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

train_df=pd.read_csv('train.csv')
test_df=pd.read_csv('test.csv')
gender_submission_df=pd.read_csv('gender_submission.csv')

#This will display first 5 rows from the train.csv dataset
train_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
				Futrelle, Mrs. Jacques Heath								

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

```
#This will display column name,number of rows in each column(count) and the datatype of each value in the columns.
train_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     891 non-null    object 
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

```
#this will display the count,mean,mode,standard devaition,min,max,quartiles of all numerical columns in the DataFrame.
train_df.describe()
```

	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

```
#This will give the number of missing values in every column
train_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     0

dtype: int64
```

```
#As there are missing values in Age and Cabin I will fill the misiing values in Age column with median.

train_df['Age']=train_df['Age'].fillna(train_df['Age'].median())
```

```
#Cabin column has over 77% of misiing values,so this column is almost unusable in its current state.So we will delete it.

train_df.drop('Cabin',axis=1,inplace=True)
```

```
#Missing data is filled and now there are no missing values
train_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
```

```
#Gives total count of males and females separately
train_df['Sex'].value_counts()
```

```
         count
   Sex
  male     577
female     314

dtype: int64
```

```
#Shows the number of passengers in each passenger class (1st, 2nd, and 3rd), ordered by count.
train_df['Pclass'].value_counts()
```

```
        count
Pclass
  3      491
  1      216
  2      184

dtype: int64
```

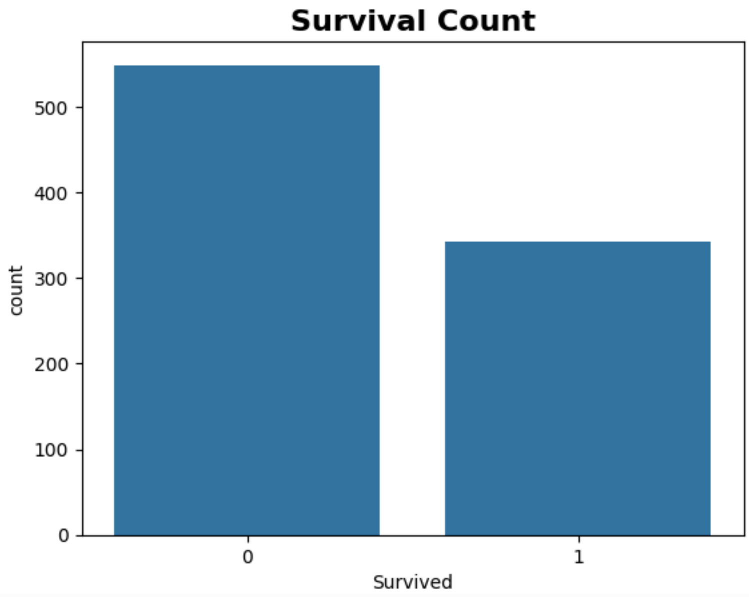
```
#Shows the average survival rate for each gender (i.e., how likely males vs. females were to survive).
train_df.groupby('Sex')['Survived'].mean()
#Shows the average survival rate for each combination of passenger class and gender (e.g., 1st class females, 3rd class males, etc.).
train_df.groupby(['Pclass', 'Sex'])['Survived'].mean()
```

```
                   Survived
Pclass    Sex
  1     female     0.968085
          male     0.368852
  2     female     0.921053
          male     0.157407
  3     female     0.500000
          male     0.135447

dtype: float64
```

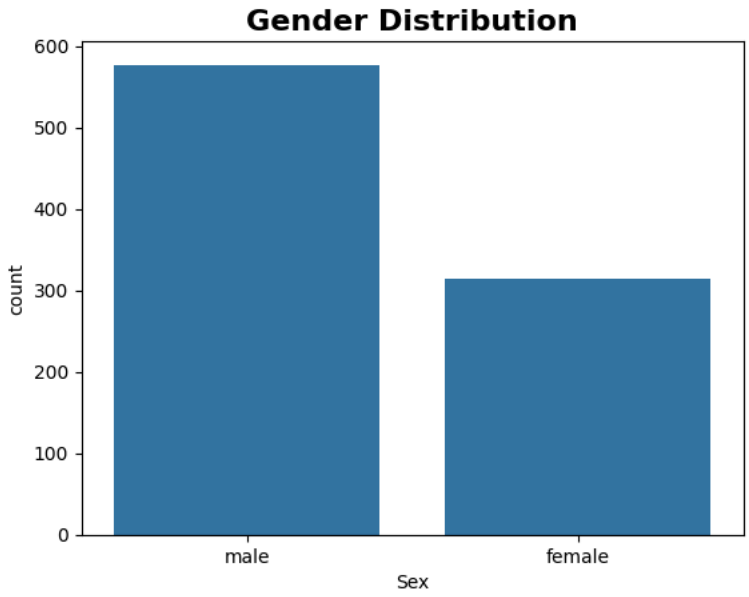
```
#Displays the count of passengers who boarded from each port (S, C, Q), helping you see which embarkation point was most common.
train_df['Embarked'].value_counts()
```

```
           count
Embarked
   S        645
   C        168
   Q         77
   SS         1

