Data Wrangling

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

import matplotlib.pyplot as  plt
import seaborn as sns
```

In [ ]:
```
titanic=sns.load_dataset("titanic")
titanic
tt1=titanic
tt2=titanic
```

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

Out[ ]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town	alive	alone
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	NaN	Southampton	no	False
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	C	Cherbourg	yes	False
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	NaN	Southampton	yes	True
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	C	Southampton	yes	False
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	NaN	Southampton	no	True

In [ ]:
```
#simple opration
(titanic["age"]+2).head(10)
```

Out[ ]:
```
0    24.0
1    40.0
2    28.0
3    37.0
4    37.0
5     NaN
6    56.0
7     4.0
8    29.0
9    16.0
Name: age, dtype: float64
```

In [ ]:
```
# missing values
titanic.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 [ ]:
```
# use dropna method for missing values
titanic.dropna(subset=["deck"], axis=0, inplace=True)
titanic.shape
```

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

In [ ]:
```
titanic.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 [ ]:
```
#titanic=titanic.dropna()
```

In [ ]:
```
titanic=titanic.dropna()
titanic.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 [ ]:
```
titanic.shape
```

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

In [ ]:
```
tt1.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
```

Replacing missing values with the average of that columns

In [ ]:
```
#finding an average (mean)
# mtlb k hum missing value bhi average k through change kr sakty han aur dropna k through bhi
mean = tt1["age"].mean()
mean
```

Out[ ]:
```
30.7794562173913
```

In [ ]:
```
#replacing nan with mean of the data (updating as well)
tt1["age"]=tt1["age"].replace(np.nan, mean)
```

In [ ]:
```
tt1.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
```

In [ ]:
```
# know the data type
titanic.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 [ ]:
```
# use this method to convert data type from one format to another
titanic["survived"]=titanic["survived"].astype("float64")
titanic.dtypes
```

```
C:\Users\MP\AppData\Roaming\Python\Python37\site-packages\ipykernel_launcher.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

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 [ ]:
```
# here we will convert the age into days instead of years
titanic["age"]=titanic["age"]*365
titanic.head(10)
```

```
C:\Users\MP\AppData\Roaming\Python\Python37\site-packages\ipykernel_launcher.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

Out[ ]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town	alive	alone
1	1.0	1	female	13870.0	1	0	71.2833	C	First	woman	False	C	Cherbourg	yes	False
3	1.0	1	female	12775.0	1	0	53.1000	S	First	woman	False	C	Southampton	yes	False
6	0.0	1	male	19710.0	0	0	51.8625	S	First	man	True	E	Southampton	no	True
10	1.0	3	female	1460.0	1	1	16.7000	S	Third	child	False	G	Southampton	yes	False
11	1.0	1	female	21170.0	0	0	26.5500	S	First	woman	False	C	Southampton	yes	True
21	1.0	2	male	12410.0	0	0	13.0000	S	Second	man	True	D	Southampton	yes	True
23	1.0	1	male	10220.0	0	0	35.5000	S	First	man	True	A	Southampton	yes	True
27	0.0	1	male	6935.0	3	2	263.0000	S	First	man	True	C	Southampton	no	False
52	1.0	1	female	17885.0	1	0	76.7292	C	First	woman	False	D	Cherbourg	yes	False
54	0.0	1	male	23725.0	0	1	61.9792	C	First	man	True	B	Cherbourg	no	False

In [ ]:
```
# always rename afterwards
titanic.rename(columns={"age":"age in days"}, inplace=True)
titanic.head()
```

```
C:\Users\MP\Desktop\python\lib\site-packages\pandas\core\frame.py:5047: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  errors=errors)
```

Out[ ]:

	survived	pclass	sex	age in days	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town	alive	alone
1	1.0	1	female	13870.0	1	0	71.2833	C	First	woman	False	C	Cherbourg	yes	False
3	1.0	1	female	12775.0	1	0	53.1000	S	First	woman	False	C	Southampton	yes	False
6	0.0	1	male	19710.0	0	0	51.8625	S	First	man	True	E	Southampton	no	True
10	1.0	3	female	1460.0	1	1	16.7000	S	Third	child	False	G	Southampton	yes	False
11	1.0	1	female	21170.0	0	0	26.5500	S	First	woman	False	C	Southampton	yes	True

In [ ]:
```
#tt4=tt1
```

Data Normalization

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

In [ ]:
```
titanic1=titanic[["age in days", "fare"]]
titanic.head()
```

Method of Normalization

In [ ]:
```
#simple feature scaling
titanic["fare"]=titanic["fare"]/titanic["fare"].max()
titanic.head()
```

```
C:\Users\MP\AppData\Roaming\Python\Python37\site-packages\ipykernel_launcher.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

Out[ ]:

	survived	pclass	sex	age in days	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town	alive	alone
1	1.0	1	female	13870.0	1	0	0.139136	C	First	woman	False	C	Cherbourg	yes	False
3	1.0	1	female	12775.0	1	0	0.103664	S	First	woman	False	C	Southampton	yes	False
6	0.0	1	male	19710.0	0	0	0.101229	S	First	man	True	E	Southampton	no	True
10	1.0	3	female	1460.0	1	1	0.032596	S	Third	child	False	G	Southampton	yes	False
11	1.0	1	female	21170.0	0	0	0.051822	S	First	woman	False	C	Southampton	yes	True

In [ ]:
```
#simple feature scaling
titanic["age in days"]=titanic["age in days"]/titanic["age in days"].max()
titanic.head()
```

```
C:\Users\MP\AppData\Roaming\Python\Python37\site-packages\ipykernel_launcher.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

Out[ ]:

	survived	pclass	sex	age in days	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town	alive	alone
1	1.0	1	female	13870.0	1	0	0.139136	C	First	woman	False	C	Cherbourg	yes	False
3	1.0	1	female	12775.0	1	0	0.103664	S	First	woman	False	C	Southampton	yes	False
6	0.0	1	male	19710.0	0	0	0.101229	S	First	man	True	E	Southampton	no	True
10	1.0	3	female	1460.0	1	1	0.032596	S	Third	child	False	G	Southampton	yes	False
11	1.0	1	female	21170.0	0	0	0.051822	S	First	woman	False	C	Southampton	yes	True

In [ ]:
```
# min and max method
titanic["fare"]=(titanic["fare"]-titanic["fare"].min())/(titanic["fare"].max()-titanic["fare"].min())
titanic.head()
```

```
C:\Users\MP\AppData\Roaming\Python\Python37\site-packages\ipykernel_launcher.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

Out[ ]:

	survived	pclass	sex	age in days	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town	alive	alone
1	1.0	1	female	13870.0	1	0	1.0	C	First	woman	False	C	Cherbourg	yes	False
3	1.0	1	female	12775.0	1	0	1.0	S	First	woman	False	C	Southampton	yes	False
6	0.0	1	male	19710.0	0	0	1.0	S	First	man	True	E	Southampton	no	True
10	1.0	3	female	1460.0	1	1	1.0	S	Third	child	False	G	Southampton	yes	False
11	1.0	1	female	21170.0	0	0	1.0	S	First	woman	False	C	Southampton	yes	True

In [ ]:
```
#z-score (standard score)
titanic["fare"]=(titanic["fare"]-titanic["fare"].mean())/titanic["fare"].std()
titanic.head()
```

```
C:\Users\MP\AppData\Roaming\Python\Python37\site-packages\ipykernel_launcher.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

Out[ ]:

	survived	pclass	sex	age in days	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town	alive	alone
1	1.0	1	female	13870.0	1	0	NaN	C	First	woman	False	C	Cherbourg	yes	False
3	1.0	1	female	12775.0	1	0	NaN	S	First	woman	False	C	Southampton	yes	False
6	0.0	1	male	19710.0	0	0	NaN	S	First	man	True	E	Southampton	no	True
10	1.0	3	female	1460.0	1	1	NaN	S	Third	child	False	G	Southampton	yes	False
11	1.0	1	female	21170.0	0	0	NaN	S	First	woman	False	C	Southampton	yes	True

In [ ]:
```
# log transformation
titanic["fare"]=np.log(titanic["fare"])
titanic.head()
```

```
C:\Users\MP\AppData\Roaming\Python\Python37\site-packages\ipykernel_launcher.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

Out[ ]:

	survived	pclass	sex	age in days	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town	alive	alone
1	1.0	1	female	13870.0	1	0	NaN	C	First	woman	False	C	Cherbourg	yes	False
3	1.0	1	female	12775.0	1	0	NaN	S	First	woman	False	C	Southampton	yes	False
6	0.0	1	male	19710.0	0	0	NaN	S	First	man	True	E	Southampton	no	True
10	1.0	3	female	1460.0	1	1	NaN	S	Third	child	False	G	Southampton	yes	False
11	1.0	1	female	21170.0	0	0	NaN	S	First	woman	False	C	Southampton	yes	True

converting categories into dummies

- Easy to use of computation
- *male* female (0,1)

In [ ]:
```
pd.get_dummies(titanic["fare"])
titanic.head()
```

Out[ ]:

	survived	pclass	sex	age in days	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town	alive	alone
1	1.0	1	female	13870.0	1	0	NaN	C	First	woman	False	C	Cherbourg	yes	False
3	1.0	1	female	12775.0	1	0	NaN	S	First	woman	False	C	Southampton	yes	False
6	0.0	1	male	19710.0	0	0	NaN	S	First	man	True	E	Southampton	no	True
10	1.0	3	female	1460.0	1	1	NaN	S	Third	child	False	G	Southampton	yes	False
11	1.0	1	female	21170.0	0	0	NaN	S	First	woman	False	C	Southampton	yes	True