\users\rutuja habib\appdata\local\programs\python\python313\lib\site-packag
Requirement already satisfied: pandas>=1.2 in c:\users\rutuja habib\appdata\local\programs\python\python313\lib\site-packages (from
Requirement already satisfied: matplotlib!=3.6.1,>=3.4 in c:\users\rutuja habib\appdata\local\programs\python\python313\lib\site-pac
Requirement already satisfied: contourpy>=1.0.1 in c:\users\rutuja habib\appdata\local\programs\python\python313\lib\site-packages
Requirement already satisfied: cycler>=0.10 in c:\users\rutuja habib\appdata\local\programs\python\python313\lib\site-packages (from
Requirement already satisfied: fonttools>=4.22.0 in c:\users\rutuja habib\appdata\local\programs\python\python313\lib\site-packages
Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\rutuja habib\appdata\local\programs\python\python313\lib\site-packages
Requirement already satisfied: packaging>=20.0 in c:\users\rutuja habib\appdata\local\programs\python\python313\lib\site-packages (f
Requirement already satisfied: pillow>=8 in c:\users\rutuja habib\appdata\local\programs\python\python313\lib\site-packages (from ma
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\rutuja habib\appdata\local\programs\python\python313\lib\site-packages
Requirement already satisfied: python-dateutil>=2.7 in c:\users\rutuja habib\appdata\local\programs\python\python313\lib\site-packag
Requirement already satisfied: pytz>=2020.1 in c:\users\rutuja habib\appdata\local\programs\python\python313\lib\site-packages (from
Requirement already satisfied: tzdata>=2022.7 in c:\users\rutuja habib\appdata\local\programs\python\python313\lib\site-packages (fr
Requirement already satisfied: six>=1.5 in c:\users\rutuja habib\appdata\local\programs\python\python313\lib\site-packages (from pyt
Note: you may need to restart the kernel to use updated packages.
```

+ Code + Text

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

```
df = sns.load_dataset('titanic')
df.head()
```

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

```
df['age'] = df['age'].fillna(df['age'].median())
df['embarked'] = df['embarked'].fillna(df['embarked'].mode()[0])
```

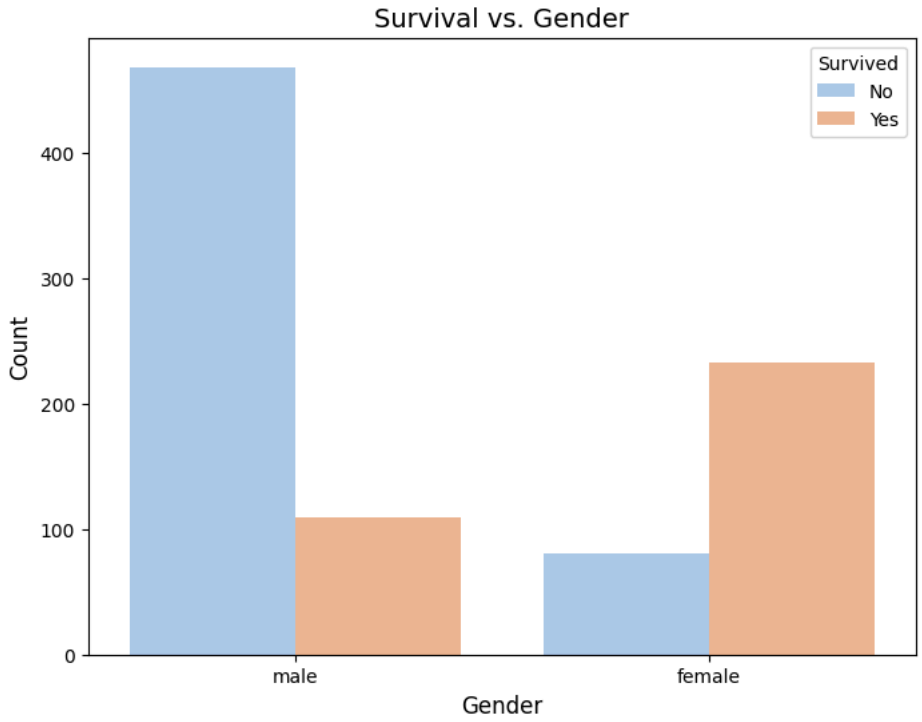
```
# Checking for any remaining missing values
print(df.isnull().sum())
```

```
survived         0
pclass           0
sex              0
age              0
sibsp            0
parch            0
fare             0
embarked         0
class            0
who              0
adult_male       0
deck           688
embark_town      2
alive            0
alone            0
dtype: int64
```

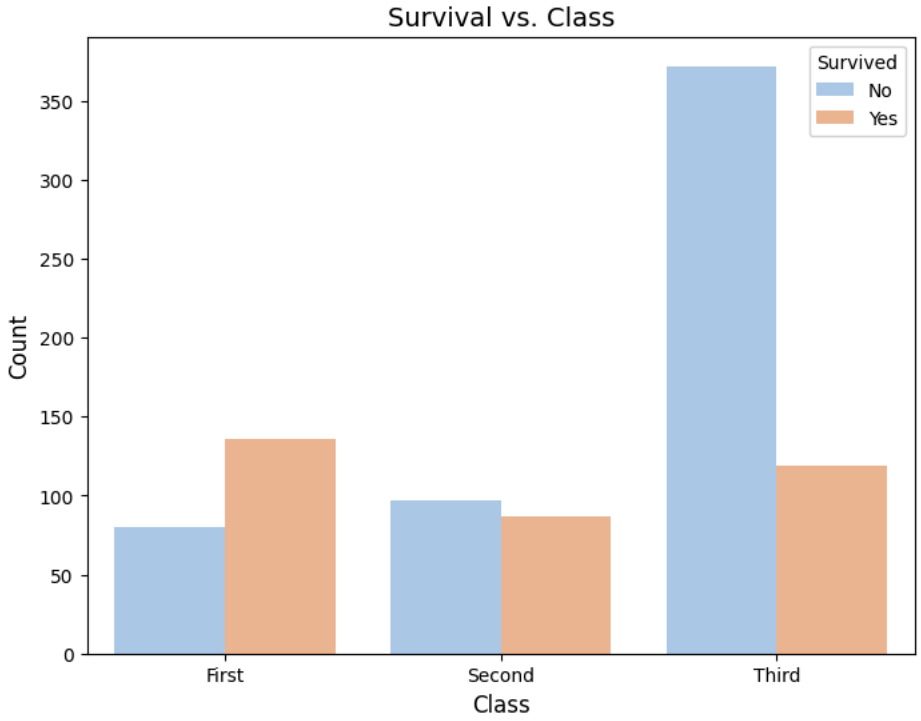
```
# Descriptive Statistics for a better understanding of the dataset
df.describe()
