P.S -> To predict whether the passenger is survived or not

1. Import packages

In [ ]:
```
import numpy
import pandas as pd
from sklearn import preprocessing
import matplotlib.pyplot as plt
```

2.Load the Dataset

In [1]:
```
import pandas as pd

# Load dataset from "concrete.csv"
dataset = pd.read_csv('concrete.csv')

# Display the first few rows of the dataset
print(dataset.head())
```

```
   cement   slag    ash  water  superplastic  coarseagg  fineagg  age  \
0   141.3  212.0    0.0  203.5           0.0      971.8    748.5   28
1   168.9   42.2  124.3  158.3          10.8     1080.8    796.2   14
2   250.0    0.0   95.7  187.4           5.5      956.9    861.2   28
3   266.0  114.0    0.0  228.0           0.0      932.0    670.0   28
4   154.8  183.4    0.0  193.3           9.1     1047.4    696.7   28

   strength
0     29.89
1     23.51
2     29.22
3     45.85
4     18.29
```

3.Analyze the dataset like shape,datatypes,missing values,describe etc..

In [2]:
```
import pandas as pd

# Load dataset from "concrete.csv"
dataset = pd.read_csv('concrete.csv')

# Display the shape of the dataset (rows, columns)
print("Shape of the dataset:", dataset.shape)

# Display the data types of each column
print("\nData types:")
print(dataset.dtypes)

# Check for missing values in each column
missing_values = dataset.isnull().sum()
```

```
print("\nMissing values:")
print(missing_values)

# Generate summary statistics for numerical columns
summary_stats = dataset.describe()
print("\nSummary Statistics:")
print(summary_stats)
```

```
Shape of the dataset: (1030, 9)

Data types:
cement          float64
slag            float64
ash             float64
water           float64
superplastic    float64
coarseagg       float64
fineagg         float64
age               int64
strength        float64
dtype: object

Missing values:
cement          0
slag            0
ash             0
water           0
superplastic    0
coarseagg       0
fineagg         0
age             0
strength        0
dtype: int64

Summary Statistics:
            cement         slag          ash        water  superplastic  \
count  1030.000000  1030.000000  1030.000000  1030.000000   1030.000000
mean    281.167864    73.895825    54.188350   181.567282      6.204660
std     104.506364    86.279342    63.997004    21.354219      5.973841
min     102.000000     0.000000     0.000000   121.800000      0.000000
25%     192.375000     0.000000     0.000000   164.900000      0.000000
50%     272.900000    22.000000     0.000000   185.000000      6.400000
75%     350.000000   142.950000   118.300000   192.000000     10.200000
max     540.000000   359.400000   200.100000   247.000000     32.200000

         coarseagg      fineagg          age     strength
count  1030.000000  1030.000000  1030.000000  1030.000000
mean    972.918932   773.580485    45.662136    35.817961
std      77.753954    80.175980    63.169912    16.705742
min     801.000000   594.000000     1.000000     2.330000
25%     932.000000   730.950000     7.000000    23.710000
50%     968.000000   779.500000    28.000000    34.445000
75%    1029.400000   824.000000    56.000000    46.135000
max    1145.000000   992.600000   365.000000    82.600000
```


distribution of survived and not survived

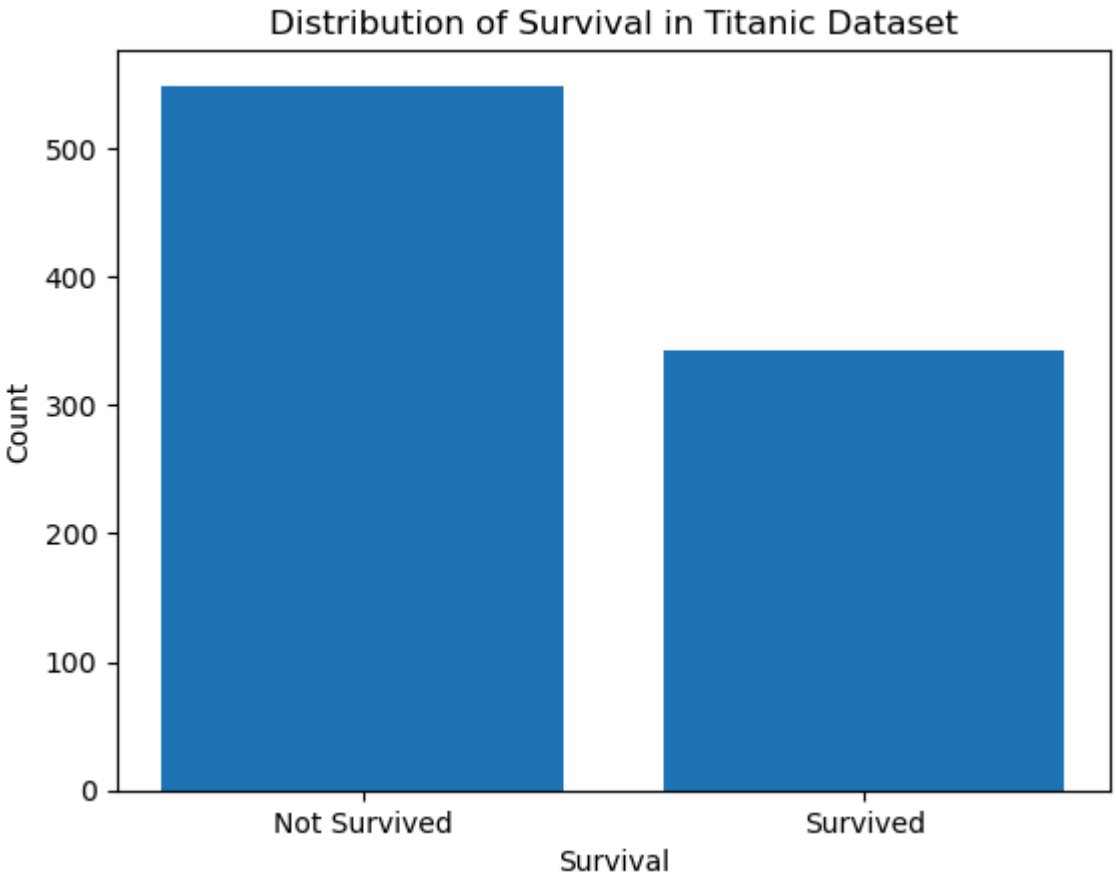
In [6]:
```
import pandas as pd
import matplotlib.pyplot as plt

# Load the Titanic dataset
dataset = pd.read_csv('titanic-training-data.csv')

# Assuming "Survived" is the column name indicating survival (0 = Not Survi
survival_distribution = dataset['Survived'].value_counts()

# Map labels for better visualization
survival_distribution.index = ['Not Survived', 'Survived']

# Create a bar plot to visualize the distribution
plt.bar(survival_distribution.index, survival_distribution.values)
plt.xlabel('Survival')
plt.ylabel('Count')
plt.title('Distribution of Survival in Titanic Dataset')
plt.show()
```



