### **Data Wrangling**

```
In [1]:
          #install libraries
          #pip install pandas
          #pip install seaborn
          #pip install numpy
In [2]:
          #import llibraries
          import pandas as pd
          import numpy as np
          import seaborn as sns
In [3]:
          kashti = sns.load dataset('titanic')
          ks1 = kashti
          ks2 = kashti
          ks = sns.load dataset('titanic')
In [4]:
          kashti.head()
Out[4]:
            survived pclass
                               sex
                                    age sibsp parch
                                                        fare
                                                              embarked class
                                                                                 who adult_male deck embark_town
         0
                  0
                         3
                              male
                                   22.0
                                                      7.2500
                                                                       Third
                                                                                            True
                                                                                                 NaN
                                                                                                        Southampton
                                            1
                                                   0
                                                                                 man
         1
                         1 female
                                   38.0
                                            1
                                                   0 71.2833
                                                                         First woman
                                                                                           False
                                                                                                    C
                                                                                                          Cherbourg
         2
                         3
                            female
                                   26.0
                                            0
                                                       7.9250
                                                                        Third woman
                                                                                           False
                                                                                                 NaN
                                                                                                        Southampton
                                                   0 53.1000
         3
                            female
                                   35.0
                                            1
                                                                         First woman
                                                                                           False
                                                                                                    C
                                                                                                        Southampton
                         3
                              male 35.0
                                            0
                                                      8.0500
                                                                       Third
                                                                                 man
                                                                                            True
                                                                                                 NaN
                                                                                                        Southampton
In [5]:
          # simple operations (Math operator)
          (kashti['age']+12).head(10)
              34.0
Out[5]:
              50.0
         1
         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 [6]:
          #where exactly missing values are?
          kashti.isnull().sum()
                           0
         survived
Out[6]:
                           0
         pclass
                           0
         sex
```

```
sibsp
                          0
         parch
                          0
         fare
                          0
                          2
         embarked
                          0
         class
         who
                          0
                          0
         adult male
                        688
         deck
         embark_town
                          2
         alive
                          0
         alone
                          0
         dtype: int64
In [7]:
         #usd drop.na method
         print(kashti.shape)
         #kashti.dropna(subset=['deck'], axis=0, inplace=True) # this will remove specifically deck
         #inplace = true modifies the data frame
         (891, 15)
In [8]:
         kashti.isnull().sum()
                          0
         survived
Out[8]:
                          0
         pclass
                          0
         sex
                        177
         age
                          0
         sibsp
                          0
         parch
                          0
         fare
         embarked
                          2
                          0
         class
                          0
         who
         adult_male
                          0
                        688
         deck
         embark_town
                          2
         alive
                          0
                          0
         alone
         dtype: int64
In [9]:
         kashti = kashti.dropna()
         kashti.isnull().sum() # remove na from whole dataframe
                        0
         survived
Out[9]:
                        0
         pclass
         sex
                        0
                        0
         age
                        0
         sibsp
         parch
                        0
         fare
                        0
                        0
         embarked
         class
                        0
                        0
         who
         adult_male
                        0
                        0
         deck
                        0
         embark_town
         alive
                        0
         alone
                        0
         dtype: int64
```

177

age

```
In [10]:
           kashti.shape
          (182, 15)
Out[10]:
In [11]:
           ks1.isnull().sum()
          survived
                            0
Out[11]:
          pclass
                            0
                            0
                          177
          age
                            0
          sibsp
                            0
          parch
          fare
                            0
                            2
          embarked
                            0
          class
                            0
          who
                            0
          adult male
                          688
          deck
          embark_town
                            2
          alive
          alone
                            0
          dtype: int64
```

