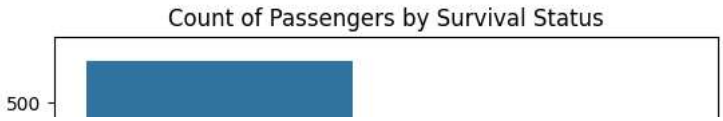
```
#/*Perform Data preprocessing on Titanic dataset
#1.Data Collection.
#   Please download the dataset from
#   https://www.kaggle.com/datasets/yasserh/titanic-dataset

#2.Data Preprocessing
#    o  Import the Libraries.
#    o  Importing the dataset.
#    o  Checking for Null Values.
#    o  Data Visualization.
#    o  Outlier Detection
#    o  Splitting Dependent and Independent variables
#    o  Perform Encoding
#    o  Feature Scaling.
#    o  Splitting Data into Train and Test


import pandas as pd
import numpy as np
import matplotlib.pyplot as plt


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


null_values = df.isnull().sum()
print(null_values)
```

```
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
```

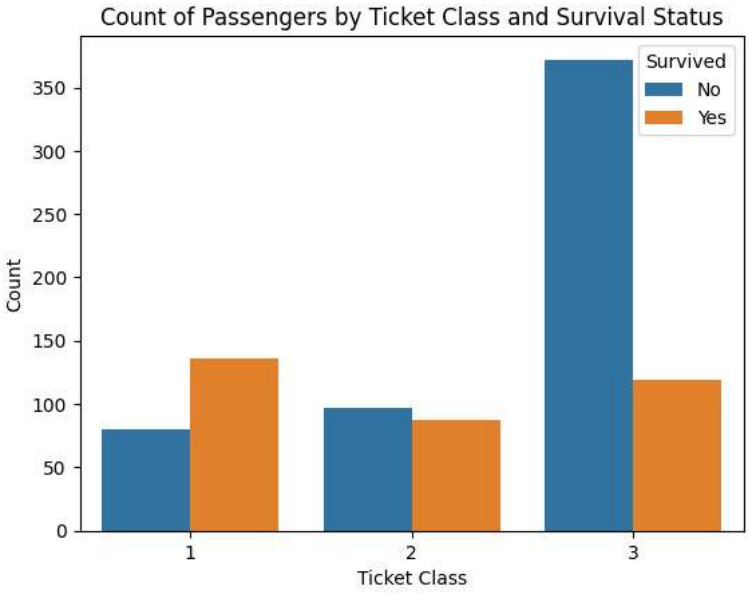
```
# Check for null values in the "Age" column
age_null_values = df['Age'].isnull().sum()
print("Number of null values in the 'Age' column:", age_null_values)
```

```
Number of null values in the 'Age' column: 177
```

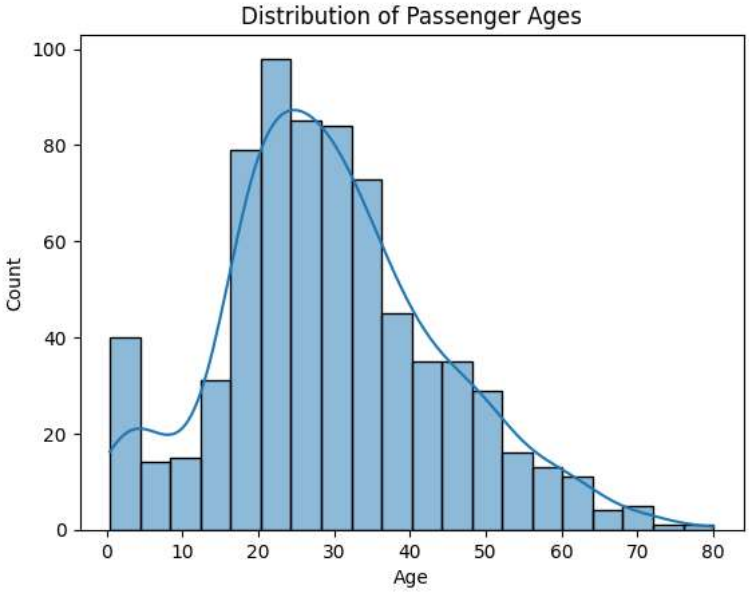
```
#countplot of passengers by survival status

import seaborn as sns
sns.countplot(data=df, x='Survived')
plt.title('Count of Passengers by Survival Status')
plt.xlabel('Survived')
plt.ylabel('Count')
plt.show()
```




```
#Countplot of Passengers by Pclass (Ticket Class)

sns.countplot(data=df, x='Pclass', hue='Survived')
plt.title('Count of Passengers by Ticket Class and Survival Status')
plt.xlabel('Ticket Class')
plt.ylabel('Count')
plt.legend(title='Survived', labels=['No', 'Yes'])
plt.show()
```

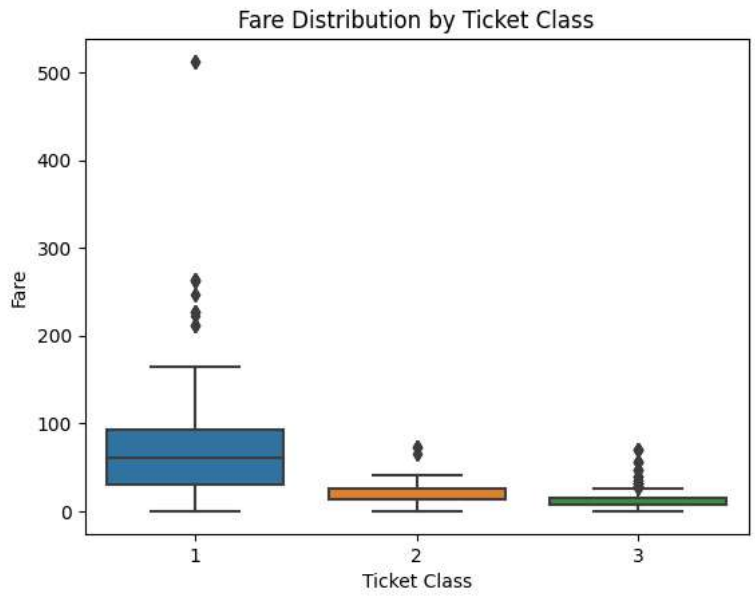



```
#Histogram of Passenger Ages

sns.histplot(data=df, x='Age', bins=20, kde=True)
plt.title('Distribution of Passenger Ages')
plt.xlabel('Age')
plt.ylabel('Count')
plt.show()
```

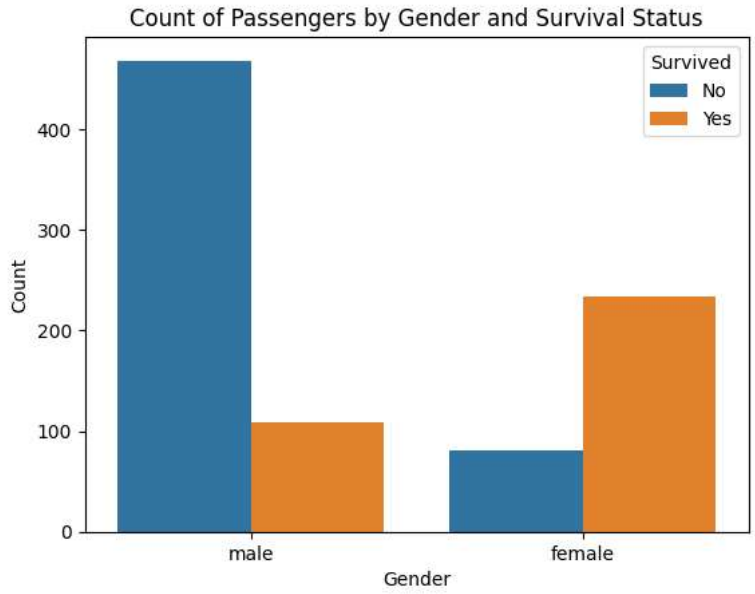



```
#Boxplot of Fare by Ticket Class

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

```
plt.title('Fare Distribution by Ticket Class')
plt.xlabel('Ticket Class')
plt.ylabel('Fare')
plt.show()
```

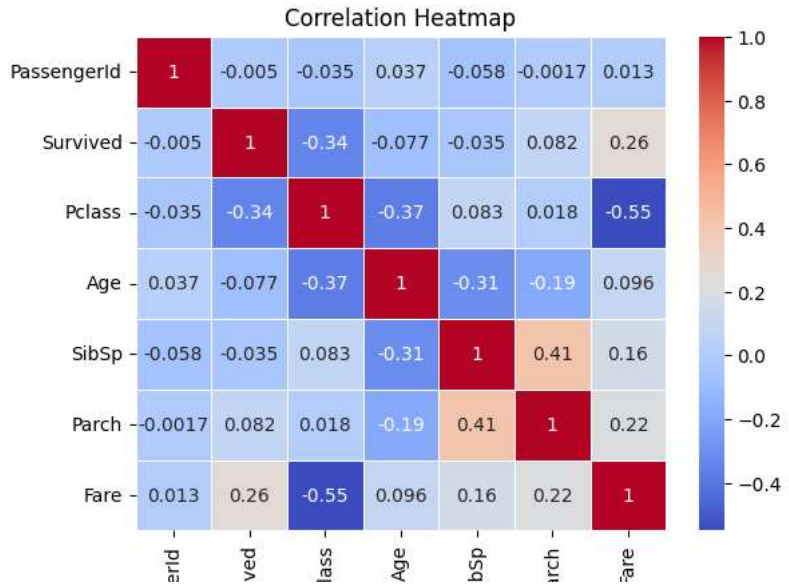



```
#Countplot of Passengers by Gender and Survival Status

sns.countplot(data=df, x='Sex', hue='Survived')
plt.title('Count of Passengers by Gender and Survival Status')
plt.xlabel('Gender')
plt.ylabel('Count')
plt.legend(title='Survived', labels=['No', 'Yes'])
plt.show()
```

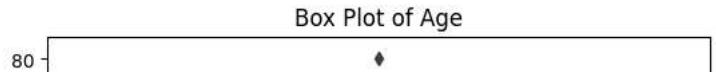



```
#Correlation Heatmap

corr_matrix = df.corr()
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', linewidths=0.5)
plt.title('Correlation Heatmap')
plt.show()
```

```
<ipython-input-10-ac3d71098086>:3: FutureWarning: The default value of numeric_only in DataFrame.corr i
  corr_matrix = df.corr()
```




```
# Box plot for Age
sns.boxplot(data=df, y='Age')
plt.title('Box Plot of Age')
plt.show()

# Box plot for Fare
sns.boxplot(data=df, y='Fare')
plt.title('Box Plot of Fare')
plt.show()
```




```
# Calculate the IQR for Age
Q1 = df['Age'].quantile(0.25)
Q3 = df['Age'].quantile(0.75)
IQR = Q3 - Q1

# Define the lower and upper bounds for Age
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

# Find outliers in Age
age_outliers = df[(df['Age'] < lower_bound) | (df['Age'] > upper_bound)]
age_outliers
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embark
33	34	0	2	Wheadon, Mr. Edward H	male	66.0	0	0	C.A. 24579	10.5000	NaN	
54	55	0	1	Ostby, Mr. Engelhart Cornelius	male	65.0	0	1	113509	61.9792	B30	
96	97	0	1	Goldschmidt, Mr. George B	male	71.0	0	0	PC 17754	34.6542	A5	
116	117	0	3	Connors, Mr. Patrick	male	70.5	0	0	370369	7.7500	NaN	
280	281	0	3	Duane, Mr. Frank	male	65.0	0	0	336439	7.7500	NaN	
456	457	0	1	Millet, Mr. Francis Davis	male	65.0	0	0	13509	26.5500	E38	

```
from scipy import stats

# Calculate the Z-score for Age
z_scores_age = np.abs(stats.zscore(df['Age']))

# Define a threshold for Z-score to identify outliers (e.g., Z-score > 3)
age_outliers = df[z_scores_age > 3]
age_outliers
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked

```
# Dependent Variable (Target)
y = df['Survived']

# Independent Variables (Features)
x = df.drop('Survived', axis=1)
```

```
X = df.drop(['Survived', 'PassengerId', 'Name', 'Ticket'], axis=1)
```

y

```
0      0
1      1
2      1
3      1
4      0
      ..
886    0
887    1
888    0
889    1
890    0
Name: Survived, Length: 891, dtype: int64
```

x

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

X

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

```
# Perform one-hot encoding for 'Sex' column
df_encoded = pd.get_dummies(df, columns=['Sex'], drop_first=True)

# Perform one-hot encoding for 'Embarked' column
df_encoded = pd.get_dummies(df_encoded, columns=['Embarked'], drop_first=True)


df_encoded
```

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

```
#Min-Max Scaling (Normalization)

from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
columns_to_scale = ['Age', 'Fare']
df[columns_to_scale] = scaler.fit_transform(df[columns_to_scale])
df[columns_to_scale]
```

	Age	Fare
0	0.271174	0.014151
1	0.472229	0.139136
2	0.321438	0.015469
3	0.434531	0.103644
4	0.434531	0.015713
...	...	...
886	0.334004	0.025374
887	0.233476	0.058556
888	NaN	0.045771
889	0.321438	0.058556
890	0.396833	0.015127

891 rows × 2 columns

```
#Standardization (Z-score Scaling)

from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
df[columns_to_scale] = scaler.fit_transform(df[columns_to_scale])


from sklearn.model_selection import train_test_split
X = df_encoded.drop('Survived', axis=1)  # Independent variables (features)
y = df_encoded['Survived']  # Dependent variable (target)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)


X
```

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

```
y
```

```
0      0
1      1
2      1
3      1
4      0
      ..
886    0
887    1
888    0
889    1
890    0
Name: Survived, Length: 891, dtype: int64
```

```
X_train, X_test, y_train, y_test
```

```
(     PassengerId  Pclass                                Name   Age  SibSp  \
 331          332       1                 Partner, Mr. Austen  45.5      0
 733          734       2          Berriman, Mr. William John  23.0      0
 382          383       3                  Tikkanen, Mr. Juho  32.0      0
 704          705       3              Hansen, Mr. Henrik Juul  26.0      1
 813          814       3  Andersson, Miss. Ebba Iris Alfrida   6.0      4
 ..           ...     ...                                 ...   ...    ...
 106          107       3    Salkjelsvik, Miss. Anna Kristine  21.0      0
 270          271       1               Cairns, Mr. Alexander   NaN      0
 860          861       3              Hansen, Mr. Claus Peter  41.0      2
 435          436       1            Carter, Miss. Lucile Polk  14.0      1
 102          103       1            White, Mr. Richard Frasar  21.0      0

      Parch            Ticket      Fare    Cabin  Sex_male  Embarked_Q  \
 331      0            113043   28.5000     C124         1           0
```

```
733      0              28425   13.0000      NaN         1           0
382      0  STON/O 2. 3101293    7.9250      NaN         1           0
704      0             350025    7.8542      NaN         1           0
813      2             347082   31.2750      NaN         0           0
..     ...                ...       ...      ...       ...         ...
106      0             343120    7.6500      NaN         0           0
270      0             113798   31.0000      NaN         1           0
860      0             350026   14.1083      NaN         1           0
435      2             113760  120.0000  B96 B98         0           0
102      1              35281   77.2875      D26         1           0

     Embarked_S
331           1
733           1
382           1
704           1
813           1
..          ...
106           1
270           1
860           1
435           1
102           1

[712 rows x 12 columns],
     PassengerId  Pclass                                          Name  \
709          710       3  Moubarek, Master. Halim Gonios ("William George")
439          440       2            Kvillner, Mr. Johan Henrik Johannesson
840          841       3                Alhomaki, Mr. Ilmari Rudolf
720          721       2             Harper, Miss. Annie Jessie "Nina"
39            40       3                Nicola-Yarred, Miss. Jamila
..           ...     ...                                         ...
433          434       3                 Kallio, Mr. Nikolai Erland
773          774       3                           Elias, Mr. Dibo
25            26       3  Asplund, Mrs. Carl Oscar (Selma Augusta Emilia...
84            85       2                       Ilett, Miss. Bertha
10            11       3              Sandstrom, Miss. Marguerite Rut

      Age  SibSp  Parch             Ticket     Fare Cabin  Sex_male  \
709   NaN      1      1               2661  15.2458   NaN         1
439  31.0      0      0        C.A. 18723  10.5000   NaN         1
840  20.0      0      0   SOTON/O2 3101287   7.9250   NaN         1
720   6.0      0      1             248727  33.0000   NaN         0
```