

# Muhammad Arham Adeel

## Pandas 3rd Assignment

### 11- Copy data from clipboard

```
In [1]: # dataset download
import seaborn as sns
import pandas as pd

df = sns.load_dataset("titanic")
df.to_excel("kashti.xlsx")
```

```
In [5]: # read clipboard in python
df = pd.read_clipboard()
display(df)
df.to_csv("kashti_half.csv")
```

	sex	age	sibsp	parch	fare	embarked	class	who
0	male	22.0	1	0	7.2500	S	Third	man
1	female	38.0	1	0	71.2833	C	First	woman
2	female	26.0	0	0	7.9250	S	Third	woman
3	female	35.0	1	0	53.1000	S	First	woman
4	male	35.0	0	0	8.0500	S	Third	man
5	male	NaN	0	0	8.4583	Q	Third	man
6	male	54.0	0	0	51.8625	S	First	man
7	male	2.0	3	1	21.0750	S	Third	child
8	female	27.0	0	2	11.1333	S	Third	woman
9	female	14.0	1	0	30.0708	C	Second	child
10	female	4.0	1	1	16.7000	S	Third	child
11	female	58.0	0	0	26.5500	S	First	woman
12	male	20.0	0	0	8.0500	S	Third	man
13	male	39.0	1	5	31.2750	S	Third	man
14	female	14.0	0	0	7.8542	S	Third	child
15	female	55.0	0	0	16.0000	S	Second	woman
16	male	2.0	4	1	29.1250	Q	Third	child

### 12- Split DataFrame into two subsets

```
In [6]: # dataset download
import seaborn as sns
import pandas as pd

df = sns.load_dataset("titanic")
df.head()
```

```
Out[6]:
```

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	e
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	NaN	5
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	5
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	C	5
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	NaN	5



```
In [7]: df.shape
```

```
Out[7]: (891, 15)
```

```
In [8]: # Splitting it at 50% randomly
kashti_1 = df.sample(frac=0.5, random_state=1)
kashti_1.shape
```

```
Out[8]: (446, 15)
```

```
In [9]: # Subtracting the 50% from the entire dataset to get the other half
kashti_2 = df.drop(kashti_1.index)
kashti_2.shape
```

```
Out[9]: (445, 15)
```

```
In [10]: kashti_1.head()
```

```
Out[10]:
```

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	e
862	1	1	female	48.0	0	0	25.9292	S	First	woman	False		
223	0	3	male	NaN	0	0	7.8958	S	Third	man	True	Na	
84	1	2	female	17.0	0	0	10.5000	S	Second	woman	False	Na	
680	0	3	female	NaN	0	0	8.1375	Q	Third	woman	False	Na	
535	1	2	female	7.0	0	2	26.2500	S	Second	child	False	Na	



```
In [11]: kashti_2.head()
```

Out[11]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	C
7	0	3	male	2.0	3	1	21.0750	S	Third	child	False	NaN
10	1	3	female	4.0	1	1	16.7000	S	Third	child	False	G
15	1	2	female	55.0	0	0	16.0000	S	Second	woman	False	NaN
18	0	3	female	31.0	1	0	18.0000	S	Third	woman	False	NaN

In [12]:

```
# Check the num of rows of the sub-datasets together
len(kashti_1) + len(kashti_2)
```

Out[12]: 891

## 13- Join two Datasets

In [13]:

```
# 1st-method to join both the sub-datasets
dataset = [kashti_1, kashti_2]
df1 = pd.concat(dataset)
df1
```

Out[13]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	dec
862	1	1	female	48.0	0	0	25.9292	S	First	woman	False	
223	0	3	male	NaN	0	0	7.8958	S	Third	man	True	Na
84	1	2	female	17.0	0	0	10.5000	S	Second	woman	False	Na
680	0	3	female	NaN	0	0	8.1375	Q	Third	woman	False	Na
535	1	2	female	7.0	0	2	26.2500	S	Second	child	False	Na
...	...	...	...	...	...	...	...	...	...	...	...	...
883	0	2	male	28.0	0	0	10.5000	S	Second	man	True	Na
884	0	3	male	25.0	0	0	7.0500	S	Third	man	True	Na
886	0	2	male	27.0	0	0	13.0000	S	Second	man	True	Na
889	1	1	male	26.0	0	0	30.0000	C	First	man	True	
890	0	3	male	32.0	0	0	7.7500	Q	Third	man	True	Na

891 rows × 15 columns

## 14- Filtering a Dataset

In [14]:

```
df.head()
```

Out[14]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	e
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	NaN	9
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	C	9
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	NaN	9
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	C	9
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	NaN	9

In [15]: `df["sex"].unique()`

Out[15]: `array(['male', 'female'], dtype=object)`

In [16]: `df[df["sex"]=="female"]`

Out[16]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	e
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False		
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	Na	
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False		
8	1	3	female	27.0	0	2	11.1333	S	Third	woman	False	Na	
9	1	2	female	14.0	1	0	30.0708	C	Second	child	False	Na	
...	...	...	...	...	...	...	...	...	...	...	...	...	
880	1	2	female	25.0	0	1	26.0000	S	Second	woman	False	Na	
882	0	3	female	22.0	0	0	10.5167	S	Third	woman	False	Na	
885	0	3	female	39.0	0	5	29.1250	Q	Third	woman	False	Na	
887	1	1	female	19.0	0	0	30.0000	S	First	woman	False		
888	0	3	female	NaN	1	2	23.4500	S	Third	woman	False	Na	

314 rows × 15 columns

In [17]: `df["class"].unique()`

Out[17]: `['Third', 'First', 'Second']`  
Categories (3, object): `['First', 'Second', 'Third']`

In [18]: `df[df["embark_town"] == "Southampton"].shape`

Out[18]: `(644, 15)`

In [19]:

```
# ANYONE WHO TRAVELLED SOUTHAMPTON OR IS A FEMALE
# df[required columns][the condition for those columns]
df[["embark_town", "sex"]][ (df["embark_town"] == "Southampton") |
    (df["sex"] == "female")]
```

Out[19]:

	embark_town	sex
0	Southampton	male
1	Cherbourg	female
2	Southampton	female
3	Southampton	female
4	Southampton	male
...	...	...
884	Southampton	male
885	Queenstown	female
886	Southampton	male
887	Southampton	female
888	Southampton	female

755 rows × 2 columns

In [20]:

```
# ONLY FEMALES WHO WENT TO SOUTHAMPTON
# df[required columns][the condition for those columns]
df[["embark_town", "sex"]][ (df["embark_town"] == "Southampton") &
    (df["sex"] == "female")]
```

Out[20]:

	embark_town	sex
2	Southampton	female
3	Southampton	female
8	Southampton	female
10	Southampton	female
11	Southampton	female
...	...	...
871	Southampton	female
880	Southampton	female
882	Southampton	female
887	Southampton	female
888	Southampton	female

203 rows × 2 columns

```
In [21]: # ONLY FEMALES WHO TRAVELLED SOUTHAMPTON OR QUEENSTOWN
df[ ((df["embark_town"] == "Southampton") |
      (df["embark_town"] == "Queenstown")) &
      (df["sex"] == "female")]
```

```
Out[21]:
```

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	dec
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	Na
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	
8	1	3	female	27.0	0	2	11.1333	S	Third	woman	False	Na
10	1	3	female	4.0	1	1	16.7000	S	Third	child	False	
11	1	1	female	58.0	0	0	26.5500	S	First	woman	False	
...	...	...	...	...	...	...	...	...	...	...	...	...
880	1	2	female	25.0	0	1	26.0000	S	Second	woman	False	Na
882	0	3	female	22.0	0	0	10.5167	S	Third	woman	False	Na
885	0	3	female	39.0	0	5	29.1250	Q	Third	woman	False	Na
887	1	1	female	19.0	0	0	30.0000	S	First	woman	False	
888	0	3	female	NaN	1	2	23.4500	S	Third	woman	False	Na

239 rows × 15 columns



```
In [22]: # ONLY VIEW THOSE WHO TRAVELLED QUEENSTOWN OR SOUTHAMPTON
df[df["embark_town"].isin(["Queenstown", "Southampton"])]
```

```
Out[22]:
```

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	dec
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	Na
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	Na
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	Na
5	0	3	male	NaN	0	0	8.4583	Q	Third	man	True	Na
...	...	...	...	...	...	...	...	...	...	...	...	...
885	0	3	female	39.0	0	5	29.1250	Q	Third	woman	False	Na
886	0	2	male	27.0	0	0	13.0000	S	Second	man	True	Na
887	1	1	female	19.0	0	0	30.0000	S	First	woman	False	
888	0	3	female	NaN	1	2	23.4500	S	Third	woman	False	Na
890	0	3	male	32.0	0	0	7.7500	Q	Third	man	True	Na

721 rows × 15 columns



```
In [23]: # ONLY THOSE WHO'RE ABOVE 30
df[df["age"] > 30]
```

```
Out[23]:
```

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	C
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
6	0	1	male	54.0	0	0	51.8625	S	First	man	True	E
11	1	1	female	58.0	0	0	26.5500	S	First	woman	False	C
...	...	...	...	...	...	...	...	...	...	...	...	...
873	0	3	male	47.0	0	0	9.0000	S	Third	man	True	NaN
879	1	1	female	56.0	0	1	83.1583	C	First	woman	False	C
881	0	3	male	33.0	0	0	7.8958	S	Third	man	True	NaN
885	0	3	female	39.0	0	5	29.1250	Q	Third	woman	False	NaN
890	0	3	male	32.0	0	0	7.7500	Q	Third	man	True	NaN

305 rows × 15 columns

## 15- Filtering by Large Categories

```
In [24]: # Counts of all the towns
df["embark_town"].value_counts()
```

```
Out[24]: Southampton    644
Cherbourg             168
Queenstown            77
Name: embark_town, dtype: int64
```

```
In [25]: # Top 3 Age Groups with Highest Count (Names Only)
df["age"].value_counts().nlargest(3).index
```

```
Out[25]: Float64Index([24.0, 22.0, 18.0], dtype='float64')
```

```
In [26]: # Top 3 Age Groups with Highest Count (Names and Count Values)
df["age"].value_counts().nlargest(3)
```

```
Out[26]: 24.0    30
22.0    27
18.0    26
Name: age, dtype: int64
```

```
In [27]: # Top 3 Towns with Highest Counts
counts = df["embark_town"].value_counts()
counts.nlargest(3)
```

```
Out[27]: Southampton    644
Cherbourg        168
Queenstown       77
Name: embark_town, dtype: int64
```

```
In [28]: # Top 3 Category of People Who were on Titanic
counts = df["who"].value_counts()
counts.nlargest(3)
```

```
Out[28]: man      537
woman    271
child     83
Name: who, dtype: int64
```

```
In [29]: # Those Category of People Who were Largest in Count on Titanic
counts = df["who"].value_counts()
df[df["who"].isin(counts.nlargest(1).index)]
```

```
Out[29]:
```

	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	
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	NaN	
5	0	3	male	NaN	0	0	8.4583	Q	Third	man	True	NaN	
6	0	1	male	54.0	0	0	51.8625	S	First	man	True	E	
12	0	3	male	20.0	0	0	8.0500	S	Third	man	True	NaN	
...	...	...	...	...	...	...	...	...	...	...	...	...	
883	0	2	male	28.0	0	0	10.5000	S	Second	man	True	NaN	
884	0	3	male	25.0	0	0	7.0500	S	Third	man	True	NaN	
886	0	2	male	27.0	0	0	13.0000	S	Second	man	True	NaN	
889	1	1	male	26.0	0	0	30.0000	C	First	man	True	C	
890	0	3	male	32.0	0	0	7.7500	Q	Third	man	True	NaN	

537 rows × 15 columns



```
In [35]: counts_embarktown = df["embark_town"].value_counts()
df[df["embark_town"].isin(counts_embarktown.nlargest(2).index)]
```

```
Out[35]:
```

	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	Na	
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False		
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	Na	
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False		
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	Na	



	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck
...	...	...	...	...	...	...	...	...	...	...	...	...
884	0	3	male	25.0	0	0	7.0500	S	Third	man	True	Na
886	0	2	male	27.0	0	0	13.0000	S	Second	man	True	Na
887	1	1	female	19.0	0	0	30.0000	S	First	woman	False	
888	0	3	female	NaN	1	2	23.4500	S	Third	woman	False	Na
889	1	1	male	26.0	0	0	30.0000	C	First	man	True	

812 rows × 15 columns

