

```
1 https://s2.dosya.tc/server26/rmdkj1/House_Rent_Dataset.csv.html
```

```
In [1]: 1 # Kullanacağımız kütüphaneleri yüklüyoruz
2 import pandas as pd
3 import numpy as np
4 import matplotlib.pyplot as plt
5 import warnings
6 warnings.simplefilter('ignore')
```

## PANDAS DATAFRAME

```
In [2]: 1 # dataframe oluşturmak(yöntem 1)
2 df_00 = pd.DataFrame(index=[0,1,2,3],columns=['sütun1','sütun2'])
3 df_00
```

```
Out[2]:
```

	sütun1	sütun2
0	NaN	NaN
1	NaN	NaN
2	NaN	NaN
3	NaN	NaN

```
In [3]: 1 #Dataframe sütunları
2 df_00.columns
```

```
Out[3]: Index(['sütun1', 'sütun2'], dtype='object')
```

```
In [4]: 1 #Dataframe sırası
2 df_00.index
```

```
Out[4]: Int64Index([0, 1, 2, 3], dtype='int64')
```

```
In [5]: 1 #Boş Dataframe oluşturmak
2 df_01 = pd.DataFrame()
3 print(df_01.columns)
4 print(df_01.index)
```

```
Index([], dtype='object')
Index([], dtype='object')
```

```
In [6]: 1 # List kullanarak dataframe oluşturmak (sıra ve sütunların index sırası aldığına dikkat edin)(yöntem 2)
2 list = [1,2,3,4,5]
3 df_02 =pd.DataFrame(list)
4 df_02
```

```
Out[6]:  0
         0 1
         1 2
         2 3
         3 4
         4 5
```

```
In [7]: 1 #dataframe oluşturmak(yöntem 3)
2 df_03 = pd.DataFrame(columns=['Column1','Column2'])
3 print(df_03)
```

```
Empty DataFrame
Columns: [Column1, Column2]
Index: []
```

```
In [8]: 1 #dictionary üzerinden dataframe oluşturmak(yöntem 3)
2 data = {'isim': ['Tom', 'nick', 'krish', 'jack'],
3         'yaş': [20, 21, 19, 18]}
4 df_04 = pd.DataFrame(data)
5 df_04
```

```
Out[8]:
```

	isim	yaş
0	Tom	20
1	nick	21
2	krish	19
3	jack	18

```
In [9]: 1 # index oluşturmak
2 data = {'isim': ['Tom', 'nick', 'krish', 'jack'],
3         'yaş': [20, 21, 19, 18]}
4 ranks=['sıra1', 'sıra2', 'sıra3', 'sıra4']
5 df_05 = pd.DataFrame(data, index=ranks)
6 df_05
```

```
Out[9]:
```

	isim	yaş
sıra1	Tom	20
sıra2	nick	21
sıra3	krish	19
sıra4	jack	18

```
In [10]: 1 df_05.index
```

```
Out[10]: Index(['sıra1', 'sıra2', 'sıra3', 'sıra4'], dtype='object')
```

```
In [11]: 1 df_05.columns
```

```
Out[11]: Index(['isim', 'yaş'], dtype='object')
```

In [12]:

```
1 # yeni sütun eklemek  
2 df_05 ['doğum_yeri'] = ['New York', 'Boston', 'San Francisco', 'New Delhi']
```

In [13]:

```
1 # yeni sütun eklerken data uzunluğu aynı olmalıdır  
2 df_05['mezun_olduğu_okul'] = ['NYU', 'Boston State University', 'San Francisco State', 'Istanbul Gelisim University']
```

```

-----
ValueError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_5376\3970332306.py in <module>
      1 # yeni sütun eklerken data uzunluğu aynı olmalıdır
----> 2 df_05['mezun_olduğu_okul'] = ['NYU', 'Boston State University', 'San Francisco State', 'Istanbul Gelisim Univer
sity', 'Stanford University']

C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\frame.py in __setitem__(self, key, value)
    3653         else:
    3654             # set column
-> 3655             self._set_item(key, value)
    3656
    3657     def _setitem_slice(self, key: slice, value):

C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\frame.py in _set_item(self, key, value)
    3830         ensure_homogeneity.
    3831         """
-> 3832         value = self._sanitize_column(value)
    3833
    3834         if (

C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\frame.py in _sanitize_column(self, value)
    4536
    4537         if is_list_like(value):
-> 4538             com.require_length_match(value, self.index)
    4539         return sanitize_array(value, self.index, copy=True, allow_2d=True)
    4540

C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\common.py in require_length_match(data, index)
    555         """
    556         if len(data) != len(index):
--> 557             raise ValueError(
    558                 "Length of values "
    559                 f"({len(data)}) "

```

**ValueError:** Length of values (5) does not match length of index (4)

```
In [14]: 1 df_05['mezun_olduğu_okul'] = ['NYU', 'Boston State University', 'San Francisco State', 'Istanbul Gelisim University']
```

```
In [15]: 1 df_05
```

```
Out[15]:
```

	isim	yaş	doğum_yeri	mezun_olduğu_okul
<b>sıra1</b>	Tom	20	New York	NYU
<b>sıra2</b>	nick	21	Boston	Boston State University
<b>sıra3</b>	krish	19	San Francisco	San Francisco State
<b>sıra4</b>	jack	18	New Delhi	Istanbul Gelisim University

```
In [16]: 1 #Dataframe datatipleri
2 df_05.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 4 entries, sıra1 to sıra4
Data columns (total 4 columns):
#   Column                Non-Null Count  Dtype
---  -
0   isim                  4 non-null     object
1   yaş                   4 non-null     int64
2   doğum_yeri            4 non-null     object
3   mezun_olduğu_okul     4 non-null     object
dtypes: int64(1), object(3)
memory usage: 160.0+ bytes
```

```
In [17]: 1 #dataframe boyutu
2 df_05.shape
```

```
Out[17]: (4, 4)
```

```
In [18]: 1 #dataframe temel istatistikler
        2 df_05.describe()
```

```
Out[18]:
```

	yaş
count	4.000000
mean	19.500000
std	1.290994
min	18.000000
25%	18.750000
50%	19.500000
75%	20.250000
max	21.000000

```
In [19]: 1 df_05
```

```
Out[19]:
```

	isim	yaş	doğum_yeri	mezun_olduğu_okul
sıra1	Tom	20	New York	NYU
sıra2	nick	21	Boston	Boston State University
sıra3	krish	19	San Francisco	San Francisco State
sıra4	jack	18	New Delhi	Istanbul Gelisim University

```
In [20]: 1 df_05
```

```
Out[20]:
```

	isim	yaş	doğum_yeri	mezun_olduğu_okul
sıra1	Tom	20	New York	NYU
sıra2	nick	21	Boston	Boston State University
sıra3	krish	19	San Francisco	San Francisco State
sıra4	jack	18	New Delhi	Istanbul Gelisim University



# DATA FRAMİ SİLMEK

```
In [21]: 1 df_06 = df_05.head(2)
         2 df_06
```

```
Out[21]:
```

	isim	yaş	doğum_yeri	mezun_olduğu_okul
<b>sıra1</b>	Tom	20	New York	NYU
<b>sıra2</b>	nick	21	Boston	Boston State University

```
In [22]: 1 del df_06
```

```
In [23]: 1 df_06
```

```
-----
NameError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_5376\2339950171.py in <module>
-----> 1 df_06

NameError: name 'df_06' is not defined
```

```
In [ ]: 1
```

```
In [ ]: 1
```

## sütun silmek drop(axis=1)

```
In [24]: 1 #sütun silmek
        2 df_05.drop(['mezun_olduğu_okul'],axis=1)
```

```
Out[24]:
```

	isim	yaş	doğum_yeri
sıra1	Tom	20	New York
sıra2	nick	21	Boston
sıra3	krish	19	San Francisco
sıra4	jack	18	New Delhi

## YUKARIDA SÜTUNU SİLDİK AMA YİNE GÖZÜKÜYOR, NİÇİN?

```
In [25]: 1 # inplace=True (kayıt eder)
        2 df_05.drop(['mezun_olduğu_okul'],axis=1,inplace=True)
```

```
In [26]: 1 df_05
```

```
Out[26]:
```

	isim	yaş	doğum_yeri
sıra1	Tom	20	New York
sıra2	nick	21	Boston
sıra3	krish	19	San Francisco
sıra4	jack	18	New Delhi

```
In [27]: 1 # Sıra silinmesi axis=0
        2 df_05.drop(['sıra4'],axis=0)
```

```
Out[27]:
```

	isim	yaş	doğum_yeri
sıra1	Tom	20	New York
sıra2	nick	21	Boston
sıra3	krish	19	San Francisco

```
In [29]: 1 #Birden fazla sıra silinmesi
        2 df_05.drop(['sıra2', 'sıra4'],axis=0)
```

```
Out[29]:
```

	isim	yaş	doğum_yeri
<b>sıra1</b>	Tom	20	New York
<b>sıra3</b>	krish	19	San Francisco

```
In [30]: 1 df_05
```

```
Out[30]:
```

	isim	yaş	doğum_yeri
<b>sıra1</b>	Tom	20	New York
<b>sıra2</b>	nick	21	Boston
<b>sıra3</b>	krish	19	San Francisco
<b>sıra4</b>	jack	18	New Delhi

```
In [33]: 1 df_05.drop(['sıra4'],axis=0,inplace=True)
        2 df_05
```

```
Out[33]:
```

	isim	yaş	doğum_yeri
<b>sıra1</b>	Tom	20	New York
<b>sıra2</b>	nick	21	Boston
<b>sıra3</b>	krish	19	San Francisco

```
In [34]: 1 df_04
```

```
Out[34]:
```

	isim	yaş
<b>0</b>	Tom	20
<b>1</b>	nick	21
<b>2</b>	krish	19
<b>3</b>	jack	18

```
In [35]: 1 # indeksleme ile sıra silinmesi  
2 df_04.drop([1,3],axis=0)
```

```
Out[35]:
```

	isim	yaş
0	Tom	20
2	krish	19

```
In [ ]:
```

```
1
```

```
In [36]: 1 from IPython.display import display, Image
          2 display(Image(filename='Pandas-selections-and-indexing.png'))
```

# Python Pandas Selections and Indexing

---

## .iloc selections - position based selection

`data.iloc[<row selection>, <column selection>]`

*Integer list of rows: [0,1,2]*

*Integer list of columns: [0,1,2]*

*Slice of rows: [4:7]*

*Slice of columns: [4:7]*

*Single values: 1*

*Single column selections: 1*

---

## loc selections - position based selection

`data.loc[<row selection>, <column selection>]`

*Index/Label value: 'john'*

*Named column: 'first\_name'*

*List of labels: ['john', 'sarah']*

*List of column names: ['first\_name', 'age']*

*Logical/Boolean index: data['age'] == 10*

*Slice of columns: 'first\_name':'address'*

```
In [37]: 1 df_07 = pd.DataFrame([[ 'Ali',18,'istanbul'],[ 'Veli',25,'istanbul'],[ 'Ayşe',20,'izmir'],[ 'Fatma',23,'ankara']],
2         columns=[ 'isimler','yaşlar','doğum_yerleri'])
```

```
In [39]: 1 print(df_07)
```

```
   isimler  yaşlar doğum_yerleri
0      Ali     18      istanbul
1      Veli     25      istanbul
2     Ayşe     20        izmir
3     Fatma     23      ankara
```

```
In [ ]: 1
```

## sütuna ulaşmak

```
In [40]: 1 # bu bir seridir
2 df_07[ 'isimler']
```

```
Out[40]: 0      Ali
1      Veli
2     Ayşe
3     Fatma
Name: isimler, dtype: object
```

```
In [41]: 1 # bu bir data framedir
2 df_07[[ 'isimler']]
```

```
Out[41]:
```

	isimler
0	Ali
1	Veli
2	Ayşe
3	Fatma

```
In [42]: 1 # bu bir seridir
          2 df_07.loc[:, 'isimler']
```

```
Out[42]: 0      Ali
          1      Veli
          2      Ayşe
          3      Fatma
          Name: isimler, dtype: object
```

```
In [43]: 1 # bu bir data framedir
          2 df_07.loc[:, ['isimler']]
```

```
Out[43]:
```

	isimler
0	Ali
1	Veli
2	Ayşe
3	Fatma

```
In [44]: 1 # bu bir seridir
          2 df_07.iloc[:, 0]
```

```
Out[44]: 0      Ali
          1      Veli
          2      Ayşe
          3      Fatma
          Name: isimler, dtype: object
```

```
In [45]: 1 # bu bir dataframedir  
2 df_07.iloc[:,[0]]
```

```
Out[45]:
```

	isimler
0	Ali
1	Veli
2	Ayşe
3	Fatma

```
In [ ]: 1
```

## çoklu sıra ve sütuna ulaşmak

```
In [46]: 1 df_07[['isimler', 'doğum_yerleri']]
```

```
Out[46]:
```

	isimler	doğum_yerleri
0	Ali	istanbul
1	Veli	istanbul
2	Ayşe	izmir
3	Fatma	ankara

```
In [47]: 1 df_07.loc[:,['isimler', 'doğum_yerleri']]
```

```
Out[47]:
```

	isimler	doğum_yerleri
0	Ali	istanbul
1	Veli	istanbul
2	Ayşe	izmir
3	Fatma	ankara



```
In [48]: 1 df_07.iloc[:,[0,2]]
```

```
Out[48]:
```

	isimler	doğum_yerleri
0	Ali	istanbul
1	Veli	istanbul
2	Ayşe	izmir
3	Fatma	ankara

```
In [49]: 1 df_07.iloc[:,[0,-1]]
```

```
Out[49]:
```

	isimler	doğum_yerleri
0	Ali	istanbul
1	Veli	istanbul
2	Ayşe	izmir
3	Fatma	ankara

```
In [ ]:
```

```
1
```

## FİLTRELEME

```
In [50]: 1 # istanbul doğumlu kişiler  
2 df_07[df_07['doğum_yerleri']=='istanbul']
```

```
Out[50]:
```

	isimler	yaşlar	doğum_yerleri
0	Ali	18	istanbul
1	Veli	25	istanbul

```
In [51]: 1 # istanbul doğumlu olmayan kişiler
        2 df_07[df_07['doğum_yerleri']!='istanbul']
```

```
Out[51]:
```

	isimler	yaşlar	doğum_yerleri
2	Ayşe	20	izmir
3	Fatma	23	ankara

```
In [52]: 1 # yaşları 20 den küçük kişiler
        2 df_07[df_07['yaşlar']<20]
```

```
Out[52]:
```

	isimler	yaşlar	doğum_yerleri
0	Ali	18	istanbul

```
In [53]: 1 # yaşları 20 veya 20 den küçük kişiler
        2 df_07[df_07['yaşlar']<=20]
```

```
Out[53]:
```

	isimler	yaşlar	doğum_yerleri
0	Ali	18	istanbul
2	Ayşe	20	izmir

```
In [54]: 1 #en yaşlı kişi
        2 df_07[df_07['yaşlar']==df_07['yaşlar'].max()]
```

```
Out[54]:
```

	isimler	yaşlar	doğum_yerleri
1	Veli	25	istanbul

```
In [55]: 1 #en yaşlı kişinin ismi (list olarak)
        2 df_07[df_07['yaşlar']==df_07['yaşlar'].max()][ 'isimler']
```

```
Out[55]: 1 Veli
Name: isimler, dtype: object
```

```
In [56]: 1 #en yaşlı kişinin ismi
2 df_07[df_07['yaşlar']==df_07['yaşlar'].max()][ 'isimler'].values[0]
```

Out[56]: 'Veli'

```
In [57]: 1 #en genç kişi
2 df_07[df_07['yaşlar']==df_07['yaşlar'].min()]
```

Out[57]:

	isimler	yaşlar	doğum_yerleri
0	Ali	18	istanbul

```
In [58]: 1 #en genç kişinin ismi (list olarak)
2 df_07[df_07['yaşlar']==df_07['yaşlar'].min()][ 'isimler']
```

Out[58]: 0 Ali  
Name: isimler, dtype: object

```
In [59]: 1 #en genç kişinin ismi
2 df_07[df_07['yaşlar']==df_07['yaşlar'].min()][ 'isimler'][0]
```

Out[59]: 'Ali'

```
In [ ]: 1
```

## ÇOKLU FİLTRELEME

```
In [60]: 1 # 20 ve 20 üstü yaşında olup ve istanbullu olmayanlar (parantez kullanmak zorundayız)
2 df_07[(df_07['yaşlar']>=20) & (df_07['doğum_yerleri']!='istanbul') ]
```

Out[60]:

	isimler	yaşlar	doğum_yerleri
2	Ayşe	20	izmir
3	Fatma	23	ankara

```
In [61]: 1 #25 ve altı yaşda olup ismi A harfi ile başlayanlar
          2 df_07[(df_07['yaşlar']<=25) & (df_07['isimler'].str.startswith('A'))]
```

```
Out[61]:
```

	isimler	yaşlar	doğum_yerleri
0	Ali	18	istanbul
2	Ayşe	20	izmir

```
In [62]: 1 #25 ve altı yaşda olup ismi i harfi ile bitenler
          2 df_07[(df_07['yaşlar']<=25) & (df_07['isimler'].str.endswith('i'))]
```

```
Out[62]:
```

	isimler	yaşlar	doğum_yerleri
0	Ali	18	istanbul
1	Veli	25	istanbul

## pd.read\_csv()

```
In [ ]: 1
```

## Eğer farklı bir dosyadan ulaşıyorsanız

```
In [63]: 1 indian_rent = pd.read_csv("C:/Users/mbenturk/Documents/python/python_begginer_course_notes/week_4/House_Rent_Data.csv")
```

In [64]: 1 indian\_rent

Out[64]:

	Posted On	BHK	Rent	Size	Floor	Area Type	Area Locality	City	Furnishing Status	Tenant Preferred	Bathroom	Point of Contact
0	5/18/2022	2.0	10000.0	1100	Ground out of 2	Super Area	Bandel	Kolkata	Unfurnished	Bachelors/Family	2.0	Contact Owner
1	5/13/2022	2.0	20000.0	800	1 out of 3	Super Area	Phool Bagan, Kankurgachi	Kolkata	Semi-Furnished	Bachelors/Family	1.0	Contact Owner
2	5/16/2022	2.0	17000.0	1000	1 out of 3	Super Area	Salt Lake City Sector 2	Kolkata	Semi-Furnished	Bachelors/Family	1.0	Contact Owner
3	7/4/2022	2.0	10000.0	800	1 out of 2	Super Area	Dumdum Park	Kolkata	Unfurnished	Bachelors/Family	1.0	Contact Owner
4	5/9/2022	2.0	7500.0	850	1 out of 2	Carpet Area	South Dum Dum	Kolkata	Unfurnished	Bachelors	1.0	Contact Owner
...	...	...	...	...	...	...	...	...	...	...	...	...
4744	5/18/2022	2.0	15000.0	1000	3 out of 5	Carpet Area	Bandam Kommu	Hyderabad	Semi-Furnished	Bachelors/Family	2.0	Contact Owner
4745	5/15/2022	3.0	29000.0	2000	1 out of 4	Super Area	Manikonda, Hyderabad	Hyderabad	Semi-Furnished	Bachelors/Family	3.0	Contact Owner
4746	7/10/2022	3.0	35000.0	1750	3 out of 5	Carpet Area	Himayath Nagar, NH 7	Hyderabad	Semi-Furnished	Bachelors/Family	3.0	Contact Agent
4747	7/6/2022	3.0	45000.0	1500	23 out of 34	Carpet Area	Gachibowli	Hyderabad	Semi-Furnished	Family	2.0	Contact Agent
4748	5/4/2022	2.0	15000.0	1000	4 out of 5	Carpet Area	Suchitra Circle	Hyderabad	Unfurnished	Bachelors	2.0	Contact Owner

4749 rows × 12 columns

In [65]: 1 indian\_rent.columns

Out[65]: Index(['Posted On', 'BHK', 'Rent', 'Size', 'Floor', 'Area Type', 'Area Locality', 'City', 'Furnishing Status', 'Tenant Preferred', 'Bathroom', 'Point of Contact'], dtype='object')

```
In [66]: 1 #sütun isimlerini Türkçeye çevirdik
2 indian_rent.columns = ['ilan_tarihi', 'oda_sayısı', 'kira', 'boyut', 'bulunduğu_kat', 'zemin_tipi', 'lokasyon_türü',
3                       'şehir', 'mobilyalı_mobilyasız', 'kiracı_tercihi', 'banyo_sayısı', 'erişim_yolu']
```

```
In [67]: 1 #ilk ve son beş satır
2 indian_rent
```

```
Out[67]:
```

	ilan_tarihi	oda_sayısı	kira	boyut	bulunduğu_kat	zemin_tipi	lokasyon_türü	şehir	mobilyalı_mobilyasız	kiracı_tercihi	banyo_s
0	5/18/2022	2.0	10000.0	1100	Ground out of 2	Super Area	Bandel	Kolkata	Unfurnished	Bachelors/Family	
1	5/13/2022	2.0	20000.0	800	1 out of 3	Super Area	Phool Bagan, Kankurgachi	Kolkata	Semi-Furnished	Bachelors/Family	
2	5/16/2022	2.0	17000.0	1000	1 out of 3	Super Area	Salt Lake City Sector 2	Kolkata	Semi-Furnished	Bachelors/Family	
3	7/4/2022	2.0	10000.0	800	1 out of 2	Super Area	Dumdum Park	Kolkata	Unfurnished	Bachelors/Family	
4	5/9/2022	2.0	7500.0	850	1 out of 2	Carpet Area	South Dum Dum	Kolkata	Unfurnished	Bachelors	
...	...	...	...	...	...	...	...	...	...	...	...
4744	5/18/2022	2.0	15000.0	1000	3 out of 5	Carpet Area	Bandam Kommu	Hyderabad	Semi-Furnished	Bachelors/Family	
4745	5/15/2022	3.0	29000.0	2000	1 out of 4	Super Area	Manikonda, Hyderabad	Hyderabad	Semi-Furnished	Bachelors/Family	
4746	7/10/2022	3.0	35000.0	1750	3 out of 5	Carpet Area	Himayath Nagar, NH 7	Hyderabad	Semi-Furnished	Bachelors/Family	
4747	7/6/2022	3.0	45000.0	1500	23 out of 34	Carpet Area	Gachibowli	Hyderabad	Semi-Furnished	Family	
4748	5/4/2022	2.0	15000.0	1000	4 out of 5	Carpet Area	Suchitra Circle	Hyderabad	Unfurnished	Bachelors	

4749 rows × 12 columns

In [68]:

```
1 # head() metodu ile ilk satırları çağırabiliriz, eğer bir rakam koymazsak ilk beş satır gelir
2 indian_rent.head(10)
```

Out[68]:

	ilan_tarihi	oda_sayısı	kira	boyut	bulunduğu_kat	zemin_tipi	lokasyon_türü	şehir	mobilyalı_mobilyasız	kiracı_tercihi	banyo_sayısı
0	5/18/2022	2.0	10000.0	1100	Ground out of 2	Super Area	Bandel	Kolkata	Unfurnished	Bachelors/Family	2.0
1	5/13/2022	2.0	20000.0	800	1 out of 3	Super Area	Phool Bagan, Kankurgachi	Kolkata	Semi-Furnished	Bachelors/Family	1.0
2	5/16/2022	2.0	17000.0	1000	1 out of 3	Super Area	Salt Lake City Sector 2	Kolkata	Semi-Furnished	Bachelors/Family	1.0
3	7/4/2022	2.0	10000.0	800	1 out of 2	Super Area	Dumdum Park	Kolkata	Unfurnished	Bachelors/Family	1.0
4	5/9/2022	2.0	7500.0	850	1 out of 2	Carpet Area	South Dum Dum	Kolkata	Unfurnished	Bachelors	1.0
5	4/29/2022	2.0	7000.0	600	Ground out of 1	Super Area	Thakurpukur	Kolkata	Unfurnished	Bachelors/Family	2.0
6	6/21/2022	2.0	10000.0	700	Ground out of 4	Super Area	Malancha	Kolkata	Unfurnished	Bachelors	2.0
7	6/21/2022	1.0	5000.0	250	1 out of 2	Super Area	Malancha	Kolkata	Unfurnished	Bachelors	1.0
8	6/7/2022	2.0	26000.0	800	1 out of 2	Carpet Area	Palm Avenue Kolkata, Ballygunge	Kolkata	Unfurnished	Bachelors	2.0
9	6/20/2022	2.0	10000.0	1000	1 out of 3	Carpet Area	Natunhat	Kolkata	Semi-Furnished	Bachelors/Family	2.0

In [69]:

```
1 #tail() metodu ile son satırları çağırabiliriz, eğer bir rakam koymazsak son beş satır gelir
2 indian_rent.tail(7)
```

Out[69]:

	ilan_tarihi	oda_sayısı	kira	boyut	bulunduğu_kat	zemin_tipi	lokasyon_türü	şehir	mobilyalı_mobilyasız	kiracı_tercihi	banyo_s
<b>4742</b>	7/6/2022	2.0	25000.0	1040	2 out of 4	Carpet Area	Gachibowli	Hyderabad	Unfurnished	Bachelors	
<b>4743</b>	6/2/2022	2.0	12000.0	1350	2 out of 2	Super Area	Old Alwal	Hyderabad	Unfurnished	Bachelors/Family	
<b>4744</b>	5/18/2022	2.0	15000.0	1000	3 out of 5	Carpet Area	Bandam Kommu	Hyderabad	Semi-Furnished	Bachelors/Family	
<b>4745</b>	5/15/2022	3.0	29000.0	2000	1 out of 4	Super Area	Manikonda, Hyderabad	Hyderabad	Semi-Furnished	Bachelors/Family	
<b>4746</b>	7/10/2022	3.0	35000.0	1750	3 out of 5	Carpet Area	Himayath Nagar, NH 7	Hyderabad	Semi-Furnished	Bachelors/Family	
<b>4747</b>	7/6/2022	3.0	45000.0	1500	23 out of 34	Carpet Area	Gachibowli	Hyderabad	Semi-Furnished	Family	
<b>4748</b>	5/4/2022	2.0	15000.0	1000	4 out of 5	Carpet Area	Suchitra Circle	Hyderabad	Unfurnished	Bachelors	



```
In [70]: 1 # İndekslemeyi kullanarak da satırları çağırabiliriz.
        2 indian_rent[0:5]
```

Out[70]:

	ilan_tarihi	oda_sayısı	kira	boyut	bulunduğu_kat	zemin_tipi	lokasyon_türü	şehir	mobilyalı_mobilyasız	kiracı_tercihi	banyo_sayısı
0	5/18/2022	2.0	10000.0	1100	Ground out of 2	Super Area	Bandel	Kolkata	Unfurnished	Bachelors/Family	2.0
1	5/13/2022	2.0	20000.0	800	1 out of 3	Super Area	Phool Bagan, Kankurgachi	Kolkata	Semi-Furnished	Bachelors/Family	1.0
2	5/16/2022	2.0	17000.0	1000	1 out of 3	Super Area	Salt Lake City Sector 2	Kolkata	Semi-Furnished	Bachelors/Family	1.0
3	7/4/2022	2.0	10000.0	800	1 out of 2	Super Area	Dumdum Park	Kolkata	Unfurnished	Bachelors/Family	1.0
4	5/9/2022	2.0	7500.0	850	1 out of 2	Carpet Area	South Dum Dum	Kolkata	Unfurnished	Bachelors	1.0

```
In [71]: 1 # ilk 5 sıradaki tüm sütunlar
        2 indian_rent.iloc[0:5,:]
```

Out[71]:

	ilan_tarihi	oda_sayısı	kira	boyut	bulunduğu_kat	zemin_tipi	lokasyon_türü	şehir	mobilyalı_mobilyasız	kiracı_tercihi	banyo_sayısı
0	5/18/2022	2.0	10000.0	1100	Ground out of 2	Super Area	Bandel	Kolkata	Unfurnished	Bachelors/Family	2.0
1	5/13/2022	2.0	20000.0	800	1 out of 3	Super Area	Phool Bagan, Kankurgachi	Kolkata	Semi-Furnished	Bachelors/Family	1.0
2	5/16/2022	2.0	17000.0	1000	1 out of 3	Super Area	Salt Lake City Sector 2	Kolkata	Semi-Furnished	Bachelors/Family	1.0
3	7/4/2022	2.0	10000.0	800	1 out of 2	Super Area	Dumdum Park	Kolkata	Unfurnished	Bachelors/Family	1.0
4	5/9/2022	2.0	7500.0	850	1 out of 2	Carpet Area	South Dum Dum	Kolkata	Unfurnished	Bachelors	1.0

```
In [72]: 1 # ilk 5 sıradaki tüm sütunlar ( loc yönteminde indexleme farklıdır dikkat edin)
        2 indian_rent.loc[0:4,:]
```

Out[72]:

	ilan_tarihi	oda_sayısı	kira	boyut	bulunduğu_kat	zemin_tipi	lokasyon_türü	şehir	mobilyalı_mobilyasız	kiracı_tercihi	banyo_sayısı
0	5/18/2022	2.0	10000.0	1100	Ground out of 2	Super Area	Bandel	Kolkata	Unfurnished	Bachelors/Family	2.0
1	5/13/2022	2.0	20000.0	800	1 out of 3	Super Area	Phool Bagan, Kankurgachi	Kolkata	Semi-Furnished	Bachelors/Family	1.0
2	5/16/2022	2.0	17000.0	1000	1 out of 3	Super Area	Salt Lake City Sector 2	Kolkata	Semi-Furnished	Bachelors/Family	1.0
3	7/4/2022	2.0	10000.0	800	1 out of 2	Super Area	Dumdum Park	Kolkata	Unfurnished	Bachelors/Family	1.0
4	5/9/2022	2.0	7500.0	850	1 out of 2	Carpet Area	South Dum Dum	Kolkata	Unfurnished	Bachelors	1.0

```
In [73]: 1 # ilk 5 sıradaki kira verisi (bu dataframe sonucu değildir dikkat edin)
        2 indian_rent[0:5]['kira']
```

Out[73]:

```
0    10000.0
1    20000.0
2    17000.0
3    10000.0
4     7500.0
Name: kira, dtype: float64
```

```
In [74]: 1 #bu dataframe sonucudur  
2 indian_rent[0:5][['kira']]
```

```
Out[74]:      kira  
0  10000.0  
1  20000.0  
2  17000.0  
3  10000.0  
4   7500.0
```

```
In [75]: 1 # son beş sıradaki kira ve şehir bilgileri  
2 indian_rent[-5:-1][['kira','şehir']]
```

```
Out[75]:      kira      şehir  
4744  15000.0  Hyderabad  
4745  29000.0  Hyderabad  
4746  35000.0  Hyderabad  
4747  45000.0  Hyderabad
```

```
In [76]: 1 #ilk 5 sıradaki kira verisi iloc ile  
2 indian_rent.iloc[0:5,2]
```

```
Out[76]: 0    10000.0  
1    20000.0  
2    17000.0  
3    10000.0  
4     7500.0  
Name: kira, dtype: float64
```

```
In [77]: 1 #ilk 5 sıradaki rent verisi loc ile
        2 indian_rent.loc[0:4,'kira']
```

```
Out[77]: 0    10000.0
        1    20000.0
        2    17000.0
        3    10000.0
        4     7500.0
        Name: kira, dtype: float64
```

```
In [78]: 1 # eğer birden fazla sütun seçersek (ikili kşeli parenteze dikkat edin)
        2 indian_rent[0:5][['kira','boyut']]# we must use double pranthesis
```

```
Out[78]:
```

	kira	boyut
0	10000.0	1100
1	20000.0	800
2	17000.0	1000
3	10000.0	800
4	7500.0	850

```
In [79]: 1 # iloc ile
        2 indian_rent[0:5].iloc[0:5,[2,3]]
```

```
Out[79]:
```

	kira	boyut
0	10000.0	1100
1	20000.0	800
2	17000.0	1000
3	10000.0	800
4	7500.0	850

```
In [80]: 1 # loc ile  
        2 indian_rent.loc[0:4,['kira','boyut']]
```

```
Out[80]:
```

	kira	boyut
0	10000.0	1100
1	20000.0	800
2	17000.0	1000
3	10000.0	800
4	7500.0	850

## MANTIKSAL İŞLEMLER

```
In [81]: 1 # SADECE rentin 10000' eşit olanlarını görmek istiyoruz
        2 indian_rent[indian_rent['kira']==10000]
```

Out[81]:

	ilan_tarihi	oda_sayısı	kira	boyut	bulunduğu_kat	zemin_tipi	lokasyon_türü	şehir	mobilyalı_mobilyasız	kiracı_tercihi	banyo_s
0	5/18/2022	2.0	10000.0	1100	Ground out of 2	Super Area	Bandel	Kolkata	Unfurnished	Bachelors/Family	
3	7/4/2022	2.0	10000.0	800	1 out of 2	Super Area	Dumdum Park	Kolkata	Unfurnished	Bachelors/Family	
6	6/21/2022	2.0	10000.0	700	Ground out of 4	Super Area	Malancha	Kolkata	Unfurnished	Bachelors	
9	6/20/2022	2.0	10000.0	1000	1 out of 3	Carpet Area	Natunhat	Kolkata	Semi-Furnished	Bachelors/Family	
17	6/20/2022	2.0	10000.0	800	Ground out of 2	Super Area	Behala	Kolkata	Unfurnished	Bachelors/Family	
...	...	...	...	...	...	...	...	...	...	...	...
4613	6/17/2022	2.0	10000.0	1200	1 out of 2	Carpet Area	Jaya Puri Colony, Nagole	Hyderabad	Unfurnished	Family	
4617	7/7/2022	2.0	10000.0	1100	5 out of 5	Super Area	Kapra	Hyderabad	Furnished	Family	
4643	5/18/2022	2.0	10000.0	110	1 out of 1	Carpet Area	Sangareddy District	Hyderabad	Unfurnished	Bachelors	
4658	5/25/2022	2.0	10000.0	1300	2 out of 3	Super Area	Moula Ali	Hyderabad	Unfurnished	Bachelors/Family	
4688	7/2/2022	2.0	10000.0	1125	2 out of 3	Carpet Area	Allwyn Colony	Hyderabad	Unfurnished	Bachelors	

248 rows × 12 columns



```
In [82]: 1 # SADECE rentin 10000' eşit olanlarını görmek istiyoruz, loc indeksleme ile de yapabiliriz
2 indian_rent.loc[indian_rent['kira']==10000]
```

Out[82]:

	ilan_tarihi	oda_sayısı	kira	boyut	bulunduğu_kat	zemin_tipi	lokasyon_türü	şehir	mobilyalı_mobilyasız	kiracı_tercihi	banyo_s
0	5/18/2022	2.0	10000.0	1100	Ground out of 2	Super Area	Bandel	Kolkata	Unfurnished	Bachelors/Family	
3	7/4/2022	2.0	10000.0	800	1 out of 2	Super Area	Dumdum Park	Kolkata	Unfurnished	Bachelors/Family	
6	6/21/2022	2.0	10000.0	700	Ground out of 4	Super Area	Malancha	Kolkata	Unfurnished	Bachelors	
9	6/20/2022	2.0	10000.0	1000	1 out of 3	Carpet Area	Natunhat	Kolkata	Semi-Furnished	Bachelors/Family	
17	6/20/2022	2.0	10000.0	800	Ground out of 2	Super Area	Behala	Kolkata	Unfurnished	Bachelors/Family	
...	...	...	...	...	...	...	...	...	...	...	...
4613	6/17/2022	2.0	10000.0	1200	1 out of 2	Carpet Area	Jaya Puri Colony, Nagole	Hyderabad	Unfurnished	Family	
4617	7/7/2022	2.0	10000.0	1100	5 out of 5	Super Area	Kapra	Hyderabad	Furnished	Family	
4643	5/18/2022	2.0	10000.0	110	1 out of 1	Carpet Area	Sangareddy District	Hyderabad	Unfurnished	Bachelors	
4658	5/25/2022	2.0	10000.0	1300	2 out of 3	Super Area	Moula Ali	Hyderabad	Unfurnished	Bachelors/Family	
4688	7/2/2022	2.0	10000.0	1125	2 out of 3	Carpet Area	Allwyn Colony	Hyderabad	Unfurnished	Bachelors	

248 rows × 12 columns



## ÇOKLU MANTIKSAL SEÇİMLERDE PARENTEZ KULLANMAK ZORUNDAYIZ

In [83]:

```
1 # rentin 10000'eşit olduğu ve bhk'nın 3 eşit olduğu veriyi görmek istiyoruz
2 indian_rent[(indian_rent['kira']==10000) & (indian_rent['oda_sayısı']==3)]
```

Out[83]:

	ilan_tarihi	oda_sayısı	kira	boyut	bulunduğu_kat	zemin_tipi	lokasyon_türü	şehir	mobilyalı_mobilyasız	kiracı_tercihi	bany
247	5/31/2022	3.0	10000.0	1200	2 out of 4	Super Area	Belghoria	Kolkata	Furnished	Bachelors/Family	
253	7/1/2022	3.0	10000.0	1250	1 out of 2	Super Area	Jaffarpore Rifle Range Road	Kolkata	Unfurnished	Bachelors/Family	
291	6/26/2022	3.0	10000.0	1190	1 out of 2	Carpet Area	Sarsuna	Kolkata	Unfurnished	Bachelors/Family	
295	5/14/2022	3.0	10000.0	800	Ground out of 4	Super Area	Brahmapur	Kolkata	Unfurnished	Bachelors/Family	
297	6/11/2022	3.0	10000.0	900	Ground out of 3	Carpet Area	Garia	Kolkata	Semi-Furnished	Bachelors/Family	
446	4/26/2022	3.0	10000.0	1000	2 out of 5	Carpet Area	Bansdroni	Kolkata	Unfurnished	Bachelors	
483	6/10/2022	3.0	10000.0	1500	11 out of 19	Super Area	Santoshpur	Kolkata	Semi-Furnished	Bachelors/Family	



In [84]:

```
1 # rentin 10000'eşit olduğu ve bhk'nın 3 eşit ve büyük olduğu veriyi görmek istiyoruz. loc indekslleme ile de yapal  
2 indian_rent.loc[(indian_rent['kira']==10000) & (indian_rent['oda_sayısı']>=3)]
```

Out[84]:

	ilan_tarihi	oda_sayısı	kira	boyut	bulunduğu_kat	zemin_tipi	lokasyon_türü	şehir	mobilyalı_mobilyasız	kiracı_tercihi	banyo_s
192	4/29/2022	4.0	10000.0	1110	2 out of 2	Carpet Area	Purbasa	Kolkata	Unfurnished	Bachelors/Family	
247	5/31/2022	3.0	10000.0	1200	2 out of 4	Super Area	Belghoria	Kolkata	Furnished	Bachelors/Family	
253	7/1/2022	3.0	10000.0	1250	1 out of 2	Super Area	Jaffarpore Rifle Range Road	Kolkata	Unfurnished	Bachelors/Family	
291	6/26/2022	3.0	10000.0	1190	1 out of 2	Carpet Area	Sarsuna	Kolkata	Unfurnished	Bachelors/Family	
295	5/14/2022	3.0	10000.0	800	Ground out of 4	Super Area	Brahmapur	Kolkata	Unfurnished	Bachelors/Family	
297	6/11/2022	3.0	10000.0	900	Ground out of 3	Carpet Area	Garia	Kolkata	Semi-Furnished	Bachelors/Family	
446	4/26/2022	3.0	10000.0	1000	2 out of 5	Carpet Area	Bansdroni	Kolkata	Unfurnished	Bachelors	
483	6/10/2022	3.0	10000.0	1500	11 out of 19	Super Area	Santoshpur	Kolkata	Semi-Furnished	Bachelors/Family	
485	5/17/2022	3.0	10000.0	800	Ground out of 2	Carpet Area	Arobindo Pally, Paschim Putiary	Kolkata	Unfurnished	Bachelors	
487	6/24/2022	4.0	10000.0	1200	5 out of 10	Super Area	Naihati	Kolkata	Furnished	Bachelors/Family	
2414	7/9/2022	3.0	10000.0	600	3 out of 4	Carpet Area	Laxmi Nagar	Delhi	Semi-Furnished	Bachelors/Family	
2509	5/5/2022	3.0	10000.0	75	4 out of 5	Super Area	Sagar Pur	Delhi	Unfurnished	Bachelors/Family	
2557	6/11/2022	3.0	10000.0	100	1 out of 4	Carpet Area	Krishna Nagar	Delhi	Furnished	Bachelors/Family	
3478	6/12/2022	3.0	10000.0	80	3 out of 3	Carpet Area	Pudupakkam	Chennai	Semi-Furnished	Bachelors/Family	
3479	5/9/2022	3.0	10000.0	1000	1 out of 8	Carpet Area	Kelambakkam, Old Mahabalipuram Road	Chennai	Semi-Furnished	Bachelors/Family	
3744	6/4/2022	3.0	10000.0	1200	1 out of 3	Carpet Area	Kovur	Chennai	Unfurnished	Bachelors/Family	

	ilan_tarihi	oda_sayısı	kira	boyut	bulunduğu_kat	zemin_tipi	lokasyon_türü	şehir	mobilyalı_mobilyasız	kiracı_tercihi	banyo_s
3748	6/12/2022	3.0	10000.0	1750	1 out of 1	Super Area	Kuthanur	Chennai	Unfurnished	Bachelors/Family	
4345	5/7/2022	3.0	10000.0	1000	2 out of 2	Carpet Area	Manikonda, Outer Ring Road	Hyderabad	Furnished	Family	
4444	7/6/2022	3.0	10000.0	1100	1 out of 4	Carpet Area	Manikonda, Outer Ring Road	Hyderabad	Semi-Furnished	Bachelors/Family	
4553	5/12/2022	4.0	10000.0	100	1 out of 3	Super Area	Old Nallakunta	Hyderabad	Unfurnished	Bachelors/Family	

## max() min() mean() kullanımları

```
In [85]: 1 #en yüksek kiraya sahip ev
          2 indian_rent[indian_rent['kira']==indian_rent['kira'].max()]
```

Out[85]:

	ilan_tarihi	oda_sayısı	kira	boyut	bulunduğu_kat	zemin_tipi	lokasyon_türü	şehir	mobilyalı_mobilyasız	kiracı_tercihi	banyo_sayısı
1840	6/8/2022	3.0	3500000.0	2500	4 out of 4	Carpet Area	Marathahalli	Bangalore	Semi-Furnished	Bachelors	

```
In [86]: 1 #en yüksek kiraya sahip ev, loc indeksi ile
          2 indian_rent.loc[indian_rent['kira']==indian_rent['kira'].max()]
```

Out[86]:

	ilan_tarihi	oda_sayısı	kira	boyut	bulunduğu_kat	zemin_tipi	lokasyon_türü	şehir	mobilyalı_mobilyasız	kiracı_tercihi	banyo_sayısı
1840	6/8/2022	3.0	3500000.0	2500	4 out of 4	Carpet Area	Marathahalli	Bangalore	Semi-Furnished	Bachelors	

In [ ]:

1

In [87]:

```
1 # en büyük ev
2 indian_rent[indian_rent['boyut']==indian_rent['boyut'].max()]
```

Out[87]:

	ilan_tarihi	oda_sayısı	kira	boyut	bulunduğu_kat	zemin_tipi	lokasyon_türü	şehir	mobilyalı_mobilyasız	kiracı_tercihi	banyo_
4188	6/6/2022	1.0	200000.0	8000	Ground out of 4	Super Area	Beeramguda, Ramachandra Puram, NH 9	Hyderabad	Unfurnished	Bachelors/Family	

In [88]:

```
1 # en büyük ev, loc indeksi ile
2 indian_rent.loc[indian_rent['boyut']==indian_rent['boyut'].max()]
```

Out[88]:

	ilan_tarihi	oda_sayısı	kira	boyut	bulunduğu_kat	zemin_tipi	lokasyon_türü	şehir	mobilyalı_mobilyasız	kiracı_tercihi	banyo_
4188	6/6/2022	1.0	200000.0	8000	Ground out of 4	Super Area	Beeramguda, Ramachandra Puram, NH 9	Hyderabad	Unfurnished	Bachelors/Family	

In [ ]:

1

In [89]:

```
1 # Hyderabad şehrindeki en pahalı ev
2 indian_rent[indian_rent['şehir']=='Hyderabad'][indian_rent[indian_rent['şehir']=='Hyderabad']['kira']==indian_rent
```

Out[89]:

	ilan_tarihi	oda_sayısı	kira	boyut	bulunduğu_kat	zemin_tipi	lokasyon_türü	şehir	mobilyalı_mobilyasız	kiracı_tercihi	banyo_
4460	7/6/2022	4.0	400000.0	7000	Lower Basement out of 2	Carpet Area	Jubilee Hills	Hyderabad	Semi-Furnished	Bachelors/Family	

```
In [90]: 1 # kolay ve hata yapma şansını azaltan yöntem
2 aa = indian_rent[indian_rent['şehir']=='Hyderabad']
3 aa[aa['kira']==aa['kira'].max()]
```

```
Out[90]:
```

	ilan_tarihi	oda_sayısı	kira	boyut	bulunduğu_kat	zemin_tipi	lokasyon_türü	şehir	mobilyalı_mobilyasız	kiracı_tercihi	banyo_
4460	7/6/2022	4.0	400000.0	7000	Lower Basement out of 2	Carpet Area	Jubilee Hills	Hyderabad	Semi-Furnished	Bachelors/Family	

```
In [91]: 1 # bu yöntem Hyderabad şehrindeki sadece en yüksek kirayı gösterir
2 indian_rent[indian_rent['şehir']=='Hyderabad']['kira'].max()
3
```

```
Out[91]: 400000.0
```

```
In [92]: 1 # bu yöntem Hyderabad şehrindeki sadece en ortalama kirayı gösterir
2 indian_rent[indian_rent['şehir']=='Hyderabad']['kira'].mean()
3
```

```
Out[92]: 20555.048387096773
```

```
In [93]: 1 # bu yöntem Hyderabad şehrindeki sadece en düşük kirayı gösterir
2 indian_rent[indian_rent['şehir']=='Hyderabad']['kira'].min()
```

```
Out[93]: 1200.0
```

```
In [94]: 1 # bu yöntem Hyderabad şehrindeki ortalama kirayı, ev boyutlarını ve banyo sayısını gösterir
2 indian_rent[indian_rent['şehir']=='Hyderabad']['[kira,boyut,banyo_sayısı]'].mean()
```

```
Out[94]: kira          20555.048387
boyut          1186.669355
banyo_sayısı    2.149770
dtype: float64
```

In [95]: 1 # aşağıda, ortalamaları istedik ama zemin tipi ortalamasını vermedi, neden?

In [96]: 1  
2 indian\_rent[indian\_rent['şehir']=='Hyderabad'][['kira', 'boyut', 'banyo\_sayısı', 'zemin\_tipi']].mean()*# It doesn't g*

Out[96]: kira 20555.048387  
boyut 1186.669355  
banyo\_sayısı 2.149770  
dtype: float64

**groupby() metodunu kullanarak bir veya bir kaç kritere göre ortalama, minimum ve maksimumları gösterebiliriz**

In [97]: 1 indian\_rent.groupby('şehir').mean()

Out[97]:

	oda_sayısı	kira	boyut	banyo_sayısı
şehir				
Bangalore	1.930023	24966.365688	985.925508	1.793454
Chennai	2.124579	21614.092031	1031.713805	1.998878
Delhi	2.114050	29461.983471	786.406612	1.852893
Hyderabad	2.206221	20555.048387	1186.669355	2.149770
Kolkata	1.965649	11645.173664	787.366412	1.431298
Mumbai	2.122302	85235.057554	904.352156	2.287037

```
In [98]: 1 indian_rent.groupby(['şehir', 'mobilyalı_mobilyasız']).mean()
```

```
Out[98]:
```

		oda_sayısı	kira	boyut	banyo_sayısı
şehir	mobilyalı_mobilyasız				
Bangalore	Furnished	2.197802	32531.868132	1199.659341	2.076923
	Semi-Furnished	1.969178	26879.452055	1013.522260	1.820205
	Unfurnished	1.706161	16408.530806	817.364929	1.597156
Chennai	Furnished	2.320988	32961.666667	1233.839506	2.209877
	Semi-Furnished	2.274554	23900.582589	1128.466518	2.145089
	Unfurnished	1.895028	16245.303867	866.748619	1.770718
Delhi	Furnished	1.814433	30214.443299	690.329897	1.649485
	Semi-Furnished	2.262590	33945.320144	936.395683	2.043165
	Unfurnished	2.060870	23725.652174	645.634783	1.708696
Hyderabad	Furnished	2.315315	24353.153153	1296.099099	2.270270
	Semi-Furnished	2.422886	25002.442786	1385.634328	2.338308
	Unfurnished	1.926761	14331.267606	927.146479	1.898592
Kolkata	Furnished	1.819672	13865.573770	793.213115	1.475410
	Semi-Furnished	1.992958	11722.535211	824.929577	1.500000
	Unfurnished	1.981308	11189.006231	769.638629	1.392523
Mumbai	Furnished	2.271967	108974.527197	1020.493724	2.414226
	Semi-Furnished	2.264484	99744.586902	990.342569	2.471033
	Unfurnished	1.848214	51454.458333	724.627976	1.979167

## MATEMATİKSEL İŞLEMLER

```
In [99]: 1 aa = indian_rent[indian_rent['şehir']=='Delhi']
```

```
In [100]: 1 #square_foot_rent_of_most_expensive_home_in_delhi
2 aa[aa['kira']==aa['kira'].max()][ 'boyut']/aa[aa['kira']==aa['kira'].max()][ 'kira']
```

```
Out[100]: 2753      0.007547
dtype: float64
```

```
In [101]: 1 indian_rent['metrekare_fiyatı'] =indian_rent['boyut']/indian_rent['kira']
```

## KAYIP (MISSING) BİLGİLERİN KONTROLÜ

```
In [102]: 1 indian_rent.isna().sum()
```

```
Out[102]: ilan_tarihi      0
oda_sayısı      1
kira            1
boyut           0
bulunduğu_kat   2
zemin_tipi      2
lokasyon_türü   3
şehir           1
mobilyalı_mobilyasız  3
kiracı_tercihi  3
banyo_sayısı    3
erişim_yolu     3
metrekare_fiyatı  1
dtype: int64
```

## EĞER MISSING BİLGİ VARSA VE ZAMAN SERİSİ ÇALIŞMASI YPMİYORSAK İKİ SEŞENEĞİMİZ VAR:

- 1) BU KISIM SATIRI SİLERİZ
- 2) INTERPOLATION (KAYIP BİLGİYİ TAHMİN EDERİZ)



# EĞER ZAMAN SERİSİ ÇALIŞMASI YAPIYORSAK INTERPOLATION YAPMAK DIŞINDA BİR ŞANSIMIZ YOK

```
1 # #pd.interpolate(method='linear',axis=0,limit=int,inplace=boolyea)
2 #method=['linear','nearest', 'zero', 'slinear', 'quadratic', 'cubic', 'spline', 'barycentric', 'polynomial']
3 # limit: int, optional Maximum number of consecutive NaNs to fill. Must be greater than 0.
```

```
In [104]: 1 indian_rent['oda_sayısı'].interpolate(method='linear',axis=0,limit=2,inplace=True)
```

```
In [105]: 1 indian_rent.isna().sum()
```

```
Out[105]: ilan_tarihi          0
oda_sayısı          0
kira                1
boyut              0
bulunduğu_kat       2
zemin_tipi         2
lokasyon_türü       3
şehir              1
mobilyalı_mobilyasız 3
kiracı_tercihi       3
banyo_sayısı        3
erişim_yolu         3
metrekare_fiyatı     1
dtype: int64
```

```
In [106]: 1 # eğer missing datayı pd.fillna() ile doldurmak istersek
2 #pd.fillna(value=None, method=None, axis=None, inplace=False, limit=None, downcast=None) Parameters valuescalar, c
```

pd.fillna(value=None, method=None, axis=None, inplace=False, limit=None, downcast=None) Parameters valuescalar, dict, Series, or DataFrame Value to use to fill holes (e.g. 0), alternately a dict/Series/DataFrame of values specifying which value to use for each index (for a Series) or column (for a DataFrame). Values not in the dict/Series/DataFrame will not be filled. This value cannot be a list.

method{'backfill', 'bfill', 'pad', 'ffill', None}, default None Method to use for filling holes in reindexed Series pad / ffill: propagate last valid observation forward to next valid backfill / bfill: use next valid observation to fill gap.

axis{0 or 'index', 1 or 'columns'} Axis along which to fill missing values.

inplacebool, default False If True, fill in-place. Note: this will modify any other views on this object (e.g., a no-copy slice for a column in a DataFrame).

limitint, default None If method is specified, this is the maximum number of consecutive NaN values to forward/backward fill. In other words, if there is a gap with more than this number of consecutive NaNs, it will only be partially filled. If method is not specified, this is the maximum number of entries along the entire axis where NaNs will be filled. Must be greater than 0 if not None.

downcastdict, default is None A dict of item->dtype of what to downcast if possible, or the string 'infer' which will try to downcast to an appropriate equal type (e.g. float64 to int64 if possible).

```
In [107]: 1 # kayıp datayı ortalama ile doldurmak
          2 ortalama_kira = indian_rent['kira'].mean()
          3 indian_rent['kira'].fillna(value=ortalama_kira,axis=0,inplace=True)
```

```
In [108]: 1 indian_rent.isna().sum()
```

```
Out[108]: ilan_tarihi          0
          oda_sayısı          0
          kira                0
          boyut               0
          bulunduğu_kat       2
          zemin_tipi          2
          lokasyon_türü       3
          şehir              1
          mobilyalı_mobilyasız 3
          kiracı_tercihi       3
          banyo_sayısı        3
          erişim_yolu         3
          metrekare_fiyatı     1
          dtype: int64
```

```
In [109]: 1 # arkadan gelen değerle doldurmak
          2 indian_rent['zemin_tipi'].fillna(method='backfill',axis=0,inplace=True)
```

```
In [110]: 1 indian_rent.isna().sum()
```

```
Out[110]: ilan_tarihi          0
          oda_sayısı          0
          kira                0
          boyut              0
          bulunduğu_kat      2
          zemin_tipi         0
          lokasyon_türü      3
          şehir              1
          mobilyalı_mobilyasız 3
          kiracı_tercihi      3
          banyo_sayısı        3
          erişim_yolu         3
          metrekare_fiyatı    1
          dtype: int64
```

```
In [111]: 1 #kayıp dayatı silmek
          2 indian_rent.dropna(inplace=True)
```

```
In [112]: 1 indian_rent.isna().sum()
```

```
Out[112]: ilan_tarihi          0
          oda_sayısı          0
          kira                0
          boyut              0
          bulunduğu_kat      0
          zemin_tipi         0
          lokasyon_türü      0
          şehir              0
          mobilyalı_mobilyasız 0
          kiracı_tercihi      0
          banyo_sayısı        0
          erişim_yolu         0
          metrekare_fiyatı    0
          dtype: int64
```

## tekrarlanan datayı silmek

```
In [113]: 1 #DataFrame.drop_duplicates(subset=None, keep='first', inplace=False, ignore_index=False)
```

```
In [114]: 1 df_06 = pd.DataFrame({  
2     'brand': ['Yum Yum', 'Yum Yum', 'Indomie', 'Indomie', 'Indomie'],  
3     'style': ['cup', 'cup', 'cup', 'pack', 'pack'],  
4     'rating': [4, 4, 3.5, 15, 5]})
```

```
In [115]: 1 df_06
```

```
Out[115]:
```

	brand	style	rating
0	Yum Yum	cup	4.0
1	Yum Yum	cup	4.0
2	Indomie	cup	3.5
3	Indomie	pack	15.0
4	Indomie	pack	5.0

```
In [116]: 1 df_06.drop_duplicates()
```

```
Out[116]:
```

	brand	style	rating
0	Yum Yum	cup	4.0
2	Indomie	cup	3.5
3	Indomie	pack	15.0
4	Indomie	pack	5.0

## çoklu index yapmak stack

```
In [117]: 1 display(Image(filename='reshaping_stack.png'))
```

# Stack

df2

		A	B
first	second		
bar	one	1	2
	two	3	4



stacked = df2.stack()

first	second		
bar	one	A	1
		B	2
	two	A	3
		B	4
bar	one	A	5

```
In [118]: 1 import yfinance as yf
2 start = '2010-12-15'
3 end = '2020-12-15'
4 ticker = ['GOOGL', 'AAPL', 'KO']
5 interval = "1d"
6 stocks =yf.download(ticker,start,end,interval)
```

[\*\*\*\*\*100%\*\*\*\*\*] 3 of 3 completed

```
In [119]: 1 stocks = stocks['Close']
          2 stocks
```

```
Out[119]:
```

	AAPL	GOOGL	KO
Date			
2010-12-15	11.441429	14.772272	32.369999
2010-12-16	11.473214	14.807558	32.669998
2010-12-17	11.450357	14.784785	32.849998
2010-12-20	11.507500	14.891391	32.654999
2010-12-21	11.578571	15.091842	32.744999
...	...	...	...
2020-12-08	124.379997	90.566498	53.180000
2020-12-09	121.779999	88.892998	53.330002
2020-12-10	123.239998	88.382500	53.049999
2020-12-11	122.410004	88.739998	53.349998
2020-12-14	121.779999	87.612999	53.270000

2517 rows × 3 columns

```
In [120]: 1 stocks.index
```

```
Out[120]: DatetimeIndex(['2010-12-15', '2010-12-16', '2010-12-17', '2010-12-20',
                          '2010-12-21', '2010-12-22', '2010-12-23', '2010-12-27',
                          '2010-12-28', '2010-12-29',
                          ...,
                          '2020-12-01', '2020-12-02', '2020-12-03', '2020-12-04',
                          '2020-12-07', '2020-12-08', '2020-12-09', '2020-12-10',
                          '2020-12-11', '2020-12-14'],
                          dtype='datetime64[ns]', name='Date', length=2517, freq=None)
```

```
In [121]: 1 new_stocks = stocks.stack()  
          2 new_stocks
```

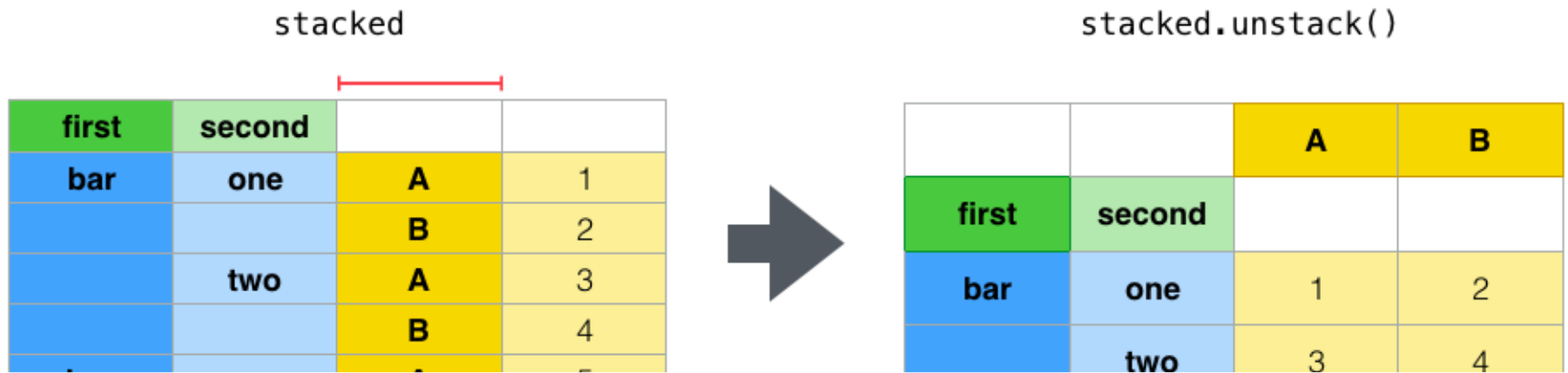
```
Out[121]: Date  
2010-12-15 AAPL      11.441429  
           GOOGL     14.772272  
           KO        32.369999  
2010-12-16 AAPL      11.473214  
           GOOGL     14.807558  
           ...  
2020-12-11 GOOGL     88.739998  
           KO        53.349998  
2020-12-14 AAPL     121.779999  
           GOOGL     87.612999  
           KO        53.270000  
Length: 7551, dtype: float64
```





```
In [123]: 1 display(Image(filename='reshaping_unstack.png'))
```

# Unstack



```
In [124]: 1 new_stocks.unstack()
```

```
Out[124]:
```

	AAPL	GOOGL	KO
Date			
2010-12-15	11.441429	14.772272	32.369999
2010-12-16	11.473214	14.807558	32.669998
2010-12-17	11.450357	14.784785	32.849998
2010-12-20	11.507500	14.891391	32.654999
2010-12-21	11.578571	15.091842	32.744999
...	...	...	...
2020-12-08	124.379997	90.566498	53.180000
2020-12-09	121.779999	88.892998	53.330002
2020-12-10	123.239998	88.382500	53.049999
2020-12-11	122.410004	88.739998	53.349998
2020-12-14	121.779999	87.612999	53.270000

2517 rows × 3 columns

## pivot table

```
In [125]: 1 display(Image(filename='reshaping_pivot.png'))
```

# Pivot

df

	foo	bar	baz	zoo
0	one	A	1	x
1	one	B	2	y
2	one	C	3	z
3	two	A	4	q
4	two	B	5	w
5	two	C	6	t



```
df.pivot(index='foo',  
          columns='bar',  
          values='baz')
```

bar	A	B	C
foo			
one	1	2	3
two	4	5	6

```
In [126]: 1 stocks = pd.read_csv('https://gist.githubusercontent.com/alexdebrie/b3f40efc3dd7664df5a20f5eee85e854/raw/ee3e6fec
2 stocks
```

Out[126]:

	date	symbol	open	high	low	close	volume
0	2019-03-01	AMZN	1655.13	1674.26	1651.00	1671.73	4974877
1	2019-03-04	AMZN	1685.00	1709.43	1674.36	1696.17	6167358
2	2019-03-05	AMZN	1702.95	1707.80	1689.01	1692.43	3681522
3	2019-03-06	AMZN	1695.97	1697.75	1668.28	1668.95	3996001
4	2019-03-07	AMZN	1667.37	1669.75	1620.51	1625.95	4957017
5	2019-03-01	AAPL	174.28	175.15	172.89	174.97	25886167
6	2019-03-04	AAPL	175.69	177.75	173.97	175.85	27436203
7	2019-03-05	AAPL	175.94	176.00	174.54	175.53	19737419
8	2019-03-06	AAPL	174.67	175.49	173.94	174.52	20810384
9	2019-03-07	AAPL	173.87	174.44	172.02	172.50	24796374
10	2019-03-01	GOOG	1124.90	1142.97	1124.75	1140.99	1450316
11	2019-03-04	GOOG	1146.99	1158.28	1130.69	1147.80	1446047
12	2019-03-05	GOOG	1150.06	1169.61	1146.19	1162.03	1443174
13	2019-03-06	GOOG	1162.49	1167.57	1155.49	1157.86	1099289
14	2019-03-07	GOOG	1155.72	1156.76	1134.91	1143.30	1166559

```
In [127]: 1 stocks.pivot(index='symbol', columns='date', values='volume')
```

Out[127]:

	date	2019-03-01	2019-03-04	2019-03-05	2019-03-06	2019-03-07
symbol						
AAPL		25886167	27436203	19737419	20810384	24796374
AMZN		4974877	6167358	3681522	3996001	4957017
GOOG		1450316	1446047	1443174	1099289	1166559

```
In [128]: 1 stocks.pivot(index='symbol', columns='date', values=['volume','close'])
```

Out[128]:

	volume					close				
date	2019-03-01	2019-03-04	2019-03-05	2019-03-06	2019-03-07	2019-03-01	2019-03-04	2019-03-05	2019-03-06	2019-03-07
symbol										
AAPL	25886167.0	27436203.0	19737419.0	20810384.0	24796374.0	174.97	175.85	175.53	174.52	172.50
AMZN	4974877.0	6167358.0	3681522.0	3996001.0	4957017.0	1671.73	1696.17	1692.43	1668.95	1625.95
GOOG	1450316.0	1446047.0	1443174.0	1099289.0	1166559.0	1140.99	1147.80	1162.03	1157.86	1143.30

```
In [ ]:
```

```
1
```