Content ¶

- **Transpose of a dataframe**
- **Conversion between numpy array and dataframe**
- **Shift Operation**

In [ ]:
```
import numpy as np
import pandas as pd
import warnings
warnings.filterwarnings('ignore')
```

Transpose of dataframe

- Change rows to cols and vice-versa
- It can be done in a similar way as that of numpy

Syntax: `df.T`

In [ ]:
```
df = pd.DataFrame({"A": [1, 10, 100],
                   "B": [2, 20, 200]
                   })
df
```

Out[4]:

	A	B
0	1	2
1	10	20
2	100	200

In [ ]:
```
df.T
```

Out[5]:

	0	1	2
A	1	10	100
B	2	20	200

Converting b/w Numpy Arrays and Pandas DataFrames

Let's **create a numpy matrix first**

Remember what `randint()` does?

- Generate random integers b/w provided range

In [ ]:
```
arr = np.random.randint(10, 100, size=(6,4))
arr
```

Out[19]:
```
array([[43, 73, 50, 97],
       [49, 88, 88, 70],
       [99, 42, 85, 98],
       [16, 66, 53, 51],
       [70, 76, 58, 13],
       [95, 22, 17, 95]])
```

- We got a 6×4 **numpy matrix**

Now, To convert it to a pandas DataFrame

- All we have to do is **use `pd.DataFrame()`**

In [ ]:
```
df = pd.DataFrame(data=arr)
df
```

Out[20]:

	0	1	2	3
0	43	73	50	97
1	49	88	88	70
2	99	42	85	98
3	16	66	53	51
4	70	76	58	13
5	95	22	17	95

- It has **created a DataFrame with 6 rows and 4 columns**

Now, Let's see how we can convert a pandas DataFrame to numpy array

- We'll use the created DataFrame `df`
- We'll convert this DataFrame back to a numpy array
- All we have to do is **use `df.values`**

In [ ]:
```
1 df.values
```

Out[21]:
```
array([[43, 73, 50, 97],
       [49, 88, 88, 70],
       [99, 42, 85, 98],
       [16, 66, 53, 51],
       [70, 76, 58, 13],
       [95, 22, 17, 95]])
```

So, very quickly we can convert b/w Numpy arrays and Pandas DataFrames

Be cautious though!

- This **works only on numerical values**
- As you know, **numpy CANNOT handle string values**

What if we try to convert a DataFrame having string values to Numpy array?

In [ ]:
```
1 df[0][0] = 'abc'
2 df
```

Out[22]:

	0	1	2	3
0	abc	73	50	97
1	49	88	88	70
2	99	42	85	98
3	16	66	53	51
4	70	76	58	13
5	95	22	17	95

In [ ]:
```
1 df.values
```

Out[23]:
```
array([['abc', 73, 50, 97],
       [49, 88, 88, 70],
       [99, 42, 85, 98],
       [16, 66, 53, 51],
       [70, 76, 58, 13],
       [95, 22, 17, 95]], dtype=object)
```

Can you tell what happened here?

- When `df.values` encountered a string value
- It **converted all values to type `object`** (string)

Shift Operation

How do we shift the values of rows or columns up and fill them efficiently?

Row Shift

In [ ]:
```
df = pd.DataFrame({"Name" : ["Sidhu", "Moose", "Jubin", "Arijit", "KK"],
df.head()
```

Out[27]:

	Name	Marks
0	Sidhu	100
1	Moose	95
2	Jubin	80
3	Arijit	96
4	KK	99

Now we realize that the marks are to be shifted, that is Moose should get 100, Jubin 95, and so on. How do we do this efficiently?

In [ ]:
```
df["Marks"] = df["Marks"].shift(periods = 1, axis = 0)
df
```

Out[28]:

	Name	Marks
0	Sidhu	NaN
1	Moose	100.0
2	Jubin	95.0
3	Arijit	80.0
4	KK	96.0

The shift function has parameters which consist of "periods" -> number of periods to shift, "axis" -> 0 for row shifting and 1 for column.

We can make use of it to efficiently shift and then we can add new value to Sidhu's marks.

In [ ]:
```
df["Marks"][0] = 99
df
```

Out[29]:

	Name	Marks
0	Sidhu	99.0
1	Moose	100.0
2	Jubin	95.0
3	Arijit	80.0
4	KK	96.0

Column shift

Suppose we fill roll numbers with names and names with marks and now want to shift it.

Also roll numbers are not yet know, so fill all as 1.

In [ ]:
```
df = pd.DataFrame({"Roll No" : ["Sidhu", "Moose", "Jubin", "Arijit", "KK'
df.head()
```

Out[41]:

	Roll No	Name	Marks
0	Sidhu	100	
1	Moose	95	
2	Jubin	80	
3	Arijit	96	
4	KK	99	

In [ ]:
```
df = df.shift(periods = 1, axis = 1, fill_value = 1)
df
```

Out[42]:

	Roll No	Name	Marks
0	1	Sidhu	100
1	1	Moose	95
2	1	Jubin	80
3	1	Arijit	96
4	1	KK	99

Changing roll numbers

In [ ]:
```
1 df["Roll No"] = df.index
```

In [ ]:
```
1 df
```

Out[44]:

	Roll No	Name	Marks
0	0	Sidhu	100
1	1	Moose	95
2	2	Jubin	80
3	3	Arijit	96
4	4	KK	99

Refer to the documentation for more :
https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.shift.html
(https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.shift.html)