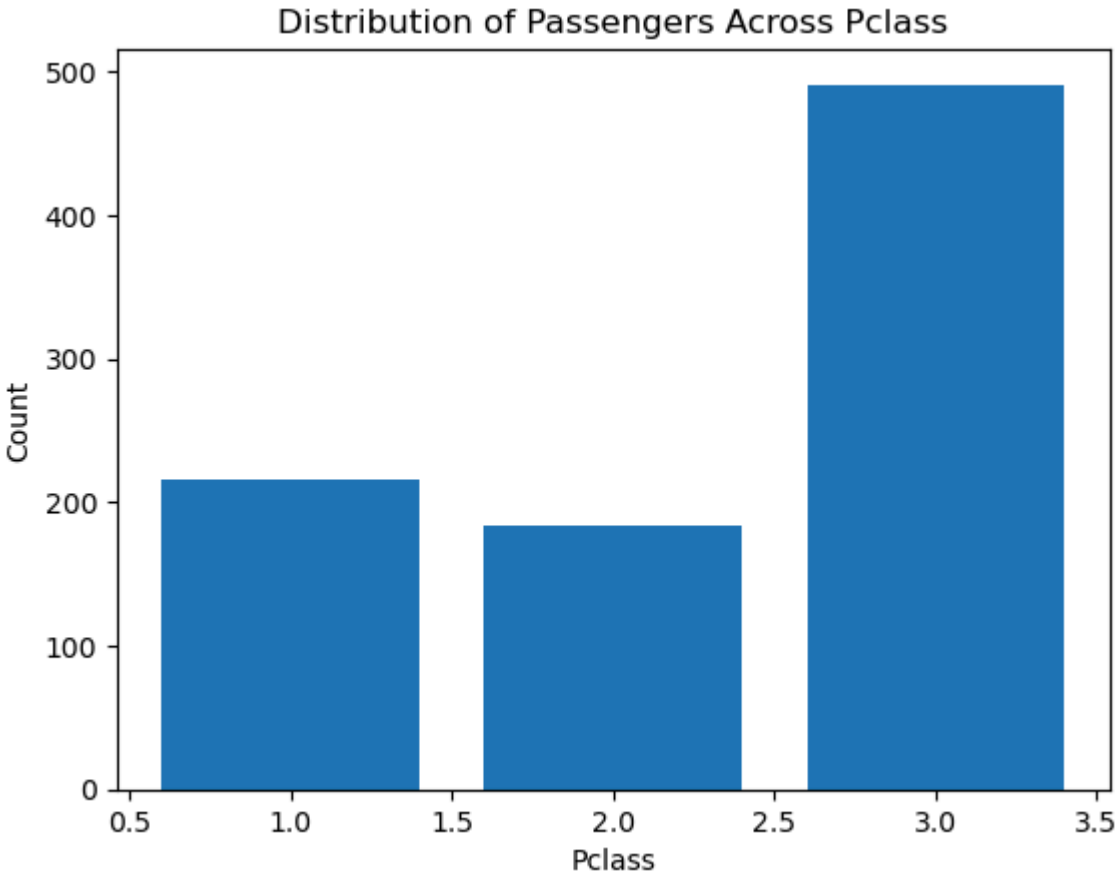
5.Show the distribution across pclass

In [5]:
```
import pandas as pd
import matplotlib.pyplot as plt

# Load the Titanic dataset
dataset = pd.read_csv('titanic-training-data.csv')
```

```
# Assuming "Pclass" is the column name for passenger classes
pclass_distribution = dataset['Pclass'].value_counts()

# Create a bar plot to visualize the distribution
plt.bar(pclass_distribution.index, pclass_distribution.values)
plt.xlabel('Pclass')
plt.ylabel('Count')
plt.title('Distribution of Passengers Across Pclass')
plt.show()
```