dtype: int64
```

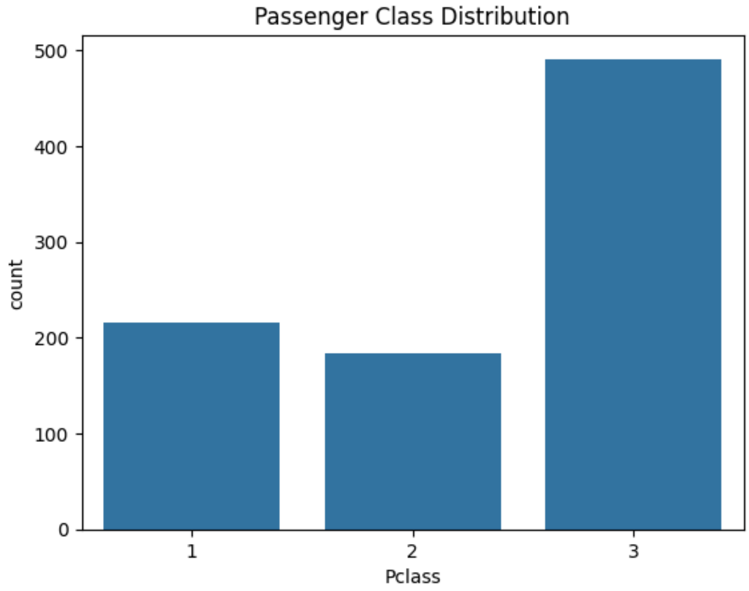
```
#Plots the number of passengers who survived (1) and who didn’t (0).
sns.countplot(x='Survived',data=train_df)
plt.title('Survival Count',fontweight='bold',fontsize=16)
plt.show()
```



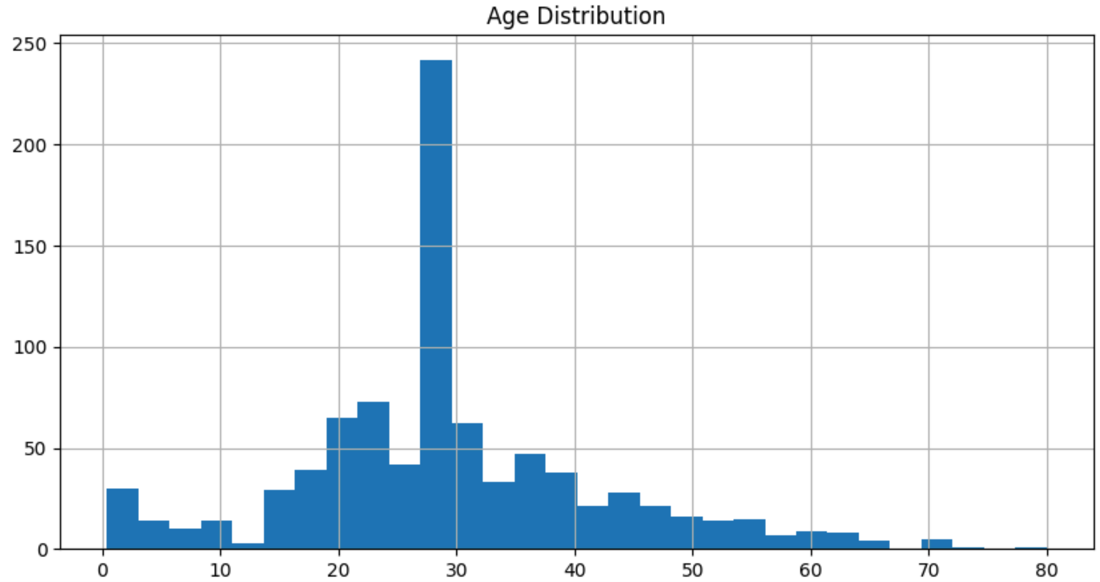
```
#Creates a bar chart showing the number of male and female passengers on the Titanic.
sns.countplot(x='Sex',data=train_df)
plt.title("Gender Distribution",fontweight='bold',fontsize=16)
plt.show()
```



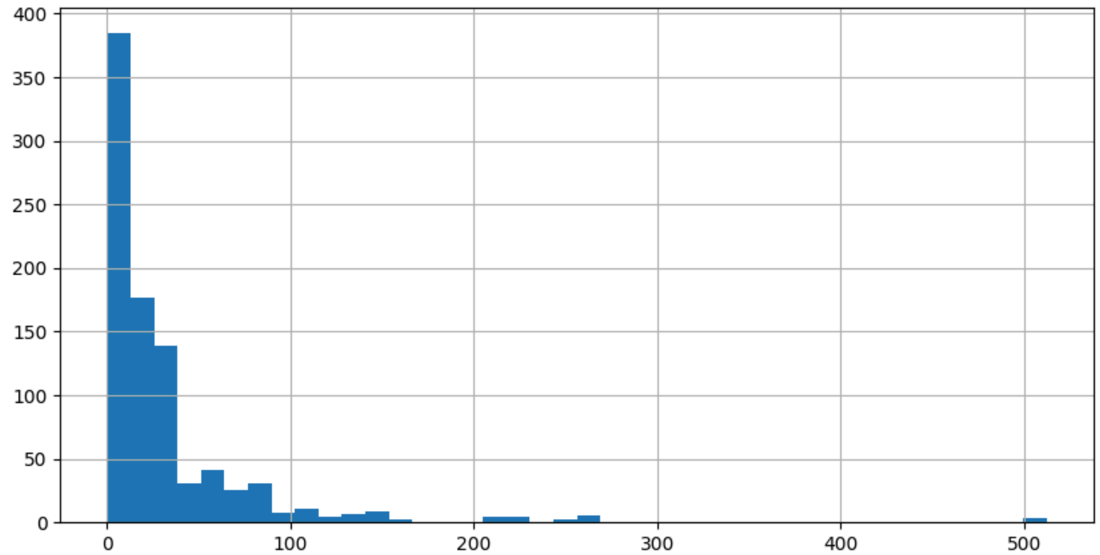
```
#Plots the number of passengers in each passenger class (1st, 2nd, 3rd), showing the class distribution on the Titanic
sns.countplot(x='Pclass', data=train_df)
plt.title("Passenger Class Distribution")
plt.show()
```



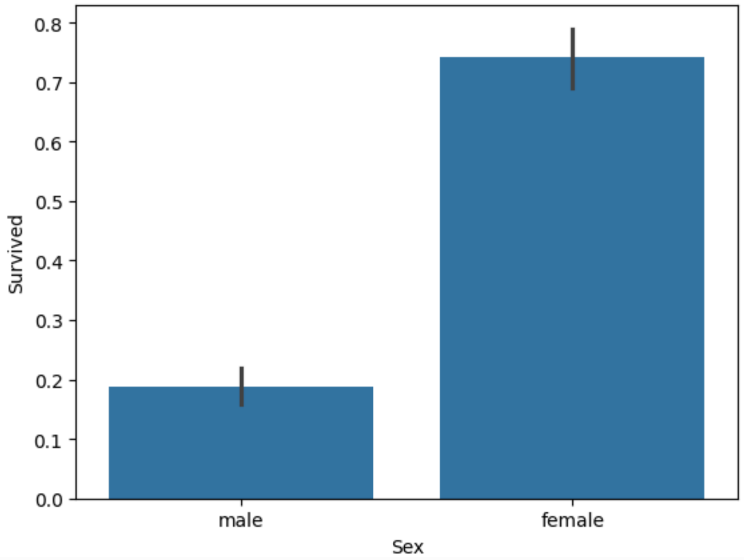
```
#This will display a histogram of the "Age" column from the `train_df` DataFrame
train_df['Age'].hist(bins=30,figsize=(10,5))
plt.title("Age Distribution")
plt.show()
```



```
## This will display a histogram of the "Fare" column from the `train_df` DataFrame with 40 bins and a figure size of 10x5 inches.
train_df['Fare'].hist(bins=40, figsize=(10,5))
plt.show()
```

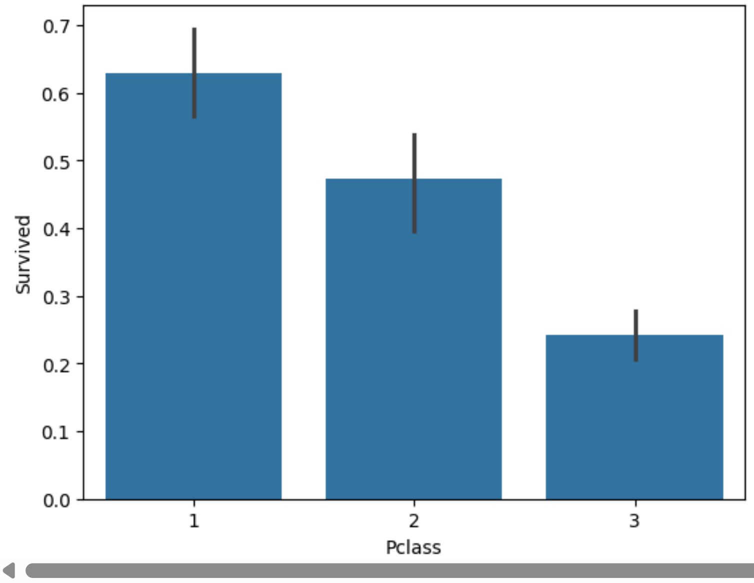


```
## This will create a bar plot showing the relationship between 'Sex' and 'Survived' in the `train_df` DataFrame.
sns.barplot(x='Sex',y='Survived',data=train_df)
plt.show()
```



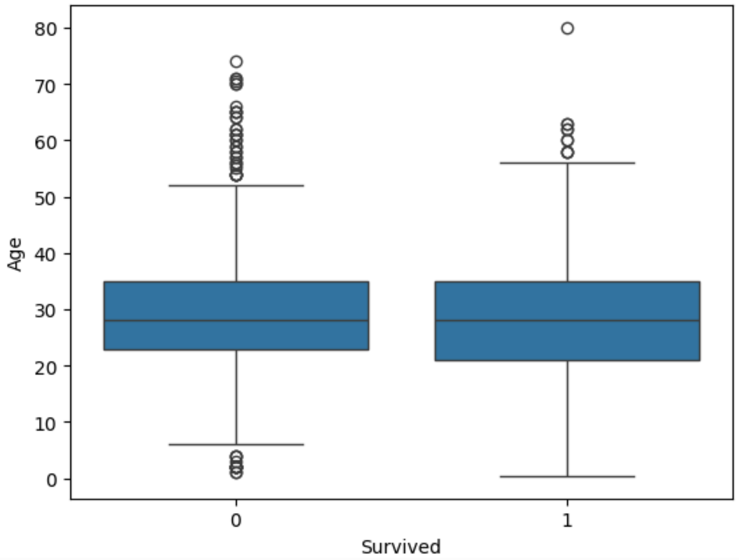
```
#This will create a bar plot showing the relationship between 'Pclass' and 'Survived' in the `train_df` DataFrame.
sns.barplot(x='Pclass', y='Survived', data=train_df)
```