```

	survived	pclass	age	sibsp	parch	fare
count	891.000000	891.000000	891.000000	891.000000	891.000000	891.000000
mean	0.383838	2.308642	29.361582	0.523008	0.381594	32.204208
std	0.486592	0.836071	13.019697	1.102743	0.806057	49.693429
min	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	0.000000	2.000000	22.000000	0.000000	0.000000	7.910400
50%	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	1.000000	3.000000	35.000000	1.000000	0.000000	31.000000
max	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

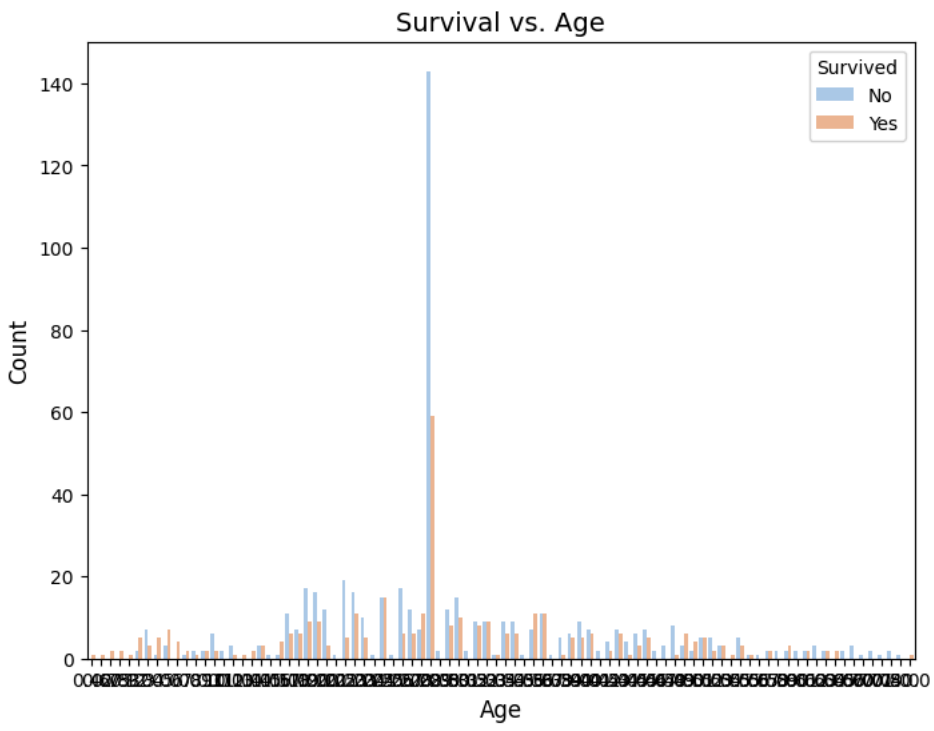
```
# Countplot: Survival vs. Gender
plt.figure(figsize=(8, 6))
sns.countplot(data=df, x='sex', hue='survived', palette='pastel')
plt.title('Survival vs. Gender', fontsize=14)
plt.xlabel('Gender', fontsize=12)
plt.ylabel('Count', fontsize=12)
plt.legend(title='Survived', labels=['No', 'Yes'])
plt.show()
