```python
import pandas as pd
data = pd.read_csv('train.csv')

data
```

```
     PassengerId  Survived  Pclass  \
0              1         0       3
1              2         1       1
2              3         1       3
3              4         1       1
4              5         0       3
..           ...       ...     ...
886          887         0       2
887          888         1       1
888          889         0       3
889          890         1       1
890          891         0       3

                                                  Name     Sex   Age  
SibSp  \
0                              Braund, Mr. Owen Harris    male  22.0   
1
1    Cumings, Mrs. John Bradley (Florence Briggs Th...  female  38.0   
1
2                               Heikkinen, Miss. Laina  female  26.0   
0
3         Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0   
1
4                             Allen, Mr. William Henry    male  35.0   
0
..                                                 ...     ...   ...   
...
886                              Montvila, Rev. Juozas    male  27.0   
0
887                       Graham, Miss. Margaret Edith  female  19.0   
0
888           Johnston, Miss. Catherine Helen "Carrie"  female   NaN   
1
889                              Behr, Mr. Karl Howell    male  26.0   
0
890                                Dooley, Mr. Patrick    male  32.0   
0

     Parch            Ticket     Fare Cabin Embarked
0        0         A/5 21171   7.2500   NaN        S
1        0          PC 17599  71.2833   C85        C
2        0  STON/O2. 3101282   7.9250   NaN        S
3        0            113803  53.1000  C123        S
4        0            373450   8.0500   NaN        S
..     ...               ...      ...   ...      ...
```

```
886      0            211536  13.0000   NaN        S
887      0            112053  30.0000   B42        S
888      2        W./C. 6607  23.4500   NaN        S
889      0            111369  30.0000  C148        C
890      0            370376   7.7500   NaN        Q

[891 rows x 12 columns]
```

```
data.head()
```

```
   PassengerId  Survived  Pclass  \
0            1         0       3
1            2         1       1
2            3         1       3
3            4         1       1
4            5         0       3

                                                Name     Sex   Age
SibSp  \
0                            Braund, Mr. Owen Harris    male  22.0
1
1  Cumings, Mrs. John Bradley (Florence Briggs Th...  female  38.0
1
2                             Heikkinen, Miss. Laina  female  26.0
0
3       Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0
1
4                           Allen, Mr. William Henry    male  35.0
0

   Parch            Ticket     Fare Cabin Embarked
0      0         A/5 21171   7.2500   NaN        S
1      0          PC 17599  71.2833   C85        C
2      0  STON/O2. 3101282   7.9250   NaN        S
3      0            113803  53.1000  C123        S
4      0            373450   8.0500   NaN        S
```

```
data.tail()
```

```
     PassengerId  Survived  Pclass
Name  \
886          887         0       2                     Montvila, Rev.
Juozas
887          888         1       1              Graham, Miss. Margaret
Edith
888          889         0       3  Johnston, Miss. Catherine Helen
"Carrie"
889          890         1       1                     Behr, Mr. Karl
Howell
890          891         0       3                       Dooley, Mr.
```

```
Patrick

        Sex   Age  SibSp  Parch      Ticket   Fare Cabin Embarked
886    male  27.0      0      0      211536  13.00   NaN        S
887  female  19.0      0      0      112053  30.00   B42        S
888  female   NaN      1      2  W./C. 6607  23.45   NaN        S
889    male  26.0      0      0      111369  30.00  C148        C
890    male  32.0      0      0      370376   7.75   NaN        Q
```

```
data.columns
```

```
Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age',
'SibSp',
       'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'],
      dtype='object')
```

```
data.shape
```

```
(891, 12)
```

```
data.size
```

```
10692
```

```
#2 ques
data.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     889 non-null    object 
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

```
# 2 ques
data.describe()
```

```
       PassengerId    Survived      Pclass         Age       SibSp  \
count   891.000000  891.000000  891.000000  714.000000  891.000000
```

```
mean    446.000000    0.383838    2.308642   29.699118    0.523008
std     257.353842    0.486592    0.836071   14.526497    1.102743
min       1.000000    0.000000    1.000000    0.420000    0.000000
25%     223.500000    0.000000    2.000000   20.125000    0.000000
50%     446.000000    0.000000    3.000000   28.000000    0.000000
75%     668.500000    1.000000    3.000000   38.000000    1.000000
max     891.000000    1.000000    3.000000   80.000000    8.000000

            Parch        Fare
count  891.000000  891.000000
mean     0.381594   32.204208
std      0.806057   49.693429
min      0.000000    0.000000
25%      0.000000    7.910400
50%      0.000000   14.454200
75%      0.000000   31.000000
max      6.000000  512.329200
```

```python
#3.1
#Data cleaning and prepocessing
df_duplicate = data.duplicated()
print(df_duplicate)
```

```
0      False
1      False
2      False
3      False
4      False
       ...  
886    False
887    False
888    False
889    False
890    False
Length: 891, dtype: bool
```

```python
#To drop duplicates
#drop_duplicates(inplace = True)
```

```python
# 3.2 ques
#Identify missing values
data.isnull().sum()
```

```
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
```

```
data.isnull()
```

```
     PassengerId  Survived  Pclass   Name    Sex    Age  SibSp  Parch
Ticket  \
0          False     False   False  False  False  False  False  False
False
1          False     False   False  False  False  False  False  False
False
2          False     False   False  False  False  False  False  False
False
3          False     False   False  False  False  False  False  False
False
4          False     False   False  False  False  False  False  False
False
..           ...       ...     ...    ...    ...    ...    ...    ...
...
886        False     False   False  False  False  False  False  False
False
887        False     False   False  False  False  False  False  False
False
888        False     False   False  False  False   True  False  False
False
889        False     False   False  False  False  False  False  False
False
890        False     False   False  False  False  False  False  False
False

      Fare  Cabin  Embarked
0    False   True     False
1    False  False     False
2    False   True     False
3    False  False     False
4    False   True     False
..     ...    ...       ...
886  False   True     False
887  False  False     False
888  False   True     False
889  False  False     False
890  False   True     False

[891 rows x 12 columns]
```

```
#Drop rows with missing values
#or
#Filling missing values with specific values or mean, meadian, mode
data['Age'].mean()
```

```
29.69911764705882
```

```
data['Age'].fillna(29.69911764705882 , inplace = True)
```

```
/var/folders/nm/jzdyc3jj6xb7z2qrn7szht600000gn/T/
ipykernel_74141/4271587288.py:1: FutureWarning: A value is trying to
be set on a copy of a DataFrame or Series through chained assignment
using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never
work because the intermediate object on which we are setting values
always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try
using 'df.method({col: value}, inplace=True)' or df[col] =
df[col].method(value) instead, to perform the operation inplace on the
original object.


  data['Age'].fillna(29.69911764705882 , inplace = True)
```

```
data['Age'].isnull().sum()
```

```
0
```

```
data['Cabin'].fillna('unknown', inplace = True)
```

```
/var/folders/nm/jzdyc3jj6xb7z2qrn7szht600000gn/T/
ipykernel_74141/4050063452.py:1: FutureWarning: A value is trying to
be set on a copy of a DataFrame or Series through chained assignment
using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never
work because the intermediate object on which we are setting values
always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try
using 'df.method({col: value}, inplace=True)' or df[col] =
df[col].method(value) instead, to perform the operation inplace on the
original object.


  data['Cabin'].fillna('unknown', inplace = True)
```

```
data['Cabin'].isnull().sum()
```

```
0
```

```
data.dropna('Survived', axis = 1, inplace = True)
```

```
----------------------------------------------------------------------
-----
TypeError                                 Traceback (most recent call
last)
Cell In[20], line 1
----> 1 data.dropna('Survived', axis = 1, inplace = True)

TypeError: DataFrame.dropna() takes 1 positional argument but 2
positional arguments (and 2 keyword-only arguments) were given
```

```
#5 To handle the outliers for fare attributes
import matplotlib.pyplot as plt
x = data['Survived']
y = data['Fare']
plt.xlabel('Passengers')
plt.ylabel('Price')

plt.scatter(x,y)

x = data['Survived']
y = data['Age']
plt.xlabel('Passengers')
plt.ylabel('Age')

plt.scatter(x,y)

#Finding outliers from IQR Method
q1 = data['Fare'].quantile(0.25)
q3 = data['Fare'].quantile(0.75)
IQR = q3 - q1
upper_limit = q3 + 1.5*IQR
lower_limit = q1 - 1.5*IQR
new = data.loc[(data['Fare'] > lower_limit ) & (data['Fare'] <
upper_limit)]
new

#after
import matplotlib.pyplot as plt
x = new['Survived']
y = new['Fare']
plt.xlabel('passengers')
plt.ylabel('cabin Fare')

plt.scatter(x,y)

new['Fare'].max()
```