## Replacing missing values with the average of that column

```
In [12]:
          #finding an average (mean)
          mean = ks1['age'].mean()
          mean
          29.69911764705882
Out[12]:
In [13]:
          # replacing nan with mean of the data (updating as well)
          ks1['age'] = ks1['age'].replace(np.nan, mean)
In [14]:
          ks1.isnull().sum()
          survived
                           0
Out[14]:
          pclass
                           0
                           0
          age
          sibsp
                           0
          parch
                           0
          fare
                           2
          embarked
          class
                           0
                           0
          who
                           0
          adult male
                         688
          deck
                           2
          embark_town
                           0
          alive
          alone
          dtype: int64
In [15]:
```

```
ks1.dropna(subset=['embarked'], axis=0, inplace=True)
          ks1.dropna(subset=['embark_town'], axis=0, inplace=True)
In [16]:
          ks1.isnull().sum()
         survived
                        0
Out[16]:
                        0
         pclass
                        0
         sex
                        0
         age
         sibsp
                        0
         parch
                        0
         fare
                        0
         embarked
         class
                        0
                        0
         who
         adult_male
                        0
         deck
         embark_town
                        0
         alive
                        0
         alone
         dtype: int64
        Data Formatting
In [17]:
          # know the data type and convert it into the known one
          kashti.dtypes
         survived
                           int64
Out[17]:
         pclass
                           int64
                          object
         sex
         age
                         float64
                           int64
         sibsp
                           int64
         parch
                         float64
         fare
         embarked
                         object
         class
                        category
         who
                          object
         adult_male
                            bool
         deck
                        category
                          object
         embark_town
         alive
                          object
         alone
                            bool
         dtype: object
In [18]:
          # use this method to convert datatype from one to another format
          kashti['survived'] = kashti['survived'].astype("float64")
          kashti.dtypes
         C:\Users\eAgLe\AppData\Local\Temp/ipykernel_9780/2526506595.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/inde
         xing.html#returning-a-view-versus-a-copy
           kashti['survived'] = kashti['survived'].astype("float64")
```

ks1.dropna(subset=['deck'], axis=0, inplace=True)

float64

survived

Out[18]:

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

In [19]:

```
# here we will convert the age into days instead of years
ks1['age'] = ks1['age']*365
ks1.head(10)
```

е	deck	adult_male	who	class	embarked	fare	parch	sibsp	age	sex	pclass	survived	[19]:
	C	False	woman	First	С	71.2833	0	1	13870.000000	female	1	1	1
•	C	False	woman	First	S	53.1000	0	1	12775.000000	female	1	1	3
•	Е	True	man	First	S	51.8625	0	0	19710.000000	male	1	0	6
•	G	False	child	Third	S	16.7000	1	1	1460.000000	female	3	1	10
•	C	False	woman	First	S	26.5500	0	0	21170.000000	female	1	1	11
•	D	True	man	Second	S	13.0000	0	0	12410.000000	male	2	1	21
	Α	True	man	First	S	35.5000	0	0	10220.000000	male	1	1	23
	C	True	man	First	S	263.0000	2	3	6935.000000	male	1	0	27
	В	False	woman	First	С	146.5208	0	1	10840.177941	female	1	1	31
	D	False	woman	First	С	76.7292	0	1	17885.000000	female	1	1	52
<b>&gt;</b>													4

In [20]:

#here is the assignment to remove multipke 0's

In [21]:

```
# always rename afterwards
ks1.rename(columns={"age": "age in days"}, inplace=True)
ks1.head()
```

Out[21]: age in fare embarked class who adult\_male deck embark\_to survived pclass sibsp parch sex days 0 71.2833 1 1 female 13870.0 C False 1 First woman 3 1 female 12775.0 1 0 53.1000 S First woman False

3	1	1	female	12775.0	1	0	53.1000	9	5	First	woman	False	C	Southamp
6	0	1	male	19710.0	0	0	51.8625	S	5	First	man	True	Е	Southamp
10	1	3	female	1460.0	1	1	16.7000	9	5	Third	child	False	G	Southamp
11	1	1	female	21170.0	0	0	26.5500	9	5	First	woman	False	С	Southamp

C

Cherbc

In [22]: ks1['age in days'] = ks1['age in days'].astype("int64") ks1.head(10) Out[22]: age adult\_male deck embark\_ survived pclass sex in sibsp parch fare embarked class days 1 1 female 13870 0 71.2833 C C Cherk 1 First woman False 3 S 1 C female 12775 1 0 53.1000 First woman **False** Southan 0 S 6 19710 0 0 Ε Southan male 51.8625 First True man 10 1 female 1460 1 1 16.7000 S Third child False G Southan

31 1 female 10840 1 146.5208 C First woman False В Cherk C 52 1 female 17885 1 76.7292 First woman False D Cherk

26.5500

13.0000

35.5000

263.0000

S

S

S

S

First

First

First

Second

woman

man

man

man

False

True

True

True

C

D

C

Southan

Southan

Southan

Southan

#### **Data Normalization**

female

male

male

male

2

1

1

21170

12410

10220

6935

0

0

0

3

0

0

0

2

In [23]: kashti.head()