```



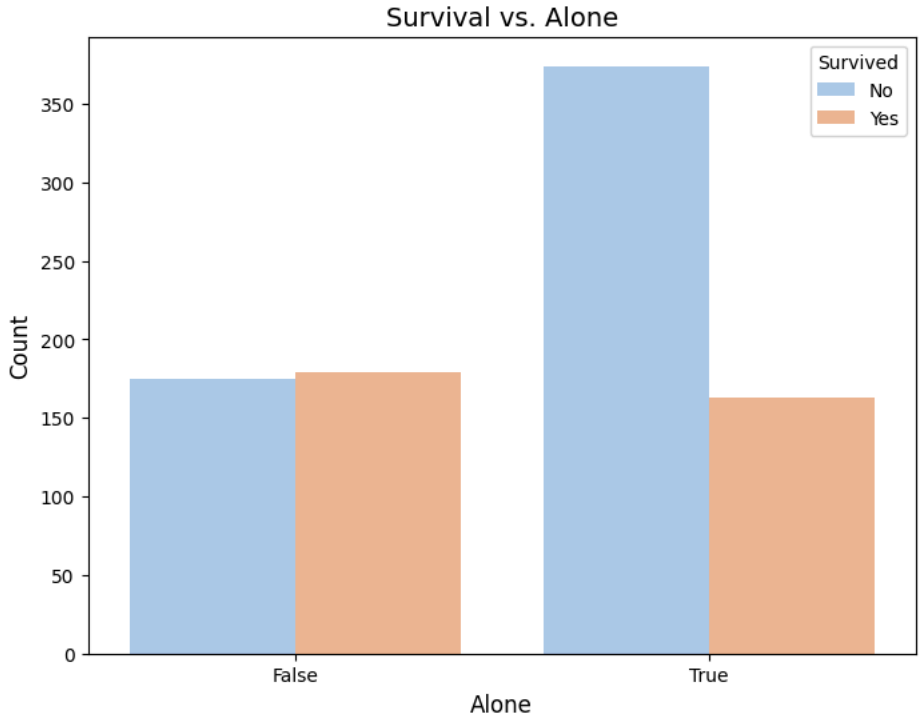

```
# Countplot: Survival vs. Class
plt.figure(figsize=(8, 6))
sns.countplot(data=df, x='class', hue='survived', palette='pastel')
plt.title('Survival vs. Class', fontsize=14)
plt.xlabel('Class', fontsize=12)
plt.ylabel('Count', fontsize=12)
plt.legend(title='Survived', labels=['No', 'Yes'])
plt.show()
```



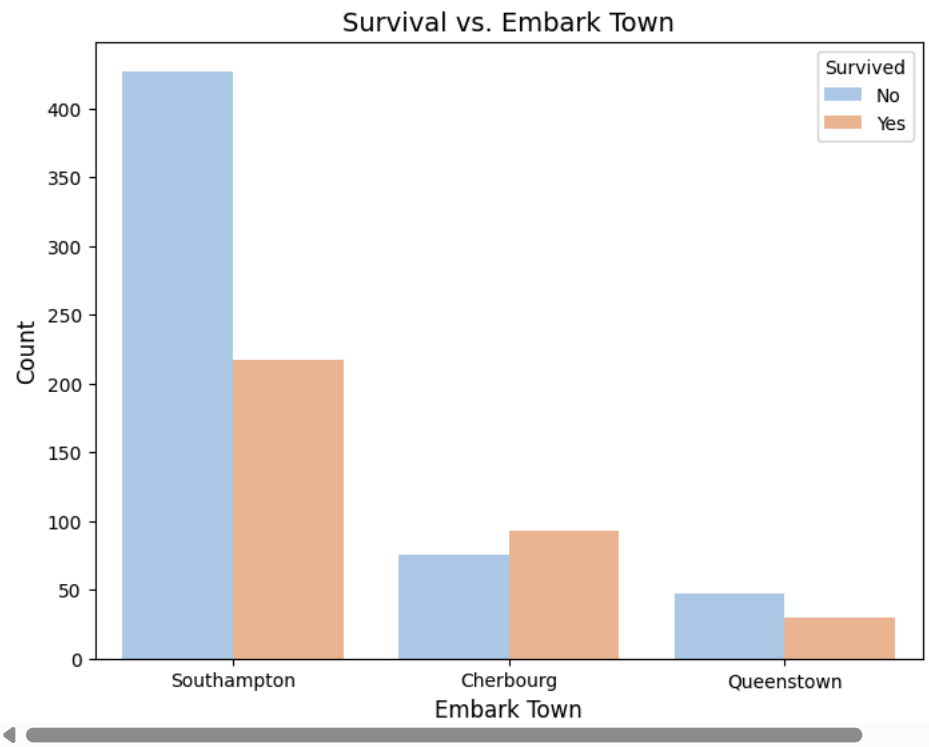

```
# Countplot: Survival vs. Age
plt.figure(figsize=(8, 6))
sns.countplot(data=df, x='age', hue='survived', palette='pastel')
plt.title('Survival vs. Age', fontsize=14)
plt.xlabel('Age', fontsize=12)
plt.ylabel('Count', fontsize=12)
plt.legend(title='Survived', labels=['No', 'Yes'])
plt.show()
```



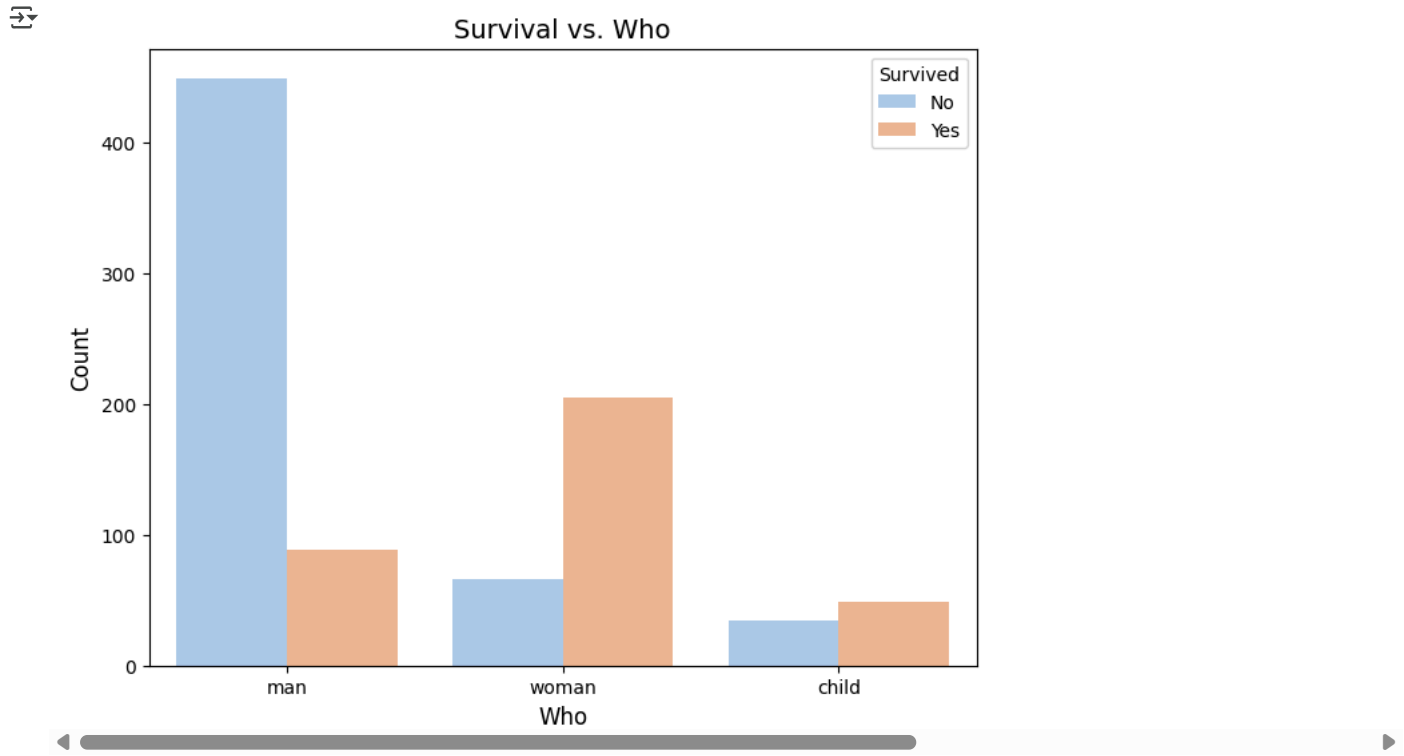

```
# Countplot: Survival vs. Alone (whether the passenger was alone or not)
plt.figure(figsize=(8, 6))
sns.countplot(data=df, x='alone', hue='survived', palette='pastel')
plt.title('Survival vs. Alone', fontsize=14)
plt.xlabel('Alone', fontsize=12)
plt.ylabel('Count', fontsize=12)
plt.legend(title='Survived', labels=['No', 'Yes'])
plt.show()
```



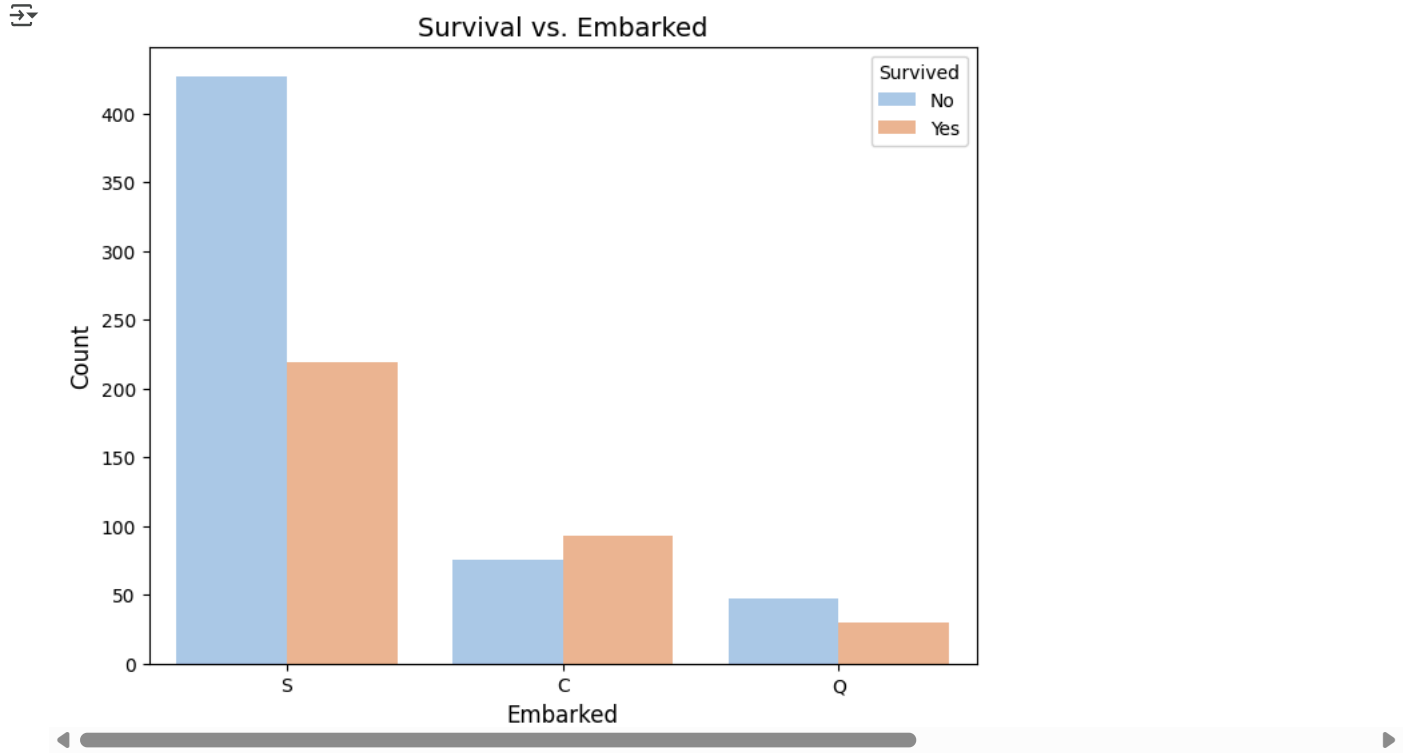

```
# Countplot: Survival vs. Embark Town
plt.figure(figsize=(8, 6))
sns.countplot(data=df, x='embark_town', hue='survived', palette='pastel')
plt.title('Survival vs. Embark Town', fontsize=14)
plt.xlabel('Embark Town', fontsize=12)
plt.ylabel('Count', fontsize=12)
plt.legend(title='Survived', labels=['No', 'Yes'])
plt.show()
```



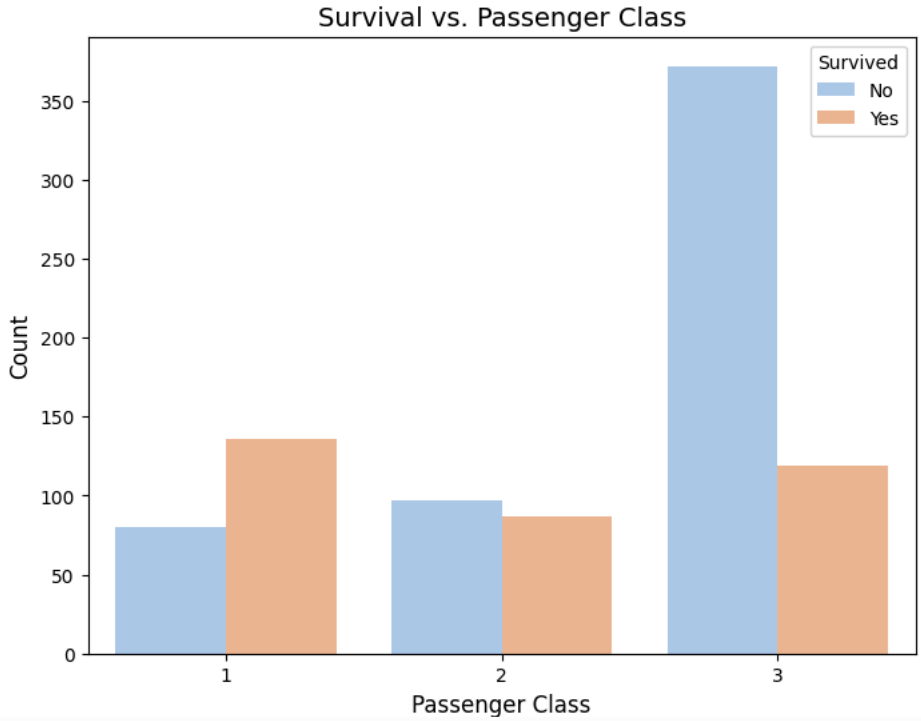

```
# Countplot: Survival vs. Who (man, woman, child)
plt.figure(figsize=(8, 6))
sns.countplot(data=df, x='who', hue='survived', palette='pastel')
plt.title('Survival vs. Who', fontsize=14)
plt.xlabel('Who', fontsize=12)
plt.ylabel('Count', fontsize=12)
