

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

%matplotlib inline
```

```
[2]: df=pd.read_csv("train.csv")
print(df)
```

	PassengerId	Survived	Pclass	\
0	1	0	3	
1	2	1	1	
2	3	1	3	
3	4	1	1	
4	5	0	3	
..	
886	887	0	2	
887	888	1	1	
888	889	0	3	
889	890	1	1	
890	891	0	3	

	Name	Sex	Age	SibSp	\
0	Braund, Mr. Owen Harris	male	22.0	1	
1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	
2	Heikkinen, Miss. Laina	female	26.0	0	
3	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	
4	Allen, Mr. William Henry	male	35.0	0	
..	
886	Montvila, Rev. Juozas	male	27.0	0	
887	Graham, Miss. Margaret Edith	female	19.0	0	
888	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	
889	Behr, Mr. Karl Howell	male	26.0	0	
890	Dooley, Mr. Patrick	male	32.0	0	

	Parch	Ticket	Fare	Cabin	Embarked
0	0	A/5 21171	7.2500	NaN	S
1	0	PC 17599	71.2833	C85	C
2	0	STON/O2. 3101282	7.9250	NaN	S
3	0	113803	53.1000	C123	S
4	0	373450	8.0500	NaN	S
..
886	0	211536	13.0000	NaN	S
887	0	112053	30.0000	B42	S
888	2	W./C. 6607	23.4500	NaN	S
889	0	111369	30.0000	C148	C
890	0	370376	7.7500	NaN	Q

[891 rows x 12 columns]

[3]: df.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
```

```
df.describe()
```

[4]:

	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

```
df.isnull()
```

[5]:

[illegible]

887	False	False	False	False	False	False	False	False	False	False	False	False
888	False	False	False	False	False	True	False	False	False	False	True	False
889	False	False	False	False	False	False	False	False	False	False	False	False
890	False	False	False	False	False	False	False	False	False	False	True	False

891 rows × 12 columns

```
[7]: # Select only the numeric columns
numeric_df = df.select_dtypes(include=[np.number])

corr_matrix = numeric_df.corr()

print(corr_matrix)
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch
PassengerId	1.000000	-0.005007	-0.035144	0.036847	-0.057527	-0.001652
Survived	-0.005007	1.000000	-0.338481	-0.077221	-0.035322	0.081629
Pclass	-0.035144	-0.338481	1.000000	-0.369226	0.083081	0.018443
Age	0.036847	-0.077221	-0.369226	1.000000	-0.308247	-0.189119
SibSp	-0.057527	-0.035322	0.083081	-0.308247	1.000000	0.414838
Parch	-0.001652	0.081629	0.018443	-0.189119	0.414838	1.000000
Fare	0.012658	0.257307	-0.549500	0.096067	0.159651	0.216225

	Fare
PassengerId	0.012658
Survived	0.257307
Pclass	-0.549500
Age	0.096067
SibSp	0.159651
Parch	0.216225
Fare	1.000000

```
[8]: df.duplicated()
```

```
[8]: 0      False
      1      False
      2      False
      3      False
      4      False
      ...
     886     False
     887     False
     888     False
     889     False
     890     False
      Length: 891, dtype: bool
```

```
[9]: df.isnull().sum()
```

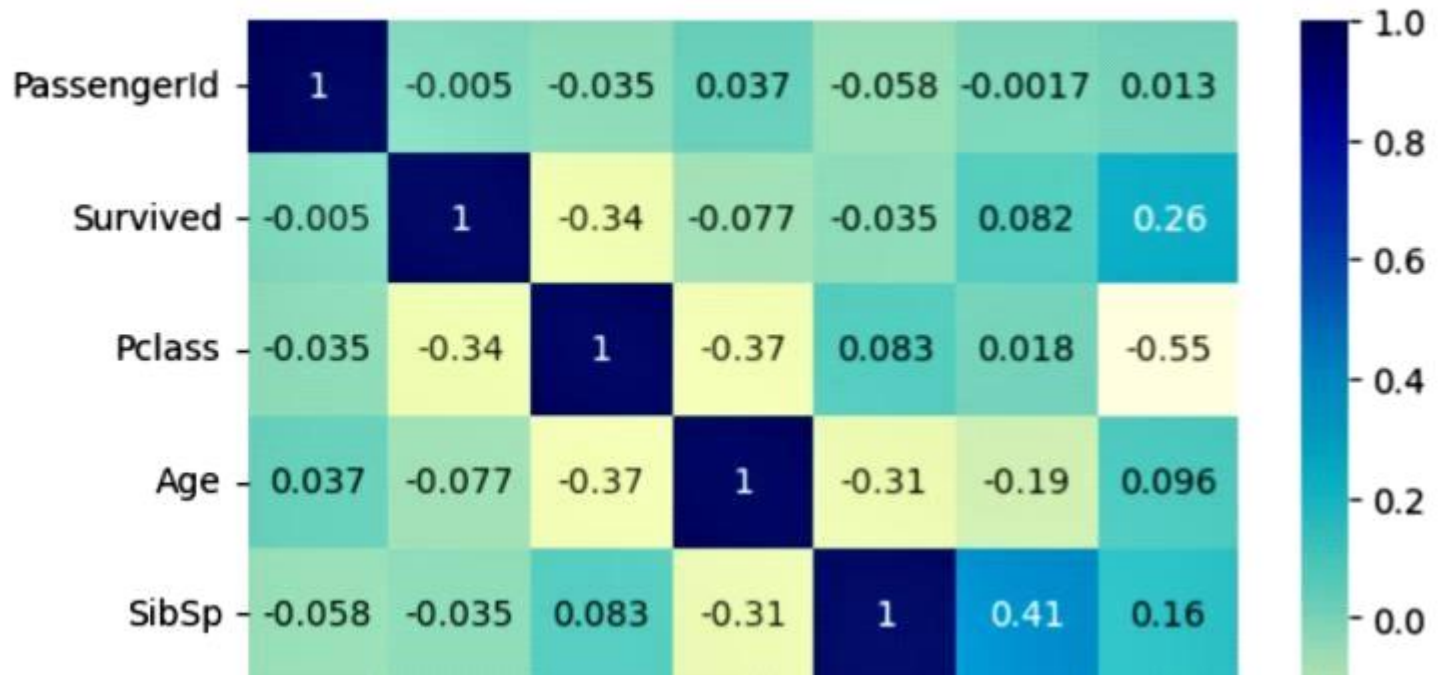
```
[9]: PassengerId      0
      Survived        0
      Pclass         0
      Name           0
      Sex            0
      Age           177
      SibSp          0
      Parch          0
      Ticket         0
      Fare           0
      Cabin         687
      Embarked       2
      dtype: int64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

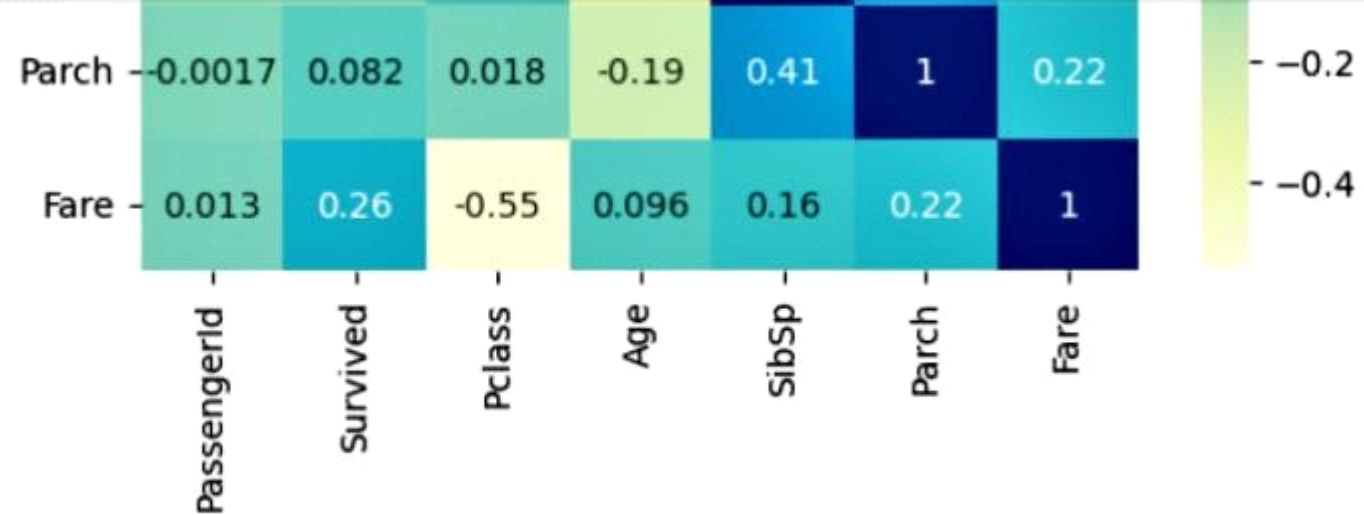
```
[10]: df.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

```
[13]: sns.heatmap(corr_matrix, annot=True, cmap="YlGnBu")
```

```
# Show the plot  
plt.show()
```



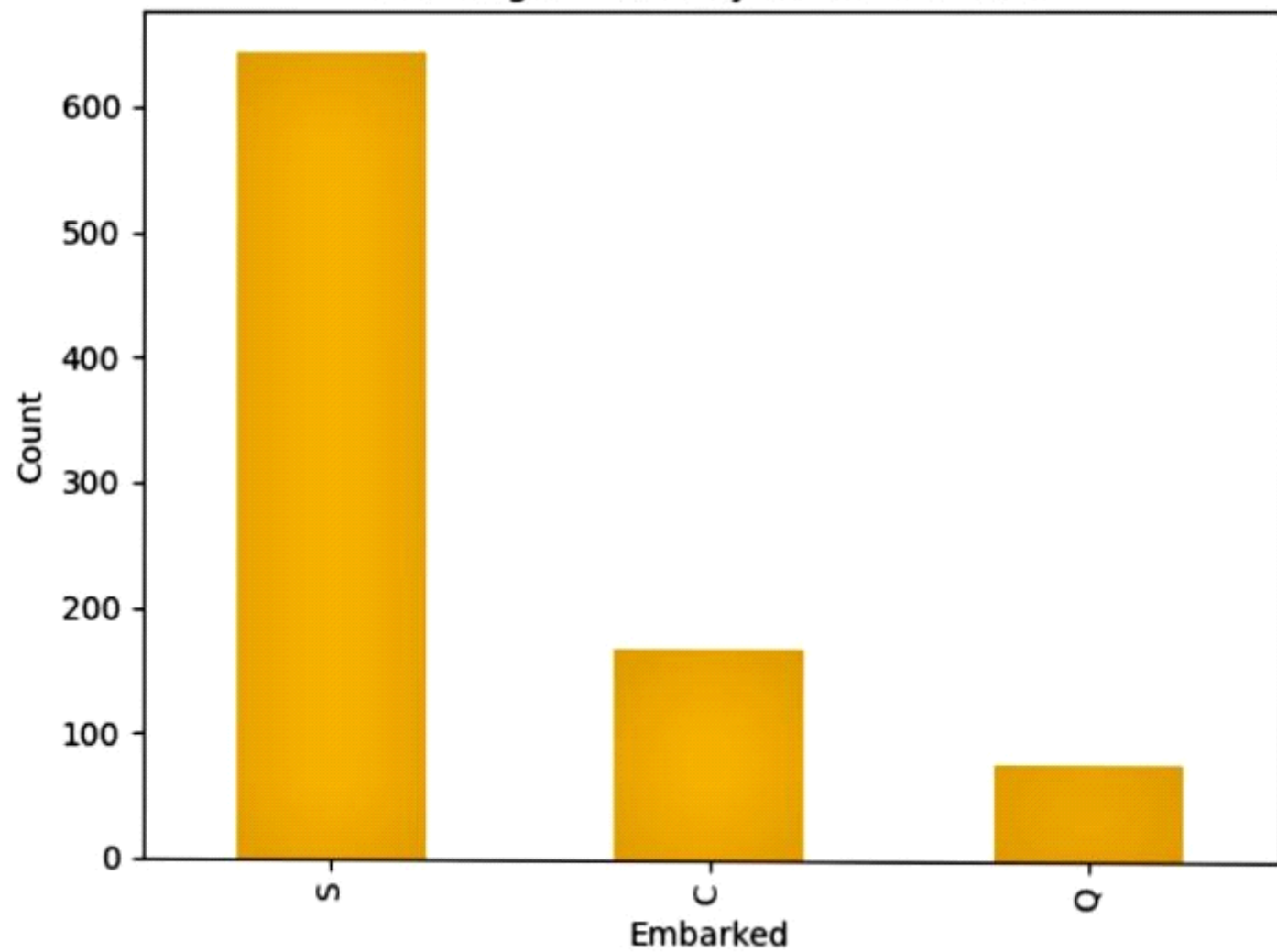


```
[14]: embarked_counts = df['Embarked'].value_counts()
print(embarked_counts)
```

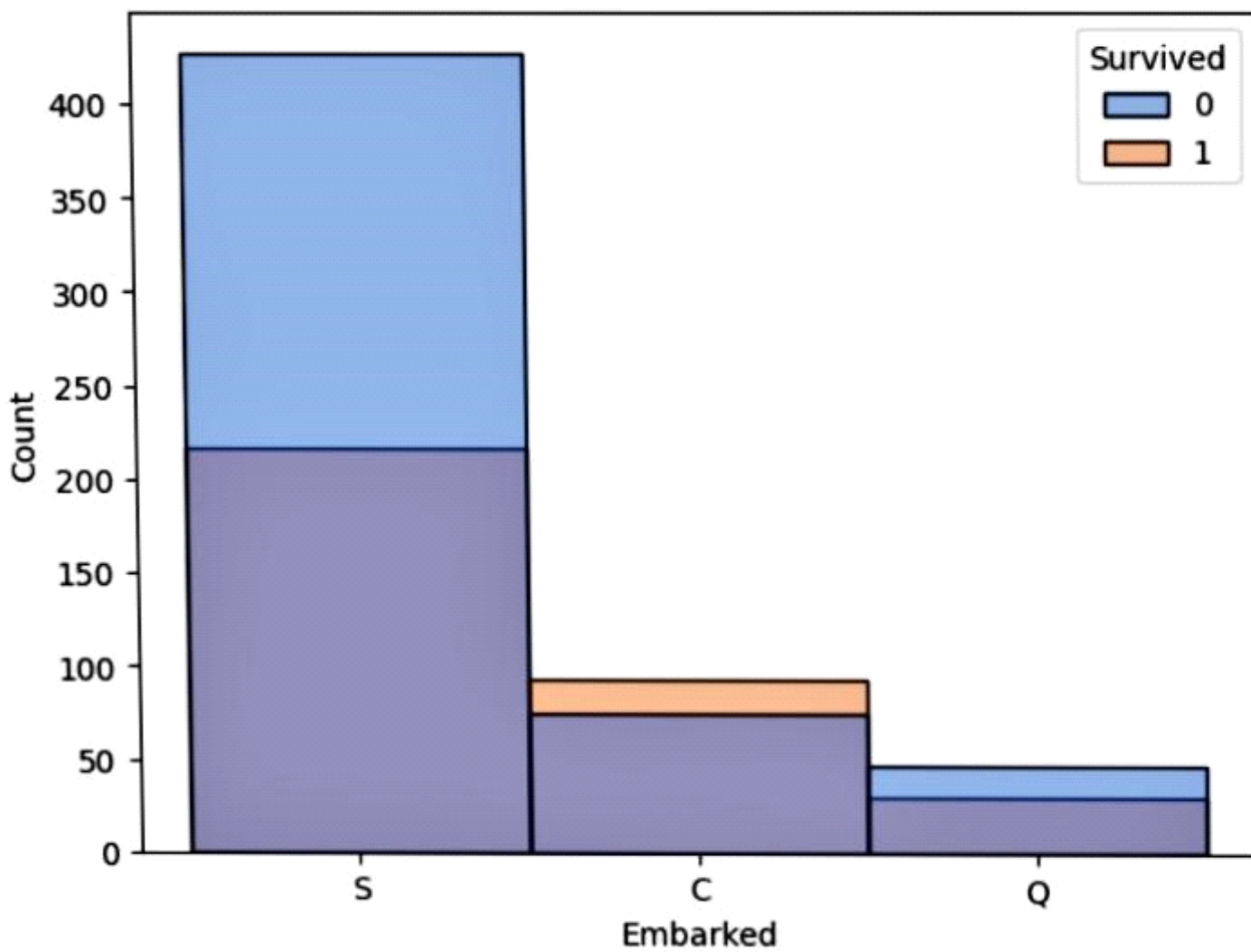
```
embarked_counts.plot(kind='bar',color='goldenrod')
plt.xlabel('Embarked')
plt.ylabel('Count')
plt.title('Passenger Count by Embarked Port')
plt.show()
```

```
Embarked
S      644
C      168
Q       77
Name: count, dtype: int64
```

Passenger Count by Embarked Port




```
ax = sns.histplot(x="Embarked", data=df, hue='Survived', palette='muted')
```



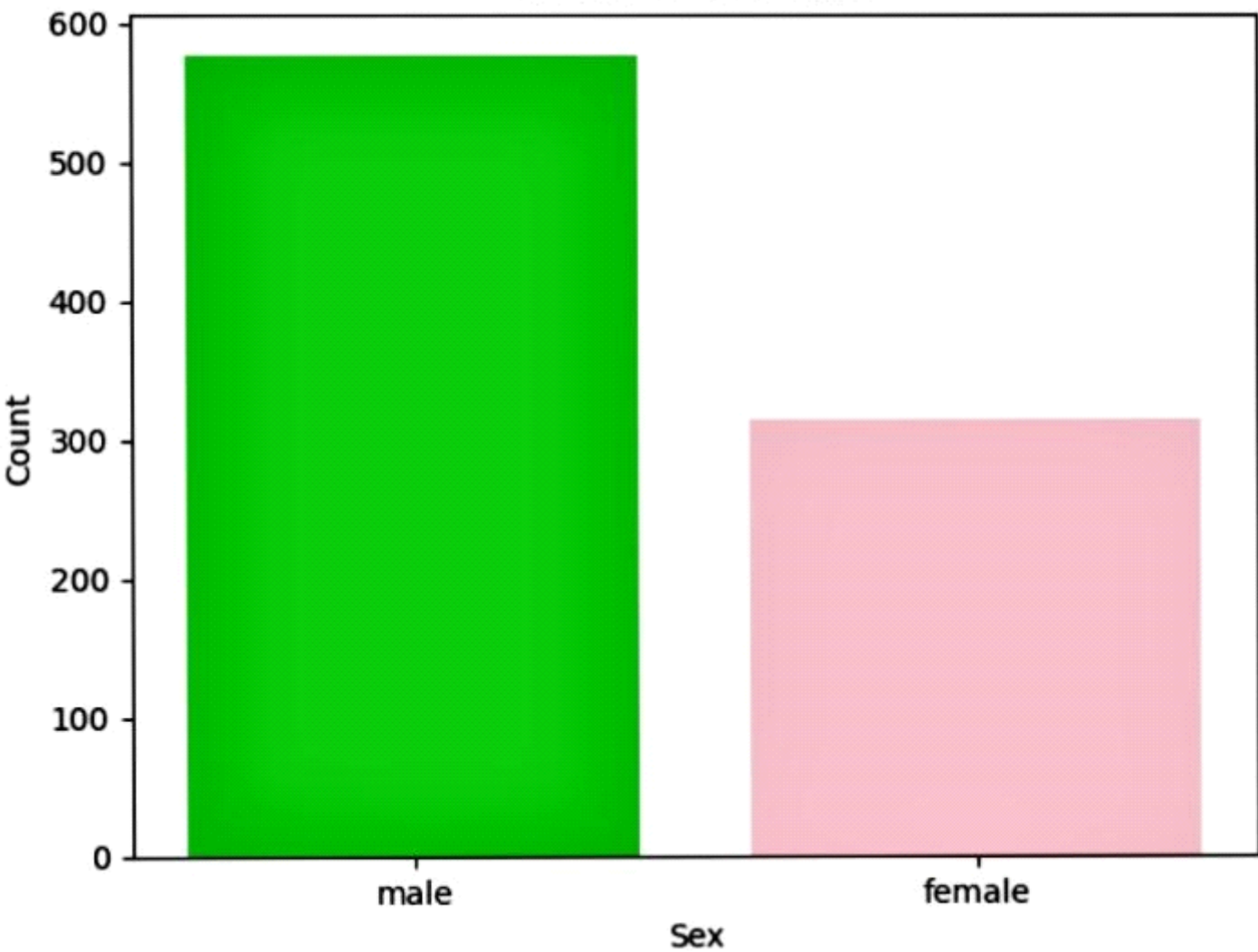
```
[16]: gender=df['Sex'].value_counts()
      print(gender)

      c=['limegreen','pink']
      sns.countplot(data=df, x='Sex',palette=c)

      plt.xlabel('Sex')
      plt.ylabel('Count')
      plt.title('Gender Distribution')
      plt.show()
```

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

Gender Distribution

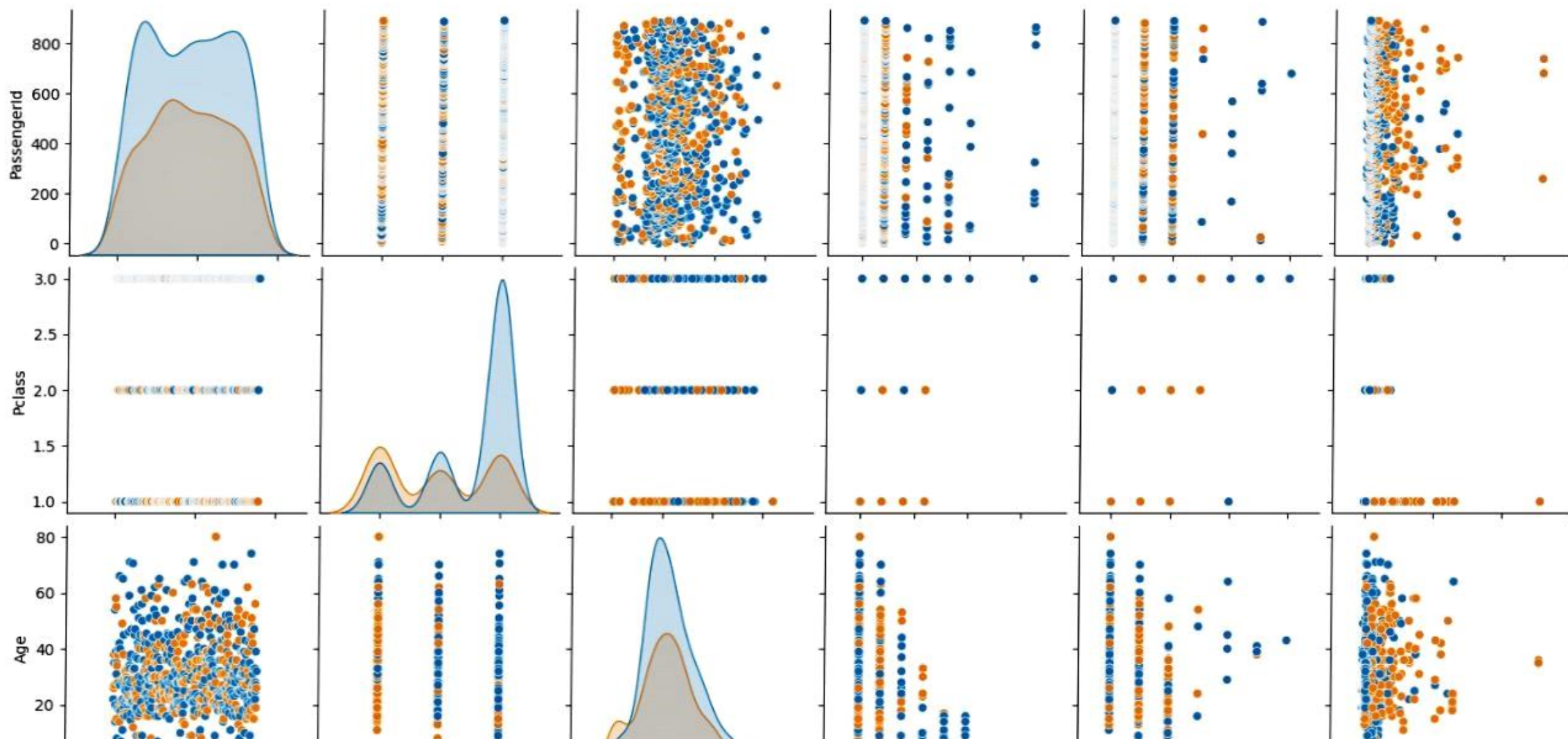


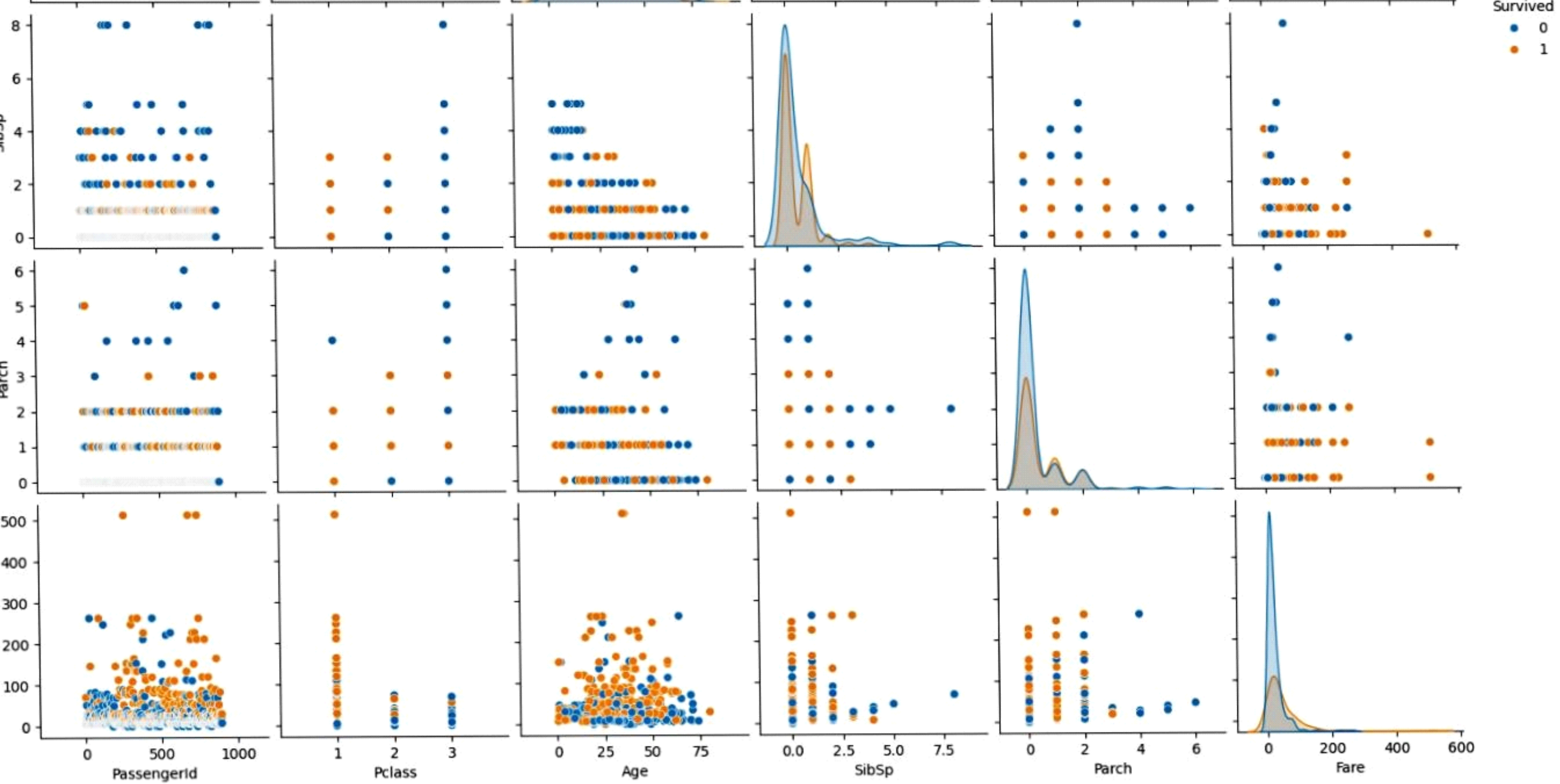
```
[17]: survived_count = df['Survived'].value_counts()[1]
print(f'The number of people who survived: {survived_count}')
```

