# Pandas2

### Groupby()

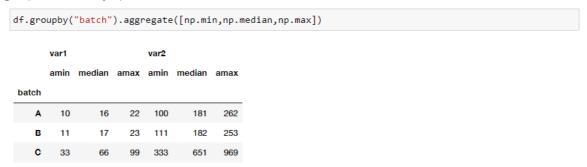
Groupby methodu genellikle math islemleri ile kullanilir.mean,standart deviation,max-min,count,sum gibi.

Groupby islemleri orjinal nesne uzerinden bazi islemleri de icerir. Splitting the Object (Nesneyi Bölme),

applying a function(İşlev uygulama), combining the results(Sonuçları birleştirme)

### Aggregation

Bir ozet istatistik hesaplar.Bir veya birden cok sutuna islemler yapmak icin kullanilir.Girilen anahtar valuelardaki unique degerlere gore yeniden bir gruplandirma yapar.



# Transform()

Gruba ozgu bazi islemleri gerceklestirir.Alt DataFrame'in sonuclarini orjinal DataFrame'ne gore getirir.Her zaman orjinal DataFrame uzunlugunda yeni bir Series dondurur.

# filter()

Bazi kosullara bagli olarak verileri atar.Bu alt Dataframe'e gelen verilere bir filtre uygular.Uygulanan kosula bagli olarak orjinal verideki verilerden uyanlari getirir.Bu durumda orjinal DataFrame'den gelen bir kucuk surum elde edilir.

# apply()

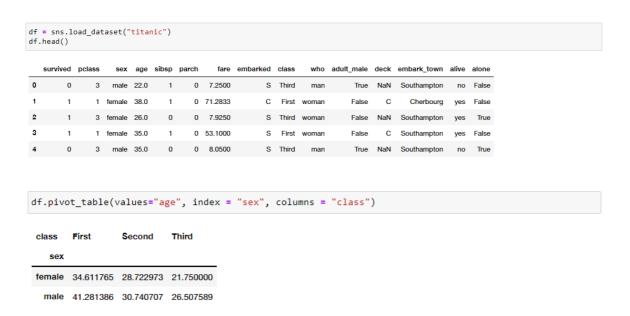
Yalniz bir isleve izin verir. Toplu sonuclar elde edilir.

#### **Pivot Table & Stack**

Özellikler arasındaki ilişki, iki değişkenli analiz yoluyla elde edilebilir.Sayısal ve diğer kategorik özelliklerle ilişkilerini çıkarmak için, groupby ve apply fonksiyonlari kullanılır.Pivot tables ve Stack/Unstack fonsiyonlari da bu konuda oldukca kullanıslidir.

#### **Pivot Table**

Orjinal DataFrame'den categorical feature'ler kullanılarak index,satir,sutunlar ve veriler kulanılarak yeni bir tablo olusturmaktir.



#### Stack/Unstack

Bir DataFrame'de, en içteki sütun dizini en içteki satır dizini olur.Unstacking ters islemdir.Yeniden eski haline getirir.

Basliklardan icteki basligi index degeri olarak atar.(level =) icerisine yazdigimiz baslik degerini index olarak atar.

# **Useful Methods**

```
d={"var1":[1, 2, 3, 4, 5],
    "var2":[111, 222, 333, 444, 555]}

df = pd.DataFrame(d)
df
```

	var1	var2
0	1	111
1	2	222
2	3	333
3	4	444
4	5	555

# apply()

DataFrame'in bir axis'i boyunca bir islemi uygular.Verilen axis'teki Seriesler fonksiyondan gecer.

```
df.apply(np.mean)

var1 3.0

var2 333.0

dtype: float64
```

### applymap()

```
df.applymap(lambda x:x*5)
```

	var1	var2
0	5	555
1	10	1110
2	15	1665
3	20	2220
4	25	2775

# map()

```
df.var1.map({3:"A"})

0  NaN
1  NaN
2  A
3  NaN
4  NaN
Name: var1, dtype: object
```

# unique()

Bir Seriesdeki benzersiz değerlerin işlem dizisi, gözlemlenen sırayla döndürülür.

# nunique()

Benzersiz değerlerin sayısını verir

## value\_counts()

Dizin olarak benzersiz değerler ve değerleri olarak frekanslar içeren bir Seri döndürür, sayım azalan düzende sıralanır

# sort\_values()

Bir DataFrame'i bir sütuna göre sıralama. Sütun adıyla bir 'by' bağımsız değişkenini kabul eder.