

What is pandas?

- Recent API based on Numpy
- Devised by Wes McKinney
- Fast and intuitive data structures
- Easy to work with messy and irregularly indexed data
- Optimized for performance, with critical code paths compiled to C
- Adopts concepts of R language

Main focus

- The two basics structures of pandas
 - Series 1d array
 - DataFrame 2d array
 - Panel nd array ($n > 2$)
- Filtering, selecting data
- Aggregating, transforming data
- Joining, concatenating, merging data
- Descriptive basics statistics

Installing pandas

- Version python 2.6 or 2.7
- Dependencies:
 - NumPy 1.6.1 or higher
- Optional dependencies:
 - Matplotlib to plot
 - SciPy for statistical functions

Exercise

- ```
> sudo apt-get install python-pandas
```
- ```
> git clone git://github.com/pydata/pandas.git
```

```
> cd pandas
```

```
> python setup.py install
```
- Header:

```
import pandas as pd
```

Series

	index		value
0	C	▶	3
1	B	▶	7
2	A	▶	4
3	D	▶	4
4	D	▶	0.3

- Subclass from `numpy.ndarray`
- Any type of data (numeric, string, boolean...)
- Index need not to be ordered
- Duplicated index are possible

Some vocabulary:

- `Series.index`: list of indices
- `Series.values`: list of values

DataFrame

columns

index

	id	country	isOver	amount
a	P255	Afg	True	300000
b	P31256	Fr	False	22354
c	P2245	Cor	False	12478
d	415	Som	False	Nan
e	P332	Esp	True	4789123

- ndarray-like
- 2D data structure (for n D data structures see Panel)
- Dictionary of series
- Row and column index
- Size mutable: insert or delete columns

DataFrame

- Some vocabulary
 - `DataFrame.index`: list of DataFrame indices
 - `DataFrame.values`: 2D array of all values contained in the DataFrame
 - `DataFrame.columns`: list of columns labels
 - `axis`: indicates the axis index for rows (`axis = 0`), columns (`axis = 1`),
or even n th axis in panels

Construction of Series and DataFrame

Exercise

- Directly editing

```
s = pd.Series([3,7,4,4,0.3] ,  
              [index = ['a','b','c','d','e']])  
  
df = pd.DataFrame(np.arange(9).reshape(3,3),  
                  [index = ['b','a','c'],  
                   columns=['Paris','Berlin','Madrid']])
```

- From a python dict

```
data = {'Paris': [0,3,6,999999999],'Berlin': [1,4,7], 'Madrid': [2,5,8]}  
  
df = pd.DataFrame(data,  
                  [index = ['b','a','c','d'],  
                   Columns = ['Paris', 'Berlin', 'Madrid'] ])
```

Warning: index array size >= max element array size

- Several methods in the API to import from databases

```
df = pd.read_csv(path/fichier.csv,  
                 [index_col = [...]])  
  
df = pd.read_table(path/fichier.txt,  
                  [sep = ','])
```

Selection of data

- Selection on series

In: <code>s</code>		In: <code>s['b']</code>		In: <code>s['a':'c']</code>		In: <code>s['d']</code>		In: <code>s[1]</code>
Out:		Out:		Out:		Out:		Out:
a	3.0	7.0		a	3.0	d	4.0	7.0
b	7.0			b	7.0	d	0.3	
c	4.0			c	4.0			
d	4.0							
d	0.3							

- The returned object is either a value, or a subset of the initial series `s`
- Select some data with integer index OR index label
 - **Warning: Work only if the index type is not numeric**

Selection of data

- Filter on DataFrame

In: df
Out:

	Paris	Berlin	Madrid
b	0	1	2
a	3	7	5
c	6	4	8

In: df[:2]
Out:

	Paris	Berlin	Madrid
b	0	1	2
a	3	7	5

In: df[df['Paris']>1]
Out:

	Paris	Berlin	Madrid
a	3	7	5
c	6	4	8

df.Berlin[df['Berlin']>1]=0
In: df
Out:

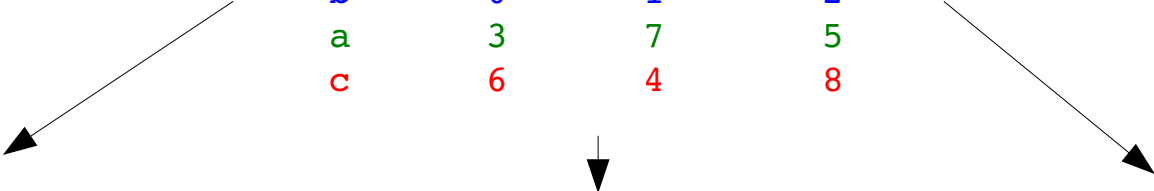
	Paris	Berlin	Madrid
b	0	1	2
a	3	0	5
c	6	0	8

- Output Object: subset of the initial DataFrame

Selection of data

- The indexing field ***ix*** enables to select a subset of the rows and columns from a DataFrame.

```
In: df
Out:
      Paris  Berlin  Madrid
b         0        1        2
a         3        7        5
c         6        4        8
```



```
In: df.ix['a', 'Berlin']
Out:
7
```

```
In: df.ix[['b', 'c'], 'Berlin']
Out:
b    1
c    4
Name: Berlin
```

```
In: df.ix[:, 'Berlin']
Out:
b    1
a    7
c    4
Name: Berlin
```

- Output Object: a value OR a Series subset of the DataFrame

Exercise

Select the rows where 'Rank' = 0