plt.legend(title='Survived', labels=['No', 'Yes'])
plt.show()
```



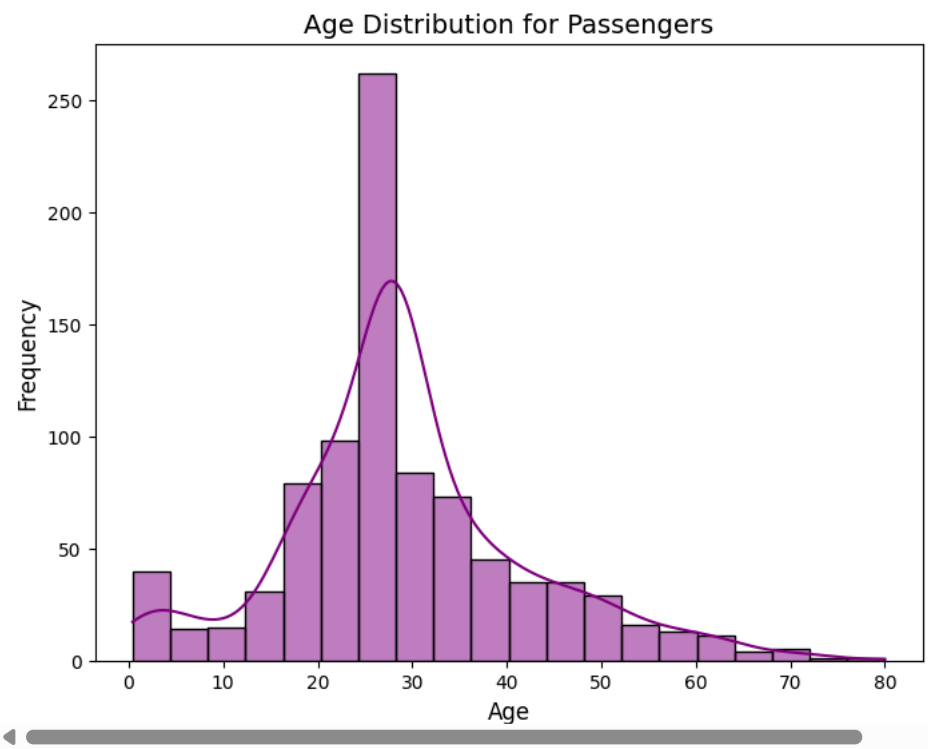

```
# Countplot: Survival vs. Embarked (Port of embarkation)
plt.figure(figsize=(8, 6))
sns.countplot(data=df, x='embarked', hue='survived', palette='pastel')
plt.title('Survival vs. Embarked', fontsize=14)
plt.xlabel('Embarked', fontsize=12)
plt.ylabel('Count', fontsize=12)
plt.legend(title='Survived', labels=['No', 'Yes'])
plt.show()
```



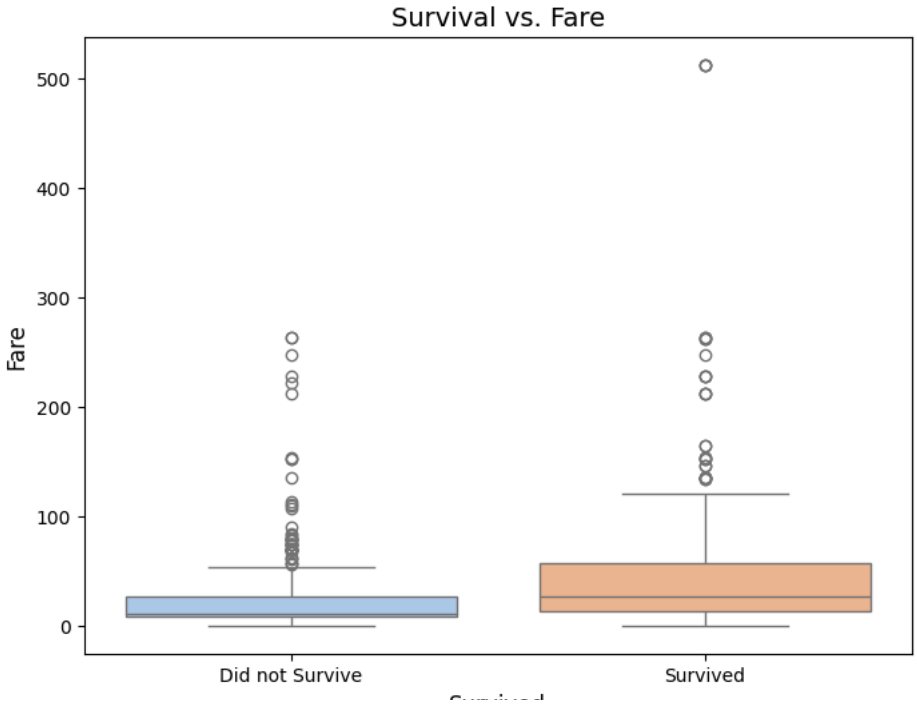

```
# Countplot: Survival vs. Pclass (Passenger Class)
plt.figure(figsize=(8, 6))
sns.countplot(data=df, x='pclass', hue='survived', palette='pastel')
plt.title('Survival vs. Passenger Class', fontsize=14)
plt.xlabel('Passenger Class', fontsize=12)
plt.ylabel('Count', fontsize=12)
plt.legend(title='Survived', labels=['No', 'Yes'])
plt.show()
```




```
# Histplot: Age Distribution for Passengers
plt.figure(figsize=(8, 6))
sns.histplot(df['age'], kde=True, color='purple', bins=20)
plt.title('Age Distribution for Passengers', fontsize=14)
plt.xlabel('Age', fontsize=12)
plt.ylabel('Frequency', fontsize=12)
plt.show()
```




```
# Boxplot: Survival vs. Fare (Visualizing fare distribution for survivors and non-survivors)
plt.figure(figsize=(8, 6))
sns.boxplot(data=df, x='survived', y='fare', hue='survived', palette='pastel', legend=False)
plt.title('Survival vs. Fare', fontsize=14)
plt.xlabel('Survived', fontsize=12)
plt.ylabel('Fare', fontsize=12)
plt.xticks([0, 1], ['Did not Survive', 'Survived'])
plt.show()
```




```
# Histplot: Fare Distribution for Passengers
plt.figure(figsize=(8, 6))
sns.histplot(df['fare'], kde=True, color='purple', bins=30)
plt.title('Fare Distribution for Passengers', fontsize=14)
plt.xlabel('Fare', fontsize=12)
plt.ylabel('Frequency', fontsize=12)
plt.show()
```