6. Show the distribution of Embarked

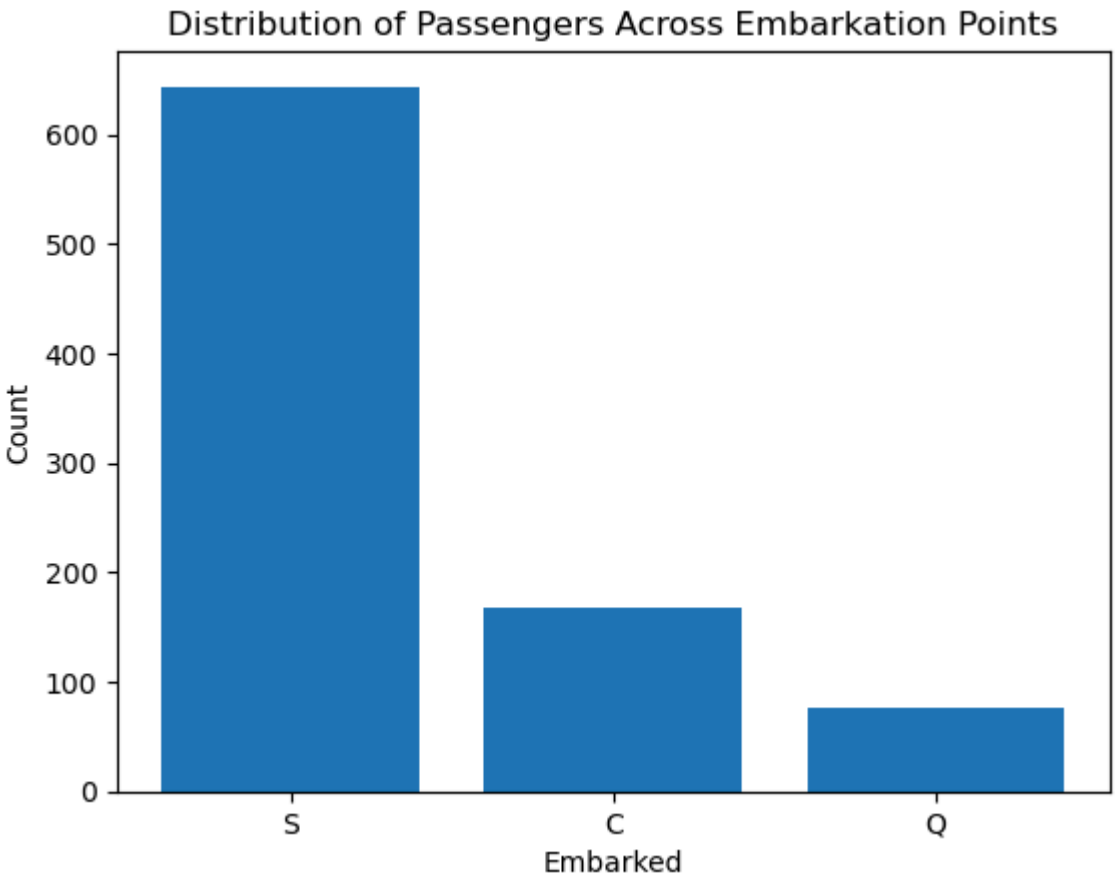
In [7]:
```
import pandas as pd
import matplotlib.pyplot as plt

# Load the Titanic dataset
dataset = pd.read_csv('titanic-training-data.csv')

# Assuming "Embarked" is the column name for embarkation points
embarked_distribution = dataset['Embarked'].value_counts()

# Create a bar plot to visualize the distribution
plt.bar(embarked_distribution.index, embarked_distribution.values)
plt.xlabel('Embarked')
plt.ylabel('Count')
```

```
plt.title('Distribution of Passengers Across Embarkation Points')
plt.show()
```



