

pandas.

ما فرقتی درس کردیم؟ منابع، آن‌ها را چگونه تحلیل کنیم؟

تحلیل: به معنای شناختن است.

متطلبات را می‌خواهیم؟ یعنی یک سری از توابعی که از قبل نوشته شده و آماده است که آن‌ها را در برنامه خودمان استفاده می‌کنیم. از طریق ماژول pandas را با `import` می‌کنیم.

ماژول `pandas` یک تابعی است که یک سری توابع در آن نوشته شده است. اولین چیزی که لازم است در دسترس داشته باشیم در محیط پایتون، ابتدا باید آن را نصب و سپس `import` کنیم.

```
import pandas as pd
```

* تابعی است که در آن یک سری توابع از قبل نوشته شده، در بعضی از مواقع برای صرفه جویی در حافظه این‌ها می‌نویسند، یعنی از ماژول‌ها فقط یک سری توابع خاص را `import` می‌کنیم.

```
from pandas import api
```

کتابخانه‌های پایتون، برای کارهای مختلف مربوط به تحلیل داده، به عنوان یک ابزار استفاده می‌شوند.

Subject:

Year:

Month:

Date:

فهرست ساختار داده‌ای در pandas درج

① Series: یک ساختار داده‌ای مثل آرایه یک بعدی

② Data Frames: جداول ماتریسی و دو بعدی از ریتا

مجموعه داده‌ها یا یک یا چند سطر یا ستون (مجموعه‌ای از سری‌ها) نشان می‌دهد

$h = ('AA', '2012-02-01', 100, 10.2)$

$S = \text{pd.Series}(h)$

$\text{type}(S)$

pandas.core.series.Series

* در مثال بالا، شماره از یک آرایه و تابع Series یک سطر ساخته
 $\text{print}(S[2]) \rightarrow 100$

ساخت سری ۱: دیکشنری:

$d = \{ 'name': 'IBM', 'date': '2010-09-08', 'shares': 100, 'price': 10.2 \}$

$ds = \text{pd.Series}(d)$

ds

name

IBM

date

2010-09-08

shares

100

price

10.2

- برای دسترسی به مقدار (این سطر از طریق دیکشنری استفاده می‌کنیم)

$d['shares']$

$\rightarrow 100$

چه نیازی است یک فایل یا دیتابیس را به سرور و در نهایت تبدیل کنیم؟
 نیازی نیست به فرض اگر
 تغییر شده یا نیاز به برسم و از قواعد آن استفاده کنیم
 سرور یعنی یک چندین رکورد طولی عدد زغال اشغال می افتد. اندیس ها ترتیب زمانی
 رو شون میدن

فرض کنید لیست داریم می خوانیم به سرور تبدیل کنیم اندیس کنیم بدویم

```
F = ['FB', '2001-08-02', 90, 3.2]
```

```
F = pd.Series(F, index = ['name', 'date', 'shares', 'price'])
```

```
Print(F)
```

یعنی طول لیست خواندن به و ترنس index متبع سرور می بینیم

```
name FB
```

```
date 2001-08-02
```

```
shares 90
```

```
price 3.2
```

```
dtype: object
```

```
F['name'] → 'FB'
```

```
F[['name', 'date']]
```

دسترسی چند متغیر از سرور:

```
name : FB
```

```
date 2001-08-02
```

```
dtype: object
```

```
f.name
```

دسترسی ریز به نام صریح:

```
f.date
```

Data Frames

همچنین می‌توانیم داده‌های خود را به یک **Data Frame** تبدیل کنیم. **Data Frame** یک ساختار داده‌ای است که به ما امکان می‌دهد تا داده‌های خود را به صورت یک جدول سازماندهی کنیم. این ساختار داده‌ای به ما امکان می‌دهد تا داده‌های خود را به صورت یک جدول سازماندهی کنیم. این ساختار داده‌ای به ما امکان می‌دهد تا داده‌های خود را به صورت یک جدول سازماندهی کنیم.

```
data = { 'name': ['AA', 'IBM', 'Google'],
         'date': ['2001-12-01', '2012-02-10', '2010-04-27'],
         'shares': [100, 30, 20],
         'price': [12.3, 10.3, 32.2]
       }
```

```
df = pd.DataFrame(data)
```

```
df
```

	name	date	shares	price
0	AA	2001-12-01	100	12.3
1	IBM	2012-02-10	30	10.3
2	Google	2010-04-27	20	32.2

افزودن یک ستون جدید:

```
df['owner'] = 'unknown'
```

```
df['owner'] = ['nan', '1', '2']
```

مقدار نام نام نام

نام‌هایی که اندیس داده‌ها را مشخص می‌کند و اندیس‌ها را به نام‌ها تبدیل می‌کند.

Subject:

Year:

Month:

Date:

به طور دقیق تر این اندیس ها از صفر شروع می شوند.

شخص و تبیین بران اندیس برای دستا فریم.

`df.index = ['one', 'two', 'three']`

df

	name	date	shares	price	open
one					
two					
three					

لایحه براساس سقون ها و دستا فریم به عنوان اندیس

`df.set_index(['name'])`

df

name	date	shares	price	open
AA				
IBM				
GOOG				

سقون نام به عنوان

دارد می شود

و می توان چندین سقون

هم زمان به عنوان اندیس

را داشته باشیم به کار

طی اصلی در دستا فریم می باشد.

$$df = df.set_index(['name'])$$

حسرت میں۔ معافی دینا فریم: یہ سب سوال

df['shares']

df, shares

۱۰. دو صورت دوم و سوم بیان به معانی لغوی که
بسیار ضعیف و سست می باشد

دستوریه؟ حیدر، سقویا اردیافرم

- df ['shares', 'price']

۱۰ این دستور برای دستورهای استقون اگر استقون هم **جمله** می باشد و باید به شکل زیر

df[['shares', 'price']]

↳ Typ = Data from

بِسْمِ اللّٰهِ الرَّحْمٰنِ الرَّحِیْمِ
اَللّٰهُمَّ صَلِّ وَسَلِّمْ عَلٰی سِرِّ سِرِّ

$$dL[\text{'shaves'}]$$

↳ type = Dat a from

سرریز نفع حاصل از ریاضت فیه است .

Subject:

Year:

Month:

Date:

location ← loc (A) } قواعد دستوری: مقادیر دستوری
index location iloc (B)

آزمون: سئوال اولی به هر صفون ندارد، index دست

df.loc['name'] سطر مربوط به index، one را برگرداند.
خروجی سطر به شکل خواسته بود.

name	AA
date	2001-12-01
shares	100
price	12.3
owner	unknown

name: one, dtype: object

df.loc[['one']] خروجی به شکل دستوری خواسته بود.

	name	date	shares	price	owner
one					

دستوری: چند سطر از دستوری:

df
df['one':'two']

	name	date	shares	price	owner
one					
two					

df.loc [سطر و ستون خاص]

Subject:

Year:

Month:

Date:

دسترسی به سطر خاص و ستون خاص

df.loc ['one', 'shares'] → ۱۰۰

دسترسی به سطرهای یک ستون

df.loc[:, 'shares']

one	AA
two	IBM
three	Google

name: name, dtype: object

دسترسی به df.loc: دسترسی به سطر خاص از ریف فریم

df.loc [ستون و سطر]

در این جا به سطر خاص اندیس ها رو می دهیم

df.iloc [3, 2] → ۱۲.۳

ستون چهارم

سطر اول

یا در این جا به سطر index سطر می دهیم

حذف یک ستون:

~~df~~ del df['owner']

ستون owner را حذف می کند و در جایگاهش یک سری ۰ قرار می دهد

Subject:

Year:

Month:

Date:

دستور `drop`

`df.drop('shares', axis=1)`

که ستون مورد نظر

داده‌ها در ستون‌ها اعمال می‌شود
سطرها $axis=0$

دستور `drop` ستون `shares` حذف می‌کند و `inplace` نیست و در عین حال نمی‌تواند

نکته: مقدار استوارات قبل برای اعمال بر روی بیشترین ستون باید برآید و فتنه کشیده اسم و کثره حذف می‌شود.

تابع `del` فقط برای حذف یک ستون است و بیشتر از یک ستون حذف می‌کند.

خطا می‌دهد `df[['shares', 'name']]`

نکته: چون تابع `drop` عمل `inplace` نمی‌کند باید مقدار در دستور را دوباره در `df` قرار دهیم.

`df = df.drop('shares', axis=1)`

نکته: بار توابع `drop` و `set_index` یکدیگر را با هم `inplace` داریم اگر ما `True` قرار دهیم به صورت درجا عمل می‌کند تغییرات.

Chapter 2

وقتی صحبت از داده‌های حجمی می‌شود یعنی مایک مجموعه‌ای از رتبه در مقیاس فلان را داریم و در این کلاس خواندن کار کنیم. دیتا به شکل آمار عربی سری فایل داریم. اول باید دیتا را بخوانیم. دیتا در فرمت‌های CSV, TSV و ... وجود دارد.

CSV ← مقایسه، کاملاً لزوم جدا می‌شوند

TSV ← tab

SSV ← space

`Ceists = pd.read_csv('cast.csv', index_col=None)`

→ دستور بالا فایل CSV را می‌خواند و به شکل اتوماتیک دیتا فریم تبدیل می‌کند.
توانع مختلفی برای `read` در پایتون داریم.

* یعنی هیچ کدام از مقول‌ها در دیتا ست رده عنوان ایندکس در نظر گرفته شود.
مستون - عنوان index را در نظر بگیر. به شماره سطر.

`Ceists.head()`

این دستور به طور پیش فرض ۵ سطر اول را نمایش می‌دهد.

`Ceists.head(2)`

دو سطر اول را نمایش می‌دهد.

`Ceists.tail()`

پنج سطر آخر را نمایش می‌دهد.

`Ceists.tail(2)`

۲ سطر آخر را نمایش می‌دهد.

Subject:

Year:

Month:

Date:

```
titles = pd.read_csv('titles.csv', index_col = None, encoding = 'utf-8')
```

برای ستون‌هایی که مقدارشان به شکل Unicode درشته باشد باید این دستور را بنویسیم و نمی‌تواند فایل را در یکدست خطی دردهد.

طول تیترها 50000 - `len(titles)`

تغییر تنظیمات اولیه و پیش‌فرض
`pd.set_option('max rows', 10, 'max columns', 10)`

نمونه‌ها: ما نمی‌توانیم تعداد سطرها و ستون‌ها را به‌طور مستقیم تنظیم کنیم. پیش‌فرض 10 است. اما اگر اول و دوم را بخواهیم.

دسترسی به ستون خاص:
`t = titles['title']`
`type(t)`

نوع یک ستون خاص از دیتا فریم
`pandas.core.series.Series`
سری می‌باشد.

نقشه: توابع `head` و `tail` به‌طور مستقیم عمل می‌کنند. (بر روی عناصر آخر اول)

عناصر دو عنصر اول
`t.head(2)`