

Khái niệm cơ bản về cấu trúc

Tim hiểu Python cho Khoa học dữ liệu một cách tương tác tại www.DataCamp.com

gấu trúc

Thư viện Pandas được xây dựng trên NumPy và cung cấp các cấu trúc dữ liệu và công cụ phân tích dữ liệu để sử dụng cho ngôn ngữ lập trình Python.

Sử dụng các quy ước nhập sau:

```
>>> nhập gấu trúc dưới dạng pd
```

Cấu trúc dữ liệu Pandas

Loạt

Mảng được gắn nhãn một chiều có khả năng chứa bất kỳ loại dữ liệu nào.

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

Khung dữ liệu

cột

Mục lục

	Vốn_nước	Dân_số
0	nước Bỉ Brussels	11190846
1	Ấn ĐộNew Delhi	1303171035
2	Brazil Brasilia	207847528

Cấu trúc dữ liệu được gắn nhãn hai chiều với các cột có thể có các kiểu khác nhau

```
>>> data = {'Quốc gia': ['Bỉ', 'Ấn Độ', 'Brazil'],
            'Thủ đô': ['Brussels', 'New Delhi', 'Brasilia'],
            'Dân số': [11190846, 1303171035, 207847528]}

>>> df = pd.DataFrame(dữ liệu,
                       cột=['Quốc gia', 'Thủ đô', 'Dân số'])
```

Yêu cầu giúp đỡ

```
>>> trợ giúp(pd.Series.loc)
```

Lựa chọn

Đồng thời xem Mảng NumPy

Nhận

```
>>> s['b']
-5

>>> df[1:]
      Quốc gia      Dân số vốn
1  Ấn Độ New Delhi 1303171035
2  Brazil Brasilia  207847528
```

Lấy một phần tử

Nhận tập hợp con của DataFrame

Chọn, lập chỉ mục Boolean và thiết lập

theo vị trí

```
>>> df.iloc[[0],[0]]
      Quốc gia
0  nước Bỉ

>>> df.iat[[0],[0]]
      Quốc gia
0  nước Bỉ
```

Theo nhãn

```
>>> df.loc[[0], ['Quốc gia']]
      Quốc gia
0  nước Bỉ

>>> df.at[[0], ['Quốc gia']]
      Quốc gia
0  nước Bỉ
```

Theo Nhãn/Vị trí

```
>>> df.ix[2]
      Quốc gia      Brazil
      Thủ đô      Brasilia
      Dân số 207847528

>>> df.ix[:, 'Capital']
0  Bruxelles
1  New Delhi
2  Brasilia

>>> df.ix[1, 'Vốn']
      New Delhi'

Lập chỉ mục Boolean
>>> s[~(s > 1)] >>>
s[(s < -1) | (s > 2)] >>>

df[df['Population']>1200000000] Sử dụng bộ lọc để điều chỉnh DataFrame

Cài đặt
>>> s['a'] = 6

Đặt chỉ mục a của Series s thành 6

Chọn một giá trị theo hàng và cột



Chọn một giá trị theo nhãn hàng và cột



Chọn một hàng của tập hợp con các hàng



Chọn một cột trong tập hợp con của các cột



Chọn hàng và cột



Chuỗi s có giá trị không >1 s trong đó giá trị là <-1 hoặc >2



Đặt chỉ mục a của Series s thành 6


```

Rời

```
>>> s.drop(['a', 'c']) >>>
df.drop('Country', axis=1) Bỏ giá trị khỏi cột(axis=1)
```

Bỏ giá trị từ các hàng (axis=0)

Sắp xếp & Xếp hạng

```
>>> df.sort_index() >>>
df.sort_values(by='Country') Sắp xếp theo các giá trị dọc theo một trục
>>> df.rank()
```

Sắp xếp theo nhãn dọc theo một trục

Chỉ định thứ hạng cho các mục

Truy xuất thông tin chuỗi/DataFrame

Thông tin cơ bản

```
>>> df.shape >>>
df.index >>>
df.columns >>>
df.info() >>>
df.count()
```

(những hàng, những cột)

Mô tả chỉ số

Mô tả các cột DataFrame

Thông tin về DataFrame

Số lượng giá trị không phải NA

Bản tóm tắt

```
trị >>> df.sum()
trị >>> df.cumsum() >>>
df.min()/df.max()
>>> df.idxmin()/df.idxmax() Giá trị chỉ mục tối thiểu/tối đa >>>
df.describe()
>>> df.mean() >>>
df.median()
```

Tổng các giá

Tổng tích lũy của các giá

Giá trị tối thiểu/tối đa

Giá trị chỉ mục tối thiểu/tối đa

thống kê tóm tắt

Ý nghĩa của các giá trị

Giá trị trung bình

Áp dụng hàm

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

Áp dụng chức năng

Áp dụng từng phần tử hàm

Căn chỉnh dữ liệu

Căn chỉnh dữ liệu nội bộ

Giá trị NA được đưa vào trong các chỉ số không trùng nhau:

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

a	10,0
b	NaN
c	5,0
d	7,0

Các phép toán số học với các phương pháp thức diễn

Bạn cũng có thể tự mình thực hiện việc căn chỉnh dữ liệu nội bộ với sự trợ giúp của các phương pháp thức diễn:

```
>>> s.add(s3, fill_value=0)
      a
0  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)
```

Vào/ra

Đọc và ghi vào CSV

```
>>> pd.read_csv('file.csv', header=None, nrows=5)
>>> df.to_csv('myDataFrame.csv')
```

Đọc và ghi vào Excel

```
>>> pd.read_excel('file.xlsx')
>>> pd.to_excel('dir/myDataFrame.xlsx', sheet_name='Sheet1')

Đọc nhiều trang từ cùng một tập tin
>>> xlsx = pd.ExcelFile('file.xls')
>>> df = pd.read_excel(xlsx, 'Sheet1')
```

Đọc và ghi vào truy vấn SQL hoặc bảng cơ sở dữ liệu

```
>>> từ nhập sqlalchemy create_engine
>>> engine = create_engine('sqlite:///memory:')
>>> pd.read_sql("CHỌN * TỪ my_table;", công cụ)
>>> pd.read_sql_table('my_table', engine)
>>> pd.read_sql_query("CHỌN * TỪ my_table;", công cụ)

read_sql() là một trình bao bọc tiện lợi xung quanh read_sql_table() và read_sql_query()

>>> pd.to_sql('myDf', động cơ )
```