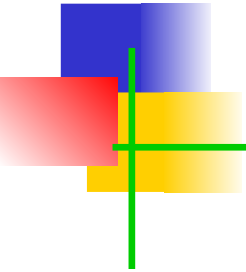


BLM442 Büyük Veri Analizine Giriş

pandas



Dr. Süleyman Eken

Bilgisayar Mühendisliği
Kocaeli Üniversitesi

Sunum Planı

- pandas veri yapıları: Series, Data frames
- Data frames attributes and functions
- İndeksleme, slicing, groupby
- Filtering, loc, iloc fonksiyonları
- Sorting
- Kayıp veriler
- Özetleme fonksiyonları

Veri bilimi için Python kütüphaneleri



Veri bilimi için Python kütüphaneleri

● numpy
Arama terimi

● scipy
Arama terimi

● pandas
Arama terimi

+ Karşılaştırma ekleyin

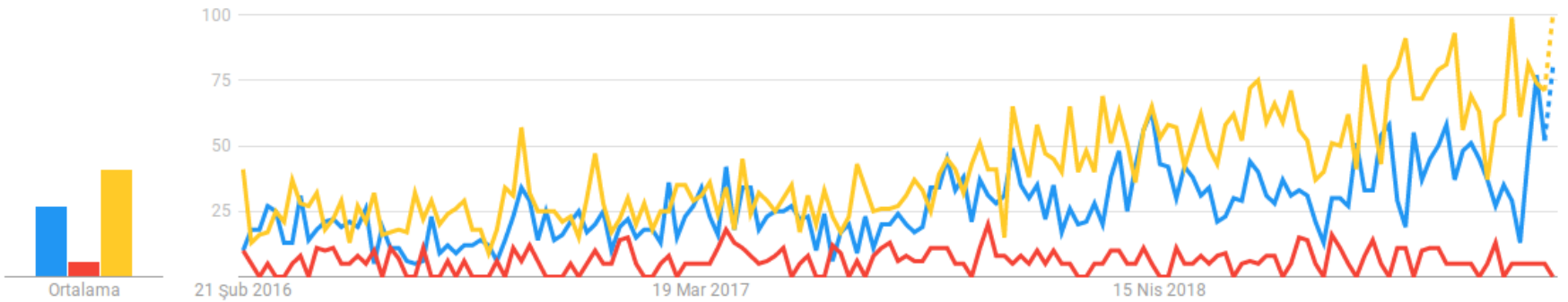
Türkiye ▾

21.02.2016 - 21.03.2019 ▾

Tüm kategoriler ▾

Google Web Arama ▾

Zaman içinde gösterilen ilgi ?



Veri bilimi için Python kütüphaneleri



<https://www.datascience.com/trends>


Python kütüphanelerini yükleme

```
#Import Python Libraries  
import numpy as np  
import scipy as sp  
import pandas as pd  
import matplotlib as mpl  
import seaborn as sns
```

pandas veri yapıları

- **Series**: 1D labelled single-type arrays
- **DataFrames**: 2D labelled multi-type arrays

	NAME	AGE	DESIGNATION
1	a	20	VP
2	b	27	CEO
3	c	35	CFO
4	d	55	VP
5	e	18	VP
6	f	21	CEO
7	g	35	MD



Series oluşturma

```
data = [1,2,3,numpy.nan,5,6]      # nan == Not a Number
unindexed = pandas.Series(data)
```

```
indices = ['a', 'b', 'c', 'd', 'e']
indexed = pandas.Series(data, index=indices)
```

```
data_dict = {'a' : 1, 'b' : 2, 'c' : 3}
indexed = pandas.Series(data_dict)
```

```
fives = pandas.Series(5, indices)      # Fill with 5s.
```

```
named = pandas.Series(data, name='mydata')
named.rename('my_data')
print(named.name)
```


pandas kullanarak veri okuma

```
#Read csv file
```

```
df = pd.read_csv("/home/ipcvlab/Downloads/Salaries.csv")
```

```
pd.read_excel('myfile.xlsx', sheet_name='Sheet1',  
index_col=None, na_values=['NA'])
```

```
pd.read_stata('myfile.dta')
```

```
pd.read_sas('myfile.sas7bdat')
```

```
pd.read_hdf('myfile.h5', 'df')
```

Veri çerçevelerini (data frames) keşfetme

```
In [3] #List first 5 records  
df.head()
```

Out[3]:

	rank	discipline	phd	service	sex	salary
0	Prof	B	56	49	Male	186960
1	Prof	A	12	6	Male	93000
2	Prof	A	23	20	Male	110515
3	Prof	A	40	31	Male	131205
4	Prof	B	20	18	Male	104800

- ilk n satırı nasıl alırız, son kısmı gösterme?

pandas veri tipleri

Pandas dtype	Python type	NumPy type	Usage
object	str	string_, unicode_	Text
int64	int	int_, int8, int16, int32, int64, uint8, uint16, uint32, uint64	Integer numbers
float64	float	float_, float16, float32, float64	Floating point numbers
bool	bool	bool_	True/False values
datetime64	NA	NA	Date and time values
timedelta[ns]	NA	NA	Differences between two datetimes
category	NA	NA	Finite list of text values

```
■ >>> df = pd.DataFrame({'float': [1.0],  
...                       'int': [1],  
...                       'datetime': [pd.Timestamp('20180310')],  
...                       'string': ['foo']})  
>>> df.dtypes  
float      float64  
int        int64  
datetime   datetime64[ns]  
string     object  
dtype: object
```

pandas veri tipleri

```
In [4] #Check a particular column type  
df['salary'].dtype
```

```
Out[4]: dtype('int64')
```

```
In [5] #Check types for all the columns  
df.dtypes
```

```
Out[4]: rank          object  
        discipline    object  
        phd           int64  
        service       int64  
        sex           object  
        salary        int64  
        dtype: object
```

Data frames attributes

T	Transpose index and columns.
at	Access a single value for a row/column label pair.
axes	Return a list representing the axes of the DataFrame.
blocks	(DEPRECATED) Internal property, property synonym for <code>as_blocks()</code> .
columns	The column labels of the DataFrame.
dtypes	Return the dtypes in the DataFrame.
empty	Indicator whether DataFrame is empty.
ftypes	Return the ftypes (indication of sparse/dense and dtype) in DataFrame.
iat	Access a single value for a row/column pair by integer position.
iloc	Purely integer-location based indexing for selection by position.
index	The index (row labels) of the DataFrame.
is_copy	Return the copy.
ix	A primarily label-location based indexer, with integer position fallback.
loc	Access a group of rows and columns by label(s) or a boolean array.
ndim	Return an int representing the number of axes / array dimensions.
shape	Return a tuple representing the dimensionality of the DataFrame.
size	Return an int representing the number of elements in this object.
style	Property returning a Styler object containing methods for building a styled HTML representation fo the DataFrame.
values	Return a Numpy representation of the DataFrame.

Data frames attributes

- Bu veri çerçevesinin kaç tane kayıt var?
- Kaç element var?
- Sütun adları nedir?
- Bu veri çerçevesinde ne tür sütunlara sahibiz?
- ..

Data frames metotlari

df.method()	description
..	..
head([n]), tail([n])	first/last n rows
describe()	generate descriptive statistics (for numeric columns only)
max(), min()	return max/min values for all numeric columns
mean(), median()	return mean/median values for all numeric columns
std()	standard deviation
sample([n])	returns a random sample of the data frame
dropna()	drop all the records with missing values
..	..

Data frames metotları

- Veri kümesindeki sayısal sütunların özetini verin.
- Tüm sayısal sütunlar için standart sapmayı hesaplayın.
- Veri kümesindeki ilk 50 kaydın ortalama değerleri nelerdir?
- ..

Veri çerçevesinde bir sütun seçme

Yöntem 1: Sütun adını index olarak kullanarak
`df['sex']`

Yöntem 2: Sütun adını attribute olarak kullanarak
`df.sex`

- Maaş (salary) sütunu için temel istatistikleri hesaplayın.
- Maaş sütununda kaç değer olduğunu bulun (count yöntemini kullanın).
- Ortalama maaş hesaplayın;
- ..

Data Frames groupby metodu

- Bazı kriterlere göre verileri gruplara ayırma
- Her gruba ait istatistikleri hesapla (veya bir fonksiyonu uygula)

```
#Group data using rank  
df_rank = df.groupby(['rank'])
```

```
#Calculate mean value for each numeric column per each group  
df_rank.mean()
```

	phd	service	salary
rank			
AssocProf	15.076923	11.307692	91786.230769
AsstProf	5.052632	2.210526	81362.789474
Prof	27.065217	21.413043	123624.804348

Data Frame: filtering

- Verilerin altkümesini elde etmek için Boolean indekslemeyi uygulayabiliriz. Bu indeksleme genellikle filtre olarak bilinir.

> greater; >= greater or equal;

< less; <= less or equal;

== equal; != not equal;

#Calculate mean salary for each professor rank:

```
df_sub = df[ df['salary'] > 120000 ]
```

#Select only those rows that contain female professors:

```
df_f = df[ df['sex'] == 'Female' ]
```

Data Frames: Slicing

- Bir (sonucu Series) veya daha fazla kolon

```
#Select column salary:  
df['salary']
```

```
#Select column salary:  
df[['rank', 'salary']]
```

- Bir veya daha fazla sütun

```
#Select rows by their position:  
df[10:20]
```

- Kolon veya sütunların bir alt parçası
 - loc, iloc

Data Frames: loc metodu

- Kolon etiketleri kullanarak bir satır aralığı seçilecekse

```
#Select rows by their labels:  
df_sub.loc[10:20, ['rank', 'sex', 'salary']]
```

	rank	sex	salary
10	Prof	Male	128250
11	Prof	Male	134778
13	Prof	Male	162200
14	Prof	Male	153750
15	Prof	Male	150480
19	Prof	Male	150500

Data Frames: iloc metodu

- Kolon veya satır aralığı pozisyonları (integer based indexing) kullanılarak seçilecekse

```
#Select rows by their labels:  
df_sub.iloc[10:20,[0, 3, 4, 5]]
```

	rank	service	sex	salary
26	Prof	19	Male	148750
27	Prof	43	Male	155865
29	Prof	20	Male	123683
31	Prof	21	Male	155750
35	Prof	23	Male	126933
36	Prof	45	Male	146856
39	Prof	18	Female	129000
40	Prof	36	Female	137000
44	Prof	19	Female	151768
45	Prof	25	Female	140096

Data Frames: iloc metodu

```
df.iloc[0]    # First row of a data frame  
df.iloc[i]    #(i+1)th row  
df.iloc[-1]   # Last row
```

```
df.iloc[:, 0] # First column  
df.iloc[:, -1] # Last column
```

```
df.iloc[0:7]          #First 7 rows  
df.iloc[:, 0:2]       #First 2 columns  
df.iloc[1:3, 0:2]     #Second through third rows and first 2 columns  
df.iloc[[0,5], [1,3]] #1st and 6th rows and 2nd and 4th columns
```

Data Frames: Sorting

- Verileri sütundaki bir değere göre sıralayabiliriz. Varsayılan olarak sıralama artan düzende gerçekleşir.

```
# Create a new data frame from the original sorted by the column Salary  
df_sorted = df.sort_values( by ='service')  
df_sorted.head()
```

	rank	discipline	phd	service	sex	salary
55	AsstProf	A	2	0	Female	72500
23	AsstProf	A	2	0	Male	85000
43	AsstProf	B	5	0	Female	77000
17	AsstProf	B	4	0	Male	92000
12	AsstProf	B	1	0	Male	88000

Data Frames: Sorting

- İki veya daha fazla sütun kullanarak verileri sıralayabiliriz:

```
df_sorted = df.sort_values( by=['service', 'salary'], ascending = [True, False])  
df_sorted.head(10)
```

	rank	discipline	phd	service	sex	salary
52	Prof	A	12	0	Female	105000
17	AsstProf	B	4	0	Male	92000
12	AsstProf	B	1	0	Male	88000
23	AsstProf	A	2	0	Male	85000
43	AsstProf	B	5	0	Female	77000
55	AsstProf	A	2	0	Female	72500
57	AsstProf	A	3	1	Female	72500
28	AsstProf	B	7	2	Male	91300
42	AsstProf	B	4	2	Female	80225
68	AsstProf	A	4	2	Female	77500

Kayıp veriler

■ Kayıp veriler NaN ile gösterilir.

```
# Read a dataset with missing values
flights = pd.read_csv("/home/ipcvlab/Downloads/flights.csv")
```

```
# Select the rows that have at least one missing value
flights[flights.isnull().any(axis=1)].head()
```

	year	month	day	dep_time	dep_delay	arr_time	arr_delay	carrier	tailnum	flight	origin	dest	air_time	distance	hour	minute
330	2013	1	1	1807.0	29.0	2251.0	NaN	UA	N31412	1228	EWR	SAN	NaN	2425	18.0	7.0
403	2013	1	1	NaN	NaN	NaN	NaN	AA	N3EHAA	791	LGA	DFW	NaN	1389	NaN	NaN
404	2013	1	1	NaN	NaN	NaN	NaN	AA	N3EVAA	1925	LGA	MIA	NaN	1096	NaN	NaN
855	2013	1	2	2145.0	16.0	NaN	NaN	UA	N12221	1299	EWR	RSW	NaN	1068	21.0	45.0
858	2013	1	2	NaN	NaN	NaN	NaN	AA	NaN	133	JFK	LAX	NaN	2475	NaN	NaN

Kayıp veriler

- Veri çerçevesindeki eksik değerlerle başa çıkmanın birkaç yöntemi vardır:

df.method()	description
dropna()	Drop missing observations
dropna(how='all')	Drop observations where all cells is NA
dropna(axis=1, how='all')	Drop column if all the values are missing
dropna(thresh = 5)	Drop rows that contain less than 5 non-missing values
fillna(0)	Replace missing values with zeros
isnull()	returns True if the value is missing
notnull()	Returns True for non-missing values

Kayıp veriler

- Veri toplanırken eksik değerler sıfır olarak kabul edilir.
- Tüm değerler eksikse, toplam NaN'a eşit olacaktır.
- `cumsum()` ve `cumprod()` yöntemleri eksik değerleri yok sayar ancak elde edilen dizilerde bunları korur.
- `GroupBy` yöntemindeki eksik değerler hariç tutulur (R'deki gibi).
- Birçok tanımlayıcı istatistik yönteminde, eksik verilerin dışlanması gerekip gerekmediğini kontrol etme seçeneği (*skipna*) bulunmaktadır.

Pandas'da özetleme fonksiyonları

- Özetleme - Her grup hakkında bir özet istatistik hesaplama yapmaktır.
 - grup toplamalarını veya ortalamalarını hesaplama
 - grup büyüklüklerini / sayısını hesaplama vb.

min, max

count, sum, prod

mean, median, mode, mad

std, var

Pandas'da özetleme fonksiyonları

- `agg()` yöntemi her sütun için birden fazla istatistik hesaplandığında kullanışlıdır.

```
flights[['dep_delay', 'arr_delay']].agg(['min', 'mean', 'max'])
```

	dep_delay	arr_delay
min	-16.000000	-62.000000
mean	9.384302	2.298675
max	351.000000	389.000000

Temel tanımlayıcı istatistikler

df.method()	description
describe	Basic statistics (count, mean, std, min, quantiles, max)
min, max	Minimum and maximum values
mean, median, mode	Arithmetic average, median and mode
var, std	Variance and standard deviation
sem	Standard error of mean
skew	Sample skewness
kurt	kurtosis

I/O

```
df = pandas.read_csv('data.csv')  
df.to_csv('data.csv')
```

```
df = pandas.read_excel(  
    'data.xlsx', 'Sheet1', index_col=None,  
    na_values=['NA'])  
df.to_excel('data.xlsx', sheet_name='Sheet1')
```

```
df = pandas.read_json('data.json')  
json = df.to_json()
```


Data frame ekleme ve çıkarma

`concat()` adds dataframes

`join()` joins SQL style

`append()` adds rows

`insert()` inserts columns at a specific location

`df.sub(df['col1'], axis=0)`

(though you might also see `df - df['col1']`)

stack & unstack

- `df.stack()`, son iki sütunu bir etiket sütunu ile birleştirir.
- `unstack()` tersini yapar.

A	B
10	20
30	40

A	10
B	20
A	30
B	40

- Pivot tablolar??

pandas üzerine inşa edilmiş paketler

- Statsmodels → Statistics and econometrics
- Sklearn-pandas → Scikit-learn (machine learning) with pandas
- Bokeh → Big data visualisation
- seaborn → Data visualisation
- yhat/ggplot → Grammar of Graphics visualisation
- Plotly → Web visualisation using D3.js
- IPython/Spyder → Both allow integration of pandas dataframes
- GeoPandas → Pandas for mapping

Uygulama

- lessons içindeki notebook'lar
<https://bitbucket.org/hrojas/learn-pandas/downloads/>
- 10 Minutes to pandas
http://pandas.pydata.org/pandas-docs/stable/getting_started/10min.html