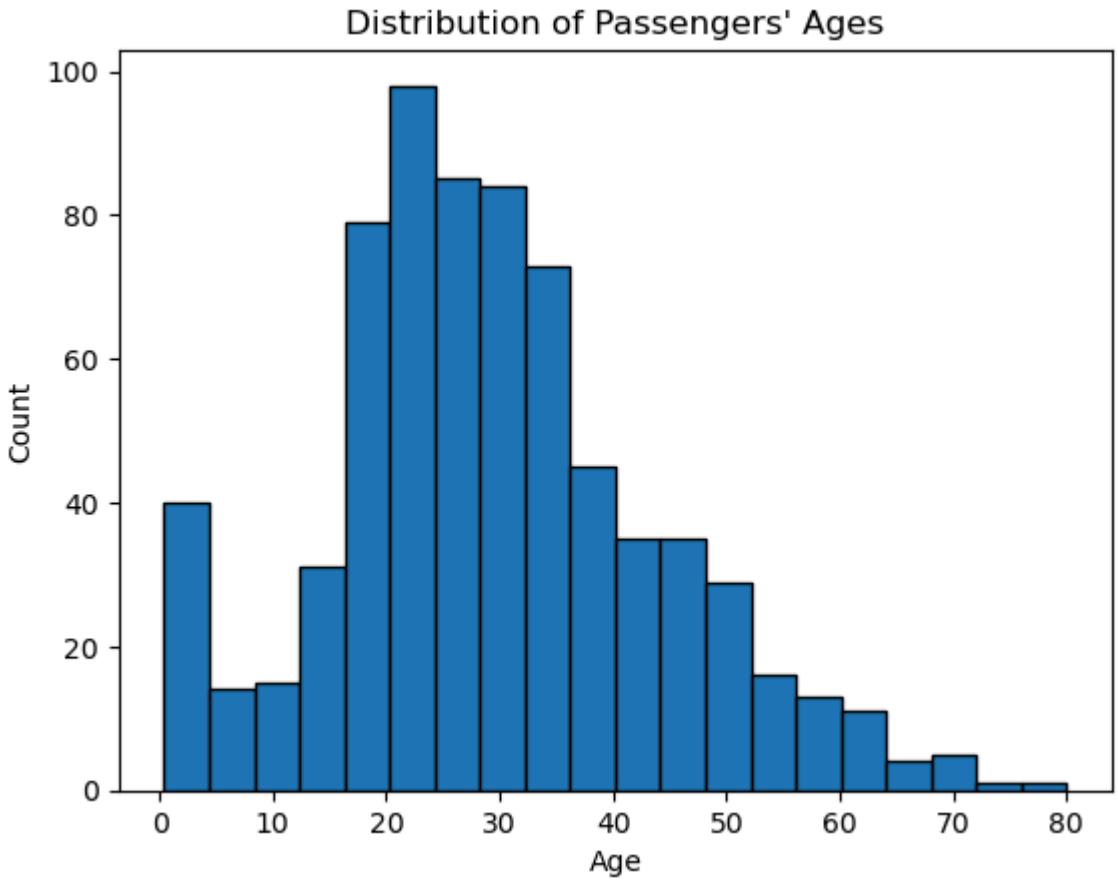
7. Show the distribution of Age

In [8]:
```
import pandas as pd
import matplotlib.pyplot as plt

# Load the Titanic dataset
dataset = pd.read_csv('titanic-training-data.csv')

# Assuming "Age" is the column name for passenger ages
# Remove rows with missing age data for better visualization
age_data = dataset['Age'].dropna()

# Create a histogram to visualize the age distribution
plt.hist(age_data, bins=20, edgecolor='black')
plt.xlabel('Age')
plt.ylabel('Count')
plt.title('Distribution of Passengers\' Ages')
plt.show()
```



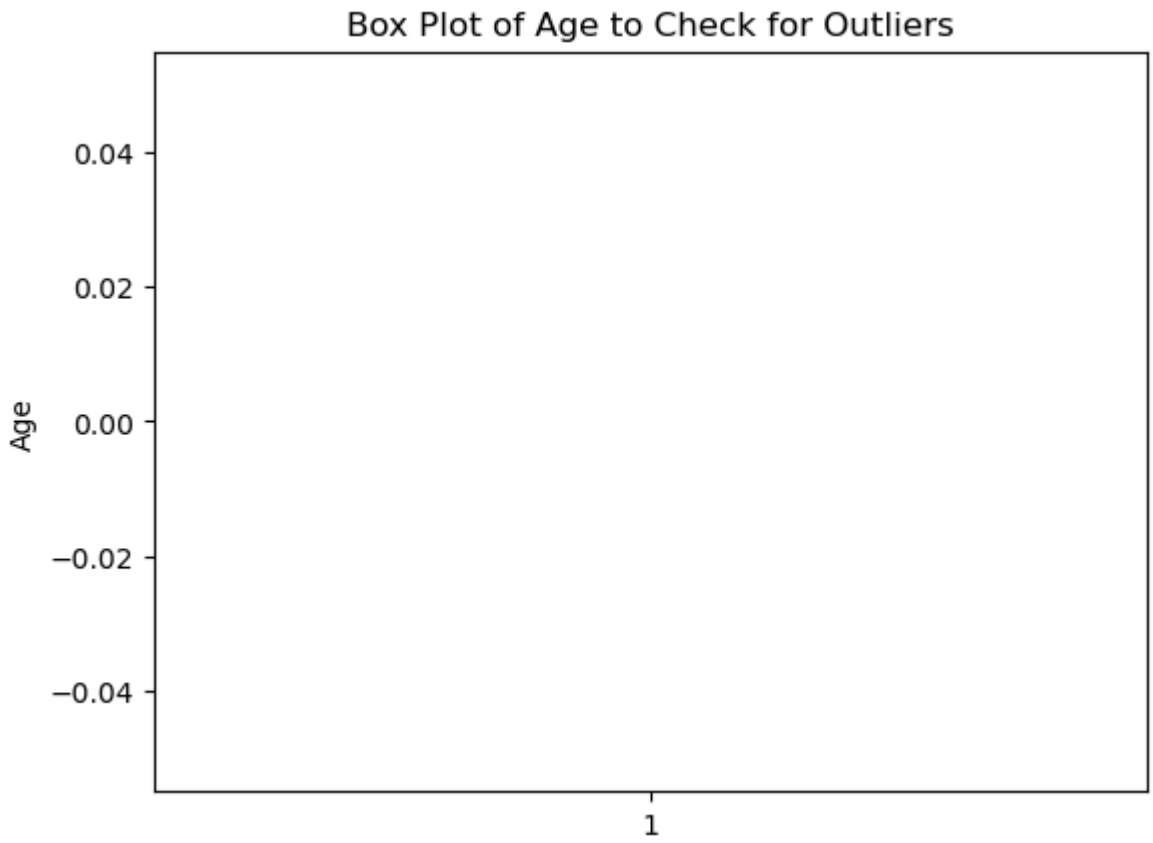
8. Check whether there are outliers in Age

In [9]:
```
import pandas as pd
import matplotlib.pyplot as plt

# Load the Titanic dataset
dataset = pd.read_csv('titanic-training-data.csv')

# Assuming "Age" is the column name for passenger ages
age_data = dataset['Age']

# Create a box plot to visualize potential outliers in Age
plt.boxplot(age_data)
plt.ylabel('Age')
plt.title('Box Plot of Age to Check for Outliers')
plt.show()
```



In [10]:
```
Q1 = age_data.quantile(0.25)
Q3 = age_data.quantile(0.75)
IQR = Q3 - Q1

lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

outliers = age_data[(age_data < lower_bound) | (age_data > upper_bound)]
print("Outliers:")
print(outliers)
```

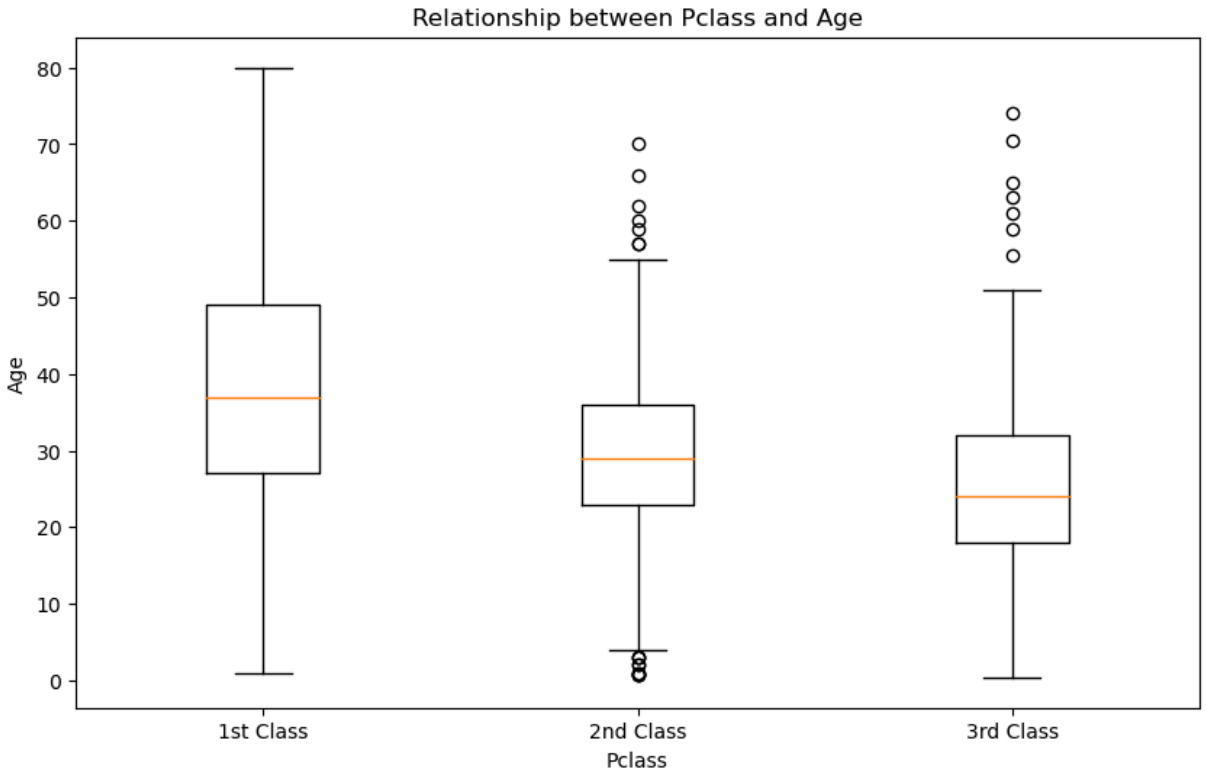
```
Outliers:
33     66.0
54     65.0
96     71.0
116    70.5
280    65.0
456    65.0
493    71.0
630    80.0
672    70.0
745    70.0
851    74.0
Name: Age, dtype: float64
```

9.Relationship between Pclass and Age

In [11]:
```
import pandas as pd
tlib.pyplot as plt
```

```
# Load the Titanic dataset
dataset = pd.read_csv('titanic-training-data.csv')

# Assuming "Pclass" is the column name for passenger classes
# Assuming "Age" is the column name for passenger ages
plt.figure(figsize=(10, 6))
plt.boxplot([dataset[dataset['Pclass'] == 1]['Age'].dropna(),
             dataset[dataset['Pclass'] == 2]['Age'].dropna(),
             dataset[dataset['Pclass'] == 3]['Age'].dropna()],
            labels=['1st Class', '2nd Class', '3rd Class'])
plt.xlabel('Pclass')
plt.ylabel('Age')
plt.title('Relationship between Pclass and Age')
plt.show()
```



10.Pairplot for all the numerical attributes

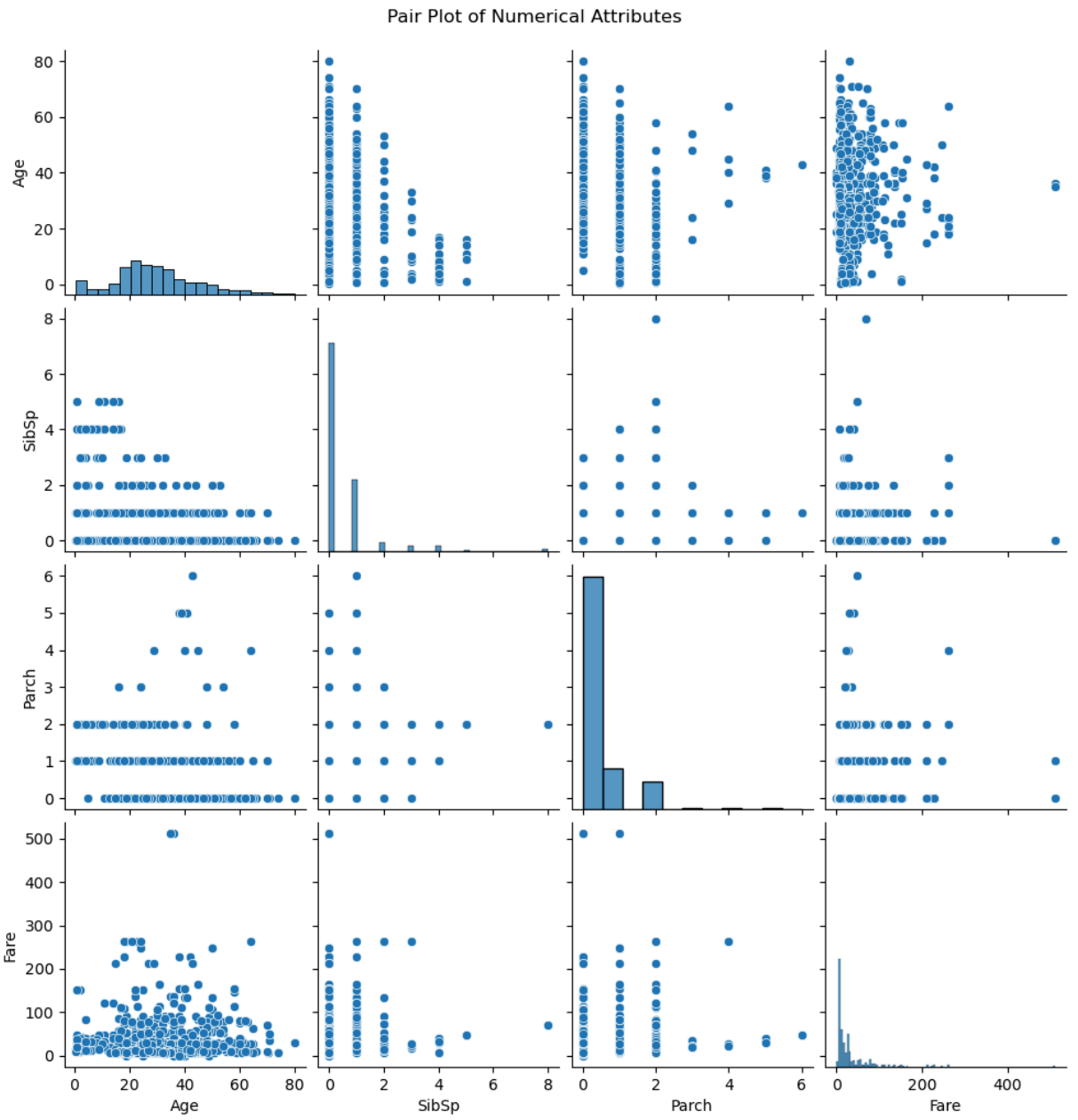
In [12]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Load the Titanic dataset
dataset = pd.read_csv('titanic-training-data.csv')

# Select numerical columns
numerical_attributes = ['Age', 'SibSp', 'Parch', 'Fare']

# Create a pair plot for all numerical attributes
sns.pairplot(dataset[numerical_attributes])
```

```
plt.suptitle('Pair Plot of Numerical Attributes', y=1.02)
plt.show()
```



In [ ]: