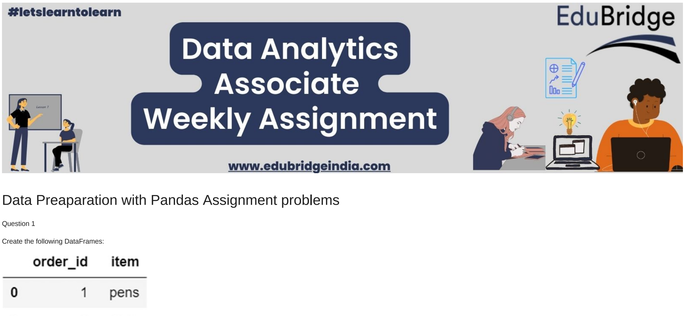
WEEK8 Assignment-2:Data Preaparation with Pandas



Data Preaparation with Pandas Assignment problems

Question 1

Create the following DataFrames:

	order_id	item
0	1	pens
1	2	shirts
2	3	coffee

DataFrame name: orders_df

	order_id	item
0	4	crayons
1	5	tea
2	6	fruits

DataFrame name: orders1_df

	order id	customer_name
0	1	anne
1	2	ben
2	3	carlos

DataFrame name:customers_df image.png

Perform the following Questionnaire operation:

Combine the details of the first two DataFramesorders_df and orders1_df? Create a DataFrame to show the customers and the items they ordered? Make the order_id column as the index for orders_df and customers_df? Which method would you now use to combine these two objects to show which orders were placed by customers?

Create the following DataFrames:

1.DataFrame name: orders_df

In [2]:
```
import pandas as pd
orders_df=pd.DataFrame({"order_id":[1,2,3],"item":["pens","shirts","coffee"]})
orders_df
```

Out[2]:

	order_id	item
0	1	pens
1	2	shirts
2	3	coffee

2.DataFrame name: orders1_df

In [3]:
```
orders1_df=pd.DataFrame({"order_id":[4,5,6],"item":["crayons","tea","fruits"]})
orders1_df
```

Out[3]:

	order_id	item
0	4	crayons
1	5	tea
2	6	fruits

3.DataFrame name:customers_df

In [4]:
```
customers_df=pd.DataFrame({"order_id":[1,2,3],"customer_name":["anne","ben","carlos"]})
customers_df
```

Out[4]:

	order_id	customer_name
0	1	anne
1	2	ben
2	3	carlos

Perform the following Questionnaire operation:

1.Combine the details of the first two DataFramesorders_df and orders1_df?

In [5]:
```
fullorders_df=pd.concat((orders_df,orders1_df),ignore_index=True)
fullorders_df
```

Out[5]:

	order_id	item
0	1	pens
1	2	shirts
2	3	coffee
3	4	crayons
4	5	tea
5	6	fruits

2.Create a DataFrame to show the customers and the items they ordered? 3.Make the order_id column as the index for orders_df and customers_df? Which method would you now use to combine these two objects to show which orders were placed by customers?

In [6]:
```
orders_df.set_index("order_id",inplace=True)
orders_df
```

Out[6]:

order_id	item
1	pens
2	shirts
3	coffee

In [7]:
```
customers_df.set_index("order_id",inplace=True)
customers_df
```

Out[7]:

order_id	customer_name
1	anne
2	ben
3	carlos

In [8]:
```
customers_df.join(orders_df)
```

Out[8]:

order_id	customer_name	item
1	anne	pens
2	ben	shirts
3	carlos	coffee

Question 2

The following DataFrame records the weight fluctuations of four people:

	Anna	Ben	Carole	Dave
0	51.0	70.0	64.0	81.0
1	52.0	70.5	64.2	81.3
2	51.4	69.1	66.8	80.5
3	52.8	69.8	66.0	80.9
4	50.5	70.5	63.4	81.4

1. Create the preceding DataFrame.
2. Convert this DataFrame into a tidy format.
3. Determine who among these four people had the least fluctuation in weight.
4. For people whose average weight is less than 65 kgs, convert their weight (on all four days) into pounds and display this data. Create the preceding DataFrame.

In [9]:
```
data=pd.DataFrame({"Anna":[51.0,52.0,51.4,52.8,50.5],"Ben":[70.0,70.5,69.1,69.8,70.5],"Carole":[64.0,64.2,66.8,66.0,63.4],"Dave":[81.0,81.3,80.5,80.9,81.4]})
data
```

Out[9]:

	Anna	Ben	Carole	Dave
0	51.0	70.0	64.0	81.0
1	52.0	70.5	64.2	81.3
2	51.4	69.1	66.8	80.5
3	52.8	69.8	66.0	80.9
4	50.5	70.5	63.4	81.4

Convert this DataFrame into a tidy format.

In [11]:
```
data.melt()
```

Out[11]:

	variable	value
0	Anna	51.0
1	Anna	52.0
2	Anna	51.4
3	Anna	52.8
4	Anna	50.5
5	Ben	70.0
6	Ben	70.5
7	Ben	69.1
8	Ben	69.8
9	Ben	70.5
10	Carole	64.0
11	Carole	64.2
12	Carole	66.8
13	Carole	66.0
14	Carole	63.4
15	Dave	81.0
16	Dave	81.3
17	Dave	80.5
18	Dave	80.9
19	Dave	81.4

Determine who among these four people had the least fluctuation in weight.

In [12]:
```
data.melt().groupby("variable")["value"].var().sort_values()[:1]
```

Out[12]:
```
variable
Dave    0.127
Name: value, dtype: float64
```

For people whose average weight is less than 65 kgs, convert their weight (on all four days) into pounds and display this data.

In [14]:
```
data.mean()
```

Out[14]:
```
Anna      51.54
Ben       69.98
Carole    64.88
Dave      81.02
dtype: float64
```

In [15]:
```
(data[list(data.mean()[data.mean()<65].index)]*2.205).round(2)
```

Out[15]:

	Anna	Carole
0	112.46	141.12
1	114.66	141.56
2	113.34	147.29
3	116.42	145.53
4	111.35	139.80

In [ ]:
```
```