

VERİ ÖN İŞLEME

```
In [3]: import pandas as pd
```

```
In [4]: import numpy as np
```

```
In [5]: data=pd.Series(np.random.randn(9),  
                        index=[['a','a','a','b','b','c','c','d','d'],  
                        [1,2,3,1,3,1,2,2,3]])  
data
```

```
Out[5]: a 1 -0.740187  
        2 -1.220759  
        3  1.769089  
       b 1 -0.082078  
        3 -1.786808  
       c 1 -0.854749  
        2  0.257084  
       d 2 -1.328351  
        3  0.901774  
dtype: float64
```

```
In [6]: data.index
```

```
Out[6]: MultiIndex([('a', 1),  
                    ('a', 2),  
                    ('a', 3),  
                    ('b', 1),  
                    ('b', 3),  
                    ('c', 1),  
                    ('c', 2),  
                    ('d', 2),  
                    ('d', 3)],  
                  )
```

```
In [7]: data['a']
```

```
Out[7]: 1 -0.740187  
        2 -1.220759  
        3  1.769089  
dtype: float64
```

```
In [8]: data['b':'d']
```

```
Out[8]: b 1 -0.082078  
        3 -1.786808  
       c 1 -0.854749  
        2  0.257084  
       d 2 -1.328351  
        3  0.901774  
dtype: float64
```

```
In [10]: data.loc[['b','d']]
```

```
Out[10]: b 1 -0.082078
          3 -1.786808
          d 2 -1.328351
            3 0.901774
          dtype: float64
```

Verileri yeniden şekillendirmek ve pivot tablo oluşturmak gibi grup tabanlı işlemlerde "unstack" modülü kullanılabilir. Tersine çevirmek için "stack" modülü kullanılır.

```
In [12]: data.unstack()
```

```
Out[12]:
```

	1	2	3
a	-0.740187	-1.220759	1.769089
b	-0.082078	NaN	-1.786808
c	-0.854749	0.257084	NaN
d	NaN	-1.328351	0.901774

```
In [13]: data.unstack().stack()
```

```
Out[13]: a 1 -0.740187
          2 -1.220759
          3 1.769089
          b 1 -0.082078
            3 -1.786808
          c 1 -0.854749
            2 0.257084
          d 2 -1.328351
            3 0.901774
          dtype: float64
```

```
In [14]: frame=pd.DataFrame(np.arange(12).reshape((4,3)),
                             index=[['a','a','b','b'], [1,2,1,2]],
                             columns=[['Ankara','Ankara','İstanbul'],
                                       ['Kırmızı','Mavi','Sarı']])
frame
```

```
Out[14]:
```

		Ankara	İstanbul	
		Kırmızı	Mavi	Sarı
a	1	0	1	2
	2	3	4	5
b	1	6	7	8
	2	9	10	11

Kod Açıklaması: np.arange(12).reshape((4,3)): 12 elemandan oluşan bir NumPy dizisini 4 satır ve 3 sütundan oluşan bir matris haline getirir.

index=[['a','a','b','b'], [1,2,1,2]]: Satırlar için iki düzeyli indeks oluşturur. İlk düzeyde 'a' ve 'b' değerleri, ikinci düzeyde ise 1 ve 2 değerleri yer alır.

columns=[['Ankara','Ankara','İstanbul'], ['Kırmızı','Mavi','Sarı']]: Sütunlar için de iki düzeyli indeks oluşturur. İlk düzeyde 'Ankara' ve 'İstanbul', ikinci düzeyde ise 'Kırmızı', 'Mavi' ve 'Sarı' değerleri yer alır.

Veri şablonu yapısında hiyerarşik seviyelere isimler verilebilir. Satır ve sütun seviyelerinde başlıkların adlandırılması :

```
In [17]: frame.index.names=[ 'Anahtar1', 'Anahtar2']
frame.columns.names=[ 'Şehir', 'Renk']
frame
```

```
Out[17]:
```

		Şehir	Ankara	İstanbul	
		Renk	Kırmızı	Mavi	Sarı
Anahtar1	Anahtar2				
a	1		0	1	2
	2		3	4	5
b	1		6	7	8
	2		9	10	11

```
In [18]: frame[ 'Ankara']
```

```
Out[18]:
```

		Renk	Kırmızı	Mavi
Anahtar1	Anahtar2			
a	1		0	1
	2		3	4
b	1		6	7
	2		9	10

DÜZEYLERİ DEĞİŞTİRME VE SIRALAMA

Veride üzerindeki bir ekseninde yer alan düzeylerin sırasını yeniden düzenlemek veya verileri belirli bir düzeydeki değerlere göre sıralamak gerekebilir. değiştirme sonucunda verilerin de yeri değişecektir.

swaplevel : iki sütunun yerini değiştirir

```
In [20]: frame.swaplevel( 'Anahtar1', 'Anahtar2')
```

Out [20]:

	Şehir	Ankara	İstanbul	
	Renk	Kırmızı	Mavi	Sarı
Anahtar2	Anahtar1			
1	a	0	1	2
2	a	3	4	5
1	b	6	7	8
2	b	9	10	11

"sort_index" modülü verileri yalnızca tek bir düzeydeki değerleri kullanarak sıralar. Veride yer alan Anahtar2 başlığı altında veriler sıranır. Swaplevel komutuyla veriler önce yer değiştirilip sonra sıralanabilir.

In [22]: `frame.sort_index(level=1)`

Out [22]:

	Şehir	Ankara	İstanbul	
	Renk	Kırmızı	Mavi	Sarı
Anahtar1	Anahtar2			
a	1	0	1	2
b	1	6	7	8
a	2	3	4	5
b	2	9	10	11

In [23]: `frame.swaplevel(0,1).sort_index(level=0)`

Out [23]:

	Şehir	Ankara	İstanbul	
	Renk	Kırmızı	Mavi	Sarı
Anahtar2	Anahtar1			
1	a	0	1	2
	b	6	7	8
2	a	3	4	5
	b	9	10	11

SEVİYE BAZINDA İŞLEM:

In [25]: `frame.groupby(level="Anahtar2").sum()`

Out[25]:

	Şehir	Ankara	İstanbul
	Renk	Kırmızı	Mavi
			Sarı
Anahtar2			

1	6	8	10
2	12	14	16

In [26]: `import pandas as pd`
`import numpy as np`

In [27]: `data=pd.DataFrame({'a':range(7),`
`'b':range(7,0,-1),`
`'c': ['bir', 'bir', 'bir', 'iki', 'iki','iki'],'`
`'d':[0,1,2,0,1,2,3]})`
`data`

Out[27]:

	a	b	c	d
0	0	7	bir	0
1	1	6	bir	1
2	2	5	bir	2
3	3	4	iki	0
4	4	3	iki	1
5	5	2	iki	2
6	6	1	iki	3

In [28]: `data2=data.set_index(['c','d'])`
`data2`

Out[28]:

	a	b
c d		
bir 0	0	7
1 1	6	
2 2	5	
iki 0	3	4
1 4	3	
2 5	2	
3 6	1	

In [29]: `data2.reset_index()`

Out[29]:

	c	d	a	b
0	bir	0	0	7
1	bir	1	1	6
2	bir	2	2	5
3	iki	0	3	4
4	iki	1	4	3
5	iki	2	5	2
6	iki	3	6	1

In []:

In []:

In []:

In []:

In []:

In []:

In []:

In []:

In []:

In []:

In []:

In []:

In []:

In []:

In []:

In []:

In []:

In []:

In []:

In []:

In []:

In []:

In []:

In []: