Introduction

Son derste, bir DataFrame veya Series'ten ilgili verileri nasıl seçeceğimizi öğrenmiştik. Görevleri tamamlayabilmek için, verilerimizi doğru bir şekilde temsil edebilmek kritik bir öneme sahiptir. Bunu, alıştırmalarımızda da görmüştük.

Ancak, veriler her zaman belleğimizden istediğimiz formatta gelmez. Bazen, elimizdeki göreve uygun olacak şekilde verileri yeniden biçimlendirmek için ek çalışmalar yapmamız gerekir. Bu derste, verilerimizi tam istediğimiz gibi hazırlamak için uygulayabileceğimiz farklı işlemleri ele alacağız.

import pandas as pd

pd.set\_option('display.max\_rows', 5)

import numpy as np

reviews = pd.read\_csv("../input/wine-reviews/winemag-data-130k-v2.csv", index\_col=0)

reviews

| country | description | designation | points | price | province | region\_1 | region\_2 | taster\_name | taster\_twitter\_handle | title | variety | winery |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | Italy | Aromas include tropical fruit, broom, brimston... | Vulkà Bianco | 87 | NaN | Sicily & Sardinia | Etna | NaN | Kerin O’Keefe | @kerinokeefe | Nicosia 2013 Vulkà Bianco (Etna) | White Blend | Nicosia |
| 1 | Portugal | This is ripe and fruity, a wine that is smooth... | Avidagos | 87 | 15.0 | Douro | NaN | NaN | Roger Voss | @vossroger | Quinta dos Avidagos 2011 Avidagos Red (Douro) | Portuguese Red | Quinta dos Avidagos |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 129969 | France | A dry style of Pinot Gris, this is crisp with ... | NaN | 90 | 32.0 | Alsace | Alsace | NaN | Roger Voss | @vossroger | Domaine Marcel Deiss 2012 Pinot Gris (Alsace) | Pinot Gris | Domaine Marcel Deiss |
| 129970 | France | Big, rich and off-dry, this is powered by inte... | Lieu-dit Harth Cuvée Caroline | 90 | 21.0 | Alsace | Alsace | NaN | Roger Voss | @vossroger | Domaine Schoffit 2012 Lieu-dit Harth Cuvée Car... | Gewürztraminer | Domaine Schoffit |

# Summary functions[¶](https://www.kaggle.com/code/residentmario/summary-functions-and-maps" \l "Summary-functions" \t "_self)

Pandas, verileri faydalı bir şekilde yeniden yapılandıran birçok basit "özet fonksiyonu" (resmi bir isim değil) sağlar. Örneğin, **describe()** metodunu ele alalım:

reviews.points.describe()

count 129971.000000

mean 88.447138

...

75% 91.000000

max 100.000000

Name: points, Length: 8, dtype: float64

Bu metot, verilen sütunun niteliklerinin üst düzey bir özetini oluşturur. **Tür duyarlıdır**, yani çıktısı, girilen verinin türüne göre değişir. Yukarıdaki çıktı yalnızca sayısal veriler için anlamlıdır; metin verileri için ise şunları elde ederiz:

reviews.taster\_name.describe()

reviews.taster\_name.describe()

count 103727

unique 19

top Roger Voss

freq 25514

Name: taster\_name, dtype: object

Eğer bir DataFrame veya Series'teki bir sütun hakkında belirli basit bir özet istatistiği elde etmek isterseniz, bunu yapmanızı sağlayan faydalı bir pandas fonksiyonu genellikle mevcuttur.

Örneğin, verilen puanların ortalamasını (örneğin, ortalama derecelendirmeye sahip bir şarabın ne kadar iyi olduğunu) görmek için **mean()** fonksiyonunu kullanabiliriz:

reviews.points.mean()

88.44713820775404

Benzersiz değerlerin listesini görmek için unique() fonksiyonunu kullanabiliriz:

reviews.taster\_name.unique()

array(['Kerin O’Keefe', 'Roger Voss', 'Paul Gregutt',

'Alexander Peartree', 'Michael Schachner', 'Anna Lee C. Iijima',

'Virginie Boone', 'Matt Kettmann', nan, 'Sean P. Sullivan',

'Jim Gordon', 'Joe Czerwinski', 'Anne Krebiehl\xa0MW',

'Lauren Buzzeo', 'Mike DeSimone', 'Jeff Jenssen',

'Susan Kostrzewa', 'Carrie Dykes', 'Fiona Adams',

'Christina Pickard'], dtype=object)

Benzersiz değerlerin bir listesini ve veri setinde ne sıklıkta geçtiklerini görmek için **value\_counts()** metodunu kullanabiliriz:

reviews.taster\_name.value\_counts()

Roger Voss 25514

Michael Schachner 15134

...

Fiona Adams 27

Christina Pickard 6

Name: taster\_name, Length: 19, dtype: int64

# Maps[¶](https://www.kaggle.com/code/residentmario/summary-functions-and-maps" \l "Maps" \t "_self)

Harita (**map**), matematikten ödünç alınmış bir terimdir ve bir değer kümesini alıp bu değerleri başka bir değer kümesine "eşleyen" bir fonksiyonu ifade eder. Veri biliminde, mevcut verilerden yeni temsiller oluşturmaya veya verileri mevcut formatından daha sonra istediğimiz formata dönüştürmeye sık sık ihtiyaç duyarız. Haritalar bu işi hallettiği için, çalışmalarınızı tamamlamanızda son derece önemlidir!

Sıklıkla kullanacağınız iki haritalama metodu vardır.

**map()** bunlardan ilki ve biraz daha basit olanıdır. Örneğin, şarapların aldığı puanları 0'a yeniden ortalamak istediğimizi varsayalım. Bunu şu şekilde yapabiliriz:

review\_points\_mean = reviews.points.mean()

reviews.points.map(lambda p: p - review\_points\_mean)

0 -1.447138

1 -1.447138

...

129969 1.552862

129970 1.552862

Name: points, Length: 129971, dtype: float64

**map()**'e ilettiğiniz fonksiyon, **Series'ten tek bir değer beklemeli** (yukarıdaki örnekte bir puan değeri gibi) ve o değerin dönüştürülmüş bir versiyonunu döndürmelidir. **map()**, tüm değerlerin sizin fonksiyonunuzla dönüştürüldüğü **yeni bir Series** döndürür.

Eğer tüm bir **DataFrame'i, her satırda özel bir metot çağırarak dönüştürmek** istersek, eşdeğer metot **apply()**'dır.

def remean\_points(row):

row.points = row.points - review\_points\_mean

return row

reviews.apply(remean\_points, axis='columns')

| country | description | designation | points | price | province | region\_1 | region\_2 | taster\_name | taster\_twitter\_handle | title | variety | winery |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | Italy | Aromas include tropical fruit, broom, brimston... | Vulkà Bianco | -1.447138 | NaN | Sicily & Sardinia | Etna | NaN | Kerin O’Keefe | @kerinokeefe | Nicosia 2013 Vulkà Bianco (Etna) | White Blend | Nicosia |
| 1 | Portugal | This is ripe and fruity, a wine that is smooth... | Avidagos | -1.447138 | 15.0 | Douro | NaN | NaN | Roger Voss | @vossroger | Quinta dos Avidagos 2011 Avidagos Red (Douro) | Portuguese Red | Quinta dos Avidagos |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 129969 | France | A dry style of Pinot Gris, this is crisp with ... | NaN | 1.552862 | 32.0 | Alsace | Alsace | NaN | Roger Voss | @vossroger | Domaine Marcel Deiss 2012 Pinot Gris (Alsace) | Pinot Gris | Domaine Marcel Deiss |
| 129970 | France | Big, rich and off-dry, this is powered by inte... | Lieu-dit Harth Cuvée Caroline | 1.552862 | 21.0 | Alsace | Alsace | NaN | Roger Voss | @vossroger | Domaine Schoffit 2012 Lieu-dit Harth Cuvée Car... | Gewürztraminer | Domaine Schoffit |

Eğer **reviews.apply()** metodunu **axis='index'** parametresiyle çağırmış olsaydık, her bir satırı dönüştürmek için bir fonksiyon iletmek yerine, her bir **sütunu** dönüştürmek için bir fonksiyon vermemiz gerekirdi.

### Önemli Not: map() ve apply()

**map()** ve **apply()** metotlarının **yeni, dönüştürülmüş Series ve DataFrames** döndürdüğüne dikkat edin. Orijinal verileri **değiştirmezler**. Eğer **reviews**'ın ilk satırına bakarsak, orijinal **points** değerini hala koruduğunu görebiliriz.

reviews.head(1)

| country | description | designation | points | price | province | region\_1 | region\_2 | taster\_name | taster\_twitter\_handle | title | variety | winery |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | Italy | Aromas include tropical fruit, broom, brimston... | Vulkà Bianco | 87 | NaN | Sicily & Sardinia | Etna | NaN | Kerin O’Keefe | @kerinokeefe | Nicosia 2013 Vulkà Bianco (Etna) | White Blend | Nicosia |

Pandas, birçok yaygın eşleme işlemini yerleşik olarak sunar. Örneğin, puan sütunumuzu yeniden anlamlandırmanın daha hızlı bir yolu şöyledir:

review\_points\_mean = reviews.points.mean()

reviews.points - review\_points\_mean

0 -1.447138

1 -1.447138

...

129969 1.552862

129970 1.552862

Name: points, Length: 129971, dtype: float64

Bu kodda, sol taraftaki birçok değer (bir **Series**'deki her şey) ile sağ taraftaki tek bir değer (ortalama değer) arasında bir işlem gerçekleştiriyoruz. Pandas bu ifadeyi inceleyerek, bu ortalama değeri veri setindeki her bir değerden çıkarmak istediğimizi anlar.

Pandas, aynı uzunluktaki **Series**'ler arasında bu tür işlemleri gerçekleştirdiğimizde ne yapacağını da bilir. Örneğin, veri setindeki ülke ve bölge bilgilerini birleştirmenin kolay bir yolu, aşağıdakini yapmaktır:

reviews.country + " - " + reviews.region\_1

0 Italy - Etna

1 NaN

...

129969 France - Alsace

129970 France - Alsace

Length: 129971, dtype: object

Bu operatörler, Pandas'ın içine yerleşik hızlandırmaları kullandıkları için **map()** veya **apply()**'dan daha hızlıdır. Standart Python operatörlerinin ( **>** , **<** , **==** vb.) hepsi bu şekilde çalışır.

Ancak, yalnızca toplama ve çıkarma ile yapılamayan, **koşullu mantık uygulama** gibi daha gelişmiş şeyleri yapabilen **map()** veya **apply()** kadar esnek değillerdir.