Pandas

Bu doküman pandas paketi ve veri madenciliği uygulamaları hakkında bilgi içerir. pandas kütüphanesi bilgisayarınızda kurulu değilse, pip install pandas komutu ile yükleyebilirsiniz.

```
pip install pandas
```

ya da

```
conda install pandas
```

Pandas Excel gibi görülebilir. Tablo şeklinde verilerle ilgili analiz ve işlemler yapabiliriz. Pandas içerisinde DataFrame nesnesi tanımlanır.

Bu derste verilerin çağrılması, gösterilmesi, dilimlenmesi gibi işlemler gerçekleştirilecek. Ayrıca eksik verilerin doldurulması gibi işlemler yapılacaktır.

Burada kullanılacak olan verileri bu adresten (https://media.geeksforgeeks.org/wp-content/uploads/employees.csv) indirebilirsiniz.

In [ ]:
```
# pandas kütüphanesinin çağrılması
import pandas as pd
import numpy as np

# Verilerin pandas objesi olarak alınması
df = pd.read_csv('employees.csv')
```

In [ ]:
```
# Verisetine ait ilk 5 satırın gösterilmesi
df.head()
```

DataFrame olarak okunması

DataFrame ile herhangi bir özelliğe şu şekilde ulaşabiliriz:

In [ ]:
```
# DataFrame.özellik_adi
print (df.Gender)
```

In [ ]:
```
# Salary özelliğinin en küçük, en büyük ve ortalamalarının gösterilmesi
print (df.Salary.min(), df.Salary.max(), df.Salary.mean())
```

Eksik Verilerin Tanımlanması

None: Python'daki boş olan veriler için kullanılır.

In [ ]:
```
degerler1 = np.array([1,2, None, 5, 2])
```

In [ ]:
```
type(degerler1[2]) # None kendine özgü bir değişken tipidir.
```

In [ ]:
```
degerler1.sum() # Aritmetik işlem yapılamaz.
```

NaN(Not a Number)

In [ ]:
```
degerler2 = np.array([1, np.nan, 2, 5])
```

In [ ]:
```
type(degerler2[1]) # Numpy float değişken tipine sahip
```

In [ ]:
```
degerler2.sum() # Aritmetik işlemleri bozar
```

In [ ]:
```
degerler2[1]+1 # Herhangi sayı ile toplanması yine NaN değerini dönderir.
# Bu nedenle NaN ile çalışılırken dikkat edilmesi gerekir.
```

In [ ]:
```
# NaN değerlerinin dışındaki değerlerin aritmetik işlemlerini yapabilmek için
np.nansum(degerler2), np.nanmin(degerler2), np.nanmax(degerler2)
```

Pandas ile Eksik Veri İşlemleri

1. `isnull()` : Eksik(`NaN`) veriler için `True` değerini dönderir.
2. `notnull()` : Eksik(`NaN`) veriler için `False` değerini dönderir.
3. `dropna()` : Eksik verileri filtreleyip dönderir.
4. `fillna()` : Eksik verileri doldurup dönderir.

In [ ]:
```
data = pd.Series([1, np.nan, 'merhaba', None])
```

In [ ]:
```
data.isnull()
```

In [ ]:
```
data.notnull()
```

In [ ]:
```
data[data.notnull()]
```

In [ ]:
```
data.dropna() # data değişmedi
```

In [ ]:
```
data
```

In [ ]:
```
data.dropna(inplace=True) # data serisini değiştirir.
```

In [ ]:
```
df = pd.DataFrame([[1, np.nan, 2],[2, 3, 5],[np.nan, 4, 6]]) # 2 boyutlu dizide NaN değer
lerini kaldırmak istediğimizde
```

In [ ]:
```
df.dropna() # NaN değeri içeren tüm satırları kaldırır.
```

In [ ]:
```
df.dropna(axis='columns') # Sütunda içerenleri kaldırır.
```

In [ ]:
```
df
```

In [ ]:
```
# Sadece tamamı NaN olan sütunları kaldırmak istersek
df[3]=np.nan
```

In [ ]:
```
df # Sadece 3. sütunu kaldıralım.
```

In [ ]:
```
df.dropna(axis='columns', how='all')
```

In [ ]:
```
df.dropna(axis='rows', thresh=3) # Thresh kalacak satırlarda kaç tane NaN olmayan sayısı
olacağını belirtir.
```

NaN verilerinin doldurulması

In [ ]:
```
data = pd.Series([1, np.nan, 2, None, 3], index=list('abcde'))
```

In [ ]:
```
data
```

In [ ]:
```
data.fillna(0) # Eksik değerleri 0 ile doldurmak istersek
```

In [ ]:
```
data.fillna(method='ffill') # Bir önceki değer ile doldurmak istersek
```

In [ ]:
```
data.fillna(method='bfill') # Bir sonraki değer ile doldurmak istersek
```

In [ ]:
```
df
```

In [ ]:
```
df.fillna(method='ffill', axis=1) # 3. satırdaki NaN değeri kalır çünkü dolduracağı öncek
i değer yok
```

Groupby

In [ ]:
```
df = pd.DataFrame({'key': ['A', 'B', 'C', 'A', 'B', 'C'], 'data': range(6)}, columns=['ke
y', 'data'])
```

In [ ]:
```
df
```

In [ ]:
```
df.groupby('key').sum()
```

In [ ]:
```
df.groupby('key').apply(lambda x:x['data']+1)
```

In [ ]:
```
df.groupby('key').sum()
```

Pivot Tablolar (Çapraz Tablolar)

In [ ]:
```
dogumlar = pd.read_csv("births.csv")
```

In [ ]:
```
dogumlar.head()
```

In [ ]:
```
dogumlar.groupby('gender')['births'].sum() # Her sene doğan kadın ve erkek sayısı ayrı ay
rı
```

In [ ]:
```
dogumlar.groupby(['gender','year'])['births'].sum()
```

In [ ]:
```
# Veya pivot_table fonksiyonu ile

dogumlar.pivot_table('births', index='gender', columns='month') # her ay kaç kişi doğmuş?
```

In [ ]:
```
dogumlar['onyil'] = 10 * (dogumlar['year'] // 10)
```

In [ ]:
```
dogumlar
```

In [ ]:
```
dogumlar.pivot_table('births', index='onyil', columns='gender', aggfunc='sum')
```

Pandas dokümantasyonları

1. Pandas sayfası (https://pandas.pydata.org/)
2. Pandas soru cevap (https://stackoverflow.com/questions/tagged/pandas)
3. Pandas Video (https://pyvideo.org/tag/pandas/)