Group A

Assignment 1

Data Wrangling I

Import all the required Python Libraries.

```
# code here
import numpy as np
import pandas as pd
```

Load the Dataset into pandas dataframe.

```
# code here
df = pd.read_csv("Titanic.csv")



# code here
df.head()

      sex   age  sibsp  parch     fare embarked  class    who  alone
survived
0    male  22.0      1      0   7.2500        S  Third    man  False
0
1  female  38.0      1      0  71.2833        C  First  woman  False
1
2  female  26.0      0      0   7.9250        S  Third  woman   True
1
3  female  35.0      1      0  53.1000        S  First  woman  False
1
4    male  35.0      0      0   8.0500        S  Third    man   True
0

df.sample()

        sex   age  sibsp  parch  fare embarked  class    who  alone
survived
654  female  18.0      0      0  6.75        Q  Third  woman   True
0

df.tail()

        sex   age  sibsp  parch   fare embarked   class    who  alone
\
886    male  27.0      0      0  13.00        S  Second    man   True
```

```
887  female  19.0      0      0  30.00         S  First  woman   True

888  female   NaN      1      2  23.45         S  Third  woman  False

889    male  26.0      0      0  30.00         C  First    man   True

890    male  32.0      0      0   7.75         Q  Third    man   True


     survived
886         0
887         1
888         0
889         1
890         0
```

Data Preprocessing

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

```
# to highlight esc+m
# we use # as h1,## as h2 and so on

df.isnull()

       sex    age  sibsp  parch   fare  embarked  class    who  alone
\
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   True  False  False  False     False  False  False  False

889  False  False  False  False  False     False  False  False  False

890  False  False  False  False  False     False  False  False  False


     survived
```

```
0       False
1       False
2       False
3       False
4       False
..        ...
886     False
887     False
888     False
889     False
890     False

[891 rows x 10 columns]
```

```python
df.isnull().sum()
```

```
sex           0
age         177
sibsp         0
parch         0
fare          0
embarked      2
class         0
who           0
alone         0
survived      0
dtype: int64
```

```python
df["age"].fillna(df["age"].mean(),inplace=True)
# if changes are seen after execution then therse changes are
temporary to do it permanent we use inplace
# to check these check above run isnull function
```

```python
df["embarked"].value_counts()
```

```
embarked
S    644
C    168
Q     77
Name: count, dtype: int64
```

```python
df["embarked"].fillna('S')
```

```
0      S
1      C
2      S
3      S
4      S
      ..
886    S
887    S
```

```
888    S
889    C
890    Q
Name: embarked, Length: 891, dtype: object
```

```
df.isnull().sum()
```

```
sex         0
age         0
sibsp       0
parch       0
fare        0
embarked    2
class       0
who         0
alone       0
survived    0
dtype: int64
```

```
df["embarked"].fillna('S',inplace=True)
```

```
df.isna().sum()
```

```
sex         0
age         0
sibsp       0
parch       0
fare        0
embarked    0
class       0
who         0
alone       0
survived    0
dtype: int64
```

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

```
# code here
df.describe()
```

```
              age       sibsp       parch        fare    survived
count  891.000000  891.000000  891.000000  891.000000  891.000000
mean    29.699118    0.523008    0.381594   32.204208    0.383838
std     13.002015    1.102743    0.806057   49.693429    0.486592
min      0.420000    0.000000    0.000000    0.000000    0.000000
25%     22.000000    0.000000    0.000000    7.910400    0.000000
50%     29.699118    0.000000    0.000000   14.454200    0.000000
75%     35.000000    1.000000    0.000000   31.000000    1.000000
max     80.000000    8.000000    6.000000  512.329200    1.000000
```

```
df.mean()
```

```
sex          0.647587
age         29.699118
sibsp        0.523008
parch        0.381594
fare        32.204208
embarked     1.536476
class        1.308642
who          1.210999
alone        0.602694
survived     0.383838
dtype: float64
```

```
df["age"].quantile(0.25)
```

```
22.0
```

Types of variables

```
# code here
df.dtypes
```

```
sex          object
age         float64
sibsp         int64
parch         int64
fare        float64
embarked     object
class        object
who          object
alone          bool
survived      int64
dtype: object
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 10 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   sex       891 non-null    object
 1   age       891 non-null    float64
 2   sibsp     891 non-null    int64
 3   parch     891 non-null    int64
 4   fare      891 non-null    float64
 5   embarked  891 non-null    object
 6   class     891 non-null    object
 7   who       891 non-null    object
 8   alone     891 non-null    bool
 9   survived  891 non-null    int64
```

```
dtypes: bool(1), float64(2), int64(3), object(4)
memory usage: 63.6+ KB
```

```
df["age"].sample(10)
```

```
90     29.000000
126    29.699118
857    51.000000
164     1.000000
842    30.000000
475    29.699118
889    26.000000
668    43.000000
319    40.000000
229    29.699118
Name: age, dtype: float64
```

Check the dimensions of the data frame

```
# code here
df.shape
```

```
(891, 10)
```

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.

```
# code here
df.nunique()
```

```
sex           2
age          89
sibsp         7
parch         7
fare        248
embarked      3
class         3
who           3
alone         2
survived      2
dtype: int64
```

```
df["sex"].value_counts()
```

```
sex
male      577
female    314
Name: count, dtype: int64
```

```
df["embarked"].value_counts()
```

```
embarked
S    646
C    168
Q     77
Name: count, dtype: int64
```

```
df["sibsp"].value_counts()
```

```
sibsp
0    608
1    209
2     28
4     18
3     16
8      7
5      5
Name: count, dtype: int64
```

```
df["parch"].value_counts()
```

```
parch
0    678
1    118
2     80
5      5
3      5
4      4
6      1
Name: count, dtype: int64
```

```
df["class"].value_counts()
```

```
class
Third     491
First     216
Second    184
Name: count, dtype: int64
```

```
df["who"].value_counts()
```

```
who
man      537
woman    271
child     83
Name: count, dtype: int64
```

```
df["alone"].value_counts()
```

```
alone
True     537
```

```
False    354
Name: count, dtype: int64
```

```python
df["survived"].value_counts()
```

```
survived
0    549
1    342
Name: count, dtype: int64
```

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

```python
# code here
# df.age.astype('int64')
```

Turn categorical variables into quantitative variables in Python.

```python
# code here
# replace function takes two parameter i.) list of string
# ii.) list of numbers to replace

df["sex"].replace(['female','male'],[0,1],inplace=True)

df["who"].replace(['child','man','woman'],[0,1,2],inplace=True)

df["embarked"].replace(['C','Q','S'],[0,1,2],inplace=True)

df["class"].replace(['First','Second','Third'],[0,1,2],inplace=True)

df["alone"].replace(['False','True'],[0,1],inplace=True)

df.dtypes
```

```
sex           int64
age         float64
sibsp         int64
parch         int64
fare        float64
embarked      int64
class         int64
who           int64
alone          bool
survived      int64
dtype: object
```

```python
df.describe()
```

```
              sex         age       sibsp       parch        fare
embarked  \
count  891.000000  891.000000  891.000000  891.000000  891.000000
891.000000
mean     0.647587   29.699118    0.523008    0.381594   32.204208
```

```
1.536476
std       0.477990   13.002015    1.102743    0.806057   49.693429
0.791503
min       0.000000    0.420000    0.000000    0.000000    0.000000
0.000000
25%       0.000000   22.000000    0.000000    0.000000    7.910400
1.000000
50%       1.000000   29.699118    0.000000    0.000000   14.454200
2.000000
75%       1.000000   35.000000    1.000000    0.000000   31.000000
2.000000
max       1.000000   80.000000    8.000000    6.000000  512.329200
2.000000

            class         who    survived
count  891.000000  891.000000  891.000000
mean     1.308642    1.210999    0.383838
std      0.836071    0.594291    0.486592
min      0.000000    0.000000    0.000000
25%      1.000000    1.000000    0.000000
50%      2.000000    1.000000    0.000000
75%      2.000000    2.000000    1.000000
max      2.000000    2.000000    1.000000
```

```
df["age"].unique()
```

```
array([22.        , 38.        , 26.        , 35.        ,
29.69911765,
       54.        ,  2.        , 27.        , 14.        ,
4.        ,
       58.        , 20.        , 39.        , 55.        ,
31.        ,
       34.        , 15.        , 28.        ,  8.        ,
19.        ,
       40.        , 66.        , 42.        , 21.        ,
18.        ,
        3.        ,  7.        , 49.        , 29.        ,
65.        ,
       28.5       ,  5.        , 11.        , 45.        ,
17.        ,
       32.        , 16.        , 25.        ,  0.83       ,
30.        ,
       33.        , 23.        , 24.        , 46.        ,
59.        ,
       71.        , 37.        , 47.        , 14.5       ,
70.5       ,
       32.5       , 12.        ,  9.        , 36.5       ,
51.        ,
       55.5       , 40.5       , 44.        ,  1.        ,
61.        ,
       56.        , 50.        , 36.        , 45.5       ,
20.5       ,
       62.        , 41.        , 52.        , 63.        ,
23.5       ,
        0.92      , 43.        , 60.        , 10.        ,
64.        ,
       13.        , 48.        ,  0.75      , 53.        ,
57.        ,
       80.        , 70.        , 24.5       ,  6.        ,
0.67      ,
       30.5       ,  0.42      , 34.5       , 74.        ])
```

```
df.sample(10)
```

```
     sex        age  sibsp  parch      fare  embarked  class  who
alone  \
293    0  24.000000      0      0    8.8500         2      2    2
True
729    0  25.000000      1      0    7.9250         2      2    2
False
685    1  25.000000      1      2   41.5792         0      1    1
False
677    0  18.000000      0      0    9.8417         2      2    2
True
288    1  42.000000      0      0   13.0000         2      1    1
True
571    0  53.000000      2      0   51.4792         2      0    2
False
356    0  22.000000      0      1   55.0000         2      0    2
False
815    1  29.699118      0      0    0.0000         2      0    1
True
334    0  29.699118      1      0  133.6500         2      0    2
False
238    1  19.000000      0      0   10.5000         2      1    1
True

     survived
293         0
729         0
685         0
677         1
288         1
571         1
356         1
815         0
334         1
238         0
```