

Pandas

Pandas, verileri analiz etmek için kullanılan bir Python kütüphanesidir.

Pandas ile veri setleri üzerinde değişiklikler yapabiliriz.

Dağınık ve düzensiz verileri temizlemeyi, düzenlemeyi ve kullanıma hazır hale getirmeyi öğreneceğiz.

Yapacağımız analize uygun verileri hazırlamak analizlerde alacağımız sonuçlar ve yapacağımız çıkarımlar bakımından çok önemlidir.

▼ Pandas Serileri

Pandas serilerini bir tablodaki sütun gibi düşünebiliriz.

Aslında bir başka deyişle tek boyutlu bir array olduğunu söylemek mümkündür.

```
import pandas as pd

a = [1, 7, 2]

myvar = pd.Series(a)

print(myvar)
```

0	1
1	7
2	2

dtype: int64

▼ Etiketler ve Etiket Oluşturmak

Eğer özellikle belirtilmemişse etiketler index değerleri olarak belirlenmiştir.

```
print(myvar[0])

import pandas as pd

a = [1, 7, 2]

myvar = pd.Series(a, index = ["x", "y", "z"])

print(myvar)
```

x	1
---	---

```
y    7
z    2
dtype: int64
```

✦ Sıra Sizde!

Sizce bu kodun çıktısı ne olur?

```
import pandas as pd

calories = {"day1": 420, "day2": 380, "day3": 390}

myvar = pd.Series(calories, index = ["day1", "day2"])

print(myvar)
```

✦ Dataframe Nedir?

Pandas Dataframe'i iki boyutlu bir veri yapısıdır.

Aslında sözlük yapısının bir Dataframe yani tablo oluşturduğunu gözlemleyebiliriz.

```
import pandas as pd

data = {
    "calories": [420, 380, 390],
    "duration": [50, 40, 45]
}

#load data into a DataFrame object:
df = pd.DataFrame(data)

print(df)
```

✦ loc ve iloc

Belki de Pandas Serileri için en önemli olan özelliklerden olan loc ve iloc yapısına göz atalım.

loc - etiket tabanlı iloc - pozisyon (index) tabanlı

	Weather	Temperature	Wind	Humidity
Day				
Mon	Sunny	12.79	13	30
Tue	Sunny	19.67	28	96
Wed	Sunny	17.51	16	20
Thu	Cloudy	14.44	11	22
Fri	Shower	10.51	26	79
Sat	Shower	11.07	27	62
Sun	Sunny	17.50	20	10

Bu özellikleri tanımak için yukarıdaki hava durumu tablosunu kullanacağız.

▼ .csv Dosyalarını Okutmak

```
import pandas as pd

df = pd.read_csv(dosya_adı)

print(df.to_string())
```

Kullanacağımız veriyi ortamımıza dahil etmek için önce csv dosyasını okutmamız gerekir.

```
df = pd.read_csv('pandas-data.csv', index_col=['Day'])
```

Önce loc özelliğini kullanalım. Bu kod ile Cuma gününe ait sıcaklık verisine ulaşmış olacağız.

```
df.loc['Fri', 'Temperature']
```

Aynı veriyi şimdi de index özellikleri ile çağıralım. Yani aslında pozisyonundan bahsediyoruz.

```
df.iloc[4, 1]
```

Slicing işlemi etiketler üzerinde de loc sayesinde gerçekleştirilebilir.

```
df.loc[:, 'Temperature']
```

```
df.loc['Fri', :]
```

```
df.loc['Tue':'Fri', 'Temperature']
```

Benzer işlemi iloc ile de gerçekleştirebiliriz.

```
df.iloc[[3, 4], 1]
```

```
df.iloc[4, [1, 2]]
```

Eğer elimizdeki tablodan yeni bir tablo oluşturmak istersek:

```
rows = [3, 4]  
cols = [1, 2]  
df.iloc[rows, cols]
```

► ✈ Sıra Sizde!

Sizce bu kodun çıktısı ne olur?

```
[ ] ↳ 5 hücre gizli
```

► ⚠ Uyarı!

iloc için direkt olarak filtreleme işlemi yapmaya çalışırsak ValueError alırız.

```
[ ] ↳ 3 hücre gizli
```

Veri Temizleme

Kirli veri ne anlama gelir?

- Boş hücreler
- Yanlış formattaki veriler
- Yanlış veri
- Tekrarlar

► Satırları Kaldırmak

Boş hücrelerden kurtulmak için **dropna()** özelliğini kullanırız.

[] ↳ 2 hücre gizli

► Boş Değerlere Yeni Değer Atamak

Boş hücrelerle başa çıkmanın bir başka yolu da **fillna()** özelliği ile boş değerleri bir değer ile doldurmaktır.

[] ↳ 5 hücre gizli

► Veriyi Doğru Bir Formata Dönüştürmek

	Duration	Date	Pulse	Maxpulse	Calories
0	60	'2020/12/01'	110	130	409.1
1	60	'2020/12/02'	117	145	479.0
2	60	'2020/12/03'	103	135	340.0
3	45	'2020/12/04'	109	175	282.4
4	45	'2020/12/05'	117	148	406.0
5	60	'2020/12/06'	102	127	300.0
6	60	'2020/12/07'	110	136	374.0
7	450	'2020/12/08'	104	134	253.3
8	30	'2020/12/09'	109	133	195.1
9	60	'2020/12/10'	98	124	269.0
10	60	'2020/12/11'	103	147	329.3
11	60	'2020/12/12'	100	120	250.7
12	60	'2020/12/12'	100	120	250.7
13	60	'2020/12/13'	106	128	345.3
14	60	'2020/12/14'	104	132	379.3
15	60	'2020/12/15'	98	123	275.0
16	60	'2020/12/16'	98	120	215.2
17	60	'2020/12/17'	100	120	300.0
18	45	'2020/12/18'	90	112	NaN
19	60	'2020/12/19'	103	123	323.0
20	45	'2020/12/20'	97	125	243.0
21	60	'2020/12/21'	108	131	364.2
22	45	NaN	100	119	282.0
23	60	'2020/12/23'	130	101	300.0
24	45	'2020/12/24'	105	132	246.0
25	60	'2020/12/25'	102	126	334.5
26	60	20201226	100	120	250.0
27	60	'2020/12/27'	92	118	241.0
28	60	'2020/12/28'	103	132	NaN
29	60	'2020/12/29'	100	132	280.0
30	60	'2020/12/30'	102	129	380.3
31	60	'2020/12/31'	92	115	243.0

[] ↳ 15 hücre gizli

► Pandas Groupby özelliği

Bir grüplama işlemi objeleri birbirinden ayırmak, bir fonksiyonu belirli bir parça koda uygulamak ve sonuçları birleştirmek için kullanılır. Büyük verilerde grüplama yaparak bu grüplar üzerinden işlemlerimizi gerçekleştirebiliriz.

↳ 1 hücre gizli

► Pandas Reshaping

[] ↳ 2 hücre gizli

► Pandas Aggregate

Aggreagate, yani kümeleme spesifik bir aks üzerinden bir ya da birden çok işlem gerçekleştirmemize olanak sağlar.

↳ 2 hücre gizli

[Colab'in ücretli ürünleri](#) - Sözleşmeleri buradan iptal edebilirsiniz

