

3. Khởi tạo và truy cập với dữ liệu kiểu series trong pandas

Series là một đối tượng giống mảng một chiều. Nó có khả năng giữ bất kỳ loại dữ liệu nào, ví dụ: Số nguyên, floats, strings, các đối tượng Python, Nó có thể được xem như là một cấu trúc dữ liệu với hai mảng: một lưu chỉ mục, cái còn lại lưu dữ liệu.

Một pandas series có thể được tạo ra bằng cách sử dụng phương thức sau: **pandas.Series(data=None, index=None, dtype=None, copy=False)**.

Trong đó:

data có thể là list, dictionary hoặc hằng số.

index là tập các giá trị chỉ mục duy nhất có thể băm và độ dài len(index) bằng với len(data).

dtype là kiểu dữ liệu của đối số data.

copy có giá trị True/False, mặc định là False. Copy input data.

Khởi tạo từ một List hoặc Array.

Chúng ta định nghĩa một đối tượng Series đơn giản trong ví dụ sau bằng cách khởi tạo đối tượng Pandas Series từ một danh sách (list).

```
>>> import pandas as pd
>>> import numpy as np
>>> S = pd.Series(np.random.randint(100,size = 5))
>>> S
0    43
1    56
2    20
3     7
4    10
dtype: int32
>>>
```

Khi chúng ta không gán chỉ mục cho đối tượng Series thì pandas sẽ tự gán cho nó giá trị mặc định. Ví dụ trước, nhưng chúng ta thấy hai cột là đầu ra: cột bên phải chứa dữ liệu, trong khi cột bên trái chứa chỉ mục. Pandas đã tạo ra một chỉ mục mặc định bắt đầu từ 0 đến 4, là chiều dài của dữ liệu trừ đi 1.

Chúng ta có thể trực tiếp lấy thông tin các chỉ mục và giá trị của Series S như sau:

```
>>> print(S.index)
RangeIndex(start=0, stop=5, step=1)
>>> print(S.values)
[43 56 20  7 10]
>>>
```

So sánh với ndarray trong numpy thì chỉ mục của nó chỉ là các con số không, trong khi đó với Series thì chúng ta hoàn toàn có thể gán bất cứ Object nào. Ví dụ:

```
>>> phones = ['Iphone','Samsung Note','Samsung S','Nokia']
```

```
>>> quantities = [10,12,30,100]

>>> S = pd.Series(quantities, index=phones)

>>> S

Iphone      10
Samsung Note 12
Samsung S    30
Nokia       100
dtype: int64

>>>
```

Series đem lại một lợi thế lớn cho mảng NumPy và rõ ràng từ ví dụ trước: Chúng ta có thể sử dụng chỉ mục tùy ý.

Cộng/Trừ hai đối tượng Series

Khi thực hiện cộng/trừ hai đối tượng Series S1 và S2, thì các giá trị tại các chỉ mục giống nhau sẽ được cộng/trừ cho nhau. Nếu một chỉ mục không tồn tại ở một trong hai Series, giá trị cho Series này sẽ là NaN. Ví dụ:

```
>>> phones = ['Iphone',"Samsung Note","Samsung S","Nokia"]

>>> quantities = [10,12,30,100]

>>> S1 = pd.Series(quantities, index=phones)

>>> S2 = pd.Series([32,14,15,16], index=phones)
```

```
>>> S1 + S2

Iphone      42
Samsung Note 26
Samsung S    45
Nokia       116
dtype: int64

>>>
```

```
>>> phones2 = ['LG',"Samsung Note","Samsung S","Nokia"]

>>> S3 = pd.Series([32,14,15,16], index=phones2)

>>>
```

```
>>> S2 + S3

Iphone      NaN
LG          NaN
```

```
Nokia          116.0
Samsung Note   26.0
Samsung S      45.0
dtype: float64
>>>
```

Có thể truy xuất các giá trị đơn của một hoặc nhiều hơn một giá trị của Series qua chỉ số hoặc một list các chỉ số:

```
>>> S["Iphone"]
10
>>> S[phones]
Iphone          10
Samsung Note    12
Samsung S       30
Nokia          100
dtype: int64
>>> S[phones2]
LG              NaN
Samsung Note    12.0
Samsung S       30.0
Nokia          100.0
dtype: float64
>>>
```

Khởi tạo từ một dictionary

Thay vì phải truyền vào gồm hai danh sách như các ví dụ phía trên ta có thể tạo ra Series bằng cách truyền dictionary như ví dụ sau:

```
>>> phones = ['Iphone', 'Samsung Note', 'Samsung S', 'Nokia']
>>> quantities = [10, 12, 30, 100]
>>> dictionary = dict(zip(phones, quantities))
>>> dictionary
{'Samsung Note': 12, 'Samsung S': 30, 'Nokia': 100, 'Iphone': 10}
>>> S = pd.Series(dictionary)
>>> S
Iphone          10
Nokia          100
Samsung Note    12
Samsung S       30
```

```
dtype: int64
```

```
>>>
```

Kết luận

Series trong Pandas là một đối tượng khá giống mảng một chiều và có khá nhiều điểm tương đồng với việc tạo ra một mảng trong Numpy. Tuy nhiên Series lại linh hoạt hơn nhiều do việc sử dụng chỉ mục tùy ý: thêm hai chuỗi có cùng chỉ số, truy cập các giá trị qua các chỉ số hay tạo Series từ dictionary. Cú pháp để tạo ra một Pandas series là `pandas.Series()`. Trong bài tiếp theo chúng tôi sẽ giới thiệu thêm các thao tác cơ bản với pandas series.