Out[23]: survived pclass sibsp parch embarked class adult\_male deck embark\_town sex age fare who 1 1.0 71.2833 C C female 38.0 First woman False Cherbourg 3 S 1.0 female 35.0 53.1000 First woman False Southampton 6 S 0.0 male 54.0 0 51.8625 First man True Southampton 10 1.0 female 4.0 16.7000 S Third child False Southampton 11 1.0 female 58.0 0 S Southampton 26.5500 First woman False

In [24]:
 ks4 = kashti[["age", "fare"]]
 ks4.head()

Out[24]: age fare

1 38.0 71.2833

3 35.0 53.1000

6 54.0 51.8625

10 4.0 16.7000

4

11

21

23

27

1

1

1

0

```
age fare
11 58.0 26.5500
```

**10** 0.0500 0.032596

#### Method of Normalization

```
In [25]:
          # simle feature scaling
          ks4['fare'] = ks4['fare']/ks4['fare'].max()
          ks4['age'] = ks4['age']/ks4['age'].max()
          ks4.head()
         C:\Users\eAgLe\AppData\Local\Temp/ipykernel 9780/1199163970.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/inde
         xing.html#returning-a-view-versus-a-copy
           ks4['fare'] = ks4['fare']/ks4['fare'].max()
         C:\Users\eAgLe\AppData\Local\Temp/ipykernel_9780/1199163970.py:3: 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/inde
         xing.html#returning-a-view-versus-a-copy
           ks4['age'] = ks4['age']/ks4['age'].max()
Out[25]:
                       fare
               age
          1 0.4750 0.139136
          3 0.4375 0.103644
          6 0.6750 0.101229
          10 0.0500 0.032596
         11 0.7250 0.051822
In [26]:
          # Min - Max Method
          ks4['fare'] = (ks4['fare']-ks4['fare'].min()) / (ks4['fare'].max()-ks4['fare'].min())
          ks4.head()
         C:\Users\eAgLe\AppData\Local\Temp/ipykernel 9780/410330791.py:3: 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/inde
         xing.html#returning-a-view-versus-a-copy
           ks4['fare'] = (ks4['fare']-ks4['fare'].min()) / (ks4['fare'].max()-ks4['fare'].min())
Out[26]:
               age
                        fare
          1 0.4750 0.139136
          3 0.4375 0.103644
          6 0.6750 0.101229
```

```
11 0.7250 0.051822
In [27]:
           # Z-score (standard score)
           ks4['fare'] = (ks4['fare']-ks4['fare'].mean()) / ks4['fare'].std()
           ks4.head()
          C:\Users\eAgLe\AppData\Local\Temp/ipykernel_9780/1430778810.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/inde
          xing.html#returning-a-view-versus-a-copy
            ks4['fare'] = (ks4['fare']-ks4['fare'].mean()) / ks4['fare'].std()
Out[27]:
                age
                         fare
           1 0.4750 -0.099835
             0.4375 -0.337554
             0.6750 -0.353732
             0.0500 -0.813428
          11 0.7250 -0.684654
In [28]:
           # log transformation
           ks['fare'] = np.log(ks['fare'])
           ks.head()
          H:\download\Anaconda\lib\site-packages\pandas\core\arraylike.py:364: RuntimeWarning: divide by zer
          o encountered in log
            result = getattr(ufunc, method)(*inputs, **kwargs)
Out[28]:
             survived pclass
                               sex age sibsp parch
                                                         fare embarked class
                                                                                 who
                                                                                      adult_male deck
                                                                                                       embark_town
          0
                   0
                          3
                                    22.0
                                                   0 1.981001
                                                                      S Third
                                                                                                  NaN
                              male
                                                                                 man
                                                                                             True
                                                                                                        Southampton
          1
                            female
                                    38.0
                                                   0 4.266662
                                                                                                    C
                          1
                                                                          First woman
                                                                                            False
                                                                                                          Cherbourg
          2
                          3
                            female
                                    26.0
                                            0
                                                   0 2.070022
                                                                         Third
                                                                               woman
                                                                                            False
                                                                                                  NaN
                                                                                                        Southampton
          3
                                   35.0
                                                   0 3.972177
                                                                      S
                                                                                                    C
                             female
                                                                          First woman
                                                                                            False
                                                                                                        Southampton
                                                                      S Third
                          3
                              male 35.0
                                            0
                                                   0 2.085672
                                                                                             True
                                                                                                  NaN
                                                                                                        Southampton
                                                                                 man
```

