```
             Name : Suryal D . Khirade
             Roll NO: T190424399
             Assignment No :01


             Data Wrangling I Perform the following operations usi
ng Python on any open source dataset
            (eg. data.csv)
            1. Import all the required Python Libraries.
            2. Locate an open-source data from the web (eg. http
s://www.kaggle.com). Provide a clear
            description of the data and its source (i.e. URL of th
e web site).
            3. Load the Dataset into pandas dataframe.
            4. Data Pre-processing: check for missing values in th
e data using pandas isnull(), describe()
            function to get some initial statistics. Provide varia
ble descriptions. Types of variables etc.
            Check the dimensions of the data frame.
            5. Data Formatting and Data Normalization: Summarize t
he types of variables by checking the
            data types (i.e., character, numeric, integer, factor,
and logical) of the variables in the data set.
            If variables are not in the correct data type, apply p
roper type conversions.
            6. Turn categorical variables into quantitative variab
les in Python In addition to the codes and
            outputs, explain every operation that you do in the ab
ove steps and explain everything that you do to
            import/read/scrape the data set.
```

#Data Wrangling 1

Import all the required Python Libraries.

In [1]:

```
import numpy as np
import pandas as pd
```

Load the Dataset into pandas dataframe.

In [11]:

```
df = pd.read_csv("C:\\Users\\alisu\\Desktop\\SIT lonvala\\TE\\6th sem\\DSBD
```

In [12]:

```
df.head()
```

Out[12]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
0	892	0	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292
1	893	1	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000
2	894	0	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875
3	895	0	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625
4	896	1	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875

In [13]:

```
df.tail()
```

Out[13]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	
413	1305	0	3	Spector, Mr. Woolf	male	NaN	0	0	A.5. 3236	8
414	1306	1	1	Oliva y Ocana, Dona. Fermina	female	39.0	0	0	PC 17758	108
415	1307	0	3	Saether, Mr. Simon Sivertsen	male	38.5	0	0	SOTON/O.Q. 3101262	7
416	1308	0	3	Ware, Mr. Frederick	male	NaN	0	0	359309	8
417	1309	0	3	Peter, Master. Michael J	male	NaN	1	1	2668	22

In [14]:
```
df.sample()
```

Out[14]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin
5	897	0	3	Svensson, Mr. Johan Cervin	male	14.0	0	0	7538	9.225	NaN

Data Preprocessing □ check for missing values in the data using pandas isnull()

In [15]:
```
df.isnull().sum()
```

Out[15]:
```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age             86
SibSp            0
Parch            0
Ticket           0
Fare             1
Cabin          327
Embarked         0
dtype: int64
```

In [16]:
```
df['Age'].fillna(df['Age'].mean(), inplace = True)
df['Age'].isna().sum()
```

Out[16]:
```
0
```

In [17]:
```
df['Embarked'].value_counts()
```

Out[17]:
```
Embarked
S    270
C    102
Q     46
Name: count, dtype: int64
```

In [18]:
```
df['Embarked'].fillna('S',inplace = True)
df['Embarked'].isna().sum()
```

Out[18]:
```
0
```

In [19]:

```
df.drop(columns = ['Cabin'],axis=1,inplace=True)
df.isnull().sum()
```

Out[19]:

```
PassengerId    0
Survived       0
Pclass         0
Name           0
Sex            0
Age            0
SibSp          0
Parch          0
Ticket         0
Fare           1
Embarked       0
dtype: int64
```

Describe() function to get some initial statistics. Provide variable descriptions.

In [20]:

```
df.describe()
```

Out[20]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	418.000000	418.000000	418.000000	418.000000	418.000000	418.000000	417.000000
mean	1100.500000	0.363636	2.265550	30.272590	0.447368	0.392344	35.627188
std	120.810458	0.481622	0.841838	12.634534	0.896760	0.981429	55.907576
min	892.000000	0.000000	1.000000	0.170000	0.000000	0.000000	0.000000
25%	996.250000	0.000000	1.000000	23.000000	0.000000	0.000000	7.895800
50%	1100.500000	0.000000	3.000000	30.272590	0.000000	0.000000	14.454200
75%	1204.750000	1.000000	3.000000	35.750000	1.000000	0.000000	31.500000
max	1309.000000	1.000000	3.000000	76.000000	8.000000	9.000000	512.329200

Types of variables

In [21]:

```
df.dtypes
```

Out[21]:

```
PassengerId      int64
Survived         int64
Pclass           int64
Name            object
Sex             object
Age            float64
SibSp            int64
Parch            int64
Ticket          object
Fare           float64
Embarked        object
dtype: object
```

In [22]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 11 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  418 non-null    int64  
 1   Survived     418 non-null    int64  
 2   Pclass       418 non-null    int64  
 3   Name         418 non-null    object 
 4   Sex          418 non-null    object 
 5   Age          418 non-null    float64
 6   SibSp        418 non-null    int64  
 7   Parch        418 non-null    int64  
 8   Ticket       418 non-null    object 
 9   Fare         417 non-null    float64
 10  Embarked     418 non-null    object 
dtypes: float64(2), int64(5), object(4)
memory usage: 36.1+ KB
```

Check the dimensions of the data frame

In [23]:

```
df.shape
```

Out[23]:

```
(418, 11)
```

In [24]:

```
df.shape[0]
```

Out[24]:

```
418
```

Data Formatting and Data Normalization Summarize the types of variables by checking the data types (i.e., character, numeric, integer, factor, and logical) of the □ variables in the data set.

In [25]:

```
df.nunique()
```

Out[25]:

```
PassengerId    418
Survived         2
Pclass           3
Name           418
Sex              2
Age             80
SibSp            7
Parch            8
Ticket         363
Fare           169
Embarked         3
dtype: int64
```

```
In [26]: df['Survived'].value_counts()
```

```
Out[26]: Survived
         0    266
         1    152
         Name: count, dtype: int64
```

```
In [27]: df['Pclass'].value_counts()
```

```
Out[27]: Pclass
         3    218
         1    107
         2     93
         Name: count, dtype: int64
```

```
In [28]: df['Sex'].value_counts()
```

```
Out[28]: Sex
         male      266
         female    152
         Name: count, dtype: int64
```

```
In [29]: df['SibSp'].value_counts()
```

```
Out[29]: SibSp
         0    283
         1    110
         2     14
         3      4
         4      4
         8      2
         5      1
         Name: count, dtype: int64
```

```
In [30]: df['Parch'].value_counts()
```

```
Out[30]: Parch
         0    324
         1     52
         2     33
         3      3
         4      2
         9      2
         6      1
         5      1
         Name: count, dtype: int64
```

```
In [31]: df['Embarked'].value_counts()
```

```
Out[31]: Embarked
         S    270
         C    102
         Q     46
         Name: count, dtype: int64
```

```
If variables are not in the correct data type, apply proper type
conversions.
```

In [33]:
```
df.dtypes
```

Out[33]:
```
PassengerId      int64
Survived         int64
Pclass           int64
Name            object
Sex             object
Age            float64
SibSp            int64
Parch            int64
Ticket          object
Fare           float64
Embarked        object
dtype: object
```

In [34]:
```
df['Age'] = df['Age'].astype('int64')
```

Turn categorical variables into quantitative variables in Python.

In [36]:
```
df["Sex"].replace(['female','male'],[0,1],inplace = True)
df['Sex'].value_counts()
```

Out[36]:
```
Sex
1    266
0    152
Name: count, dtype: int64
```

In [37]:
```
df['Embarked'].replace(['C','Q','S'],[1,2,3],inplace= True)
df['Embarked'].value_counts()
```

Out[37]:
```
Embarked
3    270
1    102
2     46
Name: count, dtype: int64
```

In [38]:
```
df.dtypes
```

Out[38]:
```
PassengerId      int64
Survived         int64
Pclass           int64
Name            object
Sex              int64
Age              int64
SibSp            int64
Parch            int64
Ticket          object
Fare           float64
Embarked         int64
dtype: object
```

In [39]:

```
df.drop(columns=['Name','PassengerId','Ticket'],axis = 1,inplace = True)
```

In [40]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 8 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   Survived  418 non-null    int64  
 1   Pclass    418 non-null    int64  
 2   Sex       418 non-null    int64  
 3   Age       418 non-null    int64  
 4   SibSp     418 non-null    int64  
 5   Parch     418 non-null    int64  
 6   Fare      417 non-null    float64
 7   Embarked  418 non-null    int64  
dtypes: float64(1), int64(7)
memory usage: 26.3 KB
```