TASK 5

I have used Titanic.csv dataset for this Task.

Input:

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

# Loading the Titanic dataset
df = pd.read_csv('Titanic.csv')

# Displaying first few rows
df.head()

# Basic information
print(df.info())

# Statistical description
print(df.describe(include='all'))

# Check data types
print(df.dtypes)

# Find missing values
print(df.isnull().sum())
```

Output:

```
<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
None
       PassengerId    Survived      Pclass              Name   Sex  \
count    891.000000  891.000000  891.000000               891   891
unique          NaN         NaN         NaN               891     2
top             NaN         NaN         NaN  Braund, Mr. Owen Harris  male
freq            NaN         NaN         NaN                 1   577
mean     446.000000    0.383838    2.308642               NaN   NaN
std      257.353842    0.486592    0.836071               NaN   NaN
```

min	1.000000	0.000000	1.000000	NaN	NaN	
25%	223.500000	0.000000	2.000000	NaN	NaN	
50%	446.000000	0.000000	3.000000	NaN	NaN	
75%	668.500000	1.000000	3.000000	NaN	NaN	
max	891.000000	1.000000	3.000000	NaN	NaN	

	Age	SibSp	Parch	Ticket	Fare	Cabin \
count	714.000000	891.000000	891.000000	891	891.000000	204
unique	NaN	NaN	NaN	681	NaN	147
top	NaN	NaN	NaN	347082	NaN	B96 B98
freq	NaN	NaN	NaN	7	NaN	4
mean	29.699118	0.523008	0.381594	NaN	32.204208	NaN
std	14.526497	1.102743	0.806057	NaN	49.693429	NaN
min	0.420000	0.000000	0.000000	NaN	0.000000	NaN
25%	20.125000	0.000000	0.000000	NaN	7.910400	NaN
50%	28.000000	0.000000	0.000000	NaN	14.454200	NaN
75%	38.000000	1.000000	0.000000	NaN	31.000000	NaN
max	80.000000	8.000000	6.000000	NaN	512.329200	NaN

	Embarked
count	889
unique	3
top	S
freq	644
mean	NaN
std	NaN
min	NaN
25%	NaN

50% NaN

75% NaN

max NaN

PassengerId int64

Survived int64

Pclass int64

Name object

Sex object

Age float64

SibSp int64

Parch int64

Ticket object

Fare float64

Cabin object

Embarked object

dtype: object

PassengerId 0

Survived 0

Pclass 0

Name 0

Sex 0

Age 177

SibSp 0

Parch 0

Ticket 0

Fare 0

Cabin 687

Embarked 2

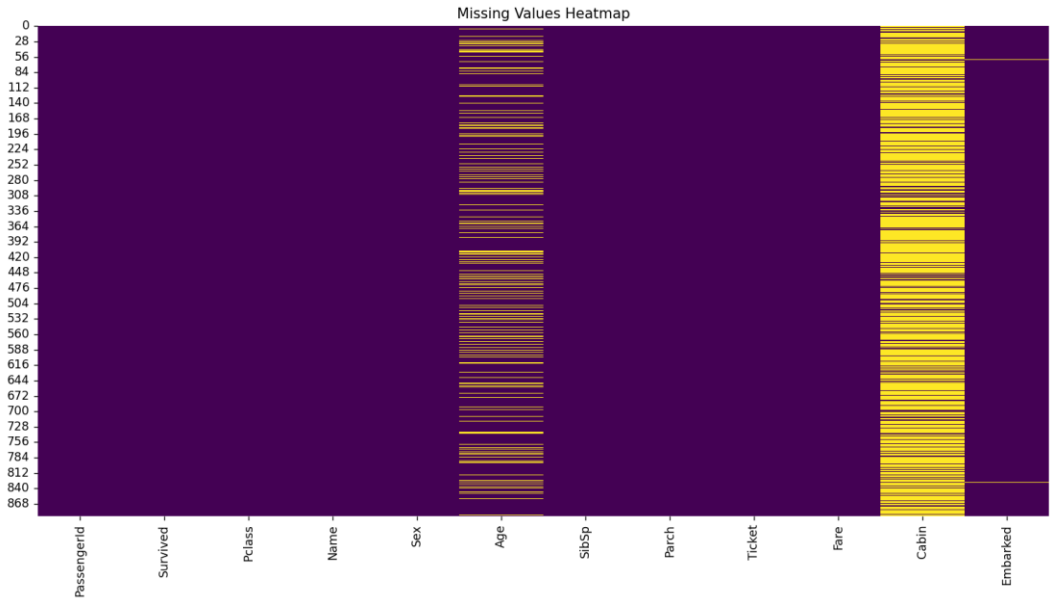
dtype: int64

Code is been continued but for better viewing the code I have divided it in parts .

Input:

