Python para Data Science
Acesso rápido Pandas básico

Biblioteca

A biblioteca Pandas é construída sob NumPy e fornece ótimas ferramentas de estruturas e de análise de dados em Python.

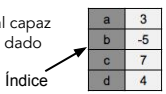
Em geral, importa-se a biblioteca dessa forma:

```
>>> import pandas as pd
```

Estruturas de dados no Pandas

Séries

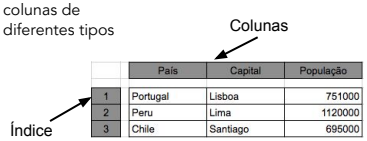
Uma matriz uni-dimensional capaz de conter qualquer tipo de dado




```
>>> s = pd.Series([3, -5, 7, 4], index=['a', 'b', 'c', 'd'])
```

Frame de dados (Data frame)

Uma estrutura de dados bi-dimensional com colunas de diferentes tipos




```
>>> data = {
'País': ['Portugal', 'Peru', 'Chile'],
'Capital': ['Lisboa', 'Lima', 'Santiago'],
'População': [751000, 1120000, 695000]
}

>>>df = pd.DataFrame(data,
        columns=['País', 'Capital', 'População'])
```

Pedindo ajuda

```
>>> help(pd.Series.loc)
```

Selecionando

Recuperar elementos

```
>>>s['b']
 -5
>>>df[1:]
  País     Capital  População
1 Portugal Lisboa   751000
2 Peru     Lima     1120000
```

Retorna um elemento

Retorna um subconjunto de um DataFrame

Seleção, indexação booleana & definição

Por posição

```
>>>df.iloc[0][0]
 'Portugal'
>>>df.at[[0], [0]]
 'Portugal'
```

Seleciona um único valor por linha & coluna

Por rótulo

```
>>>df.iloc[0]['País']
 'Portugal'
>>>df.at[[0], ['País']]
 'Portugal'
```

Seleciona um único valor por rótulo de linha & coluna

Por rótulo/posição

```
>>>df.ix[2]
 País       Chile
 Capital    Santiago
 População  695000
>>>df.ix[:, 'Capital']
 0  Lisboa
 1  Lima
 2  Santiago
>>>df.ix[1, 'Capital']
 Lima
```

Seleciona uma única linha de um subconjunto de linhas

Seleciona uma única coluna de um subconjunto de colunas

Seleciona linhas e colunas

Indexação booleana

```
>>>s[~(s > 1)]
>>>s[(s < -1) | (s >= 6)]
>>>df[df['População'] > 1000000]
```

Série *s*, valor não é > 1
s, valor é < -1 ou >=6
Filtro para ajustar o DataFrame

Definição

```
>>>s['a'] = 6
```

Define índice *a* de *s* = 6

Deletando

```
>>> s.drop(['a', 'c'])
>>> s.drop('País', axis=1)
```

Deleta valores das linhas
Deleta valores das colunas

Ordem & Classificação

```
>>> df.sort_index(by='País')
>>> s.order()
>>> df.rank()
```

Ordena por linha ou coluna
Ordena por valores
Classifica as entradas

Recuperando informações de séries e dataframes

Informações básicas

```
>>> df.shape      (linhas, colunas)
>>> df.index      Descreve o índice
>>> df.columns    Descreve as colunas
>>> df.info()     Info sobre o dataframe
>>> df.count()    Número de valores não-NA
```

Resumos

```
>>>df.sum()              Soma de valores
>>>df.cumsum()           Soma acumulada
>>>df.min/df.max()       Valores max e min
>>>df.idmin()/df.idmax() Valor max e min do índice
>>>df.describe()         Sumário estatístico
>>>df.mean()             Média dos valores
>>>df.median()           Mediana dos valores
```

Aplicando funções

```
>>> f = lambda x: x*2
>>> df.apply(f)
>>> df.applymap(f)
```

Aplica a função
Aplica a função por elemento

Alinhamento de dados

Alinhamento interno de dados

```
>>> s3 = pd.Series([7, -2, 3], index=['a', 'c', 'd'])
>>> s + s3
a    10.0
b    NaN
c    5.0
d    7.0
```

Operações aritméticas com métodos de preenchimento

É possível alinhar dados usando métodos de preenchimento.

```
>>> s.add(s3, fill_value = 0)
a    10.0
b    -5.0
c    5.0
d    7.0
>>> s.sub(s3, fill_value = 2)
>>> s.div(s3, fill_value = 4)
>>> s.mul(s3, fill_value = 3)
```

I/O

Ler e escrever em CSV

```
>>> pd.read_csv('arquivo.csv', header=None, nrows=5)
>>> pd.to_csv('meudf.csv')
```

Ler e escrever em Excel

```
>>> pd.read_excel('arquivo.xls', sheetname='Pasta1')
>>> pd.to_excel('meudf.csv', sheetname='Pasta1')
Ler múltiplas pastas de um mesmo arquivo
>>> xlsx = pd.ExcelFile(arquivo.xls')
>>> df = pd.read_excel(xlsx, 'Pasta1')
```

Ler e escrever query SQL ou tabela de dados

```
>>> from sqlalchemy import create_engine
>>> engine = create_engine('sqlite:///:memory:')
>>> pd.read_sql("SELECT * FROM m_tabela;", engine)
>>> pd.read_sql_table('m_tabela', engine)
>>> pd.read_sql_query("SELECT * FROM m_tabela;", engine)
```

read_sql() é um wrapper conveniente ao redor de read_sql_table() e de read_sql_query()

```
>>> pd.to_sql(meuDf', engine)
```