



# PANDAS DATA ANALYSIS

/imamdigi  Nov 29<sup>th</sup> 2018

# AGENDA

- Analisis deskriptif?
- Series/DataFrame? WTF!!
- Rutinitas analisis data di Pandas
- Mengurangi memori dengan mengubah tipe data
- Sort besar/kecil, atas/bawah

# ANALISIS DESKRIPTIF?

Gambaran secara umum dari sebuah data dengan memperkenalkan grafik atau visualisasi sehingga mudah ditaksir untuk mendukung keputusan selanjutnya maupun mendapatkan *insight*!



Pandas sering digunakan untuk keperluan analisis deskriptif maka dari itu, perlu diketahui tentang apa itu Analisis Deskriptif

# Intro dulu gaiss



I N E M A



Hmmm hmmm hmmm hmmmmm...

# SERIES/DATAFRAME? WTF!!

Series :

Adalah *one-dimensional array* yang dapat menampung berbagai macam tipe data

DataFrame :

Adalah struktur data array 2 dimensi berlabel yang yang dapat terdiri dari tipe data yang berbeda di setiap kolomnya



Baca selengkapnya, jangan males!

<https://pandas.pydata.org/pandas-docs/stable/dsintro.html>

# SERIES

```
</> x = pd.Series([0, 1, 4, 9, 16, 25], name='Squares')
```

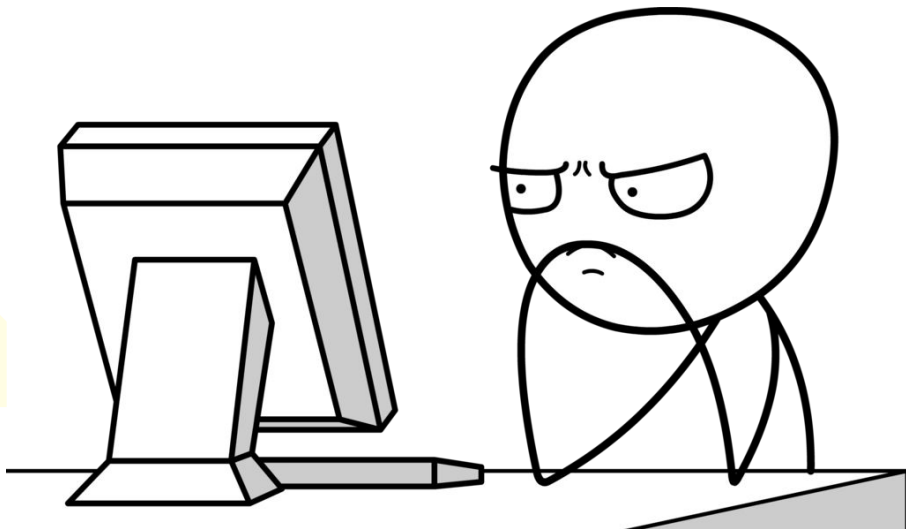
```
</> x = pd.Series([1205, 3646, 3741, 4080, 3270],  
                  index=['Arctic', 'Atlantic', 'Indian', 'Pacific', 'Southern'])
```

```
</> x = pd.Series({  
    'Arctic': 1205,  
    'Atlantic': 3646,  
    'Indian': 3741,  
    'Pacific': 4080,  
    'Southern': 3270  
})
```



Secara umum pembuatan Series adalah seperti ini:

```
s = pd.Series([data], index=[index])
```



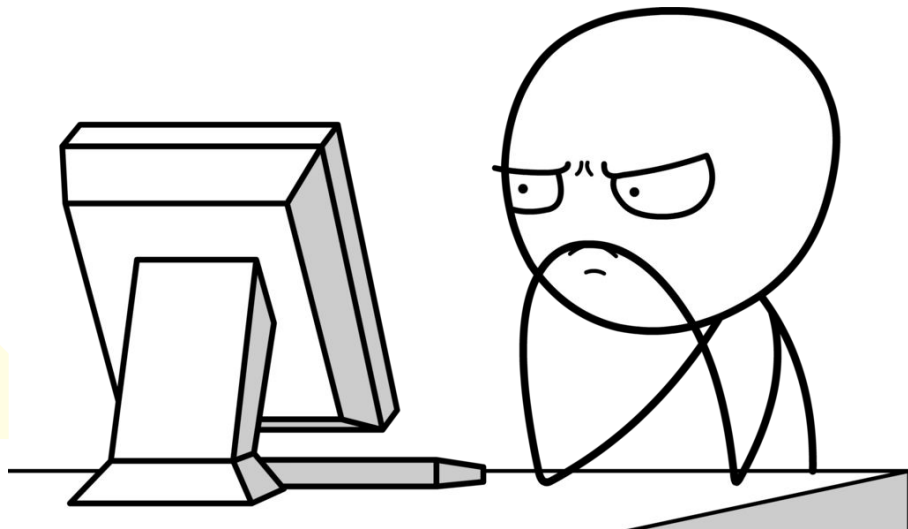
# DATAFRAME

```
</> x = pd.Series({'Arctic': 1205, 'Atlantic': 3646, 'Indian': 3741, 'Pacific': 4080,  
                  'Southern': 3270})  
y = pd.Series({'Arctic': 5567, 'Atlantic': 8486, 'Indian': 7906, 'Pacific': 10803,  
              'Southern': 7075})  
  
</> z = pd.DataFrame({'AVG': x, 'MAX': y})
```

	AVG	MAX
Arctic	1205	5567
Atlantic	3646	8486
Indian	3741	7906
Pacific	4080	10803
Southern	3270	7075




Karena series adalah 1-D maka DataFrame dapat dikonstruksi menggunakan Series.



# STATISTICAL ANALYSIS WITH DATAFRAMES

Return	Penjelasan
count	Hitungan frekuensi; berapa kali sesuatu terjadi
mean	Mean atau rata-rata
std	Standar deviasi, nilai numerik yang digunakan untuk menunjukkan seberapa luas data bervariasi
min	Jumlah minimum atau terkecil di set
25%	Persentil ke-25
50%	Persentil ke-50
75%	Persentil ke-75
max	Jumlah maksimum atau terbesar dalam set

 `DataFrame.describe()`





# MENGEMBANGKAN RUTINITAS ANALISIS DATA

Tidak ada pendekatan khusus untuk melakukan analisis data, sama halnya seperti anda melakukan pendekatan kepada lawan jenis :D

Terserah! Apapun langkah awalnya yang penting adalah tujuan anda tercapai!



Exploratory Data Analysis (EDA)

# METADATA

`</> df.head()`

	INSTNM	CITY	STABBR	HBCU	MENONLY	WOMENONLY	RELAFFIL	SATVRMID	SATMTMID	DISTANCEONLY	UGDS	UGDS_WH
0	Alabama A & M University	Normal	AL	1.0	0.0	0.0	0	424.0	420.0	0.0	4206.0	0.0
1	University of Alabama at Birmingham	Birmingham	AL	0.0	0.0	0.0	0	570.0	565.0	0.0	11383.0	0.5
2	Amridge University	Montgomery	AL	0.0	0.0	0.0	1	NaN	NaN	1.0	291.0	0.2
3	University of Alabama in Huntsville	Huntsville	AL	0.0	0.0	0.0	0	595.0	590.0	0.0	5451.0	0.6
4	Alabama State University	Montgomery	AL	1.0	0.0	0.0	0	425.0	430.0	0.0	4811.0	0.0

`</> df.describe()`

	count	unique	top	freq
INSTNM	7535	7535	San Francisco State University	1
CITY	7535	2514	New York	87
STABBR	7535	59	CA	773
MD_EARN_WNE_P10	6413	598	PrivacySuppressed	822
GRAD_DEBT_MDN_SUPP	7503	2038	PrivacySuppressed	1510

`</> df.info()`

`</> df.shape`



Bagaimana dengan data istilah/singkatan? Jangan pake DataFrame karena gak tepat, gausah protes!

# MENGURANGI MEMORI DENGAN MENGUBAH TIPE DATA

Pandas tidak secara luas mengklasifikasikan tipe data yang berbeda (kontinyu maupun kategorikal) tapi Pandas dapat dengan tepat mengenali perbedaan diantara tipe data



Di statistika secara umum ada dua jenis tipe data: kualitatif dan kuantitatif. Kuantitatif dibagi menjadi dua: diskrit dan kontinyu. Pengertiannya cari sendiri, jangan males!

# MENGUBAH TIPE DATA

```
</> cols = ['col1', 'col2', 'col3']  
columns = df.loc[:, cols]  
columns.head()
```

	RELAFFIL	SATMTMID	CURROPER	INSTNM	STABBR
0	0	420.0	1	Alabama A & M University	AL
1	0	565.0	1	University of Alabama at Birmingham	AL
2	1	NaN	1	Amridge University	AL
3	0	590.0	1	University of Alabama in Huntsville	AL
4	0	430.0	1	Alabama State University	AL

```
</> columns.dtypes
```

```
</> columns.memory_usage(deep=True)
```

```
</> columns['RELAFFIL'].astype(np.int8)
```



Untuk mengubah tipe data bisa menggunakan method **astype()** kemudian tempatkan tipe data tujuan pada argument

## Perbandingan memori

```
</> x = columns.memory_usage(deep=True)  
y = columns.memory_usage(deep=True)  
x / y
```

Index	1.000000
RELAFFIL	0.125000
SATMTMID	1.000000
CURROPER	1.000000
INSTNM	1.000695
STABBR	0.030538
dtype: float64	



Pandas menetapkan tipe integer dan float ke 64bit tanpa menghiraukan ukuran maksimum untuk DataFrame

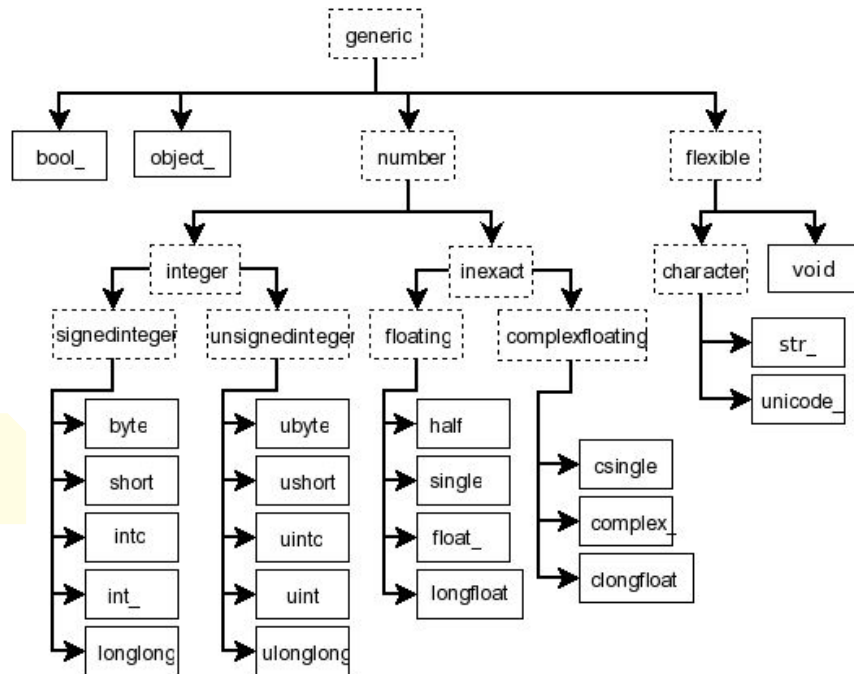
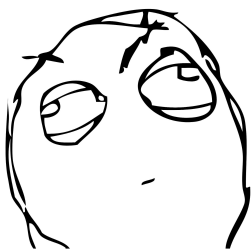
# HIRARKI DATA PADA NUMPY

Python hanya mendefinisikan satu jenis kelas data tertentu yang mana hanya ada satu jenis bilangan: bulat, floating-point, dan lain-lain.



Baca dokumentasinya di sini :

<https://docs.scipy.org/doc/numpy-1.13.0/reference/arrays.scalars.html>



# MEMILIH YANG TERKECIL DARI YANG TERBESAR

Bagian ini dapat digunakan untuk membuat sebuah informasi agar lebih menarik, contohnya:



Sekitar 100 mahasiswa dilarikan ke rumah sakit akibat terlalu sering mabar ML!, 43 diantaranya dinyatakan mengidap penyakit fakir internet, 50 diantaranya mengidap sakit telinga (karena dimarahi emak uang SPP dipake buat beli kuota internet) dan 7 mahasiswa mengidap ambeien.



Pada saat proses analisis seringkali kita perlu untuk mencari nilai terbesar maupun terkecil dari data yang sudah dikelompokkan.

# MENGURUTKAN DATA DARI GROUP BY

```
</> cols = df[['col1', 'col2', 'col3']]  
cols.sort_values('coln', ascending=False)
```

	movie_title	title_year	imdb_score
<b>3884</b>	The Veil	2016.0	4.7
<b>2375</b>	My Big Fat Greek Wedding 2	2016.0	6.1
<b>2794</b>	Miracles from Heaven	2016.0	6.8
<b>92</b>	Independence Day: Resurgence	2016.0	5.5
<b>153</b>	Kung Fu Panda 3	2016.0	7.2

```
</> cols.drop_duplicates(subset=['col1', 'coln'])
```

```
</> cols.nlargest(100, 'coln')
```

```
</> cols.nlargest(100, 'colx').smallest(50, 'coly')
```

	movie_title	title_year	content_rating	budget
<b>4026</b>	Compadres	2016.0	R	3000000.0
<b>4658</b>	Fight to the Finish	2016.0	PG-13	150000.0
<b>4661</b>	Rodeo Girl	2016.0	PG	500000.0
<b>3252</b>	The Wailing	2016.0	Not Rated	NaN
<b>4659</b>	Alleluia! The Devil's Carnival	2016.0	NaN	500000.0
<b>4731</b>	Bizarre	2015.0	Unrated	500000.0
<b>812</b>	The Ridiculous 6	2015.0	TV-14	NaN
<b>4831</b>	The Gallows	2015.0	R	100000.0
<b>4825</b>	Romantic Schemer	2015.0	PG-13	125000.0
<b>3796</b>	R.L. Stine's Monsterville: The Cabinet of Souls	2015.0	PG	4400000.0

	movie_title	imdb_score	budget
<b>4804</b>	Butterfly Girl	8.7	180000.0
<b>4801</b>	Children of Heaven	8.5	180000.0
<b>4706</b>	12 Angry Men	8.9	350000.0
<b>4550</b>	A Separation	8.4	500000.0
<b>4636</b>	The Other Dream Team	8.4	500000.0



Parameter pertama harus bertipe integer yang mana jumlah dari row, parameter kedua harus bertipe string.nama kolom



Hmmmmmm.....  
Ada pertanyaan gak?





# Udah Segitu Aja!....

Maaaaaakaaaaaasiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiihhhh