Pandas

Hafta 5

Dr. Öğr. Üyesi Nihan ÖZBALTAN

Pandas

Veri bilimi projeleri, verinin keşfedilmesi ve temizlenmesi ile başlar ve bu işlemler projelerin en çok zaman alan kısımlarıdır.

Dolayısıyla verinin keşfi ve temizlenmesi sırasında işleri kolaylaştıracak bir takım kütüphanelere ihtiyaç duyulur.

Pandas, Numpy' ın bir **alternatifi değildir**.

Pandas, Numpy'ın değişken adları ve homojen olmayan verilerle çalışamama gibi eksik kaldığı kısımlara ve daha fazlasına çözümler üretir. Pandas ile veri analizi yaparken kullanacağımız temel veri yapıları **Seriler** ve **DataFrame'**lerdir.

Seriler

NumPy ve Pandas kütüphanelerini dahil edelim ve X şirketinin maaş verileriyle seri oluşturalım

```
import numpy as np
import pandas as pd
```

Maaşları x_Maas isimli bir seri olarak oluşturalım.

```
x_Maas = pd.Series([2300, 2300, 2300, 2300, 2300, 3800, 4300, 9000, 12000, 16000])
x_Maas
```

```
0     2300
1     2300
2     2300
3     2300
4     2300
5     3800
6     4300
7     9000
8    12000
9    16000
dtype: int64
```

Serilerin Bazı Özellikleri

Tipi seri, ekseninde 10 adet eleman barındırıyor, int64 tipinde değerleri var ve boş değil.

Ayrıca tek boyutlu seri olduğu ve 10 elemandan oluştuğu da gözlenmektedir.

```
type(x_Maas) # tipini gözlemleyelim.
```

```
pandas.core.series.Series
```

```
x_Maas.axes # serinin 10 elemanlı olduğunu belirtiyor.
```

```
[RangeIndex(start=0, stop=10, step=1)]
```

```
x_Maas.dtype # int64 tipinde.
```

```
dtype('int64')
```

```
x_Maas.empty  # seri boş değildir.
```

```
False
```

```
x_Maas.ndim  # seri tek boyutludur.
```

```
1
```

```
x_Maas.size # seride 10 eleman vardır.
```

```
10
```

Serilerin Elemanlarını Gözlemleme

Listelerde uyguladığımız dilimleme (**slicing**) işlemleri seriler için de geçerlidir.

head ve **tail** fonksiyonları biz değiştirmediğimiz sürece varsayılan olarak **5** parametresini alır.

```
x_Maas.head(3)  # serinin ilk 3 maaşını yazdırır.
```

```
0    2300
1    2300
2    2300
dtype: int64
```

```
x_Maas[0:3] # serinin ilk 3 maaşını yazdırır.
```

```
0    2300
1    2300
2    2300
dtype: int64
```

```
x_Maas.tail(4) # serinin son 4 maaşını verir.
```

```
6     4300
7     9000
8    12000
9    16000
dtype: int64
```

Seriler ve NumPy Arrayler

Seriler ve NumPy arrayler arasında dönüşümler oldukça kolaydır.

```
np_array = np.array([x_Maas])

print(type(np_array))
print(np_array)
```

```
<class 'numpy.ndarray'>
[[ 2300  2300  2300  2300  2300  3800  4300  900
0 12000 16000]]
```

Sözlük Yapısınından Veri Çerçevesi Oluşturmak

sozluk isimli bir dictionary değişkeni oluşturduk.

Sözlük yapısı veri çerçevesine çok benzer. En son satırda sözlüğü veri çerçevesine çevirerek **df** değişkenine atadık.

```
sozluk = {'İsim':
          pd.Series(['Ada','Cem','Sibel','Ahmet','Mehmet','Ali','Veli',
          'Ayşe','Hüseyin','Necmi','Nalan','Namık']),

          'Meslek':pd.Series(['işçi','işçi','memur','serbest','serbest',Non
          'sigortacı','işsiz',None,None,'memur']),

          'Tarih':pd.Series(['11.11.2010','11.11.2010','11.11.2010','18.11.
          None,'11.11.2010',None,'18.11.2011','18.11.2011']),

          'Yaş':pd.Series([21, 24, 25, 44, 31, 27, 35, 33, 42, 29, 41, 43])

          'ÇocukSayısı':pd.Series([None, None, None, None, None, 1, 2, 0, N

          'Puan':pd.Series([89, 87, 77, 55, 70, 79, 73, 79, 54, 92, 61, 69]

df = pd.DataFrame(sozluk)
```

Veri Çerçevesi

Veri çerçevesini görüntülemek için değişken adını yazmak yeterlidir.

İsim, Meslek, Tarih… kısmı öznitelikleri (veya değişkenleri) temsil eder.

Sol kısımda yer alan **0,1,2,3,4…** değerleri indeksi temsil eder.

NaN : Eksik gözlem olduğunu ifade eder.

```
df
```

	İsim	Meslek	Tarih	Yaş	ÇocukSayısı	Puan
0	Ada	işçi	11.11.2010	21	NaN	89
1	Cem	işçi	11.11.2010	24	NaN	87
2	Sibel	memur	11.11.2010	25	NaN	77
3	Ahmet	serbest	18.11.2011	44	NaN	55
4	Mehmet	serbest	18.11.2011	31	NaN	70
5	Ali	None	None	27	1.0	79
6	Veli	None	None	35	2.0	73
7	Ayşe	sigortacı	None	33	0.0	79
8	Hüseyin	işsiz	11.11.2010	42	NaN	54
9	Necmi	None	None	29	NaN	92
10	Nalan	None	18.11.2011	41	NaN	61
11	Namık	memur	18.11.2011	43	NaN	69

Veri Çerçevesinin Bazı Özellikleri

Tıpkı serilerde olduğu gibi **head( )** fonksiyonu belirlediğimiz sayıda veri çerçevesinin ilk gözlemlerini getirir.

sample() fonksiyonu ise rastgele şekilde belirlediğimiz sayıda gözlem getirir.

Veri çerçevemizin kaç adet gözlemden ve değişkenden oluştuğuna bakmak istersek **shape** kullanabiliriz.

```
df.head(3)
```

	İsim	Meslek	Tarih	Yaş	ÇocukSayısı	Puan
0	Ada	işçi	11.11.2010	21	NaN	89
1	Cem	işçi	11.11.2010	24	NaN	87
2	Sibel	memur	11.11.2010	25	NaN	77

```
df.sample(3)
```

	İsim	Meslek	Tarih	Yaş	ÇocukSayısı	Puan
3	Ahmet	serbest	18.11.2011	44	NaN	55
5	Ali	None	None	27	1.0	79
4	Mehmet	serbest	18.11.2011	31	NaN	70

```
df.shape
```

```
(12, 6)
```

info() Fonksiyonu

Ekran çıktısından da anlaşılacağı üzere örneğimizdeki veri çerçevesi (yani df değişkeni) bellekte **704 byte** yer kaplamaktadır.

ÇocukSayısı özniteliğinin karşısında yazan **3** değeri, bu özniteliğin değerinin sadece **3** kişi için girildiğini ve diğer **9** kişi için eksik olduğunu ifade etmektedir.

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12 entries, 0 to 11
Data columns (total 6 columns):
İsim            12 non-null object
Meslek          8 non-null object
Tarih           8 non-null object
Yaş             12 non-null int64
ÇocukSayısı     3 non-null float64
Puan            12 non-null int64
dtypes: float64(1), int64(2), object(3)
memory usage: 704.0+ bytes
```

Bazı Öznitelikleri Görüntüleme

Değişkenleri görüntülemek için **df.columns** ifadesini kullanabiliriz.

Sadece istediğimiz değişkeni gözlemlemek için df[“degisken5”] yazarız.

İstediğimiz değişkenleri gözlemlemek için ise df[[“degisken2”, “degisken4”]] şeklinde ifadeler kullanabiliriz.

```
df.columns
```

```
Index(['İsim', 'Meslek', 'Tarih', 'Yaş', 'ÇocukSa
yısı', 'Puan'], dtype='object')
```

```
df["Meslek"]
```

```
0          işçi
1          işçi
2         memur
3       serbest
4       serbest
5          None
6          None
7     sigortacı
8         işsiz
9          None
10         None
11        memur
Name: Meslek, dtype: object
```

Veri Çerçevesinde Filtreleme İşlemleri

Sağ taraftaki görselde ilk 3 isim ve yaş değişkenlerini gözlemledik.

Aşağıdaki görselde ise yaşı 30'dan büyük ve Puanı 50 den büyük olma şartlarına uyan gözlemlerin ilk 3 tanesinin sadece isimlerini çektik.

```
df[['Yaş', 'İsim']][:3]
```

	Yaş	İsim
0	21	Ada
1	24	Cem
2	25	Sibel

```
df[(df['Yaş']>30) & (df['Puan']>50)][:3]["İsim"]
```

```
3     Ahmet
4    Mehmet
6      Veli
Name: İsim, dtype: object
```

Veri Çerçevesinde Filtreleme İşlemleri

Mesleği işçi olan veya **puanı 90'dan büyük** olan gözlemlerin **yaşlarını** görüntüledik.

```
df[(df['Meslek']=='işçi') | (df['Puan']>90)]["Yaş"]
```

```
0    21
1    24
9    29
Name: Yaş, dtype: int64
```

Meslek değişkeni ne kadar benzersiz değerler içeriyor **value_counts( )** fonksiyonu ile görüntüledik.

```
df["Meslek"].value_counts()
```

```
memur        2
serbest      2
işçi         2
işsiz        1
sigortacı    1
Name: Meslek, dtype: int64
```

Kaynakça

https://medium.com/@denizkilinc/python-ile-veri-tan%C4%B1maya-ve-temel-i%CC%87statisti%C4%9Fe-dal%C4%B1%C5%9F-7e1028270ac

https://medium.com/bili%C5%9Fim-hareketi/veri-bilimi-i%CC%87%C3%A7in-temel-python-k%C3%BCt%C3%BCphaneleri-2-pandas-dcc12ae01b7d

https://medium.com/bili%C5%9Fim-hareketi/veri-bilimi-i%CC%87%C3%A7in-temel-python-k%C3%BCt%C3%BCphaneleri-1-numpy-750429a0d8e5

https://medium.com/datarunner/veri%CC%87-bi%CC%87li%CC%87mi%CC%87-i%CC%87%C3%A7i%CC%87n-i%CC%87stati%CC%87sti%CC%87k-4c1c72c4158

https://medium.com/analytics-vidhya/the-normal-distribution-for-data-scientists-6de041a01cb9