```
# Name:Atharva Kangralkar
# Roll no : 54
# CS - AIML - A
# Colab link:- https://colab.research.google.com/drive/1DlCkItiOxM5JRqj1T4DxO1_SS9Gb5pI2?usp=sharing
# Lab Assignment 8
# Exploratory Data Analysis (EDA) -Titanic Dataset
```

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

```
# Import Dataset.
df = pd.read_csv('Titanic-Dataset.csv')
```

	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

Next steps: Generate code with df View recommended plots New interactive sheet

```
# Show preview of dataset /Show first five lines of dataset
df.head(5)
```

```
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
```

```
# Find out different column types from data.
# Numerical -
# Categorical -
# Mixed -
numerical_cols = df.select_dtypes(include=['number']).columns
categorical_cols = df.select_dtypes(include=['object']).columns
```

```
Index(['PassengerId', 'Survived', 'Pclass', 'Age', 'SibSp', 'Parch', 'Fare'], dtype='object')
```

```
numerical_cols
```

```
Index(['PassengerId', 'Survived', 'Pclass', 'Age', 'SibSp', 'Parch', 'Fare'], dtype='object')
```

```
categorical_cols
```

```
Index(['Name', 'Sex', 'Ticket', 'Cabin', 'Embarked'], dtype='object')
```

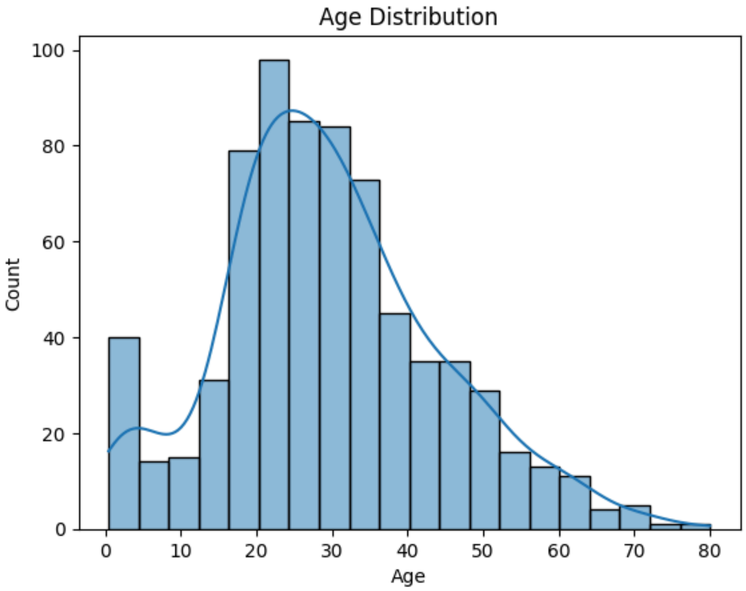
Steps of doing Univariate Analysis on Numerical columns (Age, Fare)

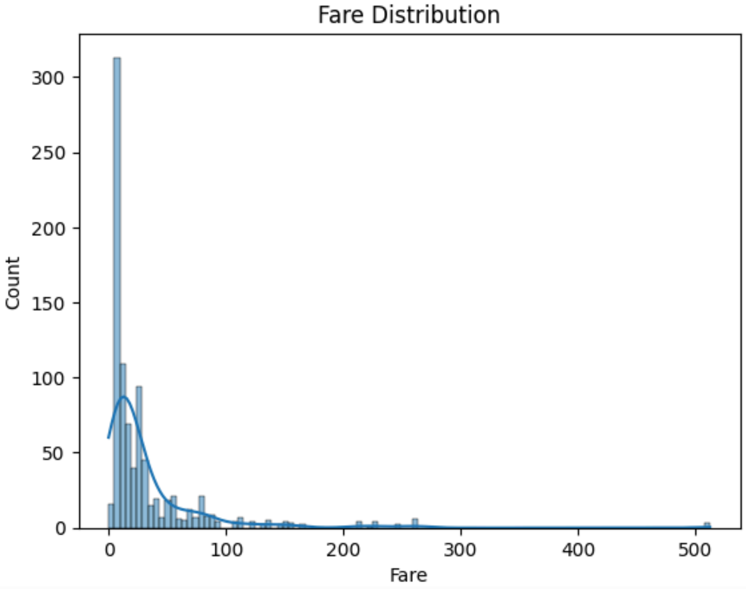
Descriptive Statistics (describe) Visualization (histogram, kde plot) Identifying Outliers (Box plot) Skewness (skew) Missing Values (isnull) Conclusion

```
# Descriptive Statistics (describe)
df[['Age', 'Fare']].describe()
```

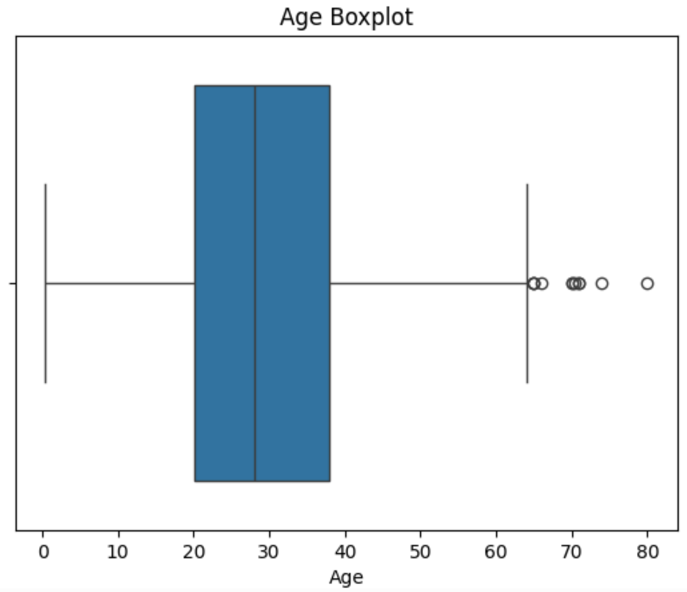
	Age	Fare
count	714.000000	891.000000
mean	29.699118	32.204208
std	14.526497	49.693429
min	0.420000	0.000000
25%	20.125000	7.910400
50%	28.000000	14.454200
75%	38.000000	31.000000
max	80.000000	512.329200

```
# Visualization (histogram, kde plot)
sns.histplot(df['Age'], kde=True)
plt.title('Age Distribution')
plt.show()

sns.histplot(df['Fare'], kde=True)
plt.title('Fare Distribution')
plt.show()
```

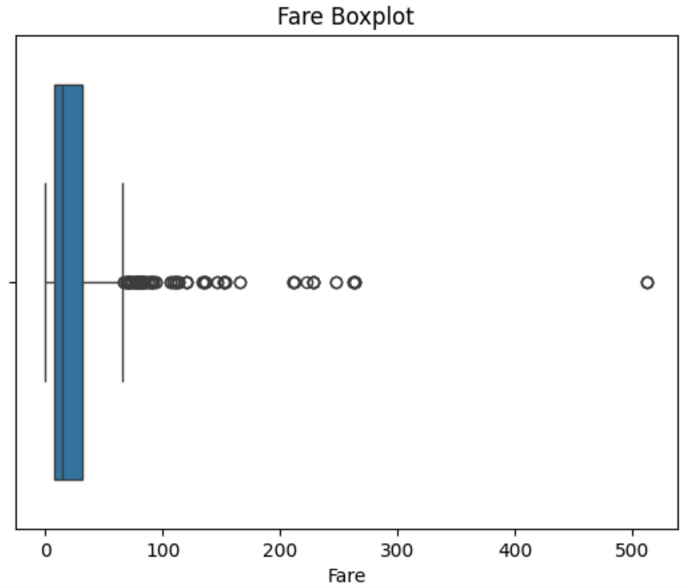




```
# Identifying Outliers (Box plot)
sns.boxplot(x=df['Age'])
plt.title('Age Boxplot')
plt.show()
```



```
# Identifying Outliers (Box plot)
sns.boxplot(x=df['Fare'])
plt.title('Fare Boxplot')
plt.show()
```



```
# Skewness (skew)
print("Skewness of Age:", df['Age'].skew())
print("Skewness of Fare:", df['Fare'].skew())
```

```
Skewness of Age: 0.38910778230082704
Skewness of Fare: 4.787316519674893
```

```
# Missing Values (isnull)
df['Age'].isna().sum()
```

```
np.int64(177)
```

```
df['Fare'].isna().sum()
```

```
np.int64(0)
```

```
# Conclusion
# Most Passengers lie in the age of 10 to 40
# Most passengers had a fare of less than 50
# The Fare distribution is highly skewed to the right.
```

Steps of doing Univariate Analysis on Categorical columns (Embarked, Sex)

Descriptive Statistics (value_count) Visualization (Bar plot,Pie Plot) Missing Values (isnull) Conclusion

```
# Descriptive Statistics (value_count)
df['Embarked'].value_counts()
```

	count
Embarked	
S	644
C	168
Q	77

dtype: int64

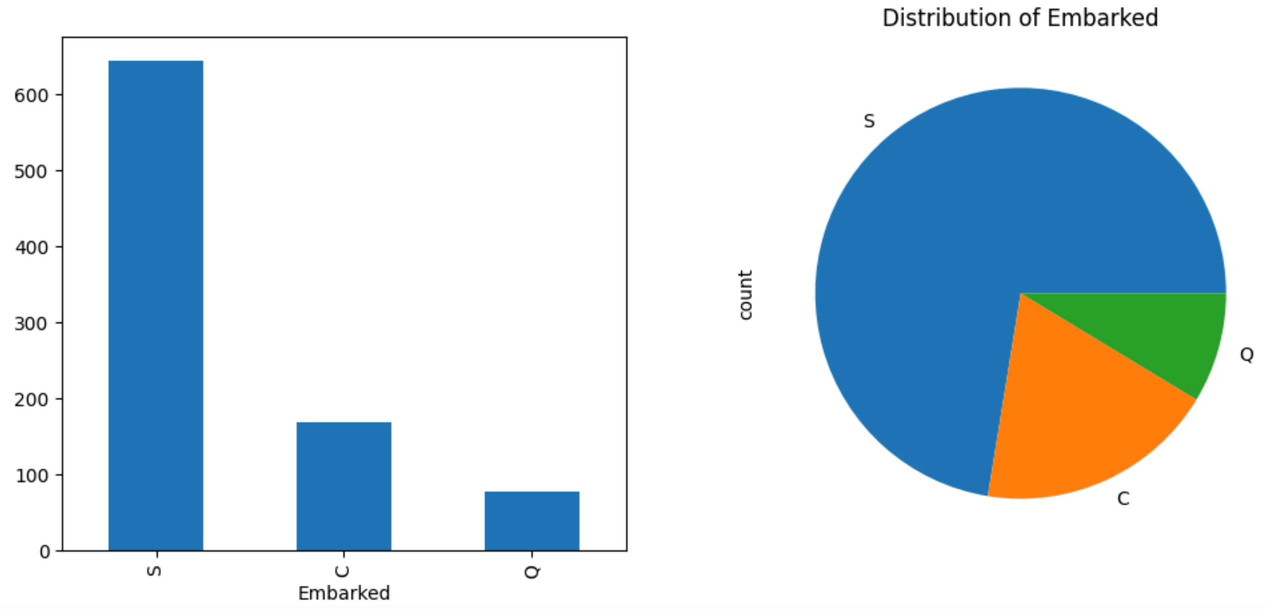
```
df['Sex'].value_counts()
```

	count
Sex	
male	577
female	314

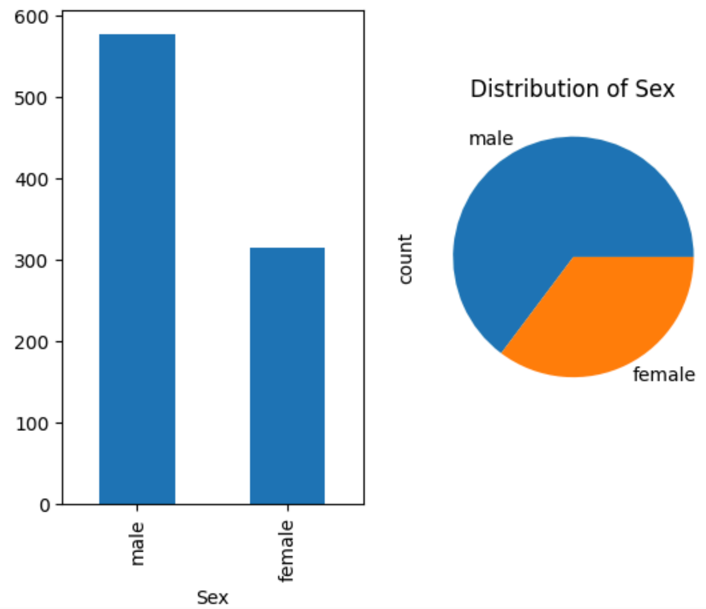
dtype: int64

```
# Visualization (Bar plot,Pie Plot)
plt.figure(figsize=(12, 5))
plt.subplot(1, 2, 1)
df['Embarked'].value_counts().plot(kind='bar')
plt.subplot(1, 2, 2)
df['Embarked'].value_counts().plot(kind='pie')
plt.title('Distribution of Embarked')
```

```
Text(0.5, 1.0, 'Distribution of Embarked')
```



```
plt.subplot(1, 2, 1)
df['Sex'].value_counts().plot(kind='bar')
plt.subplot(1, 2, 2)
df['Sex'].value_counts().plot(kind='pie')
plt.title('Distribution of Sex')
plt.show()
```



```
# Missing Values (isnull)
df['Embarked'].isna().sum()
```

```
np.int64(2)
```

```
df['Sex'].isna().sum()
```

```
np.int64(0)
```

```
# Conclusion:-
# Most passengers embarked from Southampton (S).
# There are more male passengers than female passengers.
```

Steps of doing Bivariate Analysis

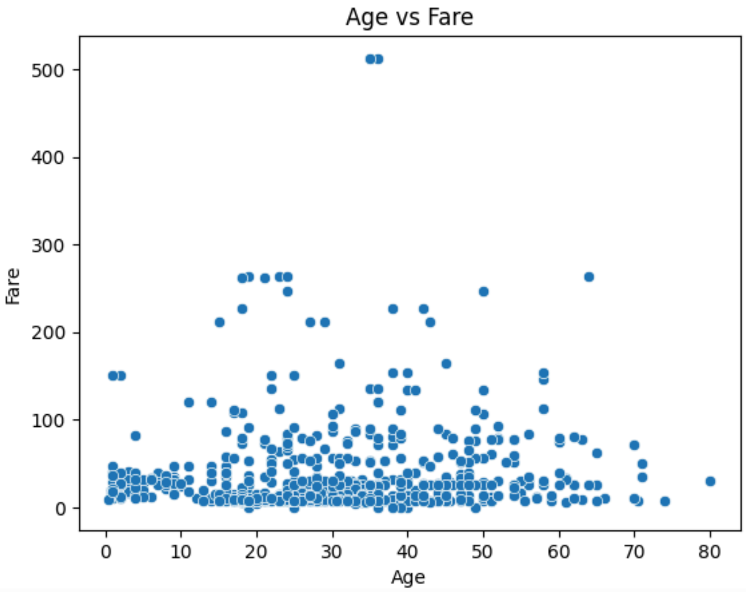
Select 2 cols Understand type of relationship Numerical – Numerical (Age and Fare) Scatterplot

Numerical - Categorical Kdeplot (Survived and Age)

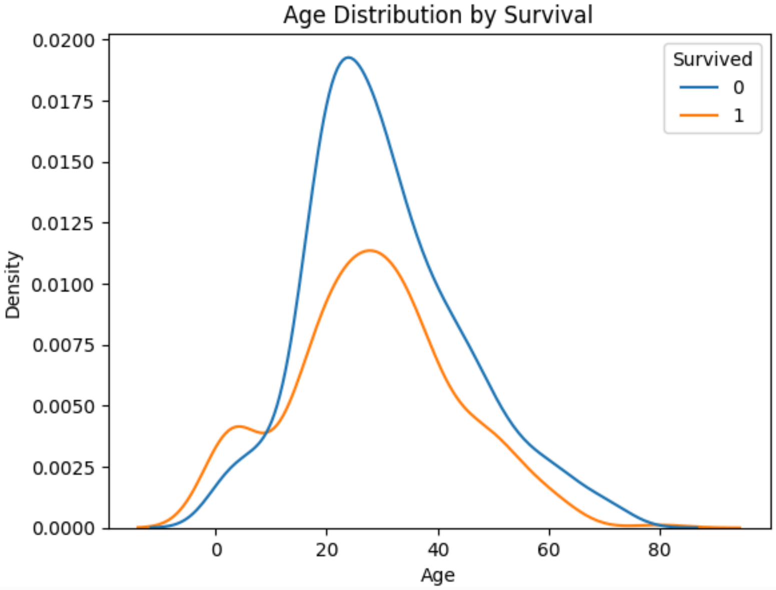
Categorical – Categorical Crosstab (Survived and Pclass / Survived and Sex /Survived and Embarked/Sex and Embarked/Pclass and Embarked)

Heatmap

```
# Numerical – Numerical (Age and Fare)
# Scatterplot
sns.scatterplot(x='Age', y='Fare', data=df)
plt.title('Age vs Fare')
plt.show()
```

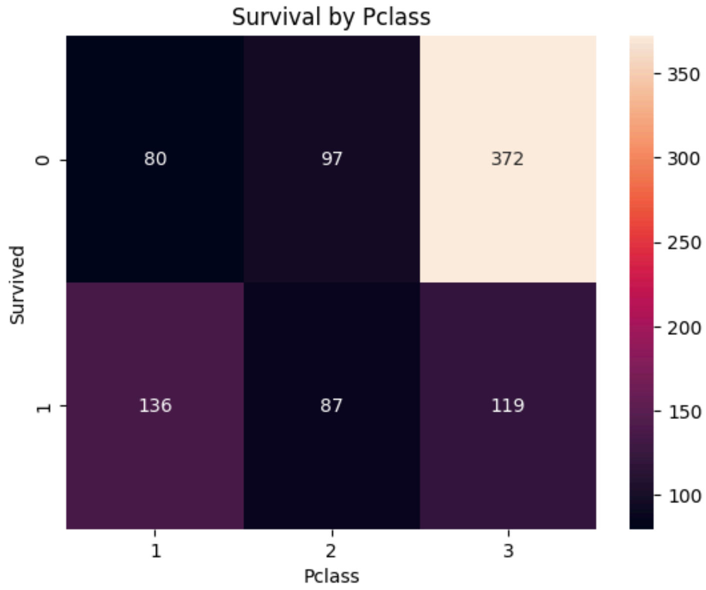


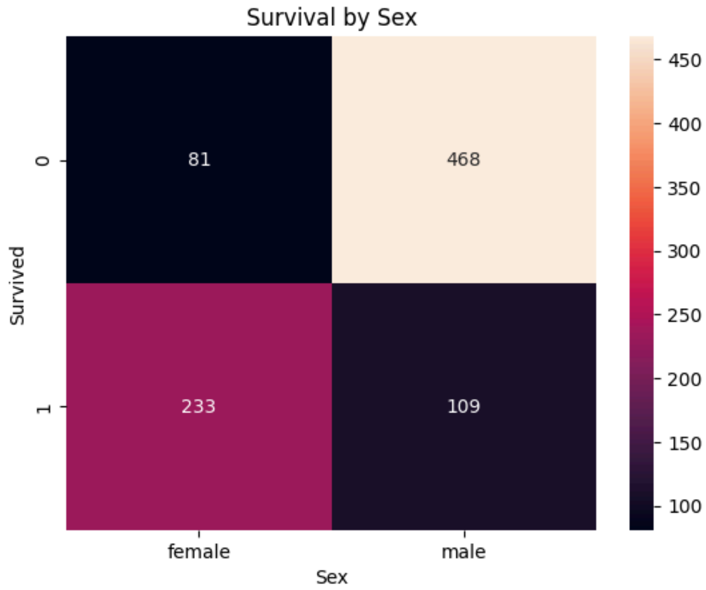
```
# Numerical - Categorical
# Kdeplot (Survived and Age)
sns.kdeplot(data=df, x='Age', hue='Survived')
plt.title('Age Distribution by Survival')
plt.show()
```

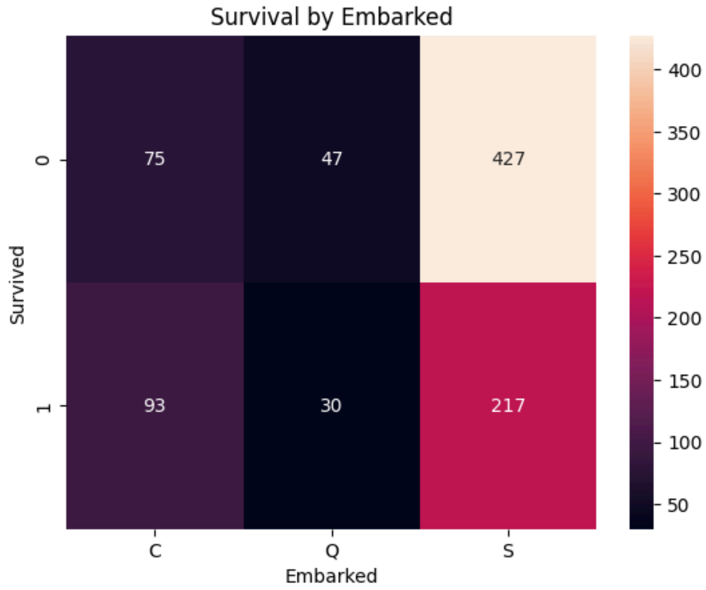


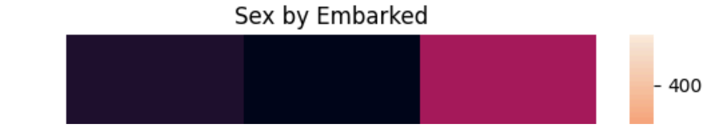
```
# Categorical - Categorical
# Crosstab (Survived and Pclass / Survived and Sex /Survived and Embarked/Sex and Embarked/Pclass and Embarked)
# Crosstab for Survived and Pclass
sns.heatmap(pd.crosstab(df['Survived'], df['Pclass']), annot=True, fmt='d')
plt.title('Survival by Pclass')
plt.show()
# fmt='d': Formats the numbers as integers (default is floating-point)

# Crosstab for Survived and Sex
sns.heatmap(pd.crosstab(df['Survived'], df['Sex']), annot=True, fmt='d')
plt.title('Survival by Sex')
plt.show()

# Crosstab for Survived and Embarked
sns.heatmap(pd.crosstab(df['Survived'], df['Embarked']), annot=True, fmt='d')
plt.title('Survival by Embarked')
plt.show()

# Crosstab for Sex and Embarked
```

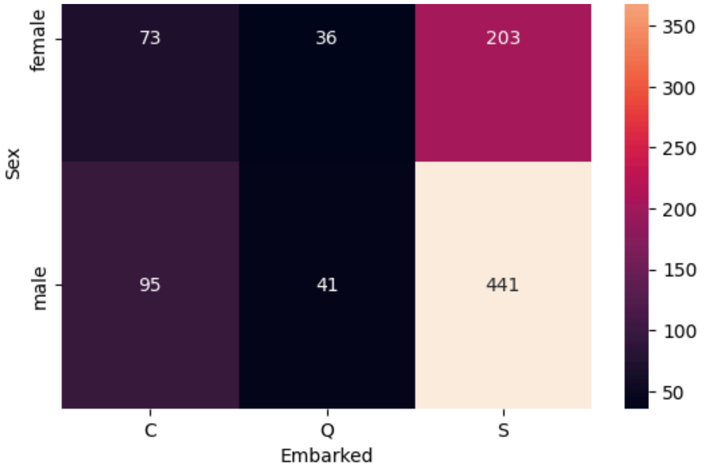
```
sns.heatmap(pd.crosstab(df['Sex'], df['Embarked']), annot=True, fmt='d')
plt.title('Sex by Embarked')
plt.show()

# Crosstab for Pclass and Embarked
sns.heatmap(pd.crosstab(df['Pclass'], df['Embarked']), annot=True, fmt='d')
plt.title('Pclass by Embarked')
plt.show()
```

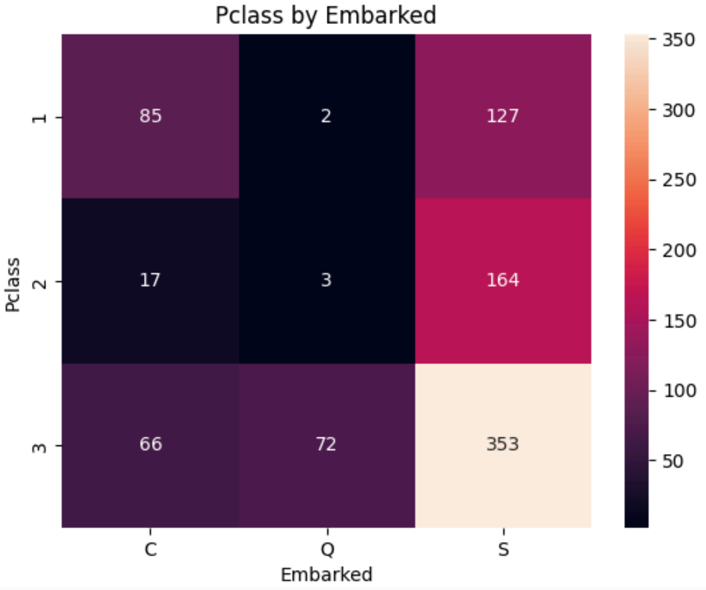

Survival by Pclass


Survival by Sex


Survival by Embarked


Sex by Embarked





```
# Find out value count for SibSp column.
# Find out Ticket of CA.2343.(It contains ticket of group -passenger, parch, sibsp .)
# Find out individual fare. Also plot box plot.

print("SibSp Value Counts:")
print(df['SibSp'].value_counts())

## Find out Ticket of CA.2343
ticket_ca2343 = df[df['Ticket'] == 'CA. 2343']
print("Passengers with Ticket CA.2343:")
print(ticket_ca2343)

## Find out individual fare
df['FamilySize'] = df['SibSp'] + df['Parch'] + 1
df['IndividualFare'] = df['Fare'] / (df['FamilySize'])
print("Individual Fare Distribution:")
```