

Get unlimited access to the best of Medium for less than \$1/week. [Become a member](#)



# Veri Bilimi İçin Temel Python Kütüphaneleri-2 : Pandas



Merve Bayram Durna · [Follow](#)

Published in Bilişim Hareketi

10 min read · Feb 16, 2019

Share

••• More



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. Hatırlayacağınız üzere Numpy verilerle çalışmayı oldukça kolaylaştırmıştı. Numpy'ın eksik kaldığı kısımlarda ise imdadımıza Pandas yetişiyor. Ancak Pandas

Numpy'ın bir alternatifi olarak değil, uzantısı olarak düşünülmelidir. Pandas, Numpy'ın sütun adları ve homojen olmayan verilerle çalışmamaya gibi eksik kaldığı kısımlara ve daha fazlasına çözümler üretir. Pandas ile veri analizi yaparken kullanacağımız temel veri yapıları Seriler ve DataFrame'lerdir.

## 1) Pandas kütüphanesinin import edilmesi

Pandas'ın özelliklerini incelemeye kütüphaneyi import ederek başlayalım.

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

Pandas serilerini Numpy dizileri ile karşılaştırabilmek için Numpy kütüphanesini de import ettik.

## 2) Pandas veri yapıları

- Series
- DataFrames

### - Seriler(Series)

Seriler Numpy dizileri baz alınarak oluşturulmuştur. Dolayısıyla tek boyutlu Numpy dizilerine çok benzerler.

Serilerin genel kullanımı :

```
my_series = pd.Series(data,index)
```

şeklindedir. Burada data parametresi

- sabit bir değer,
- liste,
- Numpy dizisi veya
- bir dictionary

olabilir. Öncelikle bir Numpy dizisi ve bir Pandas serisi oluşturup farklarına ve benzerliklerine göz atalım.

```

numbers = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

numpy_array = np.array(numbers)
print(numpy_array)
Output :
[0 1 2 3 4 5 6 7 8 9]

pandas_series = pd.Series(data=numbers)
print(pandas_series)
Output :
0    0
1    1
2    2
3    3
4    4
5    5
6    6
7    7
8    8
9    9
dtype: int64

```

Numpy dizisinden farklı olarak serilerde index sütunu da bulunur. Index sütunu belirtilmediği takdirde default olarak n uzunlukta bir dizi için 0'dan başlayıp 1,2,... şeklinde (n-1)'e kadar devam eden bir dizi olur.

```

my_index = ['a', 'b', 'c', 'd', 'e', 'f', 'g','h', 'i','j']
pandas_series = pd.Series(data=numbers, index=my_index, dtype=float)
print(pandas_series)
Output :
a    0.0
b    1.0
c    2.0
d    3.0
e    4.0
f    5.0
g    6.0
h    7.0
i    8.0
j    9.0
dtype: float64

```

Yukarıda default index yerine kendi belirlediğimiz “my\_index” listesini kullandık. Burada dikkat edilmesi gereken nokta dizinin uzunluğu ile tanımlanan index listesinin uzunluğunun eşit olmasıdır. Ayrıca serinin veri tipini float olarak belirledik. Çıktı kısmına baktığımızda da dtype: float64 olduğunu görebiliyoruz.

```

numbers_dictionary = {'a': 0, 'b':1, 'c':2, 'd':3, 'e':4, 'f':5,
'g':6,'h':7, 'i':8,'j':9}
pandas_series = pd.Series(data=numbers_dictionary)
print(pandas_series)
Output :
a    0
b    1
c    2
d    3
e    4
f    5
g    6
h    7
i    8
j    9
dtype: int64

```

Serinin data parametresi bir dictionary olduğunda ‘key’ kısımlarının index olarak kullanıldığını görebilirsiniz.

Seriler Numpy dizilerine çok benzer. Bu sebeple Numpy dizilerine ait bir çok fonksiyon ve metod seriler için de geçerlidir. Öyleyse Numpy dizilerinde ne gibi işlemler yaptığımızı hatırlayalım. Daha detaylı bilgi için Numpy ile ilgili yazıma göz atabilirsiniz.

- ndim, dtype, shape.. gibi özellikleri seriyi incelemek için Pandas'ta da kullanabiliyoruz

```

print(pandas_series.ndim)
Output :
1

print(pandas_series.dtype)
Output :
int64

print(pandas_series.shape)
Output :
(10,)

```

- max(), min(), sum(), median(), mean()... gibi özellikleri Numpy'da olduğu gibi Pandas'ta da kullanabiliyoruz.

```
print(pandas_series.sum())
```

Output :

45

Open in app ↗



Search



Output :

4.5

- Matematiksel işlemler

```
print(pandas_series+pandas_series)
```

Output :

```
a      0  
b      2  
c      4  
d      6  
e      8  
f     10  
g     12  
h     14  
i     16  
j     18
```

dtype: int64

```
print(np.sqrt(pandas_series))
```

Output :

```
a      0.000000  
b      1.000000  
c      1.414214  
d      1.732051  
e      2.000000  
f      2.236068  
g      2.449490  
h      2.645751  
i      2.828427  
j      3.000000
```

dtype: float64

```
print(pandas_series*pandas_series)
```

Output :

```
a      0  
b      1  
c      4  
d      9  
e     16  
f     25  
g     36  
h     49
```

```
i      64
j      81
dtype: int64
```

- Koşul ifadeleri ile çalışmak

```
print(pandas_series[pandas_series>pandas_series.median()])
```

Output :

```
f      5
g      6
h      7
i      8
j      9
dtype: int64
```

```
print(pandas_series[pandas_series == 5])
```

Output:

```
f      5
dtype: int64
```

### - DataFrame

DataFrame'leri farklı tipteki sütunlara ve satırlara sahip bir SQL tablosu olarak düşünebiliriz. DataFrame'ler veriyi daha kolay işleyebilmemizi sağlar.

DataFrame'lerin genel kullanımı

```
my_dataframe = pd.DataFrame(data, index)
```

şeklindedir. Serilerde olduğu gibi DataFrame'lerde farklı türden data parametresi alabilirler. Yukarıdaki kullanımda data parametresi aşağıdakilerden herhangi biri olabilir.

- Dictionary'lerden, serilerden veya listelerden oluşan bir dictionary,
- 2 boyutlu numpy dizisi,
- Başka bir DataFrame

```
#Data dictionary'lerden oluştuğunda
dict1 = dict(a=1, b=2, c=3, d=4)
dict2 = dict(a=5, b=6, c=7, d=8, e=9)
```

```
data1 = dict(first=dict1, second=dict2)
df1 = pd.DataFrame(data1)
print(df1)
Output :
    first  second
a      1.0      5
b      2.0      6
c      3.0      7
d      4.0      8
e      NaN      9

#Data serilerden oluştuğunda
s1 = pd.Series([1.1, 2.2, 3.3, 4.4])
s2 = pd.Series(['a', 'b', 'c', 'd', 'e'])

data2 = dict(first=s1, second=s2)
df2 = pd.DataFrame(data2)
print(df2)
Output :
    first  second
0      1.1      a
1      2.2      b
2      3.3      c
3      4.4      d
4      NaN      e

#Data başka bir DataFrame'den oluştuğunda
df3 = pd.DataFrame(df2)
print(df3)
```

DataFrame oluşturulması ile ilgili birkaç örneği inceledik daha detaylı bilgi için [buraya](#) bakabilirsiniz.

### 3) Pandas ile veri seçme(selecting) işlemleri

DataFrame nesneleri satır ve sütun adlarına sahip olduğu için Numpy' dan farklı olarak bu satır ve sütun adlarıyla istediğimiz veriyi seçebiliriz. Seçme işlemleri için “.loc[]” kullanılabilir. Genel kullanımı:

```
df.loc[row, column]
```

şeklindedir. “.loc[]” kullanımından farklı veri seçme teknikleri de bulunmaktadır tüm bunları özetleyen bir tablo :

| Select by Label                 | Explicit Syntax                          | Shorthand Convention               | Other Shorthand      |
|---------------------------------|--|------------------------------------|----------------------|
| Single column from dataframe    | <code>df.loc[:, "col1"]</code>           | <code>df["col1"]</code>            | <code>df.col1</code> |
| List of columns from dataframe  | <code>df.loc[:, ["col1", "col7"]]</code> | <code>df[["col1", "col7"]]</code>  |                      |
| Slice of columns from dataframe | <code>df.loc[:, "col1": "col4"]</code>   |                                    |                      |
| Single row from dataframe       | <code>df.loc["row4"]</code>              |                                    |                      |
| List of rows from dataframe     | <code>df.loc[[ "row1", "row8"]]</code>   |                                    |                      |
| Slice of rows from dataframe    | <code>df.loc["row3": "row5"]</code>      | <code>df["row3": "row5"]</code>    |                      |
| Single item from series         | <code>s.loc["item8"]</code>              | <code>s["item8"]</code>            | <code>s.item8</code> |
| List of items from series       | <code>s.loc[[ "item1", "item7"]]</code>  | <code>s[["item1", "item7"]]</code> |                      |
| Slice of items from series      | <code>s.loc["item2": "item4"]</code>     | <code>s["item2": "item4"]</code>   |                      |

Görsel [Data Quest](#) sitesinden alınmıştır daha fazla detay için siteden yararlanabilirsiniz.

Not : `df.loc[:, "col1": "col4"]` gibi DataFrame den bir dilim seçme işlemlerinde Numpy'dan farklı olarak col4 sütunu da dahil edilir. Yani col1, col2, col3 ve col4 sütunlarından oluşan DataFrame nesnesi döndürür.

## 4) Pandas ile keşifsel veri analizi (Exploratory Data Analysis-EDA)

Buraya kadar Pandas veri yapılarına ve Pandas'ın temel özelliklerine değindik. Yazının bundan sonra ki kısmında ise verilerin keşfedilmesi sırasında Pandas'ın bize sağladığı kolaylıklarını net bir şekilde görebilmek için, bir veri seti üzerinden keşif yapmakla ilgileneceğiz.

Verinin keşfedilmesi verilerin bize neler söyleyebileceğini anlama sürecidir.

Kaggle'in Datasets bölümünde araştırma yapmak için binlerce veri kümesi bulabilirsiniz. Ben veri kümesi olarak "[Google Play Store Apps](#)" verileri ile çalışmayı tercih ettim linke tıklayarak veri setini inceleyebilir ve indirebilirsiniz.

Veri seti Google Play Store'da bulunan yaklaşık 10 bin uygulamanın verilerinden oluşmaktadır. Başlamadan önce veri setinin kolon adlarına bakalım.

App : Uygulamanın adı  
 Category : Uygulamanın hangi kategoriye ait olduğu  
 Rating : Uygulamanın puanı  
 Reviews : Uygulamanın aldığı yorum sayısı  
 Size : Uygulamanın boyutu  
 Installs : Uygulamanın indirilme sayısı  
 Type : Ücretli veya ücretsiz  
 Price : Ücretli ise fiyatı  
 Content Rating : Uygulamanın hedeflediği yaşı grubu  
 Genres : Ana kategori haricindeki kategorisi  
 Last Updated : Uygulamanın son güncellenme tarihi  
 Current Ver : Uygulamanın en güncel versiyonu  
 Android Ver : Uygulama için gerekli minimum android versiyonu.

Kaggle' dan veri setimizi indirdik artık .csv formatında veri setine sahibiz ve bu dosyayı Pandas ile kullanabileceğimiz formata dönüştürmemiz gerekiyor . Bunun için “googleplaystore.csv” dosyasını bir DataFrame nesnesine okuyacağız.

Not : Pandas farklı formattaki dosyalarında DataFrame nesnesine okuma imkanı sağlıyor burada çalışacağımız dosya türü “.csv” olduğu için sadece bu tür dosyaların okunması ile ilgileneneceğiz. Diğer dosya türleri ile ilgili işlemler hakkında detaylı bilgi için buraya bakabilirsiniz.

```
import numpy as np
import pandas as pd

df = pd.read_csv("googleplaystore.csv")
```

Dosyamız artık df isminde bir DataFrame nesnesine dönüştü. Yazının önceki kısmında bahsettiğimiz özelliklerini ve daha fazlasını artık kullanabiliriz.

- df'in satır ve sütun sayısını öğrenelim.

```
print(df.shape)
```

Output :  
 (10841, 13)

- df'in sütunlarının ve veri tiplerinin neler olduğunu öğrenelim.

```
print(df.columns)
Output :
Index(['App', 'Category', 'Rating', 'Reviews', 'Size', 'Installs',
       'Type', 'Price', 'Content Rating', 'Genres', 'Last Updated', 'Current
       Ver', 'Android Ver'],
      dtype='object')

print(df.dtypes)
Output :
App                  object
Category             object
Rating               float64
Reviews              object
Size                 object
Installs             object
Type                 object
Price                object
Content Rating       object
Genres               object
Last Updated         object
Current Ver          object
Android Ver          object
dtype: object
```

- df' in ilk birkaç satırına bakalım.

head() metodu varsayılan olarak df'in ilk 5 satırını döndürür.

df.head()

|   | App   | Category       | Rating | Reviews | Size | Installs    | Type | Price | Content Rating | Genres                    | Last Updated     | Current Ver        | Android Ver  |
|---|---|----------------|--------|---------|------|-------------|------|-------|----------------|---------------------------|------------------|--------------------|--------------|
| 0 | Photo Editor & Candy Camera & Grid & ScrapBook    | ART_AND_DESIGN | 4.1    | 159     | 19M  | 10.000+     | Free | 0     | Everyone       | Art & Design              | January 7, 2018  | 1.0.0              | 4.0.3 and up |
| 1 | Coloring book moana                               | ART_AND_DESIGN | 3.9    | 967     | 14M  | 500.000+    | Free | 0     | Everyone       | Art & Design:Pretend Play | January 15, 2018 | 2.0.0              | 4.0.3 and up |
| 2 | U Launcher Lite – FREE Live Cool Themes, Hide ... | ART_AND_DESIGN | 4.7    | 87510   | 8.7M | 5.000.000+  | Free | 0     | Everyone       | Art & Design              | August 1, 2018   | 1.2.4              | 4.0.3 and up |
| 3 | Sketch - Draw & Paint                             | ART_AND_DESIGN | 4.5    | 215644  | 25M  | 50.000.000+ | Free | 0     | Teen           | Art & Design              | June 8, 2018     | Varies with device | 4.2 and up   |
| 4 | Pixel Draw - Number Art Coloring Book             | ART_AND_DESIGN | 4.3    | 967     | 2.8M | 100.000+    | Free | 0     | Everyone       | Art & Design,Creativity   | June 20, 2018    | 1.1                | 4.4 and up   |

- df' in son birkaç satırına bakalım.

tail() metodu varsayılan olarak son 5 satırı döndürür.Fakat biz burada tail() metoduna 7 parametresini göndererek son 7 satırı döndürmesini sağladık. Aynı

parametre head() metodu için de geçerlidir.

`df.tail(7)`

|       | App   | Category            | Rating | Reviews | Size               | Installs    | Type | Price | Content Rating | Genres            | Last Updated       | Current Ver        | Android Ver        |
|-------|---|---------------------|--------|---------|--------------------|-------------|------|-------|----------------|-------------------|--------------------|--------------------|--------------------|
| 10834 | FR Calculator                                 | FAMILY              | 4.0    | 7       | 2.6M               | 500+        | Free | 0     | Everyone       | Education         | June 18, 2017      | 1.0.0              | 4.1 and up         |
| 10835 | FR Forms                                      | BUSINESS            | NaN    | 0       | 9.6M               | 10+         | Free | 0     | Everyone       | Business          | September 29, 2016 | 1.1.5              | 4.0 and up         |
| 10836 | Syafa Maroc - FR                              | FAMILY              | 4.5    | 38      | 53M                | 5,000+      | Free | 0     | Everyone       | Education         | July 25, 2017      | 1.48               | 4.1 and up         |
| 10837 | Fr. Mike Schmitz Audio Teachings              | FAMILY              | 5.0    | 4       | 3.6M               | 100+        | Free | 0     | Everyone       | Education         | July 6, 2018       | 1.0                | 4.1 and up         |
| 10838 | Parkinson Exercises FR                        | MEDICAL             | NaN    | 3       | 9.5M               | 1,000+      | Free | 0     | Everyone       | Medical           | January 20, 2017   | 1.0                | 2.2 and up         |
| 10839 | The SCP Foundation DB fr nn5n                 | BOOKS_AND_REFERENCE | 4.5    | 114     | Varies with device | 1,000+      | Free | 0     | Mature 17+     | Books & Reference | January 19, 2015   | Varies with device | Varies with device |
| 10840 | iHoroscope - 2018 Daily Horoscope & Astrology | LIFESTYLE           | 4.5    | 398307  | 19M                | 10,000,000+ | Free | 0     | Everyone       | Lifestyle         | July 25, 2018      | Varies with device | Varies with device |

- df hakkında genel bir bilgi edinelim.

`df.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10841 entries, 0 to 10840
Data columns (total 13 columns):
 App           10841 non-null object
 Category       10841 non-null object
 Rating         9367 non-null float64
 Reviews        10841 non-null object
 Size           10841 non-null object
 Installs       10841 non-null object
 Type           10840 non-null object
 Price          10841 non-null object
 Content Rating 10840 non-null object
 Genres         10841 non-null object
 Last Updated   10841 non-null object
 Current Ver    10833 non-null object
 Android Ver    10838 non-null object
 dtypes: float64(1), object(12)
 memory usage: 1.1+ MB
```

info() metodu ile ;0' dan 10840 a kadar numaralandırılmış 10841 girdi olduğu, her bir sütunda null olmayan girdi sayısı ve sütunun veri tipi gibi bilgileri elde edebiliriz.

info() metodu ile eksik veriye sahip sütunlar hakkında bilgi edindik şimdi de daha anlaşırlır bir şekilde eksik veriler hakkında bilgi edinelim.

- df'deki eksik değerlerin sayısını öğrenelim ve azalan şekilde sıralayalım.

```
print(df.isnull().sum().sort_values(ascending=False))
```

| Sütun          | Count |
|----------------|-------|
| Rating         | 1474  |
| Current Ver    | 8     |
| Android Ver    | 3     |
| Content Rating | 1     |
| Type           | 1     |
| Last Updated   | 0     |
| Genres         | 0     |
| Price          | 0     |
| Installs       | 0     |
| Size           | 0     |
| Reviews        | 0     |
| Category       | 0     |
| App            | 0     |

- Sütunların istatiksel özeti bakalım.

```
print(df.describe())
```

| Sütun | Rating      |
|-------|-------------|
| count | 9367.000000 |
| mean  | 4.193338    |
| std   | 0.537431    |
| min   | 1.000000    |
| 25%   | 4.000000    |
| 50%   | 4.300000    |
| 75%   | 4.500000    |
| max   | 19.000000   |

describe() metodu sayısal verilere sahip olan sütunların max, min , std... gibi istatiksel değerlerini döndürür. Bizim veri setimizde sayısal olan tek sütun "Rating" sütunu olduğu için sadece Rating sütununun istatiksel özetini görebiliyoruz.

İstediğimiz takdirde describe() metodunu string verilerde de kullanabiliriz.

```
print(df["Category"].describe())
```

```

count:      10841
unique:      34
top:        FAMILY
freq:       1972
Name: Category, dtype: object

```

count: NaN olmayan girdi sayısı

unique : birbirinden farklı kaç kategori olduğunu bilgisi.

top : sütunda en çok bulunan kategorinin adı.

freq : en çok bulunan kategorinin sütunda bulunma sıklığı.

Sayısal olmayan tüm sütunların istatiksel özetiğini görmek istiyorsak include=['O'] parametresini kullanmalıyız.

```
df.describe(include=['O'])
```

|        | App    | Category | Reviews | Size               | Installs   | Type  | Price | Content Rating | Genres | Last Updated   | Current Ver        | Android Ver |
|--------|--------|----------|---------|--------------------|------------|-------|-------|----------------|--------|----------------|--------------------|-------------|
| count  | 10841  | 10841    | 10841   | 10841              | 10841      | 10840 | 10841 | 10840          | 10841  | 10841          | 10833              | 10838       |
| unique | 9660   | 34       | 6002    | 462                | 22         | 3     | 93    | 6              | 120    | 1378           | 2832               | 33          |
| top    | ROBLOX | FAMILY   | 0       | Varies with device | 1,000,000+ | Free  | 0     | Everyone       | Tools  | August 3, 2018 | Varies with device | 4.1 and up  |
| freq   | 9      | 1972     | 596     | 1695               | 1579       | 10039 | 10040 | 8714           | 842    | 326            | 1459               | 2451        |

- Her bir değerin sütunda bulunma sayısına bakalım.

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

|                     |      |
|---------------------|------|
| FAMILY              | 1972 |
| GAME                | 1144 |
| TOOLS               | 843  |
| MEDICAL             | 463  |
| BUSINESS            | 468  |
| PRODUCTIVITY        | 424  |
| PERSONALIZATION     | 392  |
| COMMUNICATION       | 387  |
| SPORTS              | 384  |
| LIFESTYLE           | 382  |
| FINANCE             | 366  |
| HEALTH_AND_FITNESS  | 341  |
| PHOTOGRAPHY         | 335  |
| SOCIAL              | 295  |
| NEWS_AND_MAGAZINES  | 283  |
| SHOPPING            | 268  |
| TRAVEL_AND_LOCAL    | 258  |
| DATING              | 234  |
| BOOKS_AND_REFERENCE | 231  |
| VIDEO_PLAYERS       | 175  |
| EDUCATION           | 156  |
| ENTERTAINMENT       | 149  |
| MAPS_AND_NAVIGATION | 137  |
| FOOD_AND_DRINK      | 127  |
| HOUSE_AND_HOME      | 88   |
| AUTO_AND_VEHICLES   | 85   |
| LIBRARIES_AND_DEMO  | 85   |
| WEATHER             | 82   |
| ART_AND DESIGN      | 65   |
| EVENTS              | 64   |
| PARENTING           | 68   |
| COMICS              | 68   |
| BEAUTY              | 53   |
| 1.9                 | 1    |

Name: Category, dtype: int64

value\_counts() verinin keşfi sırasında kullanılan oldukça kullanışlı bir metottur. Sütundaki NaN olmayan her bir unique değerin kaç kez kullanıldığını gösteren bir seri döndürür. Bu seri default olarak azalan şekilde sıralanmıştır ve NaN değerleri içermez. Seriyi incelediğimizde en çok uygulamanın “FAMILY” kategorisinde bulunduğuunu görebiliyoruz. En az uygulamanın bulunduğu kategori ise “1.9” kategorisi. Siz de burada bir gariplik sezmiş olmalısınız. “1.9” kategorisinde bulunan tek uygulamayı inceleyip gerçekten bir gariplik olup olmadığını bakalım.

- Belli bir koşulu sağlayan değerlerin seçilmesi.(Boolean Indexing)

```
df[df["Category"]=="1.9"]
```

|       | App                                     | Category | Rating | Reviews | Size   | Installs | Type | Price    | Content Rating | Genres            | Last Updated | Current Ver | Android Ver |
|-------|---|----------|--------|---------|--------|----------|------|----------|----------------|-------------------|--------------|-------------|-------------|
| 10472 | Life Made Wi-Fi Touchscreen Photo Frame | 1.9      | 19.0   | 3.0M    | 1,000+ | Free     | 0    | Everyone | NaN            | February 11, 2018 | 1.0.19       | 4.0 and up  | NaN         |

Uygulamanın adına baktığımızda “Life Made WI-Fi Touchscreen Photo Frame” olduğunu görüyoruz bu isimde bir uygulamanın “PHOTOGRAPHY” kategorisinde bulunabileceği karışımında bulunabiliriz. O zaman uygulamanın kategorisini “PHOTOGRAPHY” olacak şekilde değiştirelim.

- Seçilen veriye yeni bir değer atayalım .

```
df.set_value(10472, 'Category', "PHOTOGRAPHY")
```

set\_value() metodunun genel kullanımı

```
DataFrame.set_value(index, col, value)
```

şeklindedir.

- Verileri sıralayalım.

En yüksek puana sahip 10 uygulamayı görmek istediğimizi düşünelim. Bu durumda “Rating” sütununa göre uygulamaları sıralayıp ilk 10 uygulamayı ekrana yazdırımız yeterlidir. Nasıl yapılacağına bakalım:

```
df.sort_values(by='Rating', ascending=False).head(10)
```

|       | App                                     | Category    | Rating | Reviews | Size   | Installs | Type | Price    | Content Rating | Genres            | Last Updated   | Current Ver | Android Ver  |
|-------|---|-------------|--------|---------|--------|----------|------|----------|----------------|-------------------|----------------|-------------|--------------|
| 10472 | Life Made WI-Fi Touchscreen Photo Frame | PHOTOGRAPHY | 19.0   | 3.0M    | 1,000+ | Free     | 0    | Everyone | NaN            | February 11, 2018 | 1.0.19         | 4.0 and up  | NaN          |
| 9511  | Ex Bander Ne Kholi Dukan                | FAMILY      | 5.0    | 10      | 3.0M   | 10,000+  | Free | 0        | Everyone       | Entertainment     | June 28, 2017  | 1.0.9       | 4.0 and up   |
| 10168 | FA Player Essentials                    | SPORTS      | 5.0    | 7       | 68M    | 100+     | Free | 0        | Everyone       | Sports            | July 23, 2018  | 1.6.0       | 4.0.3 and up |
| 7895  | Dine In CT - Food Delivery              | SHOPPING    | 5.0    | 4       | 1.8M   | 1,000+   | Free | 0        | Everyone       | Shopping          | May 16, 2016   | 1.3         | 4.0 and up   |
| 5118  | Eternal Light AG                        | SOCIAL      | 5.0    | 30      | 13M    | 100+     | Free | 0        | Teen           | Social            | May 19, 2018   | 1.04        | 4.0.3 and up |
| 6953  | BxPort - Bitcoin Bx (Thailand)          | FINANCE     | 5.0    | 4       | 4.1M   | 50+      | Free | 0        | Everyone       | Finance           | July 14, 2018  | 1.0.4       | 4.2 and up   |
| 5125  | Ag Valley Cooperative                   | BUSINESS    | 5.0    | 6       | 74M    | 500+     | Free | 0        | Everyone       | Business          | June 26, 2017  | 2.3         | 4.0 and up   |
| 7896  | CT Checkout                             | FINANCE     | 5.0    | 1       | 8.4M   | 50+      | Free | 0        | Everyone       | Finance           | April 20, 2017 | 1.2         | 4.2 and up   |
| 5139  | Chenoweth AH                            | MEDICAL     | 5.0    | 1       | 27M    | 100+     | Free | 0        | Everyone       | Medical           | April 3, 2017  | 300000.0.78 | 4.0.3 and up |
| 5145  | Arrowhead AH App                        | MEDICAL     | 5.0    | 3       | 28M    | 100+     | Free | 0        | Everyone       | Medical           | April 21, 2017 | 300000.0.80 | 4.0.3 and up |

Evet burada da karşımıza “Life Made WI-Fi Touchscreen Photo Frame” uygulaması çıktı. Daha önce “1.9” olan kategorisini “PHOTOGRAPHY” olarak değiştirmiştik . Şimdi Rating’ inin 19.0 olduğunu NaN ve anlamsız değerler olduğunu görüyoruz

yani neresinden tutarsak tutalım elimizde kalıyor :) O zaman bu uygulamadan kurtulalım.

- df'den veri silelim.

Pandas'ta satır veya sütunları silmek için drop() fonksiyonu kullanılır. Genel kullanımı şu şekildedir:

```
DataFrame.drop(labels=None, axis=0, index=None, columns=None,
level=None, inplace=False, errors='raise')
```

— Satır(ları) silmek

```
df.drop(10472)
df.sort_values(by='Rating', ascending=False).head(10)
```

|       | App                                     | Category | Rating | Reviews | Size   | Installs | Type | Price    | Content Rating | Genres            | Last Updated   | Current Ver | Android Ver  |
|-------|---|----------|--------|---------|--------|----------|------|----------|----------------|-------------------|----------------|-------------|--------------|
| 10472 | Life Made WI-Fi Touchscreen Photo Frame | 1.9      | 19.0   | 3.0M    | 1.000+ | Free     | 0    | Everyone | NaN            | February 11, 2018 | 1.0.19         | 4.0 and up  | NaN          |
| 9511  | Ek Bander Ne Kholi Dukan                | FAMILY   | 5.0    | 10      | 3.0M   | 10.000+  | Free | 0        | Everyone       | Entertainment     | June 26, 2017  | 1.0.9       | 4.0 and up   |
| 10168 | FA Player Essentials                    | SPORTS   | 5.0    | 7       | 68M    | 100+     | Free | 0        | Everyone       | Sports            | July 23, 2018  | 1.6.0       | 4.0.3 and up |
| 7895  | Dine In CT - Food Delivery              | SHOPPING | 5.0    | 4       | 1.6M   | 1.000+   | Free | 0        | Everyone       | Shopping          | May 16, 2016   | 1.3         | 4.0 and up   |
| 5118  | Eternal Light AG                        | SOCIAL   | 5.0    | 30      | 13M    | 100+     | Free | 0        | Teen           | Social            | May 19, 2018   | 1.04        | 4.0.3 and up |
| 6953  | ExPort - Bitcoin Bix (Thailand)         | FINANCE  | 5.0    | 4       | 4.1M   | 50+      | Free | 0        | Everyone       | Finance           | July 14, 2018  | 1.0.4       | 4.2 and up   |
| 5125  | Ag Valley Cooperative                   | BUSINESS | 5.0    | 6       | 74M    | 500+     | Free | 0        | Everyone       | Business          | June 26, 2017  | 2.3         | 4.0 and up   |
| 7896  | CT Checkout                             | FINANCE  | 5.0    | 1       | 8.4M   | 50+      | Free | 0        | Everyone       | Finance           | April 20, 2017 | 1.2         | 4.2 and up   |
| 5139  | Chenoweth AH                            | MEDICAL  | 5.0    | 1       | 27M    | 100+     | Free | 0        | Everyone       | Medical           | April 3, 2017  | 300000.0.78 | 4.0.3 and up |
| 5145  | Arrowhead AH App                        | MEDICAL  | 5.0    | 3       | 28M    | 100+     | Free | 0        | Everyone       | Medical           | April 21, 2017 | 300000.0.80 | 4.0.3 and up |

Evet 10472 index'li “Life Made WI-Fi Touchscreen Photo Frame” uygulamasını silmemize rağmen hala ilk sırada onu görüyoruz. Bunun sebebi drop() fonksiyonun istenilen satırların silindiği yeni bir DataFrame döndürüyor olması. Yani asıl DataFrame (df )’de bir değişiklik olmuyor. Bu problemi çözmek için 2 alternatif var :

1. drop() fonksiyonu ile geri dönen DataFrame nesnesini orjinal df nesnesine atamak.
2. drop() fonksiyonunun default olarak False olan inplace parametresinin değerini True olarak değiştirmek.

```
#1. yöntem
df = df.drop(10472)
```

## #2.yöntem

```
#df.drop(10472, inplace = True)
```

```
df.sort_values(by='Rating', ascending=False).head(10)
```

|      | App                         | Category           | Rating | Reviews | Size | Installs | Type | Price  | Content Rating | Genres           | Last Updated      | Current Ver | Android Ver |
|------|-----------------------------|--------------------|--------|---------|------|----------|------|--------|----------------|------------------|-------------------|-------------|-------------|
| 9056 | Santa's Monster Shootout DX | GAME               | 5.0    | 4       | 33M  | 50+      | Paid | \$1.99 | Teen           | Action           | August 15, 2013   | 1.05        | 2.2 and up  |
| 8395 | DG TV                       | NEWS_AND_MAGAZINES | 5.0    | 3       | 5.7M | 100+     | Free | 0      | Everyone       | News & Magazines | May 28, 2018      | 1.2         | 4.1 and up  |
| 8493 | PK and DK Audio App         | FAMILY             | 5.0    | 2       | 3.9M | 100+     | Free | 0      | Everyone       | Entertainment    | October 25, 2017  | 5.1.4       | 4.1 and up  |
| 8330 | HON. B.J. ACS COLLEGE ALE   | FAMILY             | 5.0    | 3       | 1.8M | 100+     | Free | 0      | Mature 17+     | Education        | December 26, 2016 | 3.1         | 4.3 and up  |
| 6342 | BJ Foods                    | BUSINESS           | 5.0    | 3       | 1.5M | 10+      | Free | 0      | Everyone       | Business         | February 7, 2018  | 2.7         | 4.1 and up  |
| 6363 | Read it easy for BK         | LIFESTYLE          | 5.0    | 1       | 3.2M | 50+      | Free | 0      | Everyone       | Lifestyle        | July 15, 2018     | 1.2         | 4.1 and up  |
| 9766 | ER Assist                   | PRODUCTIVITY       | 5.0    | 3       | 28M  | 10+      | Free | 0      | Everyone       | Productivity     | December 6, 2018  | 0.1.7       | 4.1 and up  |
| 6364 | BK Video Status             | FAMILY             | 5.0    | 13      | 2.1M | 100+     | Free | 0      | Everyone       | Entertainment    | July 7, 2018      | 2.3         | 4.4 and up  |
| 6372 | BK Formula Calculator       | TOOLS              | 5.0    | 6       | 11M  | 100+     | Free | 0      | Everyone       | Tools            | August 8, 2015    | 0.1.1       | 4.2 and up  |
| 6375 | Dr BK Sachin bhai           | LIFESTYLE          | 5.0    | 19      | 3.1M | 1,000+   | Free | 0      | Everyone       | Lifestyle        | December 7, 2017  | 2.2         | 4.1 and up  |

## — Sütun(ları) silmek

drop() fonksiyonun genel kullanımına baktığımızda axis = 0 olduğunu görüyoruz  
axis = 0 olması satır bazlı işlem yapmamızı sağlar.. Kolon bazlı işlem yapabilmek için ise 2 yöntem mevcuttur.

- si axis parametresinin değerini 1 olarak ayarlamak.

```
df= df.drop("Content Rating", axis=1)
```

- ise drop() fonksiyonunun columns parametresini kullanmak.

```
df =df.drop(columns ="Last Updated")
print(df.columns)
```

```
Index(['App', 'Category', 'Rating', 'Reviews', 'Size', 'Installs', 'Type',
       'Price', 'Genres', 'Current Ver', 'Android Ver'],
      dtype='object')
```

- df' deki verileri gruplayalım.

DataFrame nesnesindeki verileri grüplamak için groupby() fonksiyonu kullanılır.

```
print(df.groupby("Category")["Rating"].mean().head(10))
```

| Category                     |          |
|------------------------------|----------|
| ART_AND DESIGN               | 4.358065 |
| AUTO_AND VEHICLES            | 4.190411 |
| BEAUTY                       | 4.278571 |
| BOOKS_AND REFERENCE          | 4.346067 |
| BUSINESS                     | 4.121452 |
| COMICS                       | 4.155172 |
| COMMUNICATION                | 4.158537 |
| DATING                       | 3.970769 |
| EDUCATION                    | 4.389032 |
| ENTERTAINMENT                | 4.126174 |
| Name: Rating, dtype: float64 |          |

Yukarıda ki kodla verileri kategorilerine göre gruplandırdık ve her bir kategorinin ortalama puanını hesaplayarak ilk 10 kategoriyi ekrana yazdırıldı.

groupby() fonksiyonu birden fazla değişkene göre gruplama imkanı da sağlar. Şimdi de DataFrame nesnemizi “Category” ve “Type” değişkenlerine göre gruplayalım ve her bir grup için maximum puan değerini bulalım.

```
print(df.groupby(["Category", "Type"])["Rating"].max().head(10))
```

| Category                     | Type |     |
|------------------------------|------|-----|
| ART_AND DESIGN               | Free | 5.0 |
|                              | Paid | 4.8 |
| AUTO_AND VEHICLES            | Free | 4.9 |
|                              | Paid | 4.6 |
| BEAUTY                       | Free | 4.9 |
|                              | Paid | 5.0 |
| BOOKS_AND REFERENCE          | Free | 5.0 |
|                              | Paid | 5.0 |
| BUSINESS                     | Free | 5.0 |
|                              | Paid | 4.8 |
| COMICS                       | Free | 5.0 |
| Name: Rating, dtype: float64 |      |     |

- df’deki eksik verilerle çalışalım.

Gerçek hayatı veriler bir çok sebepten dolayı eksik veriler içeriyor. Yani çoğu zaman “Iris Dataseti” gibi temiz datasetlerle çalışmıyor. Bu sebeple eksik verileri yönetmeyi bilmek gerekiyor. Tabi eksik verileri yönetebilmek teknik bir bilgiyi öğrenmekten çok daha zor, zaman ve tecrübe gerektiriyor.

Biz şimdilik bu bölümde `fillna()` ve `dropna()` metodlarına kısaca değineceğiz. Zaten bu yazının amacı veri analizi yapmaktan ziyade veri analizi sırasında bize yardımcı olacak Pandas kütüphanesinin işlevlerini ve özelliklerini tanımkar. Öyleyse `dropna()` metodu ile başlayalım.

`dropna()` metodu NaN değerler bulunduran satırları veya sütunları silmemizi sağlıyor. ( Tabi ki satırları silmenin ne kadar doğru olacağı tartışıılır. )

|                           |      |
|---------------------------|------|
| Rating                    | 1474 |
| Current Ver               | 8    |
| Android Ver               | 3    |
| Content Rating            | 1    |
| Type                      | 1    |
| Last Updated              | 0    |
| Genres                    | 0    |
| Price                     | 0    |
| Installs                  | 0    |
| Size                      | 0    |
| Reviews                   | 0    |
| Category                  | 0    |
| App                       | 0    |
| <code>dtype: int64</code> |      |

Sütunlara göre NaN değer sayısı

“Rating” sütununda 1474 tane NaN değer olduğunu görüyoruz. NaN değerlerin fazlalığından dolayı burada `dropna()` kullanmak doğru bir tercih olmayabilir. Öyleyse öncelikle “Rating” sütunundaki NaN değerleri `fillna()` metodu ile bulunduğu kategorinin ortalaması ile dolduralım.

```
df["Rating"].fillna(df.groupby("Category")
                     ["Rating"].transform("mean"), inplace=True)

print(df.isnull().sum().sort_values(ascending=False))
```

|                           |   |
|---------------------------|---|
| Current Ver               | 8 |
| Android Ver               | 2 |
| Type                      | 1 |
| Last Updated              | 0 |
| Genres                    | 0 |
| Price                     | 0 |
| Installs                  | 0 |
| Size                      | 0 |
| Reviews                   | 0 |
| Rating                    | 0 |
| Category                  | 0 |
| App                       | 0 |
| <code>dtype: int64</code> |   |

Evet “Rating” sütunundaki NaN değerleri hallettik şimdi 11 tane NaN değerimiz var, onları da dropna() metodu ile silelim.

```
df = df.dropna()  
print(df.isnull().sum().sort_values(ascending=False))
```

```
          Android Ver      0  
          Current Ver     0  
          Last Updated     0  
          Genres           0  
          Price            0  
          Type             0  
          Installs         0  
          Size             0  
          Reviews          0  
          Rating           0  
          Category         0  
          App              0  
          dtype: int64
```

Şu anda df' de NaN değer bulunmuyor. df' i eksik verilerden böylelikle temizlemiş olduk .

Bu yazıda genel olarak verinin analizi ve keşfi sırasında Pandas'ı nasıl kullanacağımızı öğrendik. Görselleştirme kütüphanelerinden bahsedeceğim sonraki yazınlarda ise veriden daha anlamlı çıkarımlar yapacağımızı umuyorum. Sonraki yazınlarda görüşmek üzere..

Python

Data Science

Data Analysis

Pandas

Exploratory Data Analysis

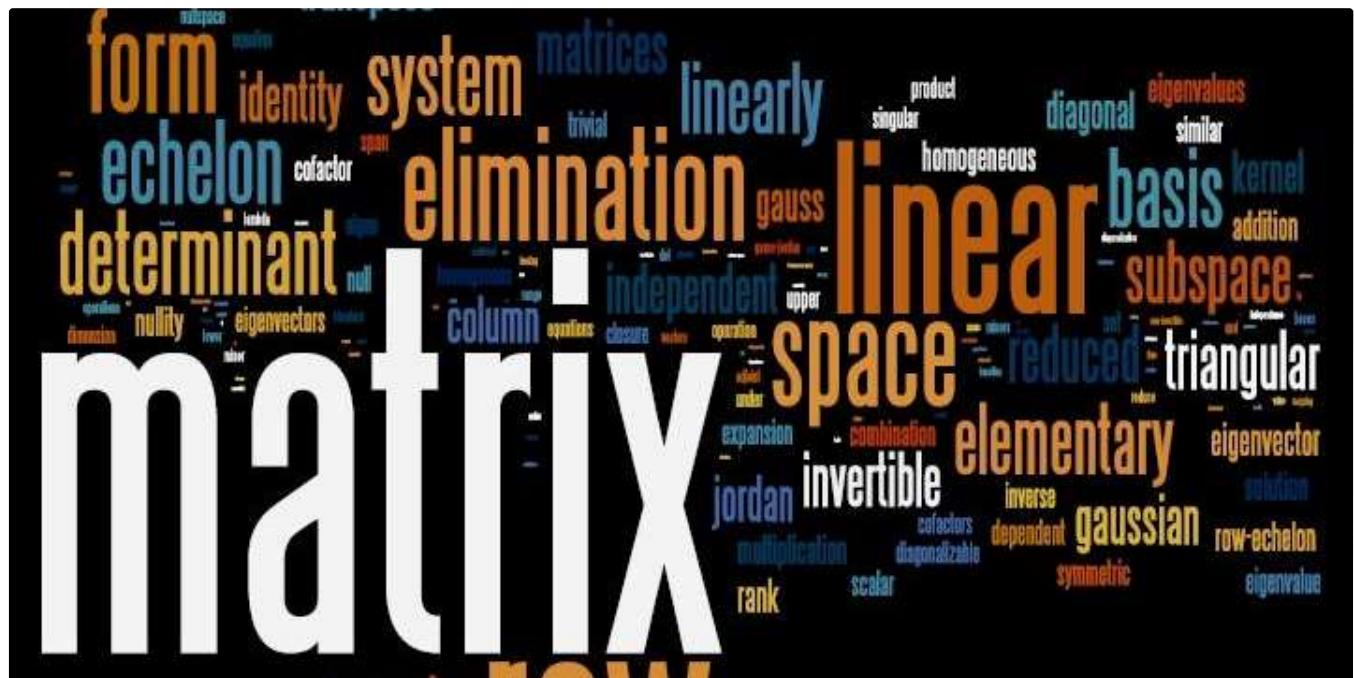
[Follow](#)

## Written by Merve Bayram Durna

1.2K Followers · Writer for Bilişim Hareketi

Data Analyst | Data Scientist

More from Merve Bayram Durna and Bilişim Hareketi



Merve Bayram Durna in Bilişim Hareketi

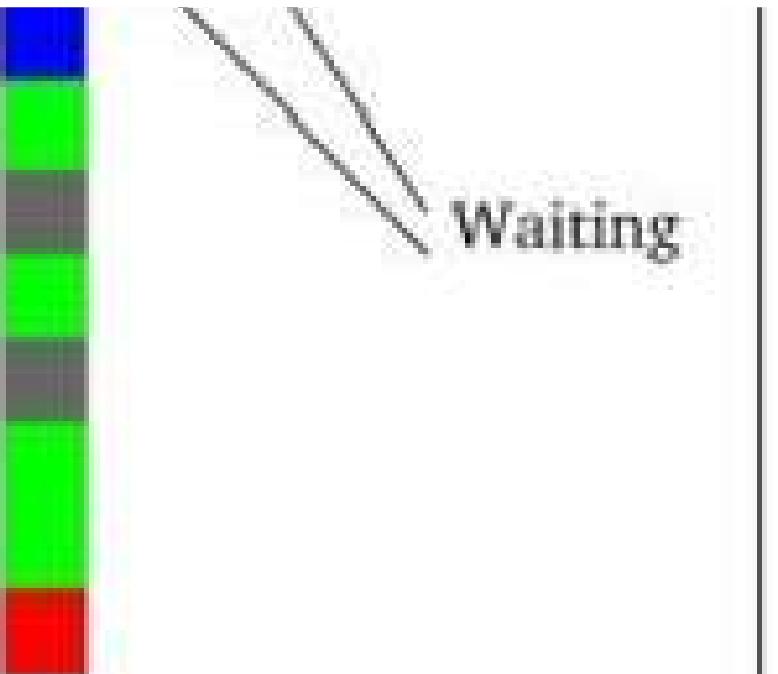
## Veri Bilimi İçin Temel Python Kütüphaneleri-1 : Numpy

NumPy (Numerical Python) bilimsel hesaplamaları hızlı bir şekilde yapmamızı sağlayan bir matematik kütüphanesidir. Numpy'ın temelini numpy...

7 min read · Feb 1, 2019



...



Task 2

Waiting

Task 3



Engin UNAL in Bilişim Hareketi

## .Net Asenkron(async & await) Programlama

Bu yazında .Net framework tarafından asenkron programlama konusu ele alınacak ve async & await kullanımları incelenecaktır.

7 min read · Sep 17, 2021



202



...



Ayhan KORKMAZ in Bilişim Hareketi

## Ücretsiz SSL Sertifikası Kurulumu—Let's Encrypt

Google belirli aralıklarla yaptığı güncellemelerle getirdiği şartları SEO kriteri olarak almaya devam ediyor. Son zamanlarda sizin de fark...

6 min read · Apr 5, 2020

👏 295

💬 2



...



Merve Bayram Durna in Bilişim Hareketi

## Cross-Validation nedir? Nasıl çalışır?

Bu yazıda cross-validation'ın ne olduğunu, neden ihtiyaç duyulduğunu ve çalışma prensibini anlamaya çalışacağız. Son olarak da Sklearn...

3 min read · May 28, 2020

👏 383

💬 2

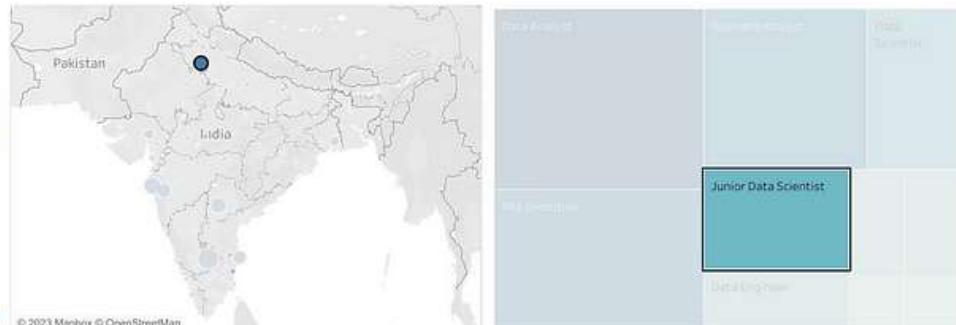


...

See all from Merve Bayram Durna

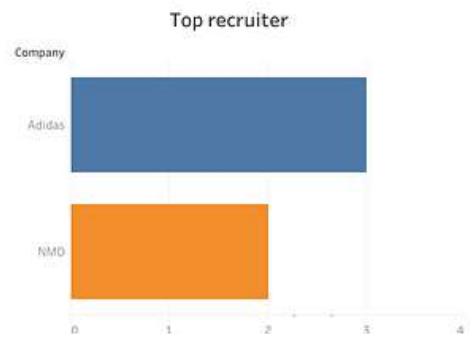
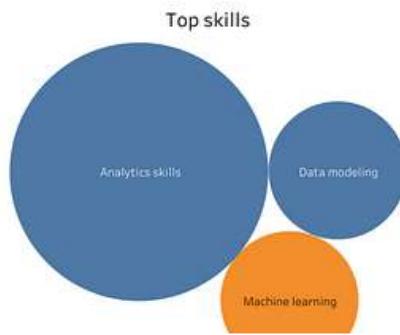
See all from Bilişim Hareketi

## Recommended from Medium



Welcome to this Tableau dashboard, designed to uncover key insights about different job roles and their most essential skills. With the ability to select a data role and adjust parameters for top jobs and skills, our dashboard offers a user-friendly interface to explore the top frequent skills required for specific professions. The interactive bar chart visually presents the data, making it easy to grasp the skill trends for each job. This dashboard provides valuable information for job seekers, career planners, and employers, aiding informed decision-making in today's competitive job market.

*Drag to the right to see more job roles*  
12  
*Drag to the right to see more skills*  
22  
*Drag to the right to see more recruiters*



Anshidtk

## Data analyst portfolio project

Scraping Job Insights from naukri.com

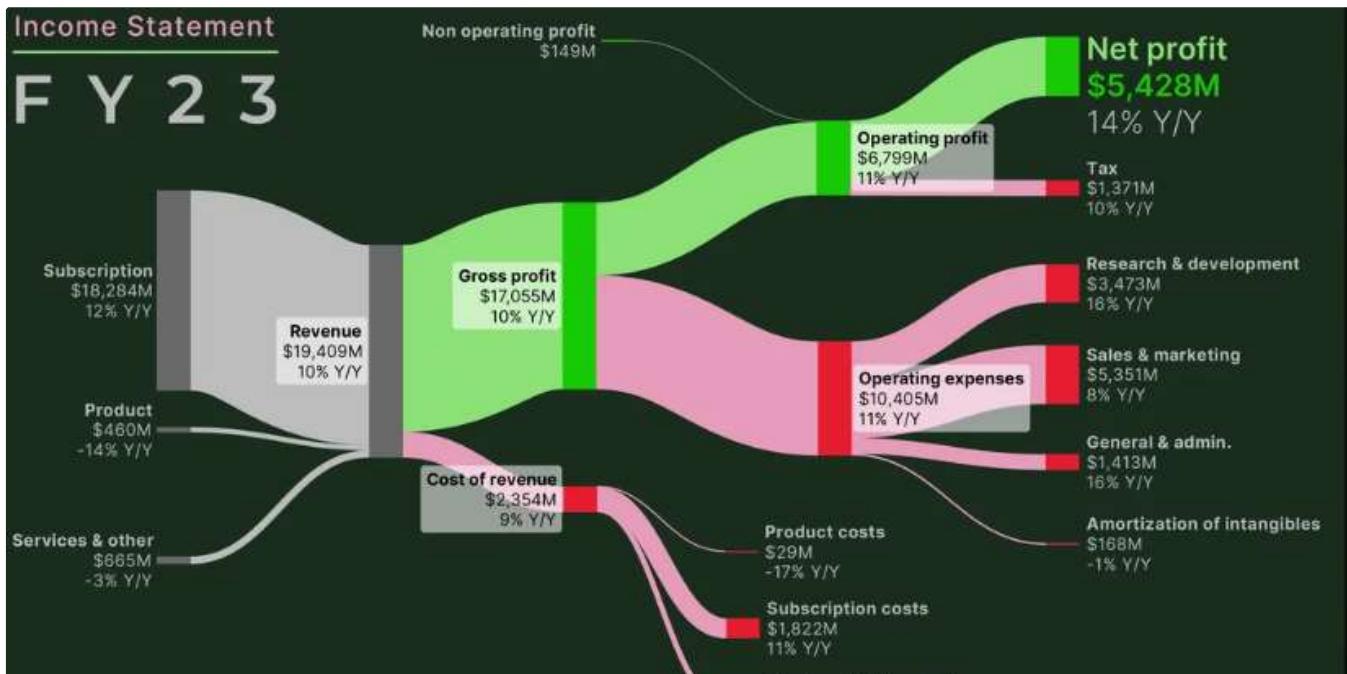
6 min read · Aug 27, 2023



46



...



Anmol Tomar in CodeX

## Top 10 Data Visualizations of 2023 Worth Looking at!

Level Up Your Visualization Game!

◆ · 4 min read · Dec 27, 2023

638 5



...

## Lists



### Predictive Modeling w/ Python

20 stories · 762 saves



### Practical Guides to Machine Learning

10 stories · 875 saves



### Coding & Development

11 stories · 363 saves



### ChatGPT

23 stories · 374 saves

## corresponding Labels

```
ls = ['20s', '30s', '40s', '50s']
```

rize each age into the defined bins

```
    ], bins=bins, labels=labels,
```

Tahera Firdose

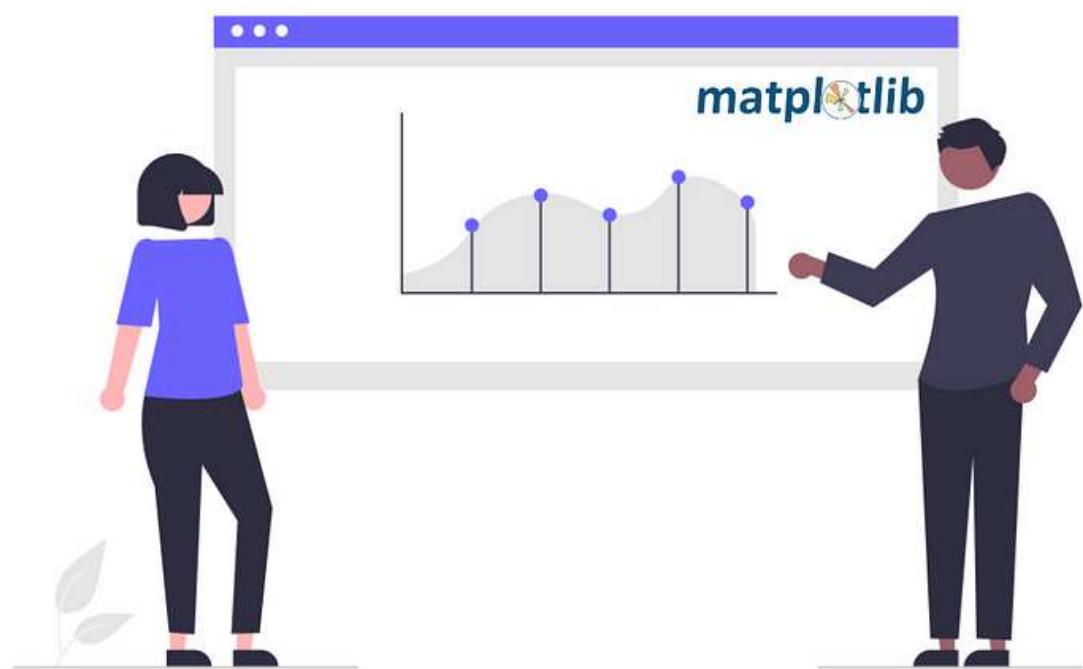
## Pandas Tips: Easy Tricks for Better Data Work

5 min read · Dec 13, 2023

5

+

...



chi

## Master Matplotlib: A Step-by-Step Guide for Beginners to Experts

Discover the power of Matplotlib, the essential data visualization library for Python. Learn the basics, explore advanced techniques, and u

26 min read · Dec 6, 2023



Amazing lifestyle

## [Python-Openpyxl-Excel] Excel in Python (Creating a New Column-Pycharm)

import and load the workbook



2 min read · Dec 19, 2023





Okan Yenigün in Level Up Coding

## Dask for Scalable Data Science: A Practical Exploration

Mastering Parallel Computing in Python with Dask: A Comprehensive Guide

14 min read · Dec 28, 2023

141



...

See more recommendations