

附录：Series 详细方法

2023 年 9 月 18 日

Pandas 的 Series

Series 是 Pandas 的两个常用的数据结构之一，还有一个是 DataFrame。

这两个常用数据结构能够解决绝大数在数据处理任务中遇到的问题。

那什么是 Series 呢? > Series is a one-dimensional labeled array capable of holding any data type (integers, strings, floating point numbers, Python objects, etc.). The axis labels are collectively referred to as the index.

——Pandas 官方

Series 是一个一维带标签的数组，可以保存任何数据类型 (整数、字符串、浮点数、Python 对象等)。轴标签统称为索引。

1. 创建 Series

1.1 使用列表进行创建

使用 `pandas.Series` 的方法来创建 Series 对象，里面包含一个列表，这里 Series 第一个字母需要大写。例如：

```
[1]: import pandas as pd
      s = pd.Series([4, 5, 6])
      s
```

```
[1]: 0    4
      1    5
      2    6
      dtype: int64
```

在交互环境里，左边一侧是索引，右边一侧是值。当我们不给 Series 指定索引时，pandas 默认生成的索引是从 0 到 N-1 (N 是数据长度)，这里 N 是 3。

通过 value 和 index 属性分别获得 Series 对象的值和属性。

```
[3]: s.values
```

```
[3]: array([4, 5, 6])
```

```
[2]: s.index
```

```
[2]: RangeIndex(start=0, stop=3, step=1)
```

我们使用 type 看下 Series.values 的数据类型，如下显示它是一个 Numpy 数组。

```
[4]: type(s.values)
```

```
[4]: numpy.ndarray
```

1.2 添加索引进行创建

在创建 Series 的时候，通过指定索引序列，来标识每个值：

```
[5]: s2 = pd.Series([4, 5, 6], index=['a', 'b', 'c'])  
s2
```

```
[5]: a    4  
     b    5  
     c    6  
     dtype: int64
```

当 Series 拥有索引时，选取数据时可以指定索引：

```
[6]: s2['a']
```

```
[6]: 4
```

```
[7]: s2[['b', 'a']]
```

```
[7]: b    5  
     a    4  
     dtype: int64
```

类似 Numpy 使用布尔值对数组进行过滤，Series 的操作也类似：

```
[8]: s2[s2>0]
```

```
[8]: a    4  
     b    5  
     c    6  
     dtype: int64
```

```
[9]: s2 * 2
```

```
[9]: a    8  
     b   10  
     c   12  
     dtype: int64
```

```
[10]: import numpy as np  
      np.exp(s2)
```

```
[10]: a    54.598150  
     b   148.413159  
     c   403.428793  
     dtype: float64
```

1.3 使用字典来创建

```
[7]: post_code = {'Shanghai':200000, 'Beijing': 100000, 'Tianjing': 300000, 'Chongqing':400000}  
      s3 = pd.Series(post_code)  
      s3
```

```
[7]: Shanghai    200000  
     Beijing    100000  
     Tianjing   300000  
     Chongqing  400000  
     dtype: int64
```

2. 修改索引

使用以下方法进行 Series 的索引修改：

```
[3]: s3.index = ["SH", "BJ", "TJ", "CQ"]
s3
```

```
[3]: SH      200000
     BJ      100000
     TJ      300000
     CQ      400000
     dtype: int64
```

3. 选取元素

我们可以使用以下三种方式来选取某一个元素：

```
[8]: s3["Shanghai"]
```

```
[8]: 200000
```

```
[9]: s3.loc["Shanghai"]
```

```
[9]: 200000
```

```
[10]: s3.iloc[0]    # 使用整数索引 [0, 1, 2, 3]
```

```
[10]: 200000
```

选取多个元素，可以使用：

```
[11]: s3[['Shanghai', 'Beijing', 'Tianjing']]
```

```
[11]: Shanghai      200000
     Beijing        100000
     Tianjing       300000
     dtype: int64
```

4. 切片

如果是作切片操作，使用以下方式：

```
[12]: s3.loc["Shanghai":"Tianjing"]
```

```
[12]: Shanghai    200000
      Beijing     100000
      Tianjing    300000
      dtype: int64
```

```
[13]: s3.iloc[:3]    # 使用整数索引 [0, 1, 2, 3]
```

```
[13]: Shanghai    200000
      Beijing     100000
      Tianjing    300000
      dtype: int64
```

5. 排序

pandas 下面的排序，使用 `sort_values`:

```
[20]: s3.sort_values()
```

```
[20]: Beijing     100000
      Shanghai    200000
      Tianjing    300000
      Chongqing   400000
      dtype: int64
```

如果是从大到小，使用参数 `ascending=False`，`ascending` 表示升序，所以令其等于 `False`

```
[21]: s3.sort_values(ascending=False)
```

```
[21]: Chongqing   400000
      Tianjing    300000
      Shanghai    200000
      Beijing     100000
      dtype: int64
```