```
sns.pairplot(df, hue = 'Survived')
```

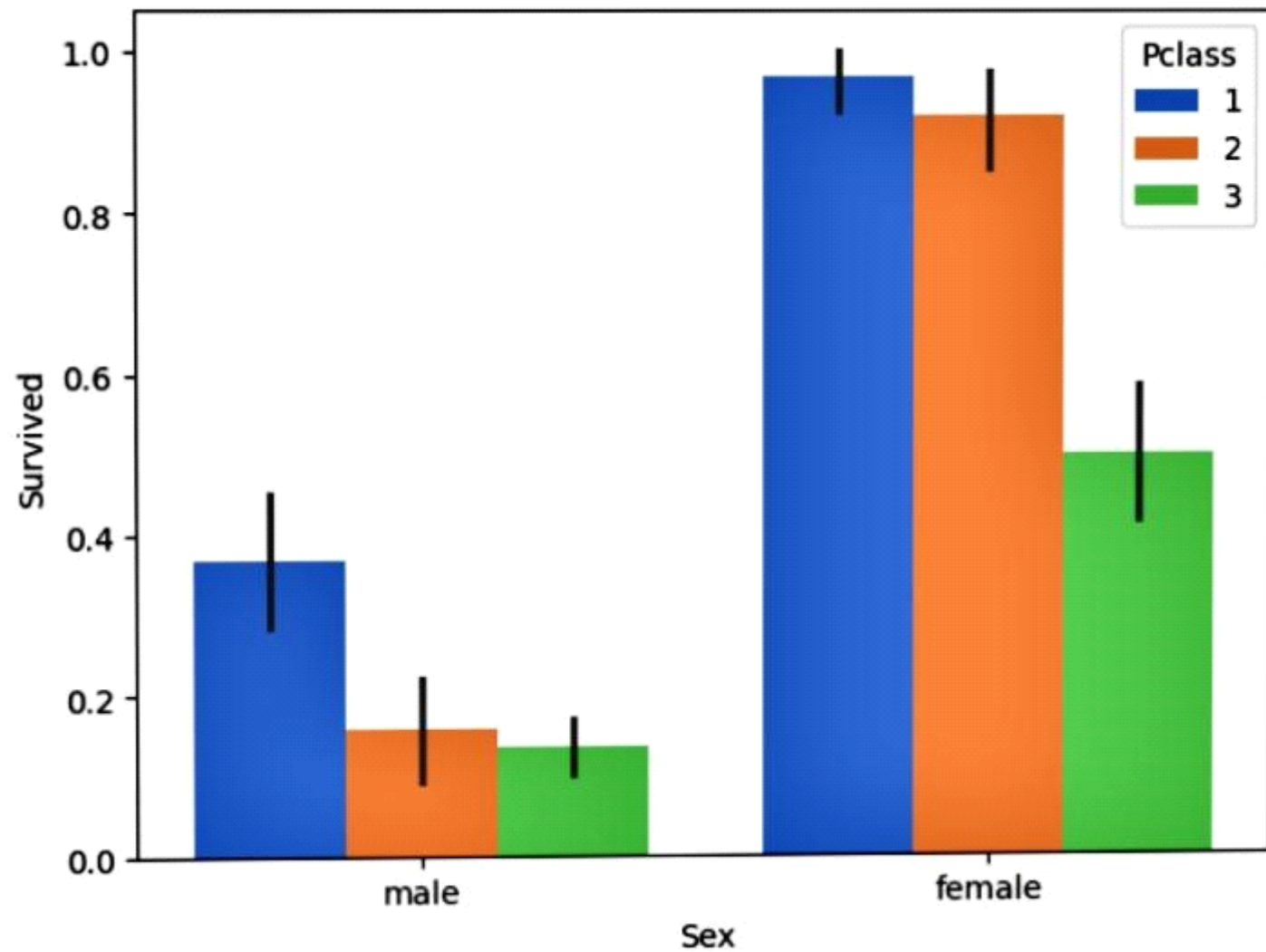
The number of people who survived: 342

```
[17]: <seaborn.axisgrid.PairGrid at 0x17fdee2b770>
```

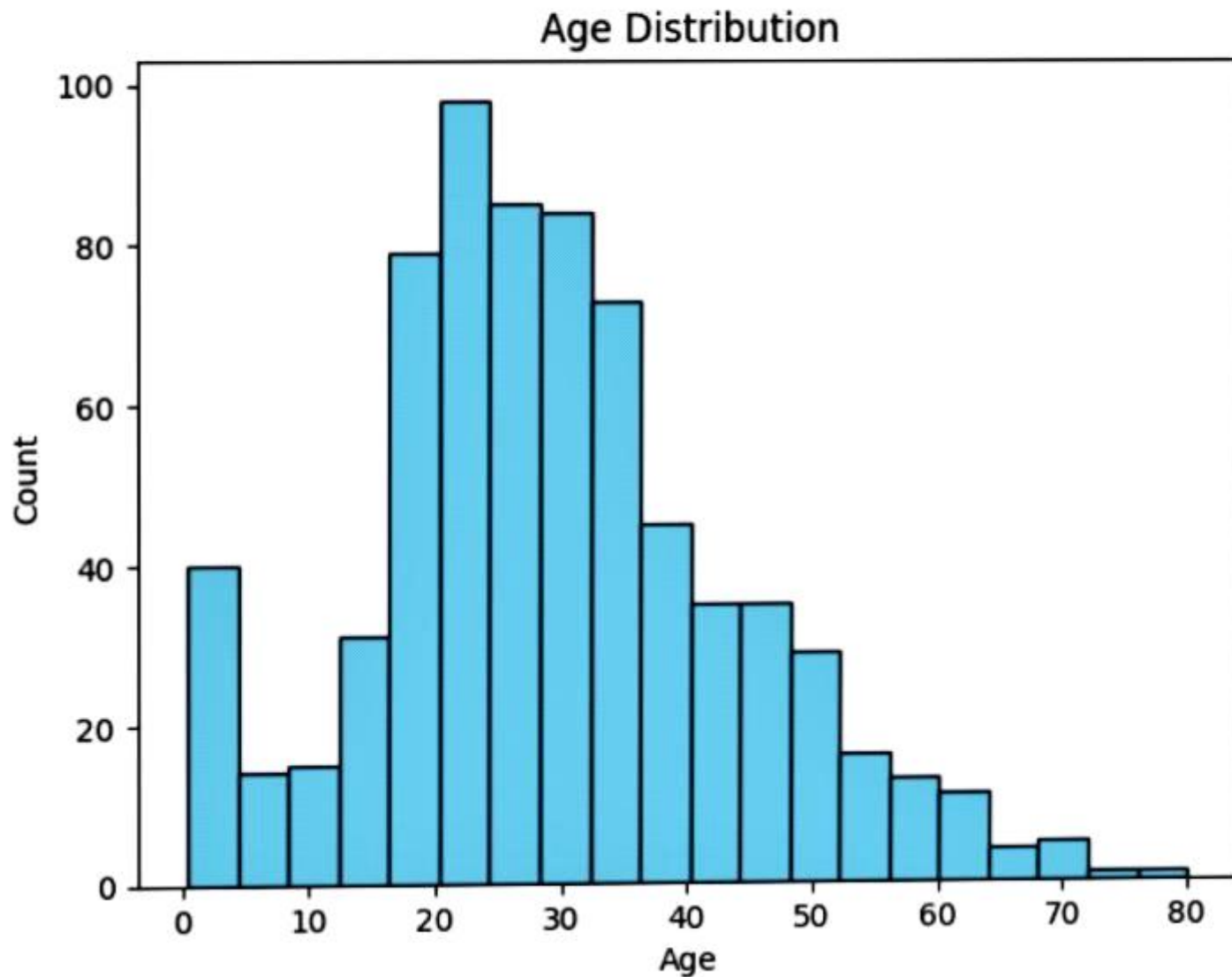




```
ax = sns.barplot(x="Sex", y="Survived", data=df, hue='Pclass', palette='muted')
```



```
plt.hist(df['Age'].dropna(), bins=20, edgecolor='k', color='skyblue')  
plt.xlabel('Age')  
plt.ylabel('Count')  
plt.title('Age Distribution')  
plt.show()
```

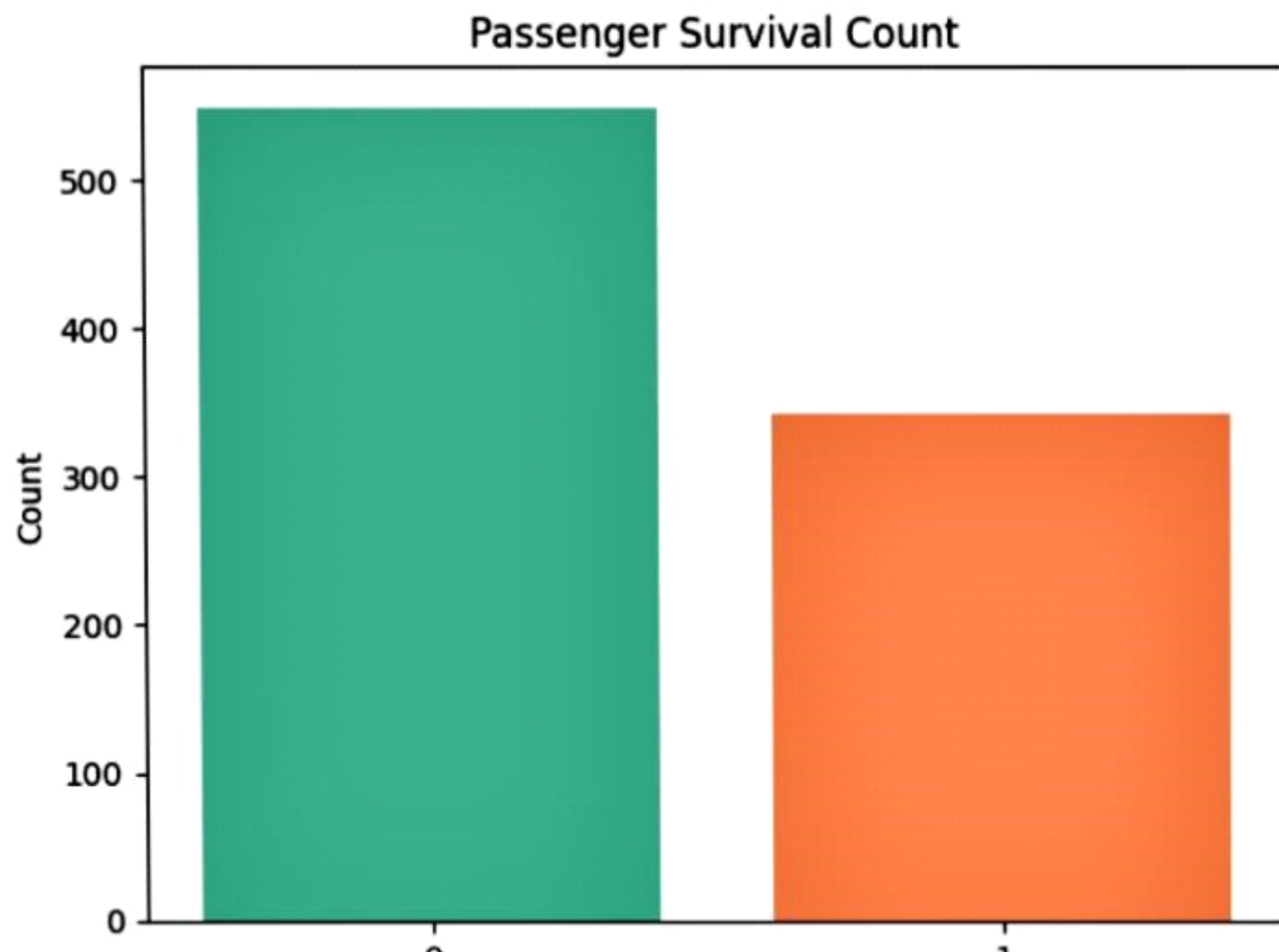


```
sns.countplot(data=df, x='Survived',palette='Set2')
plt.xlabel('Survived')
plt.ylabel('Count')
plt.title('Passenger Survival Count')
plt.show()
```

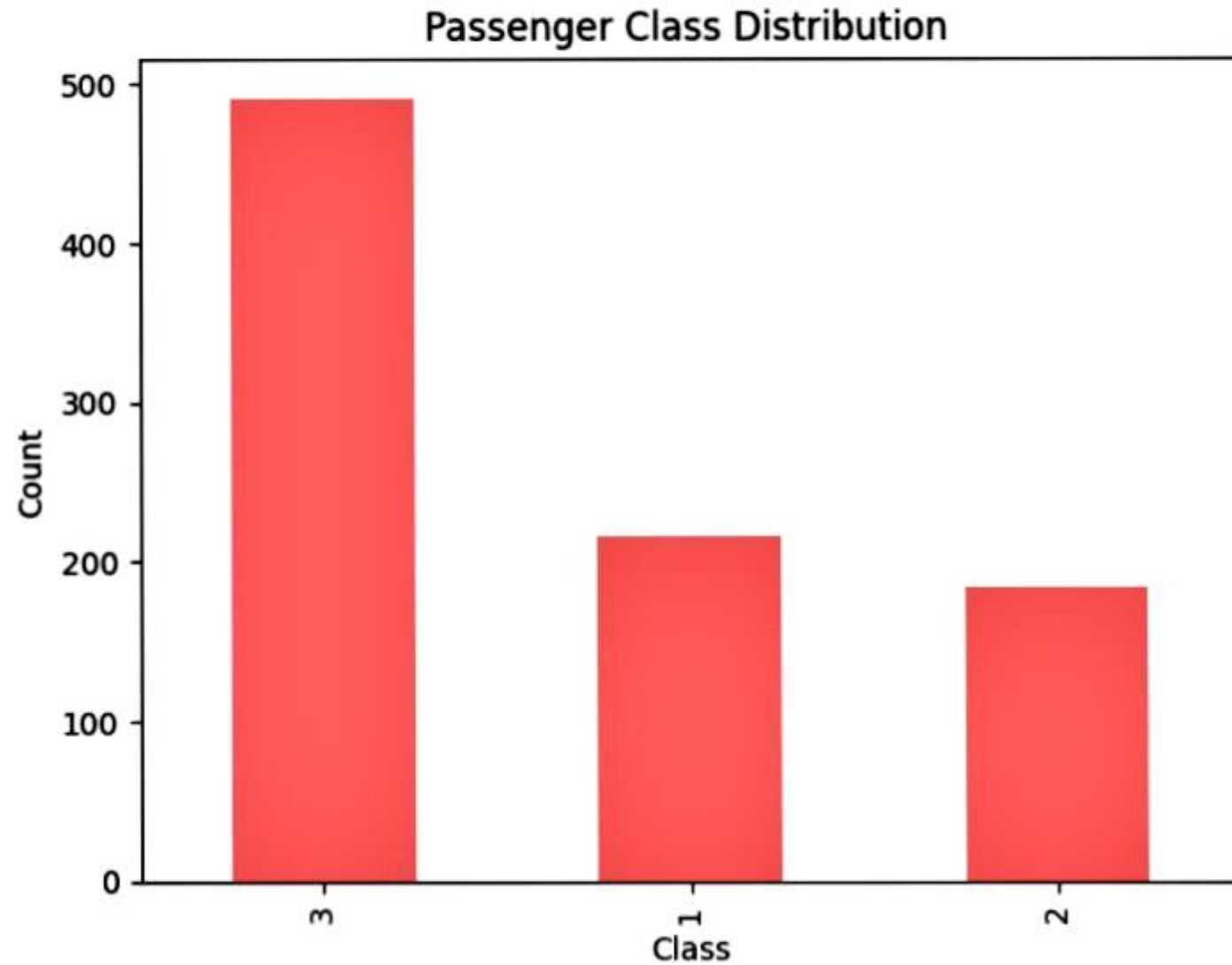
C:\Users\user\AppData\Local\Temp\ipykernel_19172\3567058290.py:1: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `hue` parameter to the same effect.

```
sns.countplot(data=df, x='Survived',palette='Set2')
```




```
[21]: df['Pclass'].value_counts().plot(kind='bar',color='lightcoral')
plt.xlabel('Class')
plt.ylabel('Count')
plt.title('Passenger Class Distribution')
plt.show()
```

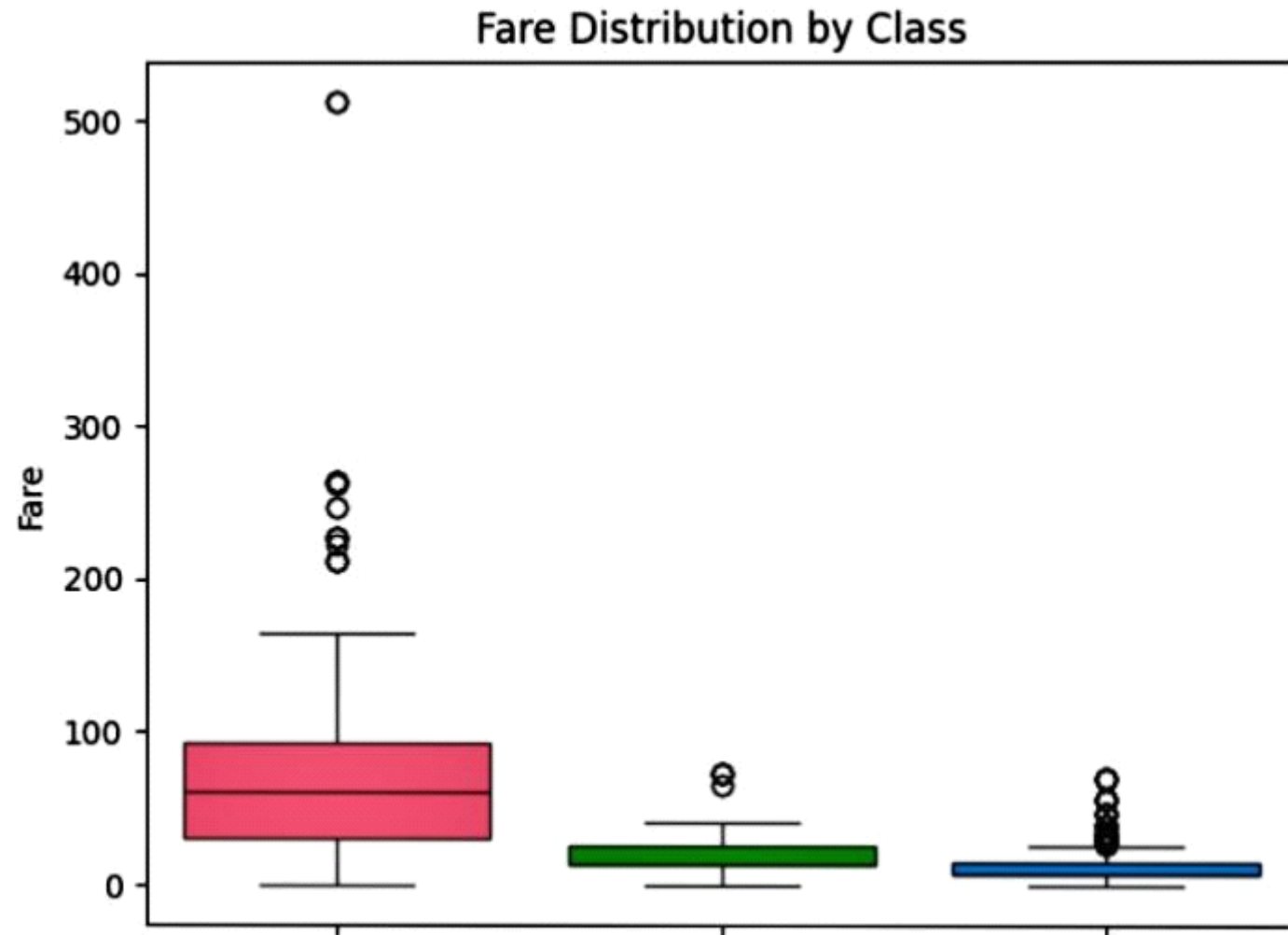


```
[22]: sns.boxplot(x='Pclass', y='Fare', data=df,palette='husl')
plt.xlabel('Class')
plt.ylabel('Fare')
plt.title('Fare Distribution by Class')
plt.show()
```

C:\Users\user\AppData\Local\Temp\ipykernel_19172\2719604965.py:1: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

```
sns.boxplot(x='Pclass', y='Fare', data=df,palette='husl')
```



```
plt.plot(x=Passenger Class, y=Survived, hue=Sex, data=df, kind="bar", palette="husl")
```

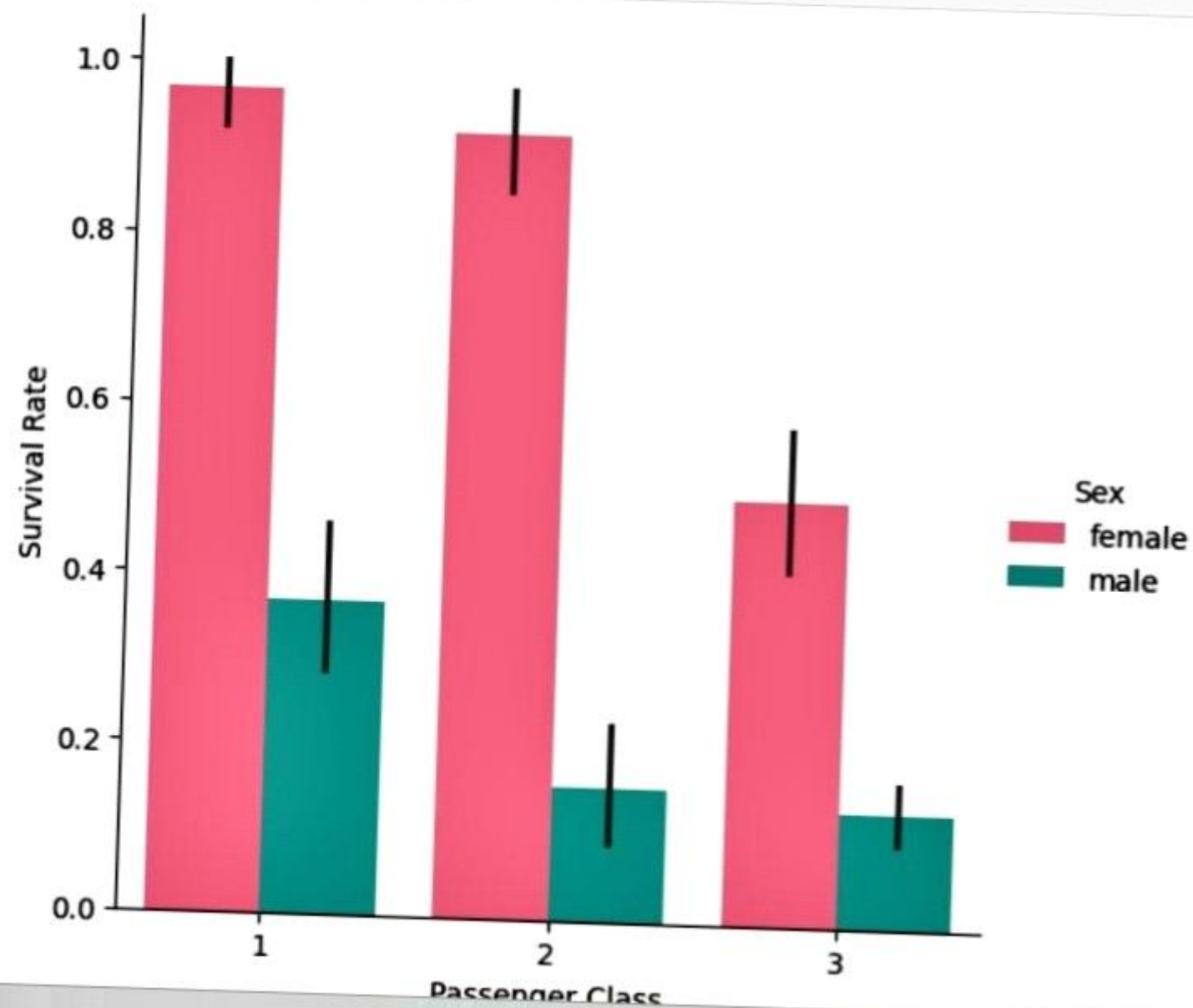
```
# Set labels
```

