Ex : 2

Use Titanic.csv and convert it into DataFrame.

AIM:

To use the Titanic dataset and do the data science process in it

PROCEDURE:

* Step 1: Load the dataset into a dataframe
* Step 2: Display the first few rows using head()
* Step 3: Explore column datatypes and check for missing value using info()
* Step 4: Apply forward use the SimpleImputer to fill the missing values
* Step 5: Analyse the passenger class feature using countplot.
* Step 6: List all the passengers who are survived (female)
* Step 7: Show all the passengers who had more than 2 siblings.
* Step 8: Split the dataset for testing and training.

Output:

(891, 12)

RangeIndex: 891 entries, 0 to 890

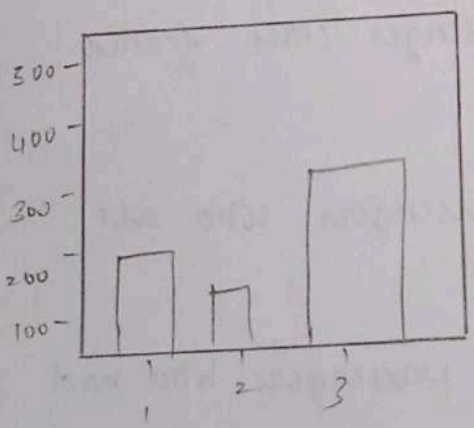
passenger Id .... Fare

count 891.000000 .... 891.000000

mean 891.000000 512.329200

passenger_id .... Embarked

0 1 2 ... 890 | 1 2 ... 891 | S ... S



(623, 11) (623,)

(260, 11) (260,)

PROGRAM:

```
import pandas as pd
data = pd.read_csv('Titanic.csv')
data.shape
data.info()
data.describe()
from sklearn.impute import SimpleImputer
imp = SimpleImputer()
data['Age'] = imp.fit_transform(data['Age'])
data.
data['cabin'] = data['cabin'].fillna[value = 'unknown']
data
data['Embarked'] = data['Emparked'].fillna[data['Embarked'
                                          .mode()]
data
import seaborn as sns
sns.countplot(data = data, x = 'pclass')
female_passengers = data[(data['sex'] == 'female') &
                         (data['survied'] == 1)]
names = female_passenger['Name']
names
old_pass = data[(data['pclass'] == 1) & (data['Age'] > 4
old_pass
combine_passenger = pd.concat([passenger, old_pass)]
survived_passenger = combine_passenger[combine_passenger
                                       ['survided' == 1]
survived_passengers.
```

2025/10/

```
male_passengers = data[(data['sex'] == 'male') &
                       (data['Fare'] > 100)]

male_passengers

Emb_passengers = data[(data['Embarked'] == 'C') &
                      (data['pclass'] == 2)]

Emb_passengers

Sib_passengers = data[data['sibsp'] > 2]

sib_passengers

passengers_not_survived = data[(data['sibsp'] == 0) &
                               (data['parch'] == 0)]

passengers_not_survived

Oldest_passengers = data.sort_values(by = 'Age',
                                     ascending = False)

oldest_passengers

zero_fare_passengers = data[data['Fare'] == 0]

zero_fare_passengers

from sklearn.model_selection import train_test_split
x = data.drop('survived', axis = 1)
y = data['survived']
x_train, x_test, y_train, y_test = train_test_split
                                   (x, y, test_size = 0.3
print(x_train.shape, y_train.shape)
print(x_test.shape, y_test.shape)
```

2025/10/