```
# Visualize missing data
plt.figure(figsize=(10,6))
sns.heatmap(df.isnull(), cbar=False, cmap='viridis')
plt.title('Missing Values Heatmap')
plt.show()
```

Output:



Input:

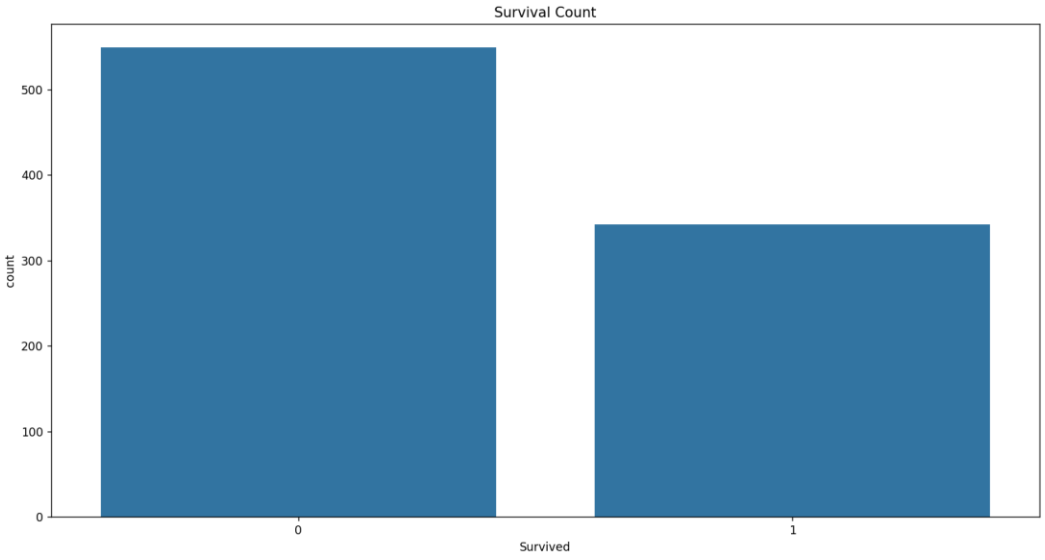
```
# Count of survival
sns.countplot(x='Survived', data=df)
plt.title('Survival Count')
plt.show()
```

Percentage

survival_rate = df['Survived'].mean() * 100

print(f"Survival Rate: {survival_rate:.2f}%")

Output:



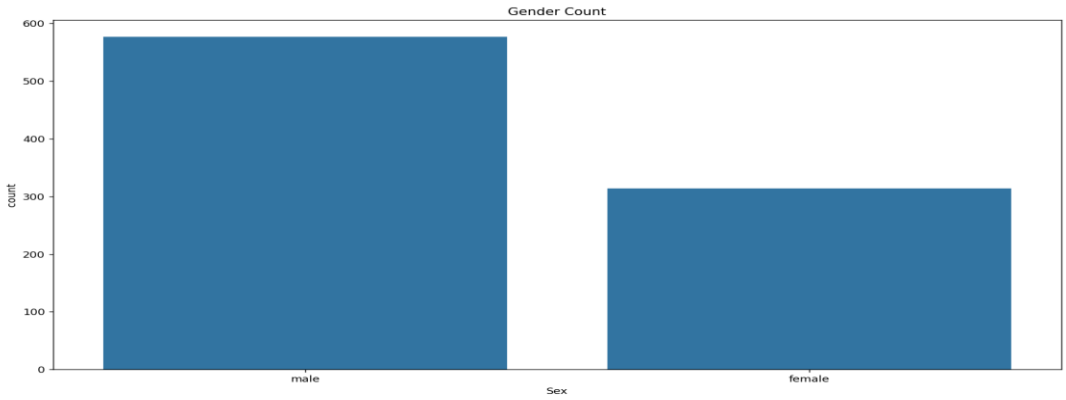
Input:

sns.countplot(x='Sex', data=df)

plt.title('Gender Count')

plt.show()

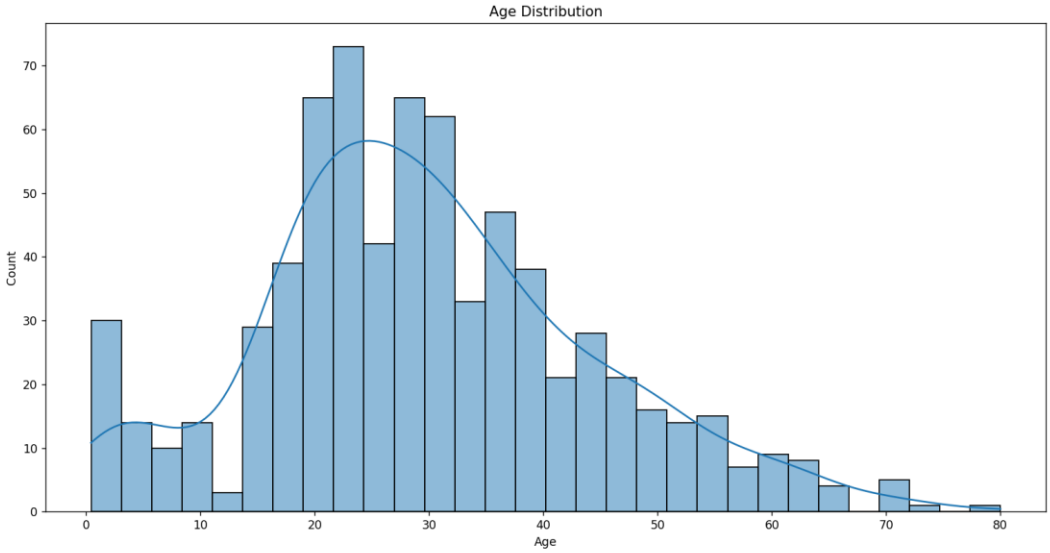
Output:



Input:

```
plt.figure(figsize=(10,6))
sns.histplot(df['Age'].dropna(), bins=30, kde=True)
plt.title('Age Distribution')
plt.show()
```

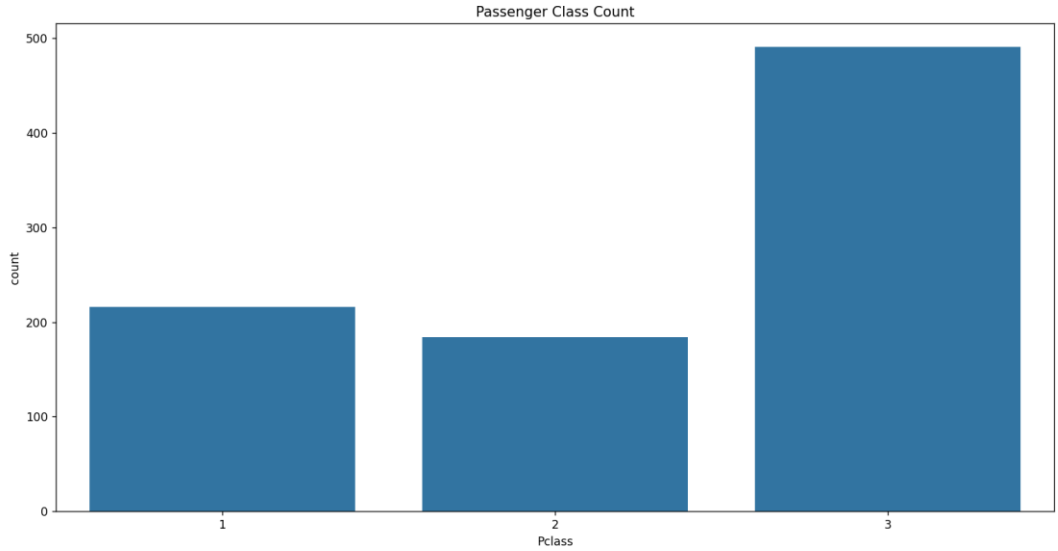
Output:



Input:

```
sns.countplot(x='Pclass', data=df)
plt.title('Passenger Class Count')
plt.show()
```

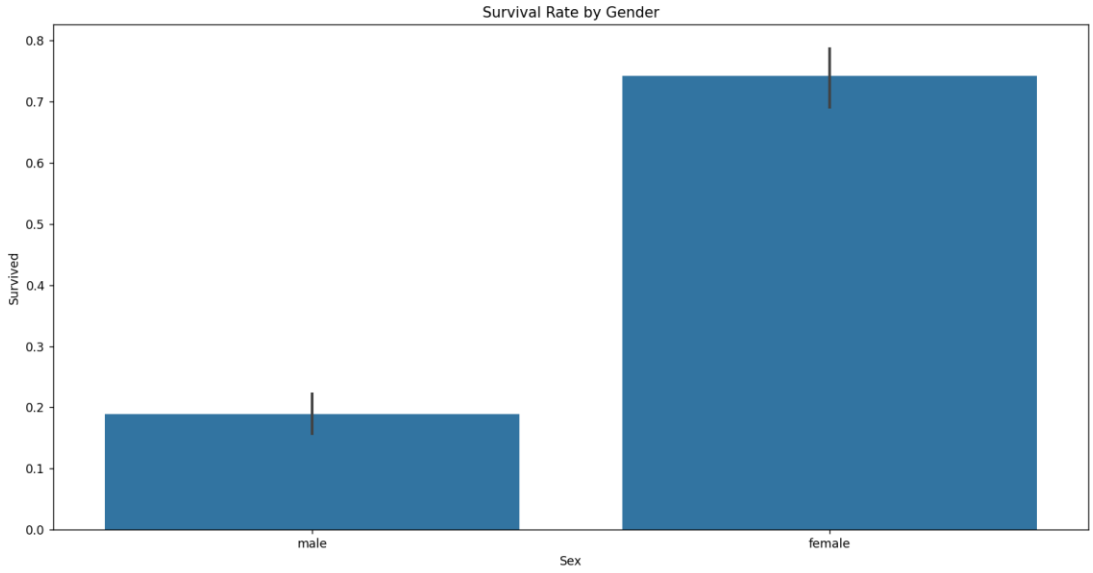
Output:



Input:

```
sns.barplot(x='Sex', y='Survived', data=df)
plt.title('Survival Rate by Gender')
plt.show()
```

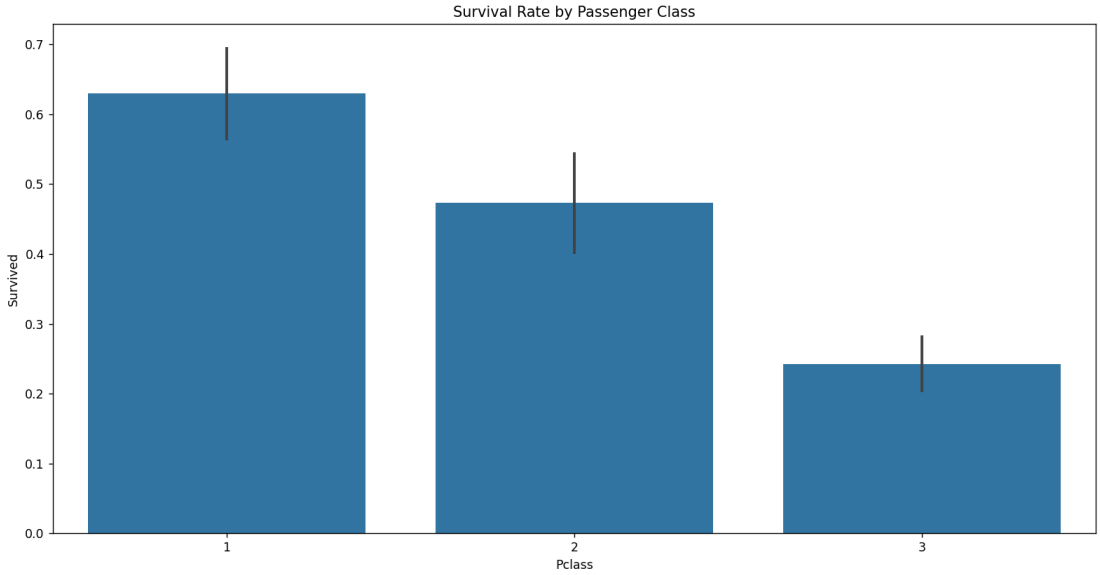
Output:



Input:

```
sns.barplot(x='Pclass', y='Survived', data=df)
plt.title('Survival Rate by Passenger Class')
plt.show()
```

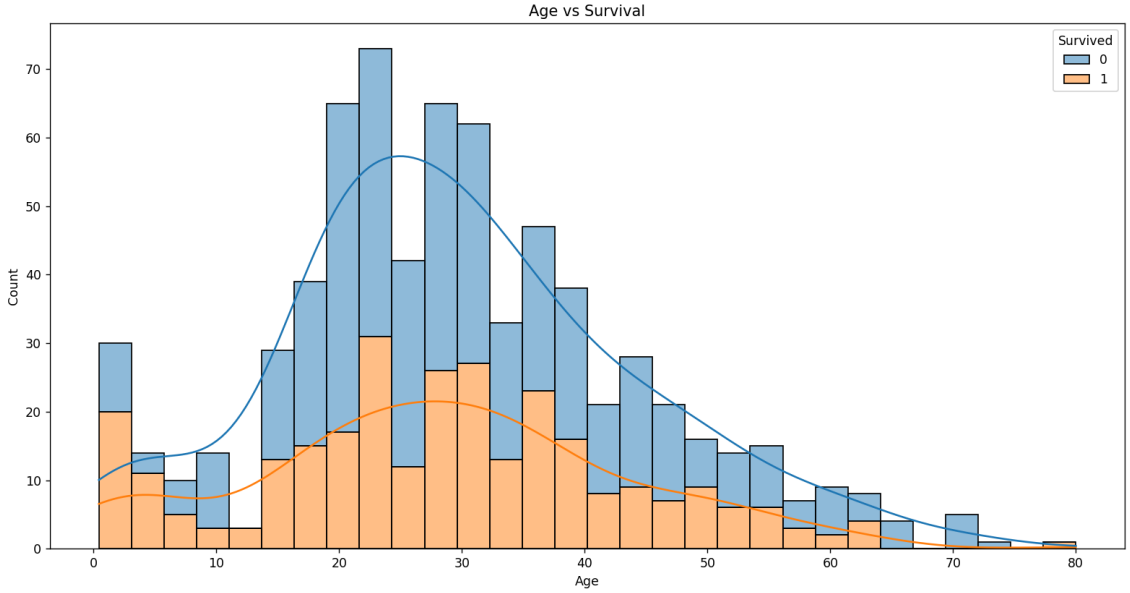
Output:



Input:

```
plt.figure(figsize=(10,6))
sns.histplot(data=df, x='Age', hue='Survived', bins=30, kde=True, multiple="stack")
plt.title('Age vs Survival')
plt.show()
```

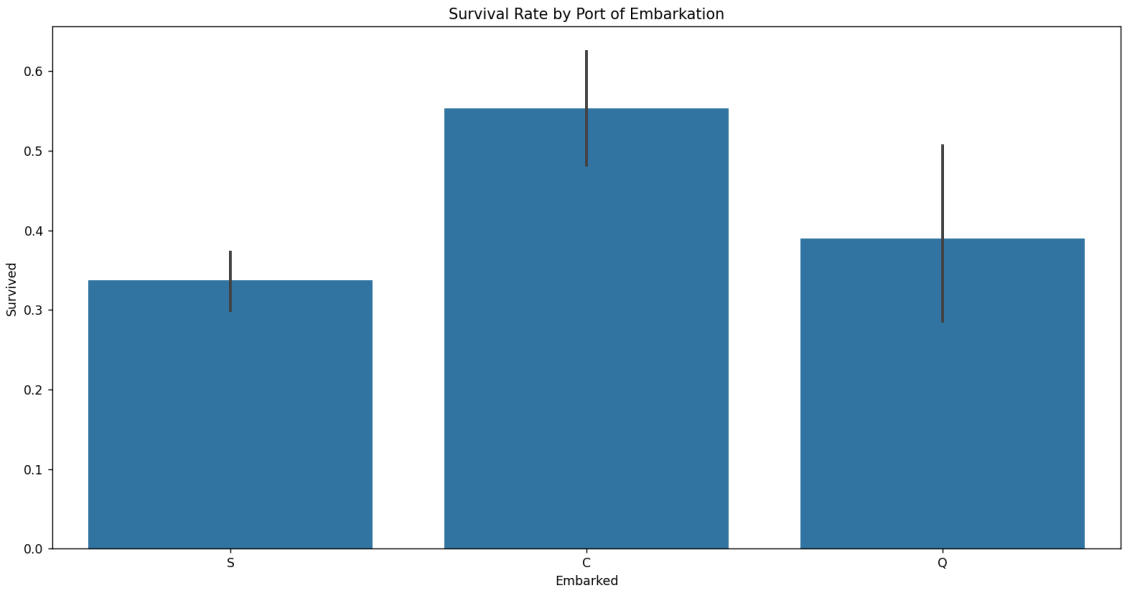
Output:



Input:

```
sns.barplot(x='Embarked', y='Survived', data=df)
plt.title('Survival Rate by Port of Embarkation')
plt.show()
```

Output:



Input:

```
# Create new feature
df['FamilySize'] = df['SibSp'] + df['Parch']
sns.barplot(x='FamilySize', y='Survived', data=df)
plt.title('Family Size vs Survival')
plt.show()
```

Output: