Introduction

# Dtypes[¶](https://www.kaggle.com/code/residentmario/data-types-and-missing-values" \l "Dtypes" \t "_self)

DataFrame veya Series'teki bir sütunun veri türü, **dtype** olarak bilinir.

### dtype ile Veri Türünü Öğrenme

Belirli bir sütunun türünü almak için **dtype** özelliğini kullanabilirsiniz. Örneğin, **reviews** DataFrame'indeki **price** sütununun veri türünü şu şekilde alabiliriz:

import pandas as pd

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

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

reviews.price.dtype

dtype('float64')

Alternatif olarak, dtypes özelliği DataFrame'deki her sütunun veri türünü döndürür:

reviews.dtypes

country object

description object

...

variety object

winery object

Length: 13, dtype: object

Veri tipleri (**dtype**), **pandas**'ın veriyi dahili olarak nasıl sakladığı hakkında bilgi verir. **float64**, verinin 64-bitlik bir ondalık sayı (**floating point**) olarak saklandığı anlamına gelirken, **int64** benzer boyutta bir tam sayı (**integer**) olduğunu gösterir.

### Veri Tipleri ve Dönüşümleri

Burada akılda tutulması gereken bir özellik (ve açıkça görülen bir durum), sadece metinlerden oluşan sütunların kendilerine özel bir tipleri olmamasıdır; bunun yerine **object** tipini alırlar.

Bir sütunun veri tipini, anlamlı olduğu durumlarda **astype()** fonksiyonunu kullanarak başka bir tipe dönüştürmek mümkündür. Örneğin, **points** sütununu mevcut **int64** veri tipinden **float64** veri tipine şu şekilde dönüştürebiliriz:

reviews.points.astype('float64')

0 87.0

1 87.0

...

129969 90.0

129970 90.0

Name: points, Length: 129971, dtype: float64

Bir DataFrame veya Series indeksinin de kendine ait veri türü vardır:

reviews.index.dtype

dtype('int64')

Pandas, kategorik veriler (**categorical data**) ve zaman serisi verileri (**timeseries data**) gibi daha nadir kullanılan veri tiplerini de destekler. Bu veri tipleri daha az kullanıldığı için, onları bu eğitimin ileriki bölümlerine kadar atlayacağız.

# Missing data[¶](https://www.kaggle.com/code/residentmario/data-types-and-missing-values" \l "Missing-data" \t "_self)

Eksik değerlere sahip girdilere "Sayı Değil" anlamına gelen **NaN** değeri verilir. Teknik nedenlerden dolayı bu **NaN** değerleri her zaman **float64** veri tipindedir.

### Eksik Verilerle Çalışma

Pandas'ın eksik verilere özel bazı metotları vardır. **NaN** girdilerini seçmek için **pd.isnull()** (veya onun tamamlayıcısı olan **pd.notnull()**) metodunu kullanabilirsiniz. Bu metotlar şu şekilde kullanılır:

reviews[pd.isnull(reviews.country)]

| country | description | designation | points | price | province | region\_1 | region\_2 | taster\_name | taster\_twitter\_handle | title | variety | winery |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 913 | NaN | Amber in color, this wine has aromas of peach ... | Asureti Valley | 87 | 30.0 | NaN | NaN | NaN | Mike DeSimone | @worldwineguys | Gotsa Family Wines 2014 Asureti Valley Chinuri | Chinuri | Gotsa Family Wines |
| 3131 | NaN | Soft, fruity and juicy, this is a pleasant, si... | Partager | 83 | NaN | NaN | NaN | NaN | Roger Voss | @vossroger | Barton & Guestier NV Partager Red | Red Blend | Barton & Guestier |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 129590 | NaN | A blend of 60% Syrah, 30% Cabernet Sauvignon a... | Shah | 90 | 30.0 | NaN | NaN | NaN | Mike DeSimone | @worldwineguys | Büyülübağ 2012 Shah Red | Red Blend | Büyülübağ |
| 129900 | NaN | This wine offers a delightful bouquet of black... | NaN | 91 | 32.0 | NaN | NaN | NaN | Mike DeSimone | @worldwineguys | Psagot 2014 Merlot | Merlot | Psagot |

Eksik değerleri değiştirmek, yaygın bir işlemdir. Pandas, bu sorun için oldukça kullanışlı bir metot sunar: **fillna()**.

### fillna() Metodu

**fillna()** metodu, bu tür verileri gidermek için birkaç farklı strateji sunar. Örneğin, her bir **NaN** değerini basitçe **"Unknown"** ile değiştirebiliriz:

reviews.region\_2.fillna("Unknown")

0 Unknown

1 Unknown

...

129969 Unknown

129970 Unknown

Name: region\_2, Length: 129971, dtype: object

Ya da her bir eksik değeri, veri tabanındaki ilgili kayıttan sonra gelen ilk boş olmayan (**non-null**) değerle doldurabiliriz. Bu strateji **backfill** olarak bilinir.

### Değerleri Değiştirme (replace())

Alternatif olarak, boş olmayan (**non-null**) bir değeri de değiştirmek isteyebiliriz. Örneğin, bu veri seti yayınlandığından beri, yorumcu **Kerin O'Keefe**'in Twitter kullanıcı adını **@kerinokeefe**'den **@kerino**'ya değiştirdiğini varsayalım. Bunu veri setine yansıtmanın bir yolu, **replace()** metodunu kullanmaktır:

reviews.taster\_twitter\_handle.replace("@kerinokeefe", "@kerino")

0 @kerino

1 @vossroger

...

129969 @vossroger

129970 @vossroger

Name: taster\_twitter\_handle, Length: 129971, dtype: object

**replace()** metodu burada anılmaya değer çünkü veri setinde eksik verilerin yerine kullanılan "Unknown", "Undisclosed", "Invalid" gibi özel işaretleme değerlerini değiştirmek için oldukça kullanışlıdır.