```
<Axes: xlabel='Pclass', ylabel='Survived'>
```



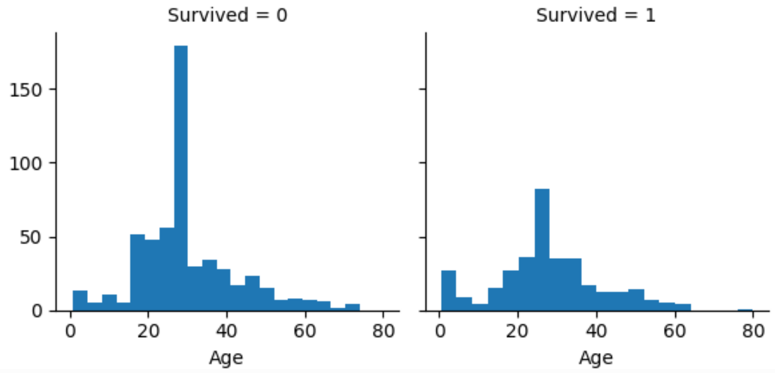
```
# This will create a box plot showing the distribution of 'Age' for each survival status ('Survived') in the `train_df` DataFrame.

sns.boxplot(x='Survived',y='Age',data=train_df)
```

```
<Axes: xlabel='Survived', ylabel='Age'>
```



```
# This will create a FacetGrid that shows the distribution of 'Age' for each survival status ('Survived'), with 20 bins in the histogra
g = sns.FacetGrid(train_df, col="Survived")
g.map(plt.hist, "Age", bins=20)
```

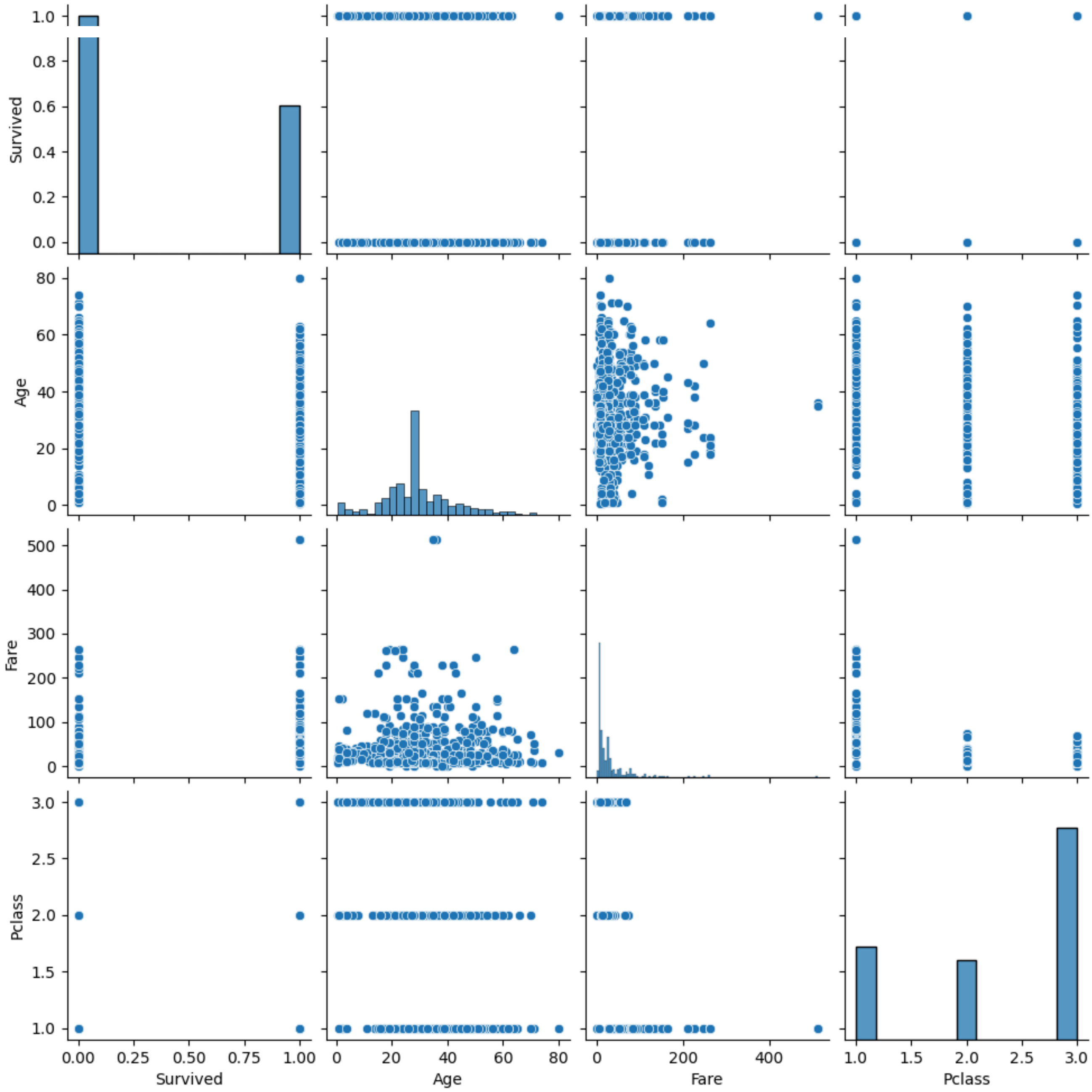
```
<seaborn.axisgrid.FacetGrid at 0x7af89358c110>
```



```
# This will create a pair plot for the columns 'Survived', 'Age', 'Fare', and 'Pclass' from the `train_df` DataFrame, showing pairwise
sns.pairplot(train_df[['Survived','Age','Fare','Pclass']])
```

```
plt.figure(figsize=(10,6))
```

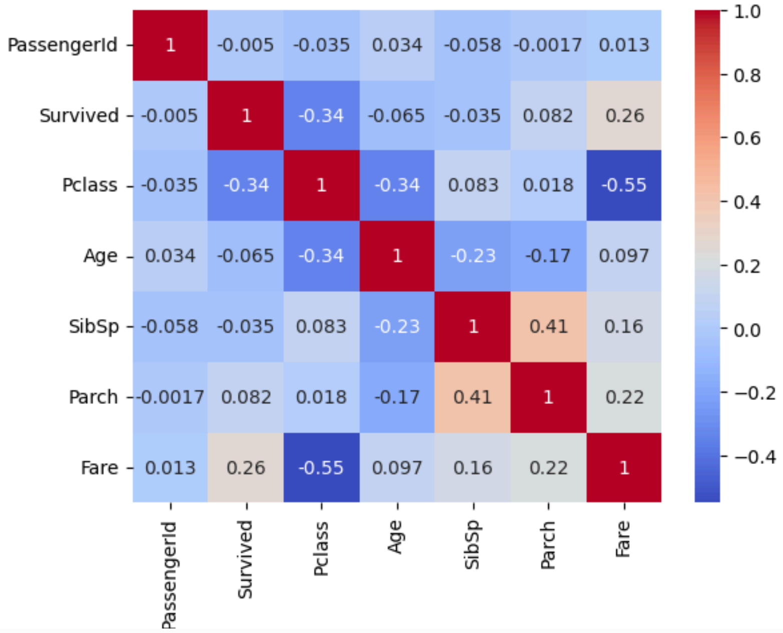
```
<Figure size 1000x600 with 0 Axes>
```



```
<Figure size 1000x600 with 0 Axes>
```

```
## This will create a heatmap of the correlation matrix for all numerical columns in the `train_df` DataFrame, with annotations and a co
sns.heatmap(train_df.select_dtypes(include='number').corr(), annot=True, cmap='coolwarm')
```

```
<Axes: >
```



```
#Display first 5 rows
gender_submission_df.head()
```

	PassengerId	Survived
0	892	0
1	893	1
2	894	0
3	895	0
4	896	1

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

```
##This will display column name,number of rows in each column(count) and the datatype of each value in the columns.
gender_submission_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 2 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  418 non-null    int64
 1   Survived     418 non-null    int64
dtypes: int64(2)
memory usage: 6.7 KB
```

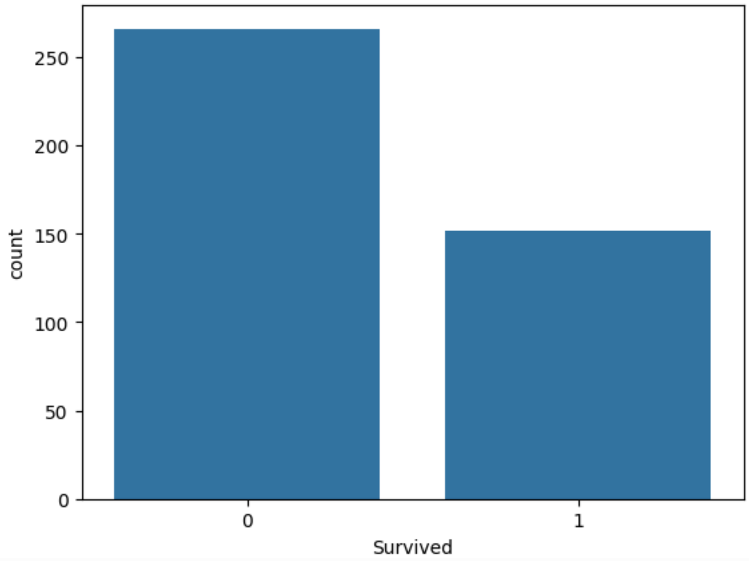
```
#this will display the count,mean,mode,standard devaition,min,max,quartiles of all numerical columns in the DataFrame.
gender_submission_df.describe()
```

	PassengerId	Survived
count	418.000000	418.000000
mean	1100.500000	0.363636
std	120.810458	0.481622
min	892.000000	0.000000
25%	996.250000	0.000000
50%	1100.500000	0.000000
75%	1204.750000	1.000000
max	1309.000000	1.000000

```
# This will display the count of unique values in the 'Survived' column of the `gender_submission_df` DataFrame,
# showing how many passengers survived and how many did not in the gender_submission.csv file.
gender_submission_df['Survived'].value_counts()
```

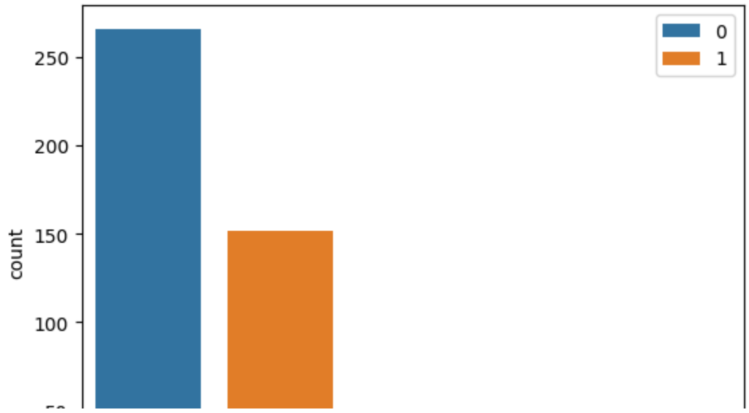
```
          count
Survived
       0    266
       1    152

dtype: int64
```

```
sns.countplot(x='Survived', data=gender_submission_df)
plt.show()
```



```
merged_df = pd.merge(test_df, gender_submission_df, on='PassengerId')
```

```
sns.countplot(x='Sex', hue='Survived', data=merged_df)
sns.barplot(x='Pclass', y='Survived', data=merged_df)
plt.show()
```


count