## **Binning**

age

fare

```
In [29]: kashti = sns.load_dataset('titanic')
```

## assignment given in the video

```
In [30]: bins = np.linspace(min(kashti['age']), max(kashti['age']), 4)
```

```
age_groups = ["Bachay", "Jawan", "Boorhay"]
           kashti['age'] = pd.cut(kashti['age'], bins, labels=age_groups, include_lowest=True)
           kashti['age']
           # how this will change the anames in dataset based on grouping? (Assignment)
                  Bachay
Out[30]:
                    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' < 'Boorhay']</pre>
In [31]:
           kashti.head(10)
Out[31]:
                                                                      embarked
                                                                                            who
                                                                                                 adult_male deck embark_t
              survived pclass
                                  sex
                                          age
                                               sibsp
                                                      parch
                                                                fare
                                                                                   class
           0
                     0
                            3
                                                              7.2500
                                                                                   Third
                                 male
                                        Bachay
                                                                                                        True
                                                                                                              NaN
                                                                                                                    Southam
                                                                                            man
                                                                              C
           1
                     1
                               female
                                        Jawan
                                                             71.2833
                                                                                    First woman
                                                                                                       False
                                                                                                                C
                                                                                                                       Cherk
                            1
                                                                              S
                                                                                   Third woman
           2
                            3
                               female
                                        Bachay
                                                   0
                                                              7.9250
                                                                                                       False
                                                                                                              NaN
                                                                                                                    Southam
           3
                                                                              S
                               female
                                                             53.1000
                                                                                    First woman
                                                                                                       False
                                                                                                                    Southam
                                        Jawan
                            1
                            3
                                male
                                        Jawan
                                                   0
                                                          0
                                                              8.0500
                                                                              S
                                                                                   Third
                                                                                                        True
                                                                                                              NaN
                                                                                                                    Southam
                                                                                            man
           5
                            3
                                                                              Q
                                                                                   Third
                                male
                                          NaN
                                                   0
                                                          0
                                                              8.4583
                                                                                                        True
                                                                                                              NaN
                                                                                                                     Queens
                                                                                            man
           6
                            1
                                                                              S
                                                                                    First
                                                                                                                    Southam
                                male
                                       Boorhay
                                                   0
                                                          0 51.8625
                                                                                                        True
                                                                                            man
           7
                            3
                                male
                                        Bachay
                                                   3
                                                             21.0750
                                                                              S
                                                                                   Third
                                                                                            child
                                                                                                       False
                                                                                                              NaN
                                                                                                                    Southam
           8
                            3
                               female
                                        Jawan
                                                   0
                                                          2 11.1333
                                                                                   Third
                                                                                         woman
                                                                                                       False
                                                                                                              NaN
                                                                                                                    Southam
                               female
                                        Bachay
                                                          0 30.0708
                                                                                Second
                                                                                            child
                                                                                                       False
                                                                                                              NaN
                                                                                                                       Cherk
```

#### converting categories into dummies

- easy to use for computation
- Male Female (0,1)

# how to use get dummies to change data inside a dataframe (Assignment)

#### 1st method

```
In [51]: pd.get_dummies(ks["sex"],prefix="Sex",columns=ks["sex"])
Out[51]: Sex_female Sex_male
```

	Sex_female	Sex_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 [56]: # df = Load_data() # reset dataframe
    df = pd.get_dummies(ks, columns=['sex'])
    df
```

Out[56]:		survived	pclass	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town	а
	0	0	3	22.0	1	0	1.981001	S	Third	man	True	NaN	Southampton	
	1	1	1	38.0	1	0	4.266662	С	First	woman	False	С	Cherbourg	
	2	1	3	26.0	0	0	2.070022	S	Third	woman	False	NaN	Southampton	
	3	1	1	35.0	1	0	3.972177	S	First	woman	False	C	Southampton	
	4	0	3	35.0	0	0	2.085672	S	Third	man	True	NaN	Southampton	
	•••											•••		
	886	0	2	27.0	0	0	2.564949	S	Second	man	True	NaN	Southampton	
	887	1	1	19.0	0	0	3.401197	S	First	woman	False	В	Southampton	
	888	0	3	NaN	1	2	3.154870	S	Third	woman	False	NaN	Southampton	
	889	1	1	26.0	0	0	3.401197	C	First	man	True	C	Cherbourg	
	890	0	3	32.0	0	0	2.047693	Q	Third	man	True	NaN	Queenstown	

891 rows × 16 columns

## 2nd method

```
In [58]: ks = sns.load_dataset('titanic')
    ks.head()
```

	survived	pclass	sex	age	sibsp	parch	fare	embarked	l clas	s who	adult_male	deck	embark_town
0	0	3	male	22.0	1	C	7.2500	9	5 Third	l man	ı True	NaN	Southampton
1	1	1	female	38.0	1	C	71.2833	C	E Firs	t woman	r False	C	Cherbourg
2	1	3	female	26.0	0	C	7.9250	9	5 Third	l woman	r False	NaN	Southampton
3	1	1	female	35.0	1	C	53.1000	S	S Firs	t woman	ralse	C	Southampton
4	0	3	male	35.0	0	C	8.0500	S	5 Third	l man	True	NaN	Southampton
4													<b> </b>
N3		pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town
0	0	3			1	0	7.2500	S	Third	man		NaN	
1								•		man	iiue	INGIN	Southampton
	1	1	NaN	38.0	1	0	71.2833	С		woman	False	С	Southampton Cherbourg
2	1	1		38.0 26.0	1	0	71.2833 7.9250	С	First				•
			NaN	26.0				С	First Third	woman	False	С	Cherbourg
2	1	3	NaN NaN	26.0	0	0	7.9250	C S S	First Third	woman woman	False False	C NaN	Cherbourg Southampton
2	1	3	NaN NaN	26.0 35.0	0	0	7.9250 53.1000	C S S	First Third First	woman woman woman	False False False	C NaN C	Cherbourg Southampton Southampton
	1 2 3 4 4 M Ks ks	0 0 1 1 2 1 3 1 4 0 4 #here i co ks['sex'] ks.head() survived	0 0 3 1 1 1 2 1 3 3 1 1 4 0 3  #here i can dire ks['sex'] = ks[' ks.head()  survived pclass	<pre>0     0     3     male 1     1     1     female 2     1     3     female 3     1     1     female 4     0     3     male  #here i can directly conditions ks['sex'] = ks['sex']. ks.head()  survived pclass sex</pre>	<pre>0     0     3     male     22.0 1     1     1     female     38.0 2     1     3     female     26.0 3     1     1     female     35.0 4     0     3     male     35.0  #here i can directly call tks['sex'] = ks['sex'].map({ks.head()}</pre> <pre>survived pclass sex age</pre>	<pre>0     0     3     male 22.0     1 1     1     1     female 38.0     1 2     1     3     female 26.0     0 3     1     1     female 35.0     1 4     0     3     male 35.0     0  #here i can directly call the funks['sex'] = ks['sex'].map({'femalks.head()}</pre> <pre>survived pclass sex age sibsp</pre>	<pre>0  0  3  male 22.0  1  0 1  1  1  female 38.0  1  0 2  1  3  female 26.0  0  0 3  1  1  female 35.0  1  0 4  0  3  male 35.0  0  0  #here i can directly call the function ks['sex'] = ks['sex'].map({'female': 1, ks.head()}</pre> <pre>survived pclass sex age sibsp parch</pre>	<pre>0     0     3     male 22.0     1     0 7.2500 1     1     1 female 38.0     1     0 71.2833 2     1     3 female 26.0     0     0 7.9250 3     1     1 female 35.0     1     0 53.1000 4     0     3 male 35.0     0     0 8.0500  #here i can directly call the function and pass ks['sex'] = ks['sex'].map({'female': 1, 'male': ks.head()}</pre> #survived pclass sex age sibsp parch fare	<pre>0     0     3     male 22.0     1     0 7.2500     5 1     1     1 female 38.0     1     0 71.2833</pre>	0 0 3 male 22.0 1 0 7.2500 S Third 1 1 1 female 38.0 1 0 71.2833 C First 2 1 3 female 26.0 0 0 7.9250 S Third 3 1 1 female 35.0 1 0 53.1000 S First 4 0 3 male 35.0 0 0 8.0500 S Third  #here i can directly call the function and pass the value ks['sex'] = ks['sex'].map({'female': 1, 'male': 0}) ks.head()  survived pclass sex age sibsp parch fare embarked class	0	0	0