```
plt.xlabel('Passenger Class')
```

```
plt.ylabel('Survival Rate')
```

```
# Show the plot
```

```
plt.show()
```

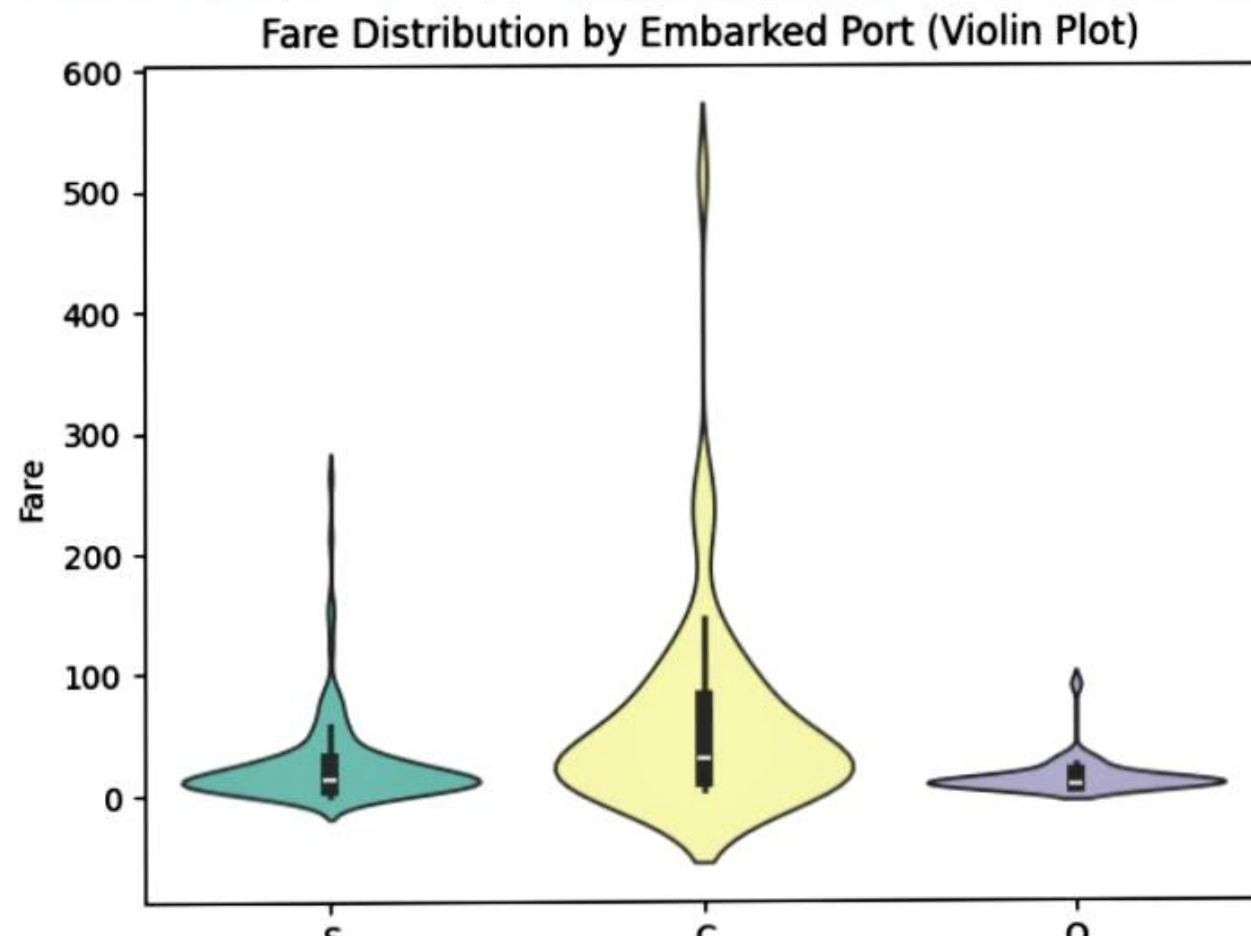


```
[25]: sns.violinplot(x='Embarked', y='Fare', data=df, palette="Set3")
plt.xlabel('Embarked')
plt.ylabel('Fare')
plt.title('Fare Distribution by Embarked Port (Violin Plot)')
plt.show()
```

C:\Users\user\AppData\Local\Temp\ipykernel_19172\401693627.py:1: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

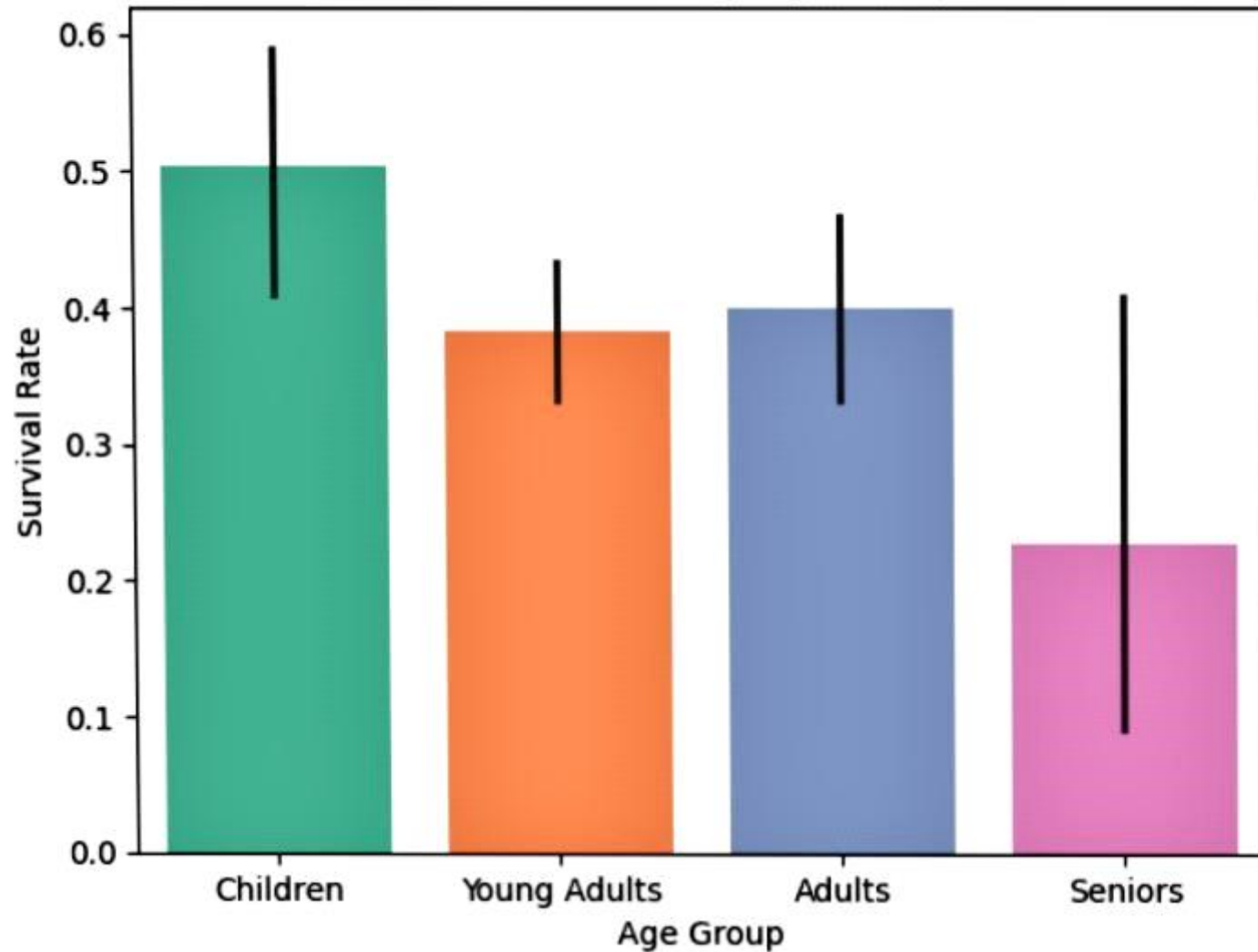
```
sns.violinplot(x='Embarked', y='Fare', data=df, palette="Set3")
```




```
[26]: bins = [0, 18, 35, 60, df['Age'].max()]
labels = ['Children', 'Young Adults', 'Adults', 'Seniors']
df['AgeGroup'] = pd.cut(df['Age'], bins=bins, labels=labels)

sns.barplot(data=df, x='AgeGroup', y='Survived', palette='Set2')
plt.xlabel('Age Group')
plt.ylabel('Survival Rate')
plt.title('Survival Rate by Age Group')
plt.show()
```

Survival Rate by Age Group



```
[28]: sns.swarmplot(data=df, x='Pclass', y='Age', palette='husl') # Example palette
plt.xlabel('Passenger Class')
plt.ylabel('Age')
plt.show()
```

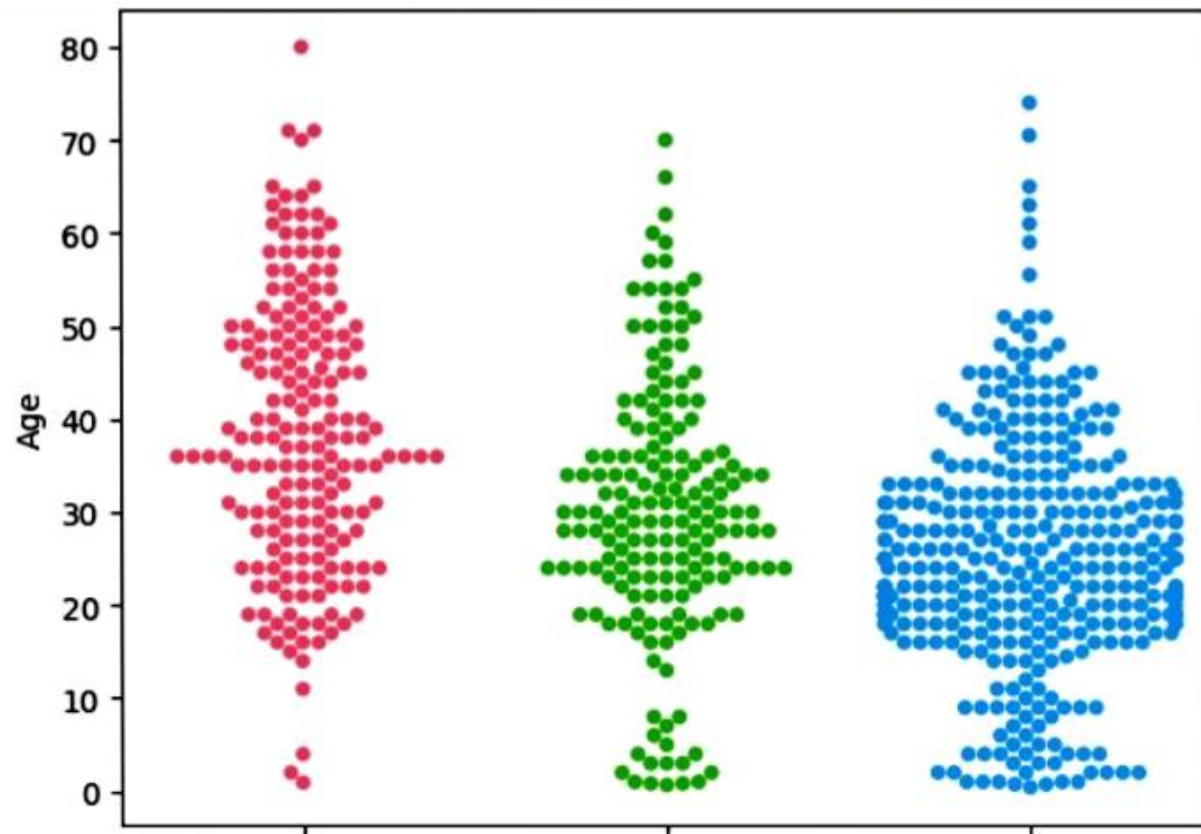
C:\Users\user\AppData\Local\Temp\ipykernel_19172\2490849921.py:1: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

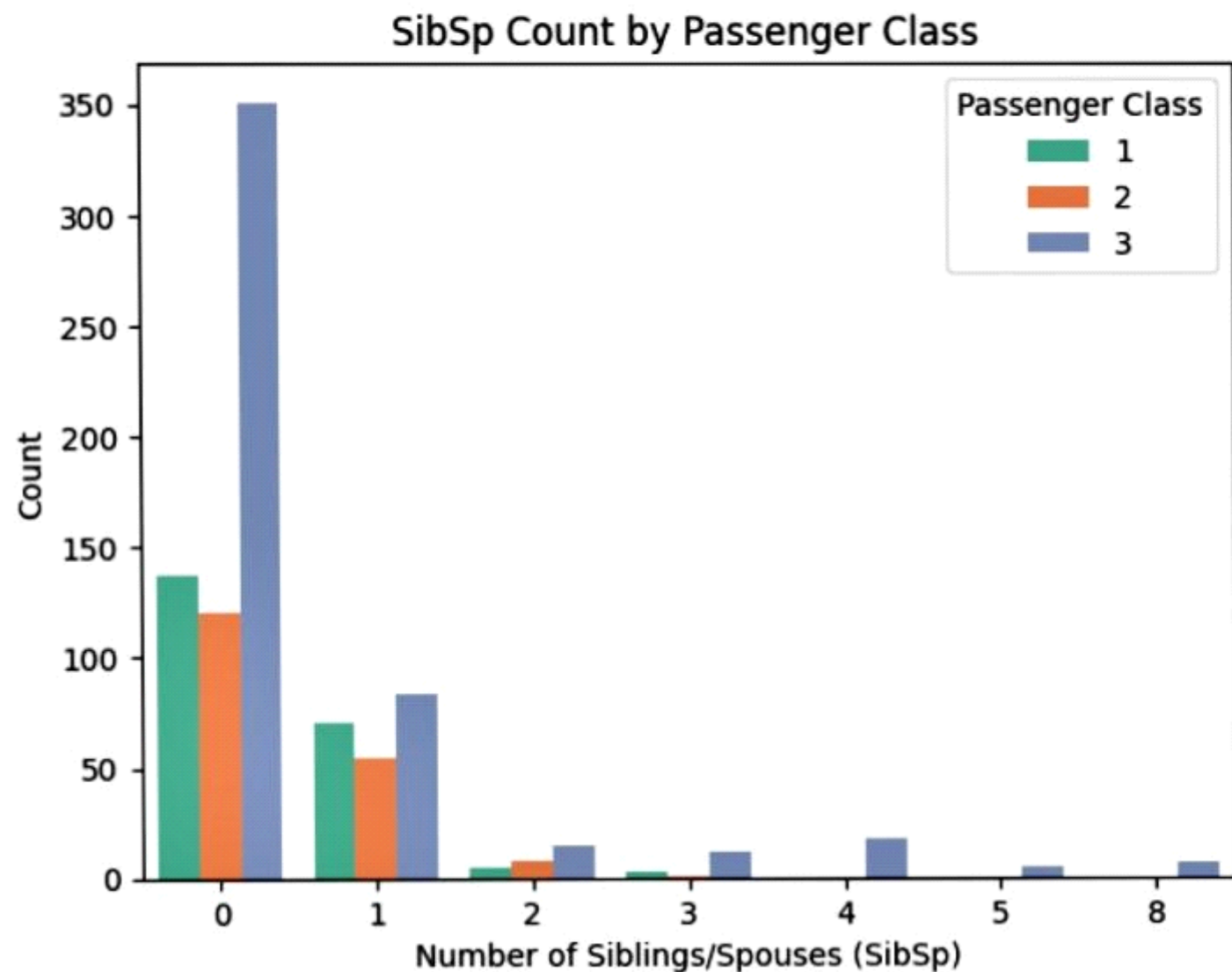
```
sns.swarmplot(data=df, x='Pclass', y='Age', palette='husl') # Example palette
```

C:\Users\user\AppData\Local\Programs\Python\Python313\Lib\site-packages\seaborn\categorical.py:3399: UserWarning: 15.2% of the points cannot be placed; you may want to decrease the size of the markers or use stripplot.

```
warnings.warn(msg, UserWarning)
```



```
sns.countplot(data=df, x='SibSp', hue='Pclass', palette='Set2')
plt.xlabel('Number of Siblings/Spouses (SibSp)')
plt.ylabel('Count')
plt.title('SibSp Count by Passenger Class')
plt.legend(title='Passenger Class')
plt.show()
```





```
[30]: df['HasCabin'] = df['Cabin'].notnull()
sns.barplot(data=df, x='HasCabin', y='Survived', palette='Set3')
plt.xlabel('Cabin Availability (1 = Has Cabin, 0 = No Cabin)')

plt.ylabel('Survival Rate')
plt.title('Survival Rate by Cabin Availability')
plt.xticks([0, 1], ['No Cabin', 'Has Cabin'])
plt.show()
```

