Data Wrangling

- Importing Libraries

In [ ]:
```
import numpy as np
import pandas as pd
import seaborn as sns
```

In [ ]:
```
kashti=sns.load_dataset("Titanic")
ks1=kashti
ks2=kashti
```

In [ ]:
```
kashti.head()
```

Out[ ]:

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

In [ ]:
```
(kashti['age']+12).head(10)
```

Out[ ]:
```
0    34.0
1    50.0
2    38.0
3    47.0
4    47.0
5     NaN
6    66.0
7    14.0
8    39.0
9    26.0
Name: age, dtype: float64
```

In [ ]:
```
kashti.isnull().sum()
```

Out[ ]:
```
survived         0
pclass           0
sex              0
age            177
sibsp            0
parch            0
fare             0
embarked         2
class            0
who              0
adult_male       0
deck           688
embark_town      2
alive            0
alone            0
dtype: int64
```

In [ ]:
```
kashti.shape
```

Out[ ]:
```
(891, 15)
```

In [ ]:
```
# dropna method
kashti.dropna(subset=['deck'],axis=0,inplace=True)
kashti.isnull().sum()
```

Out[ ]:
```
survived         0
pclass           0
sex              0
age             19
sibsp            0
parch            0
fare             0
embarked         2
class            0
who              0
adult_male       0
deck             0
embark_town      2
alive            0
alone            0
dtype: int64
```

In [ ]:
```
kashti.shape
```

Out[ ]:
```
(203, 15)
```

In [ ]:
```
# to remove null value from whole dataset
kashti=kashti.dropna()
kashti.isnull().sum()
```

Out[ ]:
```
survived       0
pclass         0
sex            0
age            0
sibsp          0
parch          0
fare           0
embarked       0
class          0
who            0
adult_male     0
deck           0
embark_town    0
alive          0
alone          0
dtype: int64
```

In [ ]:
```
kashti.shape
```

Out[ ]:
```
(182, 15)
```

Replace the missing value with the average of that column

In [ ]:
```
# finding the average(mean)
mean=ks1['age'].mean()
mean
```

Out[ ]:
```
35.77945652173913
```

In [ ]:
```
ks1['age']=ks1['age'].replace(np.nan,mean)
```

In [ ]:
```
# replacing nan with mean
ks1.isnull().sum()
```

Out[ ]:
```
survived       0
pclass         0
sex            0
age            0
sibsp          0
parch          0
fare           0
embarked       2
class          0
who            0
adult_male     0
deck           0
embark_town    2
alive          0
alone          0
dtype: int64
```

Assignment 1

In [ ]:
```
kashti=sns.load_dataset("Titanic")
```

```
ks2=kashti
ks2.head()
```

Out[ ]:

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

In [ ]:
```
ks2.isnull().sum()
```

Out[ ]:
```
survived         0
pclass           0
sex              0
age            177
sibsp            0
parch            0
fare             0
embarked         2
class            0
who              0
adult_male       0
deck           688
embark_town      2
alive            0
alone            0
dtype: int64
```

In [ ]:
```
mode=ks2['deck'].mode
mode
```

Out[ ]:
```
<bound method Series.mode of 0      NaN
1        C
2      NaN
3        C
4      NaN
      ...
886    NaN
887      B
888    NaN
889      C
890    NaN
Name: deck, Length: 891, dtype: object>
```

In [ ]:
```
ks2['deck']=ks2['deck'].replace(np.nan,mode)
```

In [ ]:
```
ks2.isnull().sum()
```

Out[ ]:
```
survived         0
pclass           0
sex              0
age            177
sibsp            0
parch            0
fare             0
embarked         2
class            0
who              0
adult_male       0
deck             0
embark_town      2
alive            0
alone            0
dtype: int64
```

In [ ]:
```
ks2['embark_town'].head()
```

Out[ ]:
```
0    Southampton
1      Cherbourg
2    Southampton
3    Southampton
4    Southampton
Name: embark_town, dtype: object
```

In [ ]:
```
ks2['embarked'].head()
```

Out[ ]:
```
0    S
1    C
2    S
3    S
4    S
Name: embarked, dtype: object
```

In [ ]:
```
mod=ks2['embark_town'].mode
mod
```

Out[ ]:
```
<bound method Series.mode of 0      Southampton
1        Cherbourg
2      Southampton
3      Southampton
4      Southampton
          ...
886    Southampton
887    Southampton
888    Southampton
889      Cherbourg
890     Queenstown
Name: embark_town, Length: 891, dtype: object>
```

In [ ]:
```
ks2['embark_town']=ks2['embark_town'].replace(np.nan,mod)
```

In [ ]:
```
ks2.isnull().sum()
```

Out[ ]:
```
survived         0
pclass           0
sex              0
age            177
sibsp            0
parch            0
fare             0
embarked         2
class            0
who              0
adult_male       0
deck             0
embark_town      0
alive            0
alone            0
dtype: int64
```

In [ ]:
```
mo=ks2['embarked'].mode
mo
```

Out[ ]:
```
<bound method Series.mode of 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>
```

In [ ]:
```
ks2['embarked']=ks2['embarked'].replace(np.nan,mo)
```

In [ ]:
```
ks2.isnull().sum()
```

Out[ ]:
```
survived         0
pclass           0
sex              0
age            177
sibsp            0
parch            0
fare             0
embarked         0
class            0
who              0
adult_male       0
deck             0
embark_town      0
alive            0
alone            0
dtype: int64
```

In [ ]:
```
mean=ks2['age'].mean()
mean
```

Out[ ]:
```
29.69911764705882
```

In [ ]:
```
ks2['age']=ks2['age'].replace(np.nan,mean)
```

In [ ]:
```
ks2.isnull().sum()
```

Out[ ]:
```
survived       0
pclass         0
sex            0
age            0
sibsp          0
parch          0
fare           0
embarked       0
class          0
who            0
adult_male     0
deck           0
embark_town    0
alive          0
alone          0
dtype: int64
```

In [ ]:
```
ks2.shape
```

Out[ ]:
```
(891, 15)
```

Data Formatting

In [ ]:
```
kashti=sns.load_dataset('titanic')
kashti.dtypes
```

Out[ ]:
```
survived          int64
pclass            int64
sex              object
age             float64
sibsp             int64
parch             int64
fare            float64
embarked         object
class          category
who              object
adult_male         bool
deck           category
embark_town      object
alive            object
alone              bool
dtype: object
```

In [ ]:
```
kashti['survived']=kashti['survived'].astype('float64')
kashti.dtypes
```

Out[ ]:
```
survived        float64
pclass            int64
sex              object
age             float64
sibsp             int64
parch             int64
fare            float64
embarked         object
class          category
who              object
adult_male         bool
deck           category
embark_town      object
alive            object
alone              bool
dtype: object
```

In [ ]:
```
kashti['survived']=kashti['survived'].astype('int64')
kashti.dtypes
```

Out[ ]:
```
survived          int64
pclass            int64
sex              object
age             float64
sibsp             int64
parch             int64
fare            float64
embarked         object
class          category
who              object
adult_male         bool
deck           category
embark_town      object
alive            object
alone              bool
dtype: object
```

In [ ]:
```
kashti['age']=kashti['age']*365
kashti['age'].head()
```

Out[ ]:
```
0     8030.0
1    13870.0
2     9490.0
3    12775.0
4    12775.0
Name: age, dtype: float64
```

In [ ]:
```
kashti.rename(columns={'age':'age in days'},inplace=True)
kashti.head()
```

Out[ ]:

	survived	pclass	sex	age in days	sibsp	parch	fare	embarked	class	who	adult_male	de
0	0	3	male	8030.0	1	0	7.2500	S	Third	man	True	N
1	1	1	female	13870.0	1	0	71.2833	C	First	woman	False	
2	1	3	female	9490.0	0	0	7.9250	S	Third	woman	False	N
3	1	1	female	12775.0	1	0	53.1000	S	First	woman	False	
4	0	3	male	12775.0	0	0	8.0500	S	Third	man	True	N

Assignment 2

In [ ]:
```
kashti['fare'].round(1)
```

Out[ ]:
```
0       7.2
1      71.3
2       7.9
3      53.1
4       8.0
       ...
886    13.0
887    30.0
888    23.4
889    30.0
890     7.8
Name: fare, Length: 891, dtype: float64
```

Data Normalization

- Simple feature Scaling
  - x(new)=x(old)/x(max)
- Min-Max method
- Z-score(Standard Score -3 to +3)
- Log Transformation

In [ ]:
```
kashti[['age in days','fare']].head()
```

Out[ ]:

	age in days	fare
0	8030.0	7.2500
1	13870.0	71.2833
2	9490.0	7.9250
3	12775.0	53.1000
4	12775.0	8.0500

In [ ]:
```
# Simple Feature Scaling
```

```
kashti['fare']=kashti['fare']/kashti['fare'].max()
kashti['fare'].head()
```

Out[ ]:
```
0    0.014151
1    0.139136
2    0.015469
3    0.103644
4    0.015713
Name: fare, dtype: float64
```

In [ ]:
```
# Min-Max method
kashti['fare']=(kashti['fare']-kashti['fare'].min())/(kashti['fare'].max()-kashti['fa
kashti['fare'].head()
```

Out[ ]:
```
0    0.014151
1    0.139136
2    0.015469
3    0.103644
4    0.015713
Name: fare, dtype: float64
```

In [ ]:
```
kashti['age in days']=kashti['age in days']/kashti['age in days'].max()
kashti['age in days']=(kashti['age in days']-kashti['age in days'].min())/(kashti['ag
kashti['age in days'].head()
```

Out[ ]:
```
0    0.271174
1    0.472229
2    0.321438
3    0.434531
4    0.434531
Name: age in days, dtype: float64
```

In [ ]:
```
kashti[['age in days','fare']].head()
```

Out[ ]:

	age in days	fare
0	0.271174	0.014151
1	0.472229	0.139136
2	0.321438	0.015469
3	0.434531	0.103644
4	0.434531	0.015713

In [ ]:
```
i=sns.load_dataset('titanic')
i.head(3)
```

Out[ ]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	NaN
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	C
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	NaN

In [ ]:
```
# Log Transformation
i['age']=np.log(i['age'])
i['age'].head(3)
```

```
C:\Users\786\AppData\Local\Programs\Python\Python39\lib\site-packages\pandas\core\arr
aylike.py:364: RuntimeWarning: divide by zero encountered in log
  result = getattr(ufunc, method)(*inputs, **kwargs)
C:\Users\786\AppData\Local\Programs\Python\Python39\lib\site-packages\pandas\core\arr
aylike.py:364: RuntimeWarning: invalid value encountered in log
  result = getattr(ufunc, method)(*inputs, **kwargs)
```

Out[ ]:
```
0    1.128508
1    1.291320
2    1.181143
Name: age, dtype: float64
```

Bining

In [ ]:
```
import seaborn as sns
import pandas as pd
import numpy as np
k=sns.load_dataset('titanic')
k.head(3)
```

Out[ ]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	NaN
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	C
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	NaN

In [ ]:
```
min(k['age'])
```

Out[ ]:
```
0.42
```

In [ ]:
```
max(k['age'])
```

Out[ ]:
```
80.0
```

In [ ]:
```
bins=np.linspace(min(k['age']),max(k['age']),4)
age_groups=['bachay','jawan','bhuray']
k['age']=pd.cut(k['age'],bins,labels=age_groups,include_lowest=True)
k['age']
```

Out[ ]:
```
0      bachay
1       jawan
2      bachay
3       jawan
4       jawan
        ...
886     jawan
887    bachay
888       NaN
889    bachay
890     jawan
Name: age, Length: 891, dtype: category
Categories (3, object): ['bachay' < 'jawan' < 'bhuray']
```

In [ ]:
```
k.head()
```

Out[ ]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	de
0	0	3	male	bachay	1	0	7.2500	S	Third	man	True	Na
1	1	1	female	jawan	1	0	71.2833	C	First	woman	False	
2	1	3	female	bachay	0	0	7.9250	S	Third	woman	False	Na
3	1	1	female	jawan	1	0	53.1000	S	First	woman	False	
4	0	3	male	jawan	0	0	8.0500	S	Third	man	True	Na

Converting Categories into dummies

1. Ease to use for computation
2. Male Female(0,1)

In [ ]:
```
pd.get_dummies(k['sex'])
```

Out[ ]:

	female	male
0	0	1
1	1	0
2	1	0
3	1	0
4	0	1
...	...	...
886	0	1
887	1	0
888	1	0
889	0	1
890	0	1

891 rows × 2 columns

In [ ]:
```
k.head()
```

Out[ ]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	de
0	0	3	male	bachay	1	0	7.2500	S	Third	man	True	Na
1	1	1	female	jawan	1	0	71.2833	C	First	woman	False	
2	1	3	female	bachay	0	0	7.9250	S	Third	woman	False	Na
3	1	1	female	jawan	1	0	53.1000	S	First	woman	False	
4	0	3	male	jawan	0	0	8.0500	S	Third	man	True	Na

In [ ]:
```
pd.get_dummies(k['sex'],drop_first=True,sparse=True)
```

Out[ ]:

	male
0	1
1	0
2	0
3	0
4	1
...	...
886	1
887	0
888	0
889	1
890	1

891 rows × 1 columns

Assignment 3

In [ ]:
```
k=sns.load_dataset('titanic')
k.head(3)
```

Out[ ]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	NaN
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	C
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	NaN

In [ ]:
```
dummies = pd.get_dummies(k[['sex']], drop_first=False)
k = pd.concat([k.drop(['sex'],axis=1), dummies],axis=1)
```

In [ ]:
```
k.head()
```

Out[ ]:

	survived	pclass	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embarl
0	0	3	22.0	1	0	7.2500	S	Third	man	True	NaN	Southa
1	1	1	38.0	1	0	71.2833	C	First	woman	False	C	Che
2	1	3	26.0	0	0	7.9250	S	Third	woman	False	NaN	Southa
3	1	1	35.0	1	0	53.1000	S	First	woman	False	C	Southa
4	0	3	35.0	0	0	8.0500	S	Third	